Problem Features vs. Algorithm Performance on Rugged Multi-objective Combinatorial Fitness Landscapes

Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernan Aguirre, Kiyoshi Tanaka

Problem Features vs. Algorithm Performance on Rugged Multi-objective Combinatorial Fitness Landscapes

Fabio Daolio fdaolio@shinshu-u.ac.jp
Shinshu University, Faculty of Engineering, Nagano, Japan

Arnaud Liefooghe arnaud.liefooghe@univ-lille1.fr
Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000, Lille, France
Inria Lille–Nord Europe, F-59650 Villeneuve d'Ascq, France

Sébastien Verel verel@lisic.univ-littoral.fr
Univ. Littoral Côte d'Opale, EA 4491 – LISIC – F-62228 Calais, France

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

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

Abstract
In this paper, we attempt to understand and to contrast the impact of problem features on the performance of randomized search heuristics for black-box multi-objective combinatorial optimization problems. At first, we measure the performance of two conventional dominance-based approaches with unbounded archive on a benchmark of enumerable binary optimization problems with tunable ruggedness, objective space dimension, and objective correlation (ρMNK-landscapes). Precisely, we investigate the expected runtime required by a global evolutionary optimization algorithm with an ergodic variation operator (GSEMO) and by a neighborhood-based local search heuristic (PLS), to identify a $(1+\varepsilon)-$approximation of the Pareto set. Then, we define a number of problem features characterizing the fitness landscape, and we study their intercorrelation and their association with algorithm runtime on the benchmark instances. At last, with a mixed-effects multi-linear regression we assess the individual and joint effect of problem features on the performance of both algorithms, within and across the instance classes defined by benchmark parameters. Our analysis reveals further insights into the importance of ruggedness and multi-modality to characterize instance hardness for this family of multi-objective optimization problems and algorithms.

Keywords
Evolutionary multi-objective optimization, black-box 0–1 multi-objective problems, feature-based analysis, fitness landscape and problem difficulty, empirical performance modeling, multi-level multi-variate analysis, random-effects mixed models.

1 Introduction

1.1 Motivation

Many optimization problems arising in real-world applications are characterized by a discrete solution space, and by multiple objective functions, such as cost, profit, or risk,

This article is an extended version of the one presented in GECCO 2015 (Daolio et al., 2015).

that are ill-defined, computationally expensive, or for which an analytical form is not available (Coello and Lamont, 2004). A difficult task is then to identify or approximate a set of solutions, known as the Pareto set, representing the possible optimal trade-offs among the objectives. But these black-box multi-objective combinatorial optimization problems severely limit the number of evaluations an optimizer can perform. Despite the increasing number of available general-purpose heuristics for evolutionary multi-objective optimization (EMO), see e.g., Deb (2001); Coello et al. (2007), we argue that their design and configuration is still mainly based on intuition, and that the understanding of their performance and design principles is still in its infancy and comparatively less advanced than in the single-objective case. A challenging issue is to identify a number of general-purpose features characterizing problem hardness, and to understand which problem features have an influence on the performance of EMO algorithms, as well as the differences and the similarities across different algorithm and problem classes.

1.2 Related Work

In single-objective combinatorial optimization, research on fitness landscape analysis aims at characterizing the fitness landscape associated with the optimization problem where the algorithm operates (Merz, 2004; Richter and Engelbrecht, 2014). Contrary to a complexity-theoretical perspective of convergence properties and runtime analysis, a fitness landscape analysis rather relies on a mathematical model that helps to understand the relation between the geometry of an optimization problem and the dynamics of a randomized search heuristic. Tools from graph theory, feature-based analysis, or correlation and regression analysis, are means to investigate the difficulties that an algorithm faces when solving a particular problem. This paradigm is particularly relevant for black-box optimization, for which problem-specific expert knowledge is usually difficult to obtain. More recently, benchmark parameters as well as problem and fitness landscape features have been used as input variables in statistical regression analysis in order to estimate, and then understand, their relationship with the performance of randomized search heuristics for single-objective optimization problems of continuous and combinatorial nature; see e.g., Mersmann et al. (2011); Bischl et al. (2012); Daolio et al. (2012); Mersmann et al. (2012). In particular, although they only consider benchmark parameters in their analysis, it is worth noticing that Chiarandini and Goegebeur (2010) investigate mixed models in order to separate the effects of algorithm components and problem characteristics when analyzing local search algorithms for the 2-edge-connectivity augmentation problem.

In multi-objective combinatorial optimization, few attempts have been made at designing and analyzing problem and fitness landscape features, whereas there is a strong evidence that problem-related properties are known to largely affect the properties of the Pareto set (Mote et al., 1991) and the behavior of multi-objective optimization algorithms (Paquete and Stützle, 2006). One of the first studies on the distribution of local optima for the multi-objective traveling salesman problem is due to Borges and Hansen (1998), which shows the existence of a global convexity under a common neighborhood structure while covering the whole Pareto front by varying the scalarizing function parameters. Similarly, Paquete and Stützle (2009) have shown that non-dominated solutions are strongly clustered with respect to the same neighborhood for the same problem class, while the degree of clustering highly depends on the instance structure for the multi-objective quadratic assignment problem. Knowles and Corne (2003) have proposed and investigated multiple fitness landscape features in order

to distinguish the degree of difficulty of multi-objective quadratic assignment problems. As well, tools from fitness landscape analysis have been reviewed and adapted to multi-objective optimization by Garrett and Dasgupta (2007) in terms of scalarizing local optima. In another study, Garrett and Dasgupta (2009) have proposed to visualize a multi-objective fitness landscape as a neutral landscape divided into different fronts containing solutions within the same dominance rank. Aguirre and Tanaka (2007) have defined several problem features such as the number of fronts, the number of solutions within each front, the probability to pass from one front to another, and the hypervolume of the (exact) Pareto set, and related them with the design of EMO algorithms on enumerable multi-objective NK-landscapes. At last, the impact of the problem dimension, the degree of non-linearity, the number of objectives, and the objective correlation of multi-objective NK-landscapes has been related to the structure of the Pareto set and to the number of Pareto local optima by Verel et al. (2013).

Overall, previous work on multi-objective fitness landscapes has often investigated one characteristic at a time, and rarely related problem and fitness landscape features to algorithm performance. Furthermore, we are not aware of any research on feature-based statistical or machine learning modeling aiming at estimating and analyzing the performance of EMO algorithms.

1.3 Contributions

This paper attempts to bridge the gap between a fully theoretical work on runtime analysis and a more practical work on the performance analysis of multi-objective randomized search heuristics. By introducing new features, by explicitly defining features from the literature on multi-objective fitness landscape analysis, and by considering multiple problem and fitness landscape features simultaneously, a fundamental general-purpose statistical framework is proposed to better understand the difficulties that EMO algorithms might have to face. To the best of our knowledge, such a feature-based performance analysis is novel in the context of multi-objective optimization, in particular in that is uses a multi-level mixed-effects linear regression to model the performance of EMO algorithms. It is our hope that a systematic and thorough empirical study could bring valuable meta-knowledge to the practitioner and to the algorithm designer. The research questions motivating and guiding the paper are as follows:

Question #1: *What features might characterize multi-objective combinatorial landscapes?*

Question #2: *How do features relate to benchmark parameters? How do they relate to one another? Are they linearly dependent?*

Question #3: *Which features are ordinally associated with algorithm performance?*

Question #4: *How much of algorithm performance variance can features explain?*

Question #5: *What is the conditional impact of each feature on algorithm performance? What are the significant common trends across instance groups?*

Question #6: *Which features are relevant predictors of algorithm performance?*

Question #7: *Does the impact of features on algorithm performance change with landscape ruggedness? Can ruggedness be used to explain changes across instance groups?*

In order to address these issues, we first identify a substantial number of existing and original problem properties and fitness landscape features for black-box multi-objective combinatorial optimization. They include benchmark instance parameters, such as variable correlation, objective correlation, and objective space dimension, as

well as problem features from the Pareto set, the Pareto graph and the ruggedness and multi-modality of the fitness landscape. We report all these measures, together with a correlation analysis between them, on a large number of enumerable multi-objective NK-landscapes with objective correlation, i.e. ρMNK-landscapes (Verel et al., 2013). As in the single-objective case, the model of NK-landscapes allows one to describe and generalize a large family of unconstrained multi-modal 0–1 optimization problems (Heckendorn and Whitley, 1997).

Next, we propose a general methodology to investigate the impact of problem features on the performance of multi-objective randomized search heuristics. More particularly, we analyze how strongly those proposed features are associated with algorithm performance, how the algorithm performance changes when varying each problem feature, and what is the relative importance of features in explaining the performance variance. To this end, we conduct an experimental analysis on the performance of two prototypical dominance-based EMO algorithms, namely the global simple EMO optimizer (GSEMO) (Laumanns et al., 2004) and the Pareto local search (PLS) algorithm (Paquete et al., 2004), of which we measure the estimated runtime to find a $(1+\varepsilon)-$approximation of the Pareto set. Overall, the runtime of both approaches is impacted by each of the identified multi-objective problem features, and particularly by the ruggedness and the multi-modality of the fitness landscape, and by the hypervolume value of the optimal Pareto set. Our study shows the relative influence of problem features on algorithm efficiency as well as the differences and similarities between both algorithms. As such, the emphasis of the present paper is more on making inferences, see e.g., Chiarandini and Goegebeur (2010), rather than making predictions, see e.g., Hutter et al. (2014). That is, we are concerned with modeling the empirical data to test hypothesis in the context of an appropriate statistical model. Hence, we value model interpretability over predictive power as long as model assumptions are acceptable. On these lines, we attempt to provide both insightful understandings and methodological suggestions about the performance analysis of multi-objective optimization algorithms.

1.4 Outline

The remainder of the paper is organized as follows. In Section 2, we detail the background information about fitness landscape analysis, multi-objective optimization, ρMNK-landscapes, EMO algorithms and their rating of performance. In Section 3, we analyze the empirical impact of ρMNK-landscape benchmark parameters on the performance of GSEMO and PLS. In Section 4, we identify relevant problem features, and report quantitative results and a correlation analysis for ρMNK-landscapes (Questions #1–2). In Section 5, we conduct an association and regression analysis in order to point out how problem features influence the performance of EMO algorithms (Questions #3–7). In Section 6, we conclude and suggest further research into feature-based performance analysis in EMO.

2 Preliminaries

In this section, we give a brief methodological context and the relevant definitions about the multi-objective combinatorial optimization problem under study, the EMO algorithms applied to it, and our performance measure of choice.

2.1 Fitness Landscape Analysis

In single-objective optimization, fitness landscape analysis allows one to study the topology of a combinatorial optimization problem by gathering important informa-

tion such as ruggedness or multi-modality (Weinberger, 1990; Merz, 2004). A fitness landscape is defined by a triplet $(X, \mathcal{N}, \phi)$, where X is a set of admissible solutions (the solution space), $\mathcal{N} : X \to 2^X$ is a neighborhood relation, and $\phi : X \to \mathbb{R}$ is a (scalar) black-box fitness function, here assumed to be maximized. A *walk* over the fitness landscape is an ordered sequence $\langle x_0, x_1, \ldots, x_\ell \rangle$ of solutions from the solution space such that $x_0 \in X$, and $x_t \in \mathcal{N}(x_{t-1})$ for all $t \in \{1, \ldots, \ell\}$.

An *adaptive walk* is a walk such that for all $t \in \{1, \ldots, \ell\}$, $\phi(x_t) > \phi(x_{t-1})$, as performed by a conventional hill-climbing algorithm. The number of iterations, or steps, of the hill-climbing algorithm defines the length of the adaptive walk. This length is an estimator of the diameter of local optima's basins of attraction, characterizing a problem instance multi-modality. Roughly speaking and assuming isotropy in the search space, the longer the length of adaptive walks, the larger the basins size, the lower the number of local optima. This allows us to estimate their number when the whole solution space cannot be enumerated exhaustively.

Let $\langle x_0, x_1, \ldots \rangle$ be an infinite *random walk* over the solution space. The autocorrelation function and the correlation length of such a random walk allow one to measure the ruggedness of a fitness landscape (Weinberger, 1990). The random walk autocorrelation function $r : \mathbb{N} \to \mathbb{R}$ of a (scalar) fitness function ϕ is defined as follows:

$$r(k) = \frac{\mathbb{E}[\phi(x_t) \cdot \phi(x_{t+k})] - \mathbb{E}[\phi(x_t)] \cdot \mathbb{E}[\phi(x_{t+k})]}{\mathrm{Var}(\phi(x_t))}$$

where $\mathbb{E}[\phi(x_t)]$ and $\mathrm{Var}(\phi(x_t))$ are the expected value and the variance of $\phi(x_t)$, respectively. The autocorrelation coefficients $r(k)$ can be estimated within a finite random walk $\langle x_0, x_1, \ldots, x_\ell \rangle$ of length ℓ:

$$\hat{r}(k) = \frac{\sum_{t=1}^{\ell-k} (\phi(x_t) - \bar{\phi}) \cdot (\phi(x_{t+k}) - \bar{\phi})}{\sum_{t=1}^{\ell} (\phi(x_t) - \bar{\phi})^2}$$

where $\bar{\phi} = \frac{1}{\ell} \sum_{t=1}^{\ell} \phi(x_t)$, and $\ell \gg 0$. The longer the length of the random walk ℓ, the better the estimation. The correlation length τ measures how the autocorrelation function decreases. This characterizes the ruggedness of the landscape: the larger the correlation length, the smoother the landscape. Following Weinberger (1990), we define the correlation length by $\tau = -\frac{1}{\ln(r(1))}$. This is based on the assumption that the autocorrelation function decreases exponentially.

2.2 Multi-objective Optimization

We are interested in maximizing a black-box objective function vector $f : X \to Z$, which maps any solution from the solution space $x \in X$ to a vector in the objective space $z \in Z$, with $Z \subseteq \mathbb{R}^M$, such that $z = f(x)$. We assume that the solution space is a discrete set $X = \{0, 1\}^N$, where N is the problem size, i.e. the number of binary (zero-one) variables. An objective vector $z \in Z$ is dominated by an objective vector $z' \in Z$, denoted by $z \prec z'$, iff $\forall i \in \{1, \ldots, M\}$ $z_i \leqslant z'_i$, and there is a $j \in \{1, \ldots, M\}$ such that $z_j < z'_j$. Similarly, a solution $x \in X$ is dominated by a solution $x' \in X$ iff $f(x) \prec f(x')$. An objective vector $z^\star \in Z$ is non-dominated if there does not exist any objective vector $z \in Z$ such that $z^\star \prec z$. A solution $x^\star \in X$ is non-dominated, or Pareto-optimal, if $f(x)$ is non-dominated. The set of Pareto-optimal solutions is the Pareto set (PS) ; its mapping in the objective space is the Pareto front (PF). The goal of multi-objective optimization is to identify the Pareto set/front, or a good approximation of it for large-size and difficult problems.

2.3 ρMNK-Landscapes

The family of ρMNK-landscapes constitutes a synthetic problem model used for constructing tunable multi-objective multi-modal landscapes with objective correlation (Verel et al., 2013). They extend single-objective NK-landscapes (Kauffman, 1993) and multi-objective NK-landscapes with independent objective functions (Aguirre and Tanaka, 2007). Candidate solutions are binary strings of size N, i.e. the solution space is $X = \{0,1\}^N$. The objective function vector $f = (f_1, \ldots, f_i, \ldots, f_M)$ is defined as $f : \{0,1\}^N \rightarrow [0,1]^M$ such that each objective function f_i is to be maximized. As in the single-objective case, each separate objective function value $f_i(x)$ of a solution $x = (x_1, \ldots, x_j, \ldots, x_N)$ is an average value of the individual contributions associated with each variable x_j. Indeed, for each objective f_i, $i \in \{1, \ldots, M\}$, and each variable x_j, $j \in \{1, \ldots, N\}$, a component function $f_{ij} : \{0,1\}^{K+1} \rightarrow [0,1]$ assigns a real-valued contribution to every combination of x_j and its K *epistatic interactions* $\{x_{j_1}, \ldots, x_{j_K}\}$. These f_{ij}-values are uniformly distributed in the range $[0,1]$. Thus, the individual contribution of a variable x_j depends on its value and on the values of $K < N$ other variables $\{x_{j_1}, \ldots, x_{j_K}\}$. The problem can be formalized as follows:

$$\begin{array}{lll} \max & f_i(x) = \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} f_{ij}(x_j, x_{j_1}, \ldots, x_{j_K}) & i \in \{1, \ldots, M\} \\ \text{s.t.} & x_j \in \{0,1\} & j \in \{1, \ldots, N\}\,. \end{array}$$

In this work, the epistatic interactions, i.e. the K variables that influence the contribution of x_j, are set uniformly at random among the $(N-1)$ variables other than x_j, following the random neighborhood model from Kauffman (1993). By increasing the number of epistatic interactions K from 0 to $(N-1)$, problem instances can be gradually tuned from smooth to rugged. In ρMNK-landscapes, f_{ij}-values additionally follow a multi-variate uniform distribution of dimension M, defined by an $M \times M$ positive-definite symmetric covariance matrix (c_{pq}) such that $c_{pp} = 1$ and $c_{pq} = \rho$ for all $p, q \in \{1, \ldots, M\}$ with $p \neq q$, where $\rho > \frac{-1}{M-1}$ defines the correlation among the objectives; see Verel et al. (2013) for details. The positive (respectively, negative) objectives correlation ρ decreases (respectively, increases) the degree of conflict between the different objective function values. The correlation coefficient ρ is the same between all pairs of objectives, and the same epistatic degree K and epistatic interactions are set for all the objectives.

2.4 Multi-objective Randomized Search Heuristics

In this paper, we consider two randomized search heuristics: (i) Global SEMO (GSEMO) proposed by Laumanns et al. (2004), a simple elitist steady-state *global* EMO algorithm (see Algorithm 1); and (ii) Pareto local search (PLS) proposed by Paquete et al. (2004), a multi-objective *local* search (Algorithm 2). These algorithms extend to the multi-objective case two conventional search heuristics, namely the $(1+1)-$evolutionary algorithm and the hill-climbing local search algorithm, which are often investigated in the theoretical literature on single-objective optimization. Though their design is guided by simple heuristic rules, these algorithms are components of many state-of-the-art multi-objective combinatorial optimization approaches (Andersen et al., 1996; Paquete et al., 2004; Paquete and Stützle, 2006; Liefooghe et al., 2013a), and their search dynamics typically reveal the complex behavior that we aim to further understand in this paper. Both algorithms maintain an *unbounded* archive A of mutually non-dominated solutions. This archive is initialized with one random solution

Algorithm 1: GSEMO

Choose an initial solution x_0 uniformly from X;
$A \leftarrow \{x_0\}$;
repeat
  Select an element x out of A uniformly;
  Create x' by flipping each bit of x with probability $1/N$;
  $A \leftarrow$ non-dominated solutions from $A \cup \{x'\}$;
until *success* $\vee$ *maxeval*;

Algorithm 2: PLS

Choose an initial solution x_0 uniformly from X;
$A \leftarrow \{x_0\}$;
repeat
  Select a non-visited element x out of A uniformly;
  Create $\mathcal{N}(x)$ by flipping each bit of x in turns;
  Flag x as visited;
  $A \leftarrow$ non-dominated solutions from $A \cup \mathcal{N}(x)$;
until *all-visited* $\vee$ *success* $\vee$ *maxeval*;

from the solution space. Then, at each iteration, one solution is selected at random from the archive, $x \in A$. In GSEMO, each binary variable from x is independently flipped with rate $\frac{1}{N}$ in order to produce an offspring solution x'. The archive is then updated by keeping the non-dominated solutions from $A \cup \{x'\}$. In PLS, the solutions located in the neighborhood of x are evaluated. Let $\mathcal{N}(x)$ denote the set of solutions located at a Hamming distance 1. The non-dominated solutions from $A \cup \mathcal{N}(x)$ are stored in the archive, and the current solution x is then tagged as *visited* in order to avoid a useless revaluation of its neighborhood. This process is iterated until a stopping condition is satisfied. While for GSEMO there does not exist any explicit stopping rule (Laumanns et al., 2004), PLS has a natural stopping condition which is satisfied when all the solutions in the archive are tagged as *visited*.

In other words, while PLS is based on the exploration of the whole 1-bit-flip neighborhood from x, GSEMO rather uses an ergodic operator, i.e. an independent bit-flip mutation. This means that there is a non-zero probability of reaching any solution from the solution space at every GSEMO iteration. This makes GSEMO a *global* optimizer rather than a *local* optimizer as PLS. In this paper, we are interested in the runtime, in terms of a number of function evaluations, until a $(1+\varepsilon)-$approximation of the Pareto set is identified and is contained in the internal memory A of the algorithm, subject to a maximum budget of function evaluations.

2.5 Estimated Runtime (`ert`)

Let ε be a constant value such that $\varepsilon \geqslant 0$. The (multiplicative) ε-dominance relation ($\preceq_\varepsilon$) can be defined as follows (Laumanns et al., 2002). For $x, x' \in X$, x is ε-dominated by x' ($x \preceq_\varepsilon x'$) iff $f_i(x) \leqslant (1+\varepsilon) \cdot f_i(x')$, $\forall i \in \{1, \ldots, M\}$. The ε-value then stands for a relative tolerance that we allow within objective values. A set $X^\varepsilon \subseteq X$ is a $(1+\varepsilon)-$approximation of the Pareto set if for any solution $x \in X$, there is one solution $x' \in X^\varepsilon$ such that $x \preceq_\varepsilon x'$. This is equivalent to finding an approximation set

whose multiplicative epsilon quality indicator value with respect to the (exact) Pareto set is lower than $(1+\varepsilon)$, see e.g., (Zitzler et al., 2003). Interestingly, under some general assumptions, there always exists a $(1+\varepsilon)$-approximation, for any given $\varepsilon \geqslant 0$, whose cardinality is both polynomial in the problem size and in $\frac{1}{\varepsilon}$ (Papadimitriou and Yannakakis, 2000).

Following a conventional methodology from single-objective continuous black-box optimization (Auger and Hansen, 2005), we measure algorithm performance in the expected number of function evaluations to identify a $(1+\varepsilon)-$approximation. However, as any heuristic, GSEMO or PLS can either succeed or fail to reach an accuracy of ε in a single run. In case of success, we record the number of function evaluations until a $(1+\varepsilon)-$approximation was found. In case of failure, we simply *restart* the algorithm at random. Thus we obtain a "*simulated runtime*" (Auger and Hansen, 2005) from a set of independent trials on each instance. Such performance measure allows us to take into account both the success rate $p_s \in (0,1]$ and the convergence speed of the algorithm with restarts. Precisely, after $(t-1)$ failures, each one requiring T_f evaluations, and the final successful run of T_s evaluations, the total runtime is $T = \sum_{i=1}^{t-1} T_f + T_s$. By taking the expectation and by considering independent trials as a Bernoulli process stopping at the first success, we have:

$$\mathbb{E}[T] = \left(\frac{1-p_s}{p_s}\right)\mathbb{E}[T_f] + \mathbb{E}[T_s]$$

In our case, the success rate p_s is estimated with the ratio of successful runs over the total number of executions ($\hat{p}_s$), the expected runtime for unsuccessful runs $\mathbb{E}[T_f]$ is set as a constant limit on the number of function evaluation calls T_{max}, and the expected runtime for successful runs $\mathbb{E}[T_s]$ is estimated with the average number of function evaluations performed by successful runs:

$$\texttt{ert} = \left(\frac{1-\hat{p}_s}{\hat{p}_s}\right) T_{max} + \frac{1}{t_s}\sum_{i=1}^{t_s} T_i$$

where t_s is the number of successful runs, and T_i is the number of evaluations for successful run i. For more details, we refer to Auger and Hansen (2005).

3 Experimental Analysis

In this section, we examine the performance of GSEMO and PLS depending on the epistatis, the objective space dimension, and the objective correlation of ρMNK-landscapes.

3.1 Experimental Setup

As problem instances, we consider ρMNK-landscapes with an epistatic degree $K \in \{2,4,6,8,10\}$, an objective space dimension $M \in \{2,3,5\}$, and an objective correlation $\rho \in \{-0.9,-0.7,-0.4,-0.2,0.0,0.2,0.4,0.7,0.9\}$, such that $\rho > \frac{-1}{M-1}$. This restriction on ρ-values comes from the fact that the contributions of objective components $f_{ij} : \{0,1\}^{K+1} \to [0,1]$ (see section 2.3), are sampled from a multi-variate normal distribution whose covariance matrix has to be symmetric and positive-definite (Verel et al., 2013). The problem size is set to $N = 18$ in order to enumerate the solution space exhaustively. The solution space size is then $|X| = 2^{18}$. A set of 30 different landscapes are independently generated at random for each parameter combination ρ, M, and K, for a total of 3 300 instances. They are made available at `http://mocobench.sf.net`.

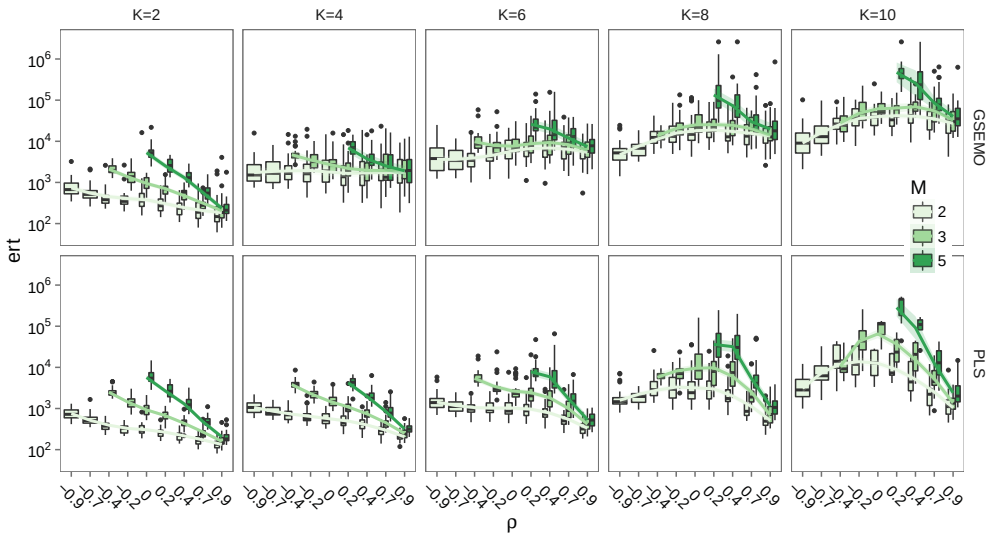

Figure 1: Distribution of the estimated runtime ert (y-axis, fixed log-scale) w.r.t. objective correlation ρ (x-axis) for both algorithms (see right labels). Results are grouped by problem non-linearity K (see top labels) and by number of objectives M (see legend). Box-and-whisker plots give median and inter-quantile range; LOESS smooth curves show trends.

The target tolerance is set at $\varepsilon = 0.1$. The time limit is set to $T_{max} = 2^N \cdot 10^{-1} < 26\,215$ function evaluations without identifying a $(1+\varepsilon)-$approximation. Each algorithm is executed 100 times *per* instance. From these 100 repetitions, the success rate and the expected number of evaluations for successful runs, hence the estimated runtime on the given instance, are computed. For the comparative analysis, we only consider pairwise-complete cases, i.e. instances that have been solved by both algorithms. This brings the total number of available observations to $2\,874$ *per* algorithm.

The algorithms have been implemented in C++ within the Paradiseo software framework (Liefooghe et al., 2011), and the statistical analysis has been performed with R (R Core Team, 2015).

3.2 Exploratory Analysis

The estimated runtime (`ert`) distribution across the experimental blocks that are defined by each combination of benchmark parameters is presented in Figure 1. For both algorithms, the `ert` clearly increases with the non-linearity (K) and the number of objectives (M), whereas the trend w.r.t. the objective correlation (ρ) is a bit more complex. Indeed, for a small K and a large M, the `ert` decreases when ρ increases. On the contrary, for large K, problem instances seem to get harder when the objectives are independent ($\rho \approx 0$) rather than anti-correlated ($\rho < 0$). This shows in the inverted u-shape observed on the right side of the figure, which is particularly pronounced for PLS. This is surprising because the cardinality of the Pareto set increases when objectives are conflicting (Verel et al., 2013). However, we observed that the ε-value of random approximation sets follows a similar u-shaped trend w.r.t. objective correlation. Also, this ε-value tends to increase with K. This holds for approximation sets

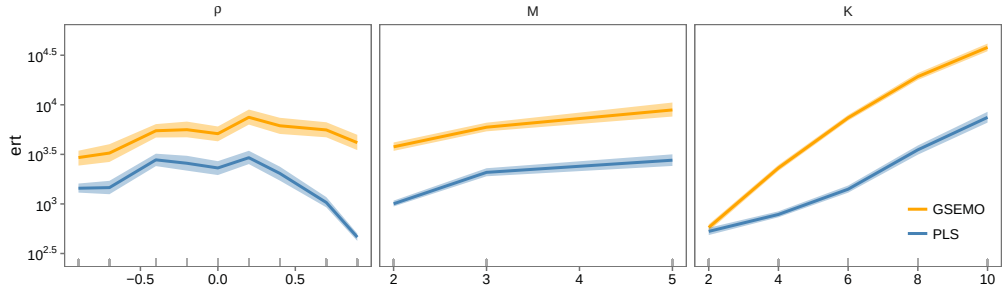

Figure 2: Interaction plots between the average estimated runtime ert (y-axis, log-scale) and the benchmark parameters, i.e. the objective correlation ρ (left), the number of objectives M (center), and the problem non-linearity K (right, see titles). Results are grouped by algorithm (see legend). The average value (lines) and the 0.95 confidence interval (shaded areas) are evaluated through bootstrapping.

containing a constant number of randomly generated solutions. Moreover, we know from Verel et al. (2013) that the number of Pareto local optimal solutions increases with K and decreases with ρ. This could explain the relative advantage of PLS on problem instances with positively correlated objectives. Notice also that the opposite is true for the connectedness of the Pareto set: the smaller K and the larger ρ, the more clustered Pareto optimal solutions are in the solution space.

Figure 2 displays the runtime aggregated over all the instance parameters (ρ, M, K) but one. We clearly see that PLS is significantly outperforming GSEMO overall, and in particular for positively correlated objectives. In fact, the runtime of PLS is shorter than that of GSEMO in 88% of the instances. Compared against PLS, GSEMO requires more than $17\,000$ additional function evaluations on average to identify a $1.1-$approximation of the Pareto set. The performance difference between the two algorithms seems to be constant, except for large ρ and w.r.t. K. Notably, the ruggedness of the underlying single-objective objective functions appears to have the highest impact on the search performance. In particular, the ruggedness seems to have more impact on the performance of GSEMO than PLS. In general, finding a $(1+\varepsilon)-$approximation becomes harder as the number of objectives grows and much harder for highly-rugged instances, whereas the trend w.r.t. objective correlation is less clear, more algorithm-dependent.

In the following, we list the problem features that *intuitively* impact the performance of randomized search heuristics for the class of ρMNK-landscapes, and we explicitly assess their separate and joint effect on the runtime of PLS and GSEMO.

4 Features Characterizing Problem Difficulty

Question #1: *What features might characterize multi-objective combinatorial landscapes?*

In this section, we identify a number of general-purpose features, either directly extracted from the problem instance, or computed from the fitness landscape. Then, we conduct a correlation analysis of feature pairs, showing how features relate to benchmark parameters and anticipating the interplay of those features in capturing the difficulties of a problem instance.

4.1 Benchmark Parameters

First, we consider the following parameters related to the definition of ρMNK-landscapes. Recall that in this analysis the problem size is kept constant to $N = 18$.

- **Number of variable interactions (K):** This gives the number of variable correlations in the construction of ρMNK-landscapes. As it will be detailed later, although the value of K cannot be retrieved directly from a black-box instance, it can be precisely estimated by some of the problem features described below.

- **Number of objective functions (M):** This parameter represents the dimension of the objective space in the definition of ρMNK-landscapes.

- **Correlation between the objective function values (ρ):** This parameter allows us to tune the correlation between the objective function values in ρMNK-landscapes. In our analysis, the objective correlation is the same between all pairs of objectives.

4.2 Problem Features from the Pareto Set

The fitness landscape features considered in our analysis are described below. We start with some general features related to the Pareto set.

- **Number of Pareto optimal solutions (`npo`):** The number of Pareto optimal solutions enumerated in the instance under consideration simply corresponds to the cardinality of the (exact) Pareto set, i.e. $\texttt{npo} = |\text{PS}|$. The approximation set manipulated by any EMO algorithm is directly related to the cardinality of the Pareto optimal set. For ρMNK-landscapes, the number of Pareto optimal solutions typically grows exponentially with the problem size, the number of objectives and with the degree of conflict between the objectives (Verel et al., 2013).

- **Hypervolume of the Pareto set (`hv`):** The hypervolume value gives the portion of the objective space that is dominated by the Pareto set (Zitzler et al., 2003). We take the origin as a reference point $z^\star = (0.0, \ldots, 0.0)$.

- **Average distance between Pareto optimal solutions (`avgd`):** This metric corresponds to the average distance, in terms of Hamming distance, between any pair of Pareto optimal solutions.

- **Maximum distance between Pareto optimal solutions (`maxd`):** This metric is the maximum distance between two Pareto optimal solutions in terms of Hamming distance. This feature is denoted as the diameter of the Pareto set by Knowles and Corne (2003).

- **Proportion of supported solutions (`supp`):** Supported solutions are Pareto optimal solutions whose corresponding objective vectors are located on the convex hull of the Pareto front. Notably, non-supported solutions are *not* optimal with respect to a weighted-sum aggregation of the objective functions, whatever the setting of the (positive) weighting coefficient vector. As a consequence, the proportion of supported solutions on the Pareto set has a direct impact on the ability of scalar approaches to find a proper Pareto set approximation. However, this feature is expected to have a low impact on the performance of dominance-based EMO approaches like GSEMO and PLS.

4.3 Problem Features from the Pareto Graph

In the following, we describe some problem features related to the *connectedness* of the Pareto set (Ehrgott and Klamroth, 1997; Gorski et al., 2011). If all Pareto optimal solutions are connected with respect to a given neighborhood structure, the Pareto set is said to be *connected*, and local search algorithms would be able to identify all Pareto optimal solutions by starting with at least one of them; see e.g., Andersen et al. (1996); Paquete et al. (2008); Paquete and Stützle (2009); Liefooghe et al. (2013a). We follow the definition of *d-Pareto graph* from Paquete et al. (2008). The d-Pareto graph is defined as a graph $PG_d = (V, E)$, where the set of vertices V contains all Pareto optimal solutions, and there is an edge $e_{ij} \in E$ between two nodes i and j if and only if the shortest distance between solutions x_i and $x_j \in X$ is below a bound d, i.e. $d(x_i, x_j) \leqslant d$. The distance $d(x_i, x_j)$ is taken as the Hamming distance for ρMNK-landscapes. This corresponds to the *bit-flip* neighborhood operator. The connectedness-related problem features under investigation are given below.

- **Relative number of connected components (`ncomp`):** This metric is the number of connected components in the 1-Pareto graph ($PG_{d=1}$), normalized by the number of Pareto optimal solutions.
- **Proportional size of the largest connected component (`lcomp`):** This corresponds to the proportion of Pareto optimal solutions that belong to the largest connected component in the 1-Pareto graph $PG_{d=1}$.
- **Minimum distance to connect the Pareto graph (`dconn`):** This measure corresponds to the smallest distance d such that the d-Pareto graph is connected, i.e. for all pairs of vertices $(x_i, x_j) \in V^2$ in PG_d, there exists a path between x_i and x_j.

4.4 Problem Features from Ruggedness and Multi-modality

At last, we consider problem features related to the number of local optima, the length of adaptive walks, and the autocorrelation functions.

- **Number of Pareto local optima (`nplo`):** A solution $x \in X$ is a *Pareto local optimum* with respect to a neighborhood structure $\mathcal{N}$ if there does not exist any neighboring solution $x' \in \mathcal{N}(x)$ such that $x \prec x'$; see e.g., Paquete et al. (2007). For ρMNK-landscapes, the neighborhood structure is taken as the *1-bit-flip*, which is directly related to a Hamming distance of 1. This metric reports the number of Pareto local optima enumerated on the ρMNK-landscape under consideration.
- **Length of a Pareto-based adaptive walk (`ladapt`):** We compute here the length of adaptive walks by means of a very basic single solution-based *Pareto-based Hill-Climbing* (PHC) algorithm. The PHC algorithm is initialized with a random solution. At each iteration, the current solution is replaced by a random dominating neighboring solution. As a consequence, PHC stops on a Pareto local optimum. The number of iterations, or steps, of the PHC algorithm is the length of the Pareto-based adaptive walk. As in the single-objective case, the number of Pareto local optima is expected to increase exponentially when the adaptive length decreases for ρMNK-landscapes (Verel et al., 2013).
- **First autocorrelation coefficient of solution hypervolume (`corhv`):** The ruggedness is measured here in terms of the autocorrelation of the hypervolume along a random walk. As explained in Section 2.1, the correlation length τ measures how

Table 1: Summary of ρMNK-landscape benchmark parameters and problem instance features investigated in the paper.

	Benchmark parameters (3)	
ρ	Correlation between the objective function values	
M	Number of objective functions	
K	Number of variable interactions (epistatis)	
	Problem features (12)	
`npo`	Number of Pareto optimal solutions	(Knowles and Corne, 2003)
`hv`	Hypervolume (Zitzler et al., 2003) of the Pareto set	(Aguirre and Tanaka, 2007)
`avgd`	Average distance between Pareto optimal solutions	(Liefooghe et al., 2013b)
`maxd`	Maximum distance between Pareto optimal solutions	(Knowles and Corne, 2003)
`supp`	Proportion of supported solutions in the Pareto set	(Knowles and Corne, 2003)
`nplo`	Number of Pareto local optima	(Paquete et al., 2007)
`ladapt`	Length of a Pareto-based adaptive walk	(Verel et al., 2013)
`ncomp`	Relative number of connected components	(Paquete and Stützle, 2009)
`lcomp`	Proportional size of the largest connected component	(Verel et al., 2011)
`dconn`	Minimal distance to connect the Pareto graph	(Paquete and Stützle, 2009)
`corhv`	First autocorrelation coefficient of solution hypervolume	(Liefooghe et al., 2013b)
`corlhv`	First autocorrelation coefficient of local hypervolume	(Liefooghe et al., 2013b)

the autocorrelation function, estimated with a random walk, decreases. The autocorrelation coefficients are computed here with the following scalar fitness function $\phi : X \rightarrow \mathbb{R}$: $\phi(x) = \texttt{hv}(\{x\})$, where $\texttt{hv}(\{x\})$ is the hypervolume of solution $x \in X$, the reference point being set to the origin. The random walk length is set to $\ell = 10^4$, and the neighborhood operator is the *1-bit-flip*.

- **First autocorrelation coefficient of local hypervolume (`corlhv`):** This metric is similar to the previous one, except that the fitness function is based here on a local hypervolume measure. The local hypervolume is the portion of the objective space covered by non-dominated neighboring solutions, i.e. for all $x \in X$, $\phi(x) = \texttt{hv}(\mathcal{N}(x) \cup \{x\})$. Similarly to `corhv`, the random walk length is set to $\ell = 10^4$, and the neighborhood operator $\mathcal{N}$ is the *1-bit-flip*.

The benchmark parameters defining ρMNK-landscapes and a total of twelve general-purpose problem features are summarized in Table 1. Notice that most of those features require the solution space to be enumerated exhaustively, with the exception of benchmark parameters as well as `ladapt`, `corhv` and `corlhv`. For this reason, they are not practical for a performance prediction purpose. However, we decided to include them because our aim is to examine their impact on the algorithm performance.

4.5 Correlation between Problem Features

Question #2: *How do features relate to benchmark parameters? How do they relate to one another? Are they linearly dependent?*

The correlation matrix between each pair of features is reported in Figure 3. In view of fitting linear models in the next stage of our analysis, we use here the Pearson linear

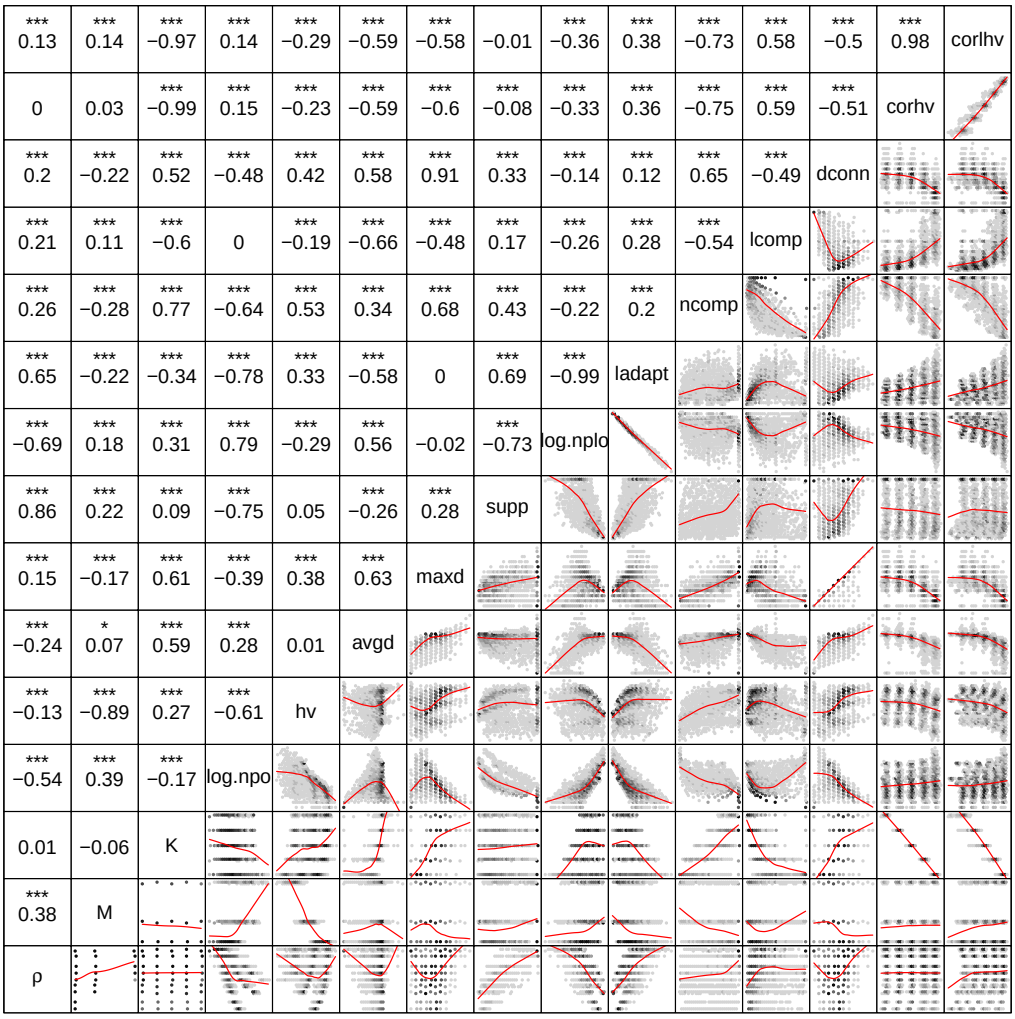

Figure 3: Correlation matrix between all pairs of features. The feature names are reported on the diagonal. For each pair of features, scatter plots and smoothing splines are displayed below the diagonal, and the corresponding linear correlation coefficients are reported above the diagonal. In the upper panel, the correlation coefficient is tested against the null hypothesis of zero correlation. The resulting Bonferroni-corrected $p-$value is symbolically encoded at the levels 0.05 (*), 0.01 (**), and 0.001 (***).

correlation coefficient. First, the number of objectives M is moderately correlated with the cardinality of the Pareto set $\log(\texttt{npo})$. So is the objective correlation ρ (the absolute correlation coefficient is around 0.5 in both cases). Surprisingly, none of these features alone (M or ρ) can explain the number of non-dominated solutions. This means that the objective space dimension does not justify by itself the large amount of non-dominated solutions found in many-objective optimization problems (Wagner et al., 2007). As pointed out by Verel et al. (2013), the degree of conflict between the objective function

values has also to be taken into account. In fact, a multi-linear regression to predict $\log(\texttt{npo})$ based on both M and ρ can explain 70% of the $\log(\texttt{npo})$ variance, with a high correlation coefficient (0.84) between measured and fitted values. The regression coefficients show that the number of Pareto optimal solutions increases with the number of objectives and decreases with the objective correlation. As the combination of M and ρ allows one to predict a large part of the (log-transformed) Pareto set cardinality variance, we believe that, more generally, the impact of many-objective fitness landscapes on the search process cannot be analyzed properly without taking the objective correlation into account. Furthermore, the number of objectives M is also highly correlated with the hypervolume `hv` of the Pareto set (the absolute correlation coefficient is 0.89), whereas the features having the highest absolute correlation with ρ are the fraction of supported solutions `supp` and the number of Pareto local optima $\log(\texttt{nplo})$ (the correlation coefficients are 0.86 and -0.69, respectively).

Problem features from the Pareto graph aim to characterize the topology of Pareto optimal solutions. The relative number of connected components `ncomp` is positively correlated with the minimal distance to connect all the components `dconn` (the correlation coefficient is 0.65), whereas the relative size of the largest component `lcomp` is negatively correlated with the number of components (-0.54). The minimal distance is also highly correlated with the maximal Hamming distance between Pareto optimal solutions `maxd` (over 0.9). Connectedness metrics appear to be mostly more correlated with benchmark parameter K. For instance, the correlation coefficient between the relative number of components and K is 0.77. The number of Pareto optimal solutions or the hypervolume of the Pareto front are not clearly correlated with Pareto graph features. Indeed, the logarithm of the Pareto set size is negatively correlated with the relative number of connected components (-0.64), but it is not correlated with the size of the largest component.

Several features relate to and can characterize the ruggedness and the multimodality of the fitness landscape. For instance, the number of Pareto optimal solutions `npo` and of Pareto local optima `nplo` are highly correlated (the correlation coefficient of their log-transformed values is 0.79). Not surprisingly, the number of local optima increases with the number of non-dominated solutions. Unfortunately, the number of Pareto local optima cannot be computed without the full enumeration of the set of feasible solutions. Nevertheless, its log-transformed value is highly linearly correlated to the length of a Pareto-based adaptive walk `ladapt` (the absolute correlation coefficient is 0.99). This potentially allows one to estimate the number of Pareto local optima for large-size problem instances; see Verel et al. (2013). On the contrary, the correlation between the number of variable interactions (epistatis) K and the number of Pareto global or local optima is low. Although those important features certainly depend on the benchmark parameter K, that is not a direct linear relation. Other features are required to fully explain the number of optima. On these benchmark instances, the number of epistatic interactions K can be estimated by hypervolume-based autocorrelation measures from random walks `corhv` and `corlhv` (the absolute correlation coefficients are close to 1.0). Indeed, the autocorrelation coefficient is higher when the number of epistatic interactions is low, as in the single-objective case (Weinberger, 1991).

This correlation matrix gives a "big picture" of some of the features that can describe the search space structure of a multi-objective combinatorial optimization problem. In the next section, we relate the value of those features for enumerable ρMNK-landscapes to the performance of both GSEMO and PLS on the same landscapes.

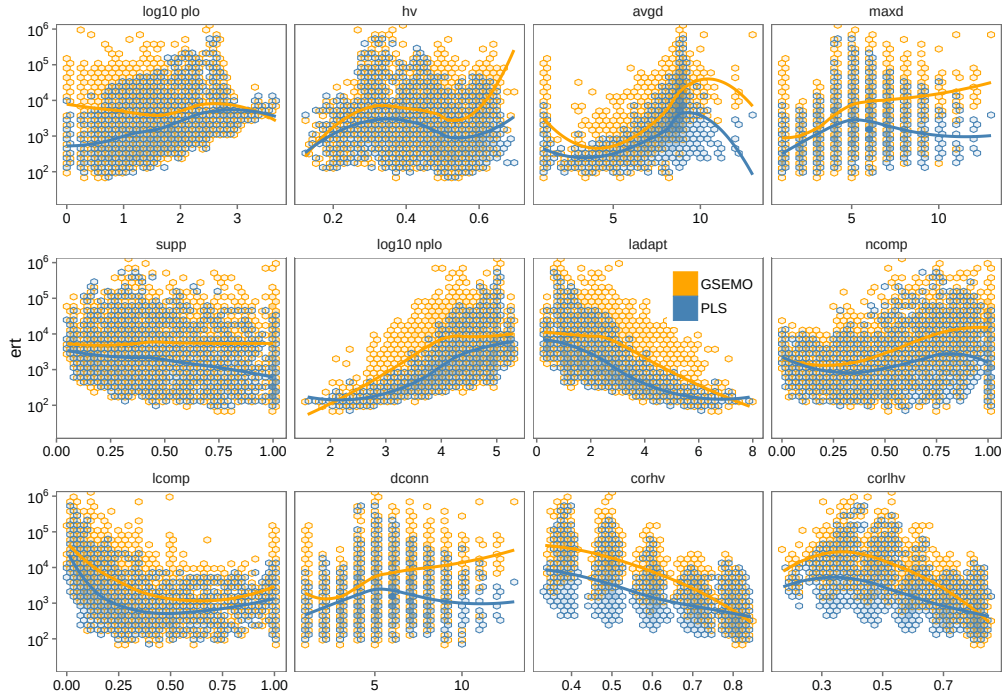

Figure 4: Interaction plots between the average estimated runtime ert (y-axis, fixed log-scale) and the instance features (x-axis, see titles). Results are grouped by algorithm (see legend). Lines represent a locally-fitted polynomial function (LOESS).

5 Feature-based Analysis

In this section we link problem features to the estimated runtime of EMO algorithms. First, we measure how strongly each individual feature is associated to performance via a correlation analysis. Then, aiming to disentangle features contributions and their importance in explaining search performance, we assess their conditional impact on algorithm runtime by means of a multi-level, multi-linear regression model.

5.1 Correlation between Problem Features and Algorithm Performance

Question #3: *Which features are ordinally associated with algorithm performance?*

A first assessment of the dependency of the search performance on instance features can be done through visual inspection of scatter plots, supported by a correlation analysis. Naturally, *correlation* does not imply *causation* and we do not draw any *direct* link between each considered feature and the algorithm runtime, even if in our case the eventual link could only go in one direction. Instead, we restrict ourself to measure the association of each feature to the performance metric (`ert`). We quantify the strength of this dependency via the Kendall's τ statistic (McLeod, 2011), since we want to assess the accordance between the variation in algorithm performance and the variation in problem features. This non-linear rank-correlation measure is based on the ordering of all possible pairs, and its value is proportional to the difference between the number of concordant and discordant pairs among all possible pairwise comparisons. As such,

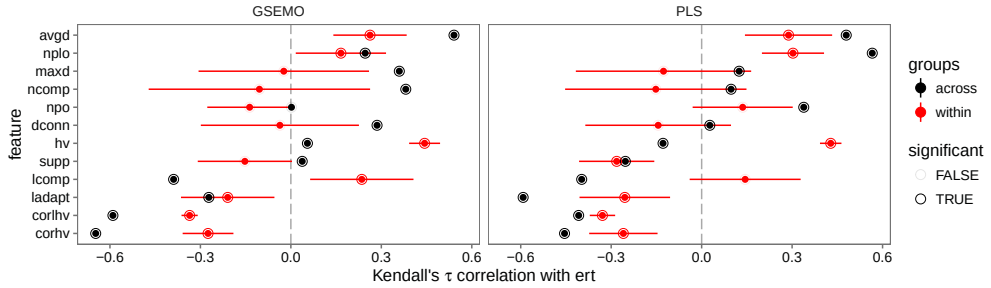

Figure 5: Performance-feature association. Points give Kendall's τ statistic for the correlation between runtime and instance feature, evaluated on the whole set of instances (red) or within instance groups (black, see legend). Group average values, 0.95 confidence intervals, and significance, are estimated with a t-test considering only statistics that were significant at the group level. Group and overall significance are based on Kendall's test $p-$value at the 0.05 level ($H_0 : \tau_i = 0$); see McLeod (2011).

when the null hypothesis of mutual independence ($H_0 : \tau = 0$) is rejected, τ can be directly interpreted as the probability of observing agreement ($\tau > 0$) or disagreement ($\tau < 0$) between the ranks of paired values.

The scatter plots and the regressions (with local polynomial fitting) of the average runtime of both algorithms as a function of the instance features are provided in Figure 4. In addition, Kendall's τ coefficients are given by the red points in Figure 5. For both algorithms, the average distance between Pareto optimal solutions (`avgd`) is highly positively correlated with `ert`: the larger this distance, the longer the runtime. On the contrary, both ruggedness-related features based on measures of hypervolume autocorrelation (`corhv`, `corlhv`), and one feature related to the connectedness, i.e. the size of the largest cluster in the Pareto set (`lcomp`), are highly negatively correlated. As expected, ruggedness and connectedness play a major role for both algorithms: the runtime decreases with `corhv` and `corlhv`, and when a large number of Pareto optimal solutions are connected in the solution space.

Some features have a different impact on the two algorithms, possibly highlighting their respective strengths and weaknesses. In particular, the runtime of PLS increases with the number of Pareto optimal and locally optimal solutions. Contrastingly, the scatter plots show that having a high number of Pareto local optima has less impact on GSEMO than on PLS. Moreover, the runtime of GSEMO is correlated with three other features related to the distance and the connectedness of Pareto optimal solutions (`maxd`, `ncomp`, `dconn`). Indeed, topological relationships between Pareto optimal solutions have a large effect on the runtime of GSEMO, especially when the distance between those solutions is large. Surprisingly, the runtime of PLS does not increase when non-dominated solutions are disconnected.

However, we have to be careful when drawing conclusions by aggregating data from different areas of the instance parameters space, since feature values and their range depend, in turn, on the levels of ρ, M, and K. This can be visually appreciated in Figure 4: the autocorrelation measures `corhv` and `corlhv`, for example, are clearly clustered around five levels that actually correspond to the different K-values. Similarly, we are able to distinguish three clusters in the hypervolume metric `hv`, which actually follow the objective space dimension M.

Therefore, we deepen the analysis by evaluating the correlation *within* the instance groups defined by each possible combination of the $\langle \rho, M, K \rangle$-values under study. Black points and lines in Figure 5 display the average τ-value within groups, together with the confidence interval associated with the mean. By comparing them with red points, we can clearly notice how data aggregation slightly enhances the correlation statistic in the `corhv` and `corlhv` case, leading to the same inference nonetheless. On the contrary, although hypervolume is very weakly associated with runtime overall, and that its impact is contradictory between GSEMO and PLS, group results are more consistent, showing a strong positive association between `ert` and `hv` for both algorithms. Unfortunately, as for features related to the connectedness of the Pareto set, our confidence on the average correlation within groups is too low to make further comments, mainly due to the fact that, in many cases, we could not reject the null hypothesis of mutual independency at the group level. Nevertheless, the previous observations on instance groups and their possible effect motivate the remainder of our analysis.

5.2 Linear Mixed Model

Question #4: *How much of algorithm performance variance can features explain?*

In this section, we aim to quantify the impact of instance features, and possibly disentangle their individual contribution to the performance variance, by taking into account the dependency among measurements that is induced by the experimental plan. Our goal is precisely to generalize from it as much as possible, in order to make inferences about the effect and relative importance of features. Let y_i be the log-transformed estimated runtime (`ert`) on the i-th problem instance. We treat y_i as an observation from a random variable Y with expectation $\mathbb{E}(Y) = \mu$, i.e. $y_i = \mu + \varepsilon_i$ where ε_i are taken to be independent, identically distributed, and zero-mean.

In a classical multi-linear regression, we would model μ as a linear combination of p predictors, notably (a subset of) the p problem instance features that we can measure. Thus, the performance observation on instance i can be written as:

$$y_i = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ki} + \varepsilon_i \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

where ε_i is the usual random term, i.e. the regression residual. In this model, performance observations are supposed to be i.i.d. from a normal distribution $y_i \sim \mathcal{N}(\mu, \sigma^2)$. However, as discussed in the previous sections, our observations are mostly clustered around the different combinations of benchmark parameters; see Figure 1. In fact, a simple linear regression on a dummy categorical predictor having a different level for each combination of ρ, M, and K, would explain 84.51% and 86.85% of the `ert` variance of GSEMO and PLS, respectively.

Since we rather want to investigate the impact of instance features, we need to decompose that global performance variance into what is due to the grouping of benchmark parameters, from which we would like to generalize, and what is due to the randomness involved in the instance generation process; namely for ρMNK-landscapes, the epistatic interaction links and their contributions to the objective values. That conveys the feature variance within the blocks of our experimental design. To this end, instead of fitting an independent regression model for each instance group, we build a linear mixed model with random effects for experimental blocks. In such a framework,

the performance on instance i from group j can then be modeled as:

$$y_{ij} = \beta_0 + \sum_{k=1}^{p} \beta_k x_{kij} + \alpha_j + \varepsilon_{ij} \qquad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

where α_j are i.i.d random variables, with $\alpha_j \sim \mathcal{N}(0, \sigma_\alpha^2)$ denoting the group effect. By doing so, we suppose that features have the same impact across groups; in linear terms, constant (*fixed*) slopes and *random* intercepts. Notice that estimates $\hat{\beta}$ and residuals ε will likely not be the same as in the previous model. Notice also that performance observations are now i.i.d. *conditionally* on the grouping factor $y_{ij}|\alpha_j \sim \mathcal{N}(\mu + \alpha_j, \sigma^2)$, whereas the unconditional model $y_{ij} \sim \mathcal{N}(\mu, \sigma^2 + \sigma_\alpha^2)$ carries a dependency between measurements of the same group. Hereby we also obtain the aforementioned variance decomposition that shows which part of the observed performance variance can be ascribed to the grouping of problem instances (Chiarandini and Goegebeur, 2010).

The usual approach to estimate the parameters of the unconditional model is the restricted maximum likelihood method; see Faraway (2005) and Bates et al. (2014) for theoretical and implementation details. In the following, we present the results of such estimation on the full model comprising all instance features and for both algorithms. In particular, each estimated β_i is tested against the null hypothesis $H_0 : \beta_i = 0$, whereas as for the group effect we only need to check $H_0 : \sigma_\alpha^2 = 0$. In the multi-linear framework, the regression coefficients that are statistically significant allow us to assess the runtime effect of each feature conditionally on all the others and, given the random effect formulation, across experimental blocks.

Finally, in order to assess the accuracy of a regression model, the conventional R^2 (ratio of variance explained by the regression) can be extended by taking into account the variance decomposition that is specific to mixed models (Nakagawa and Schielzeth, 2013). We obtain a *marginal* R^2 yielding the proportion of variance explained by instance features, and a *conditional* R^2 which, despite its name, gives the proportion of variance explained by the entire model, i.e. including the random effect of benchmark parameter combinations. Marginal and conditional R^2 are respectively 0.617 and 0.919 for the regression modeling GSEMO's `ert`, respectively 0.482 and 0.911 for PLS.

A note on multicollinearity. Multi-linear regression modeling rests on few key assumptions, namely the usual normality of residuals and homogeneity of variance, but also on predictors (linear) independence. In fact, linear correlation between two or more predictors (collinearity) may produce unstable parameter estimates with high standard errors. That is all the more problematic when the analysis goal is to determine the individual contribution of each predictor in explaining the response variable. The astute reader might have spotted collinearity in Section 4.5 and will be skeptical of the regression results hereafter. Hence, we need to address this issue before going further.

In order to assess the degree of multi-collinearity, we calculate the widely-used Variance Inflation Factor (VIF) (Fox and Monette, 1992). Classically, the VIF of a predictor p_i is computed from the R^2 of the multi-linear model predicting p_i from the remaining covariates. That is to measure its redundancy w.r.t. all other predictors. However, in the presence of clustered data, this redundancy has to be assessed conditionally on the (random) group effects. Therefore, following Davis et al. (1986) and Harrell Jr (2016), we compute the VIFs from the variance-covariance matrix of the regression coefficients estimates, which in the case of a mixed model does take random effects into account. Results are reported in Figure 6. It appears that considering group effects mitigates the consequences of collinearity to a degree that is reasonably accept-

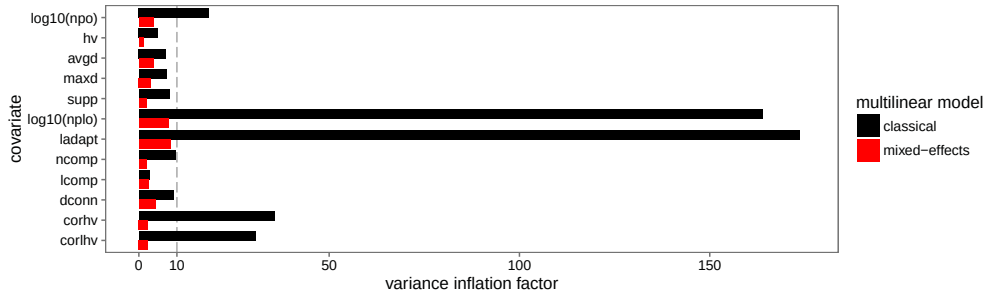

Figure 6: Variance inflation factor. For both regression strategies (see color legend), bars show the multiplicative increase in uncertainty around each coefficient estimate w.r.t. an ideal case in which predictors were linearly unrelated (Harrell Jr, 2016).

able, especially as compared to what would result from fitting a traditional multi-linear model (O'brien, 2007). As a possible explanation for this, since some of the values of certain predictors might only be observed in certain groups, what appears as collinearity in multi-linear models might not pose problems in mixed models. Notably, such has to be the case of the two most problematic predictors, the number of Pareto local optima (`nplo`) and the length of adaptive walks (`ladapt`).

5.3 Predictor Effect Size

Question #5: *What is the conditional impact of each feature on algorithm performance? What are the significant common trends across instance groups?*

We report in Figure 7 (Top) the values of the regression coefficients estimated from the mixed model. In a multi-linear regression, each coefficient $\hat{\beta}_i$ predicts the change in the conditional mean of the response variable after a unitary change of the corresponding covariate x_i. Since predictors have values in different ranges, in order to be able to compare their impact we need to standardize the regression estimates, which are reported in Figure 7 (Bottom). Standardized $\hat{\beta}_i$ predict by how many standard deviations the conditional mean of the response variable will change if predictor i shifts by one standard deviation. As such, higher values correspond to steeper slopes of the partial regression lines where the given predictor is the only covariate and all other predictors are held constant at their respective median value. This can be appreciated on Figure 8. Partial residuals are also added to the plots for a visual assessment of the model fit (Larsen and McCleary, 1972). For a given covariate x_i, the corresponding partial residuals are obtained by adding the vector $x_i\hat{\beta}_i$ to the vector of residuals from the complete regression, such that the slope of a simple regression of the partial residuals on x_i would equal $\hat{\beta}_i$. This simple interpretation motivates the choice of a multi-linear model.

For both algorithms, the features having the highest individual impact on the runtime are the hypervolume (`hv`) and its autocorrelation measures (`corhv` and `corlhv`, in order of importance). For GSEMO, the number of Pareto optimal solutions has a significant effect on estimated runtime, the larger `npo` the shorter `ert`, whereas we cannot reject the hypothesis that the number of Pareto local optimal solutions has no effect on GSEMO's `ert`. Still, the effect of the length of an adaptive walk (`ladapt`), which is a good estimator for `nplo` (see Section 4.5), is significant for GSEMO. Conversely, the

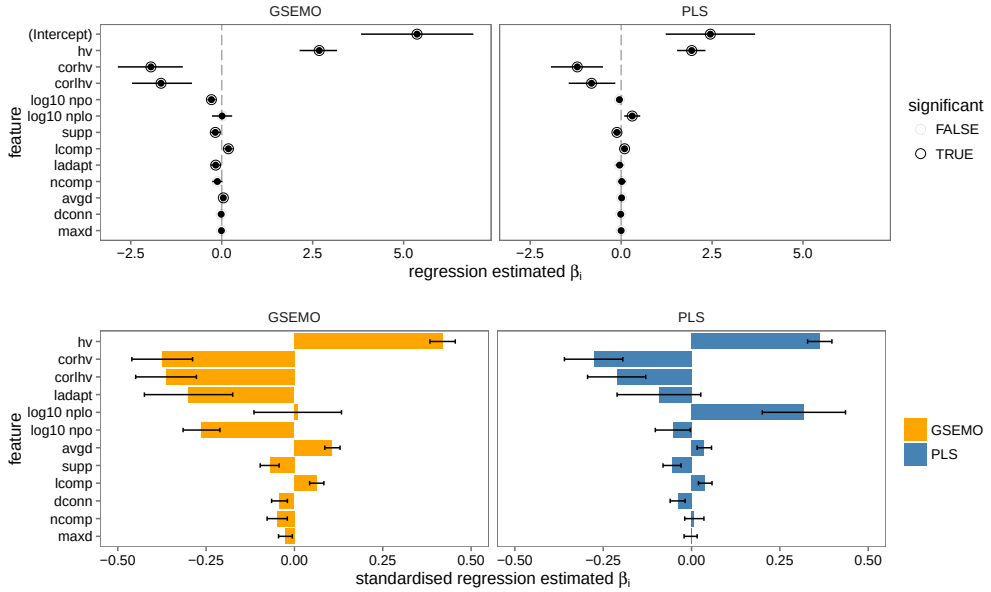

Figure 7: Conditional impact of instance features on the $\log_{10}$-transformed estimated runtime in a mixed-effects multi-linear model. Top: Points and bars give, respectively, the estimates and 0.95 confidence intervals of model parameters for intercept and fixed effects β_i ($H_0 : \beta_i = 0$); see Bates et al. (2014). Bottom: Bars and error bars give, respectively, the standardized coefficients and standard errors of features' effect size.

opposite is true for PLS: its runtime is more impacted by the problem multi-modality than GSEMO. Indeed, the number of Pareto local optima `nplo` has one of the largest effect sizes on PLS runtime (cf. Figure 7 – Bottom). This suggests that GSEMO could be more appropriate than PLS when tackling highly multi-modal instances. This also shows that, by taking group effect into account (i.e. conditionally on the problem instance class), mixed models are able to distinguish between the effects of `ladapt` and `nplo`, which could be taken as surrogates for one another at the aggregate level.

Furthermore, the relative size of the largest connected component of the Pareto set (`lcomp`) impacts the performance of both algorithms. However, when controlling for all other features (i.e. conditionally on all other predictors), we find that an increase in the Pareto set connectedness yields a small increase in the estimated runtime. Surprisingly also, the fraction of supported solutions (`supp`) is a significant predictor for both algorithms, even if none of them explicitly exploits this feature during the search process and the impact on runtime is very small indeed. Finally, despite being highly correlated with `ert`, the average distance between non-dominated solutions (`avgd`) has only a moderate impact on GSEMO and no significant impact on PLS.

5.4 Relative Importance of Predictors

Question #6: *Which features are relevant predictors of algorithm performance?*

Variable importance is commonly assessed via feature selection. However, stepwise selection can be misleading: intercorrelated predictors have a confounding effect on each other, but what the multi-linear regression tries to measure is precisely the effect of one

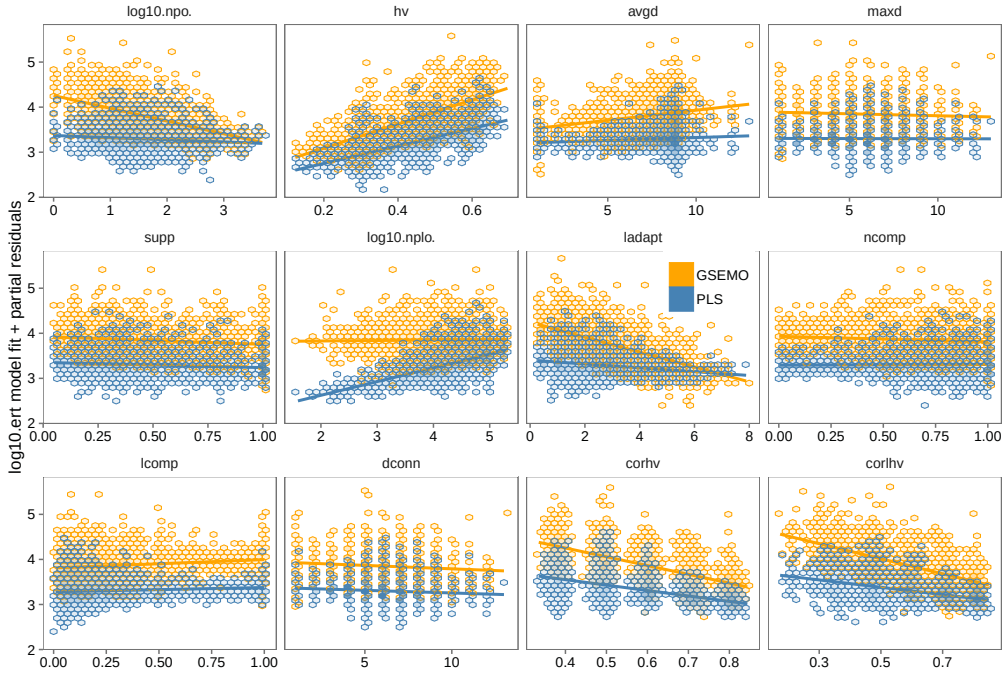

Figure 8: Partial regression plots, also called conditional plots, displaying the result of the mixed model fitting for each explanatory variable (Wickham, 2009; Breheny and Burchett, 2015). The regression models the log-transformed estimated runtime (y-axis) as a multi-linear function of problem instance features (x-axis, see titles). Lines represent partial regressions, points represent partial residuals from the multi-linear model fit. Results are grouped by algorithm (see color legend).

variable *controlling* for the others (i.e. fixing all the others), which leads to a poor estimation in the presence of collinearity. In an information theoretical framework, the Akaike Information Criterion (AIC) (Akaike, 1973) measures the relative quality of a model on a given dataset, not in terms of accuracy, but in terms of likelihood, i.e. the relative distance to the unknown true mechanism generating the observed data. The difference in AIC-values between two models can then be used to estimate the strength of evidence for one model against the other. On a set of alternate models, AIC differences can be transformed into so-called Akaike weights, which can be directly interpreted as the probability for each model to be the "best" one, conditionally on the considered set of models (Burnham and Anderson, 2002). In this context, instead of performing feature selection, variable importance can be better assessed by making inference from all candidate models (Burnham and Anderson, 2004). We perform an exhaustive search in the space of all 2^p models that can be built with our p predictors. The sum of Akaike weights of the $2^{(p-1)}$ models that contain a given predictor can give us an estimation of the relative importance of that particular variable. Admittedly, intercorrelated predictors can still be confounded depending on the regression model under consideration, but this methodology avoids the biases of stepwise feature selection.

Results are reported in Figure 9. The hypervolume of the Pareto front (`hv`), and the

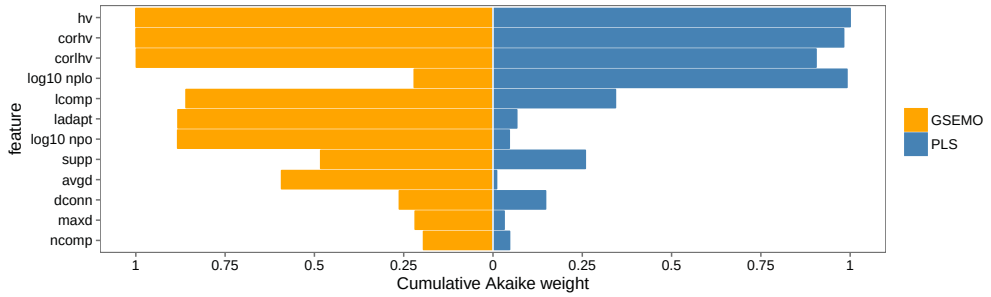

Figure 9: Relative importance of instance features as performance predictors in a linear mixed model. Each bar displays, among all 2^p possible models, the sum of the Akaike's weights of the $2^{(p-1)}$ models including the given feature (Bartoń, 2014).

first autocorrelation coefficient of solution and local hypervolume (`corhv`, `corlhv`) are strong explanatory features for both algorithms, albeit in different order of relative importance. Indeed, notice that the two autocorrelation mesures of ruggedness are considered to be both important in the prediction. These two features are highly correlated with each other (see Figure 3), and we can observe that their respective coefficients in the linear models have nearly the same value (see Figure 7). It is interesting to highlight that, as in single-objective optimization, the ruggedness of the multi-objective landscape is an important performance predictor for both algorithms.

More features are relatively important predictors for GSEMO than for PLS. Features reflecting the connectedness of the Pareto set are important for GSEMO, which navigates the search space with an ergodic variation operator. On the contrary, the same features carry little information about the runtime of PLS, which is constrained by the exploration of a finite neighborhood. Moreover, the number of Pareto optimal solutions (`npo`), the adaptive walk length (`ladapt`), and the relative size of the largest component (`lcomp`) can also be considered as important features for GSEMO, whereas the number of Pareto local optima (`nplo`) is the second most-important predictor for PLS. Let us remind that the log-transformed number of Pareto local optima and the length of an adaptive walk are highly correlated (see Figure 3). In this case, only one of these two features is preferred by the regression models: the number of Pareto local optima is more important for PLS, and the length of adaptive walk is more important for GSEMO. In any case, our results show that, in multi-objective optimization as well, the problem multi-modality actually gives a good indication of the problem difficulty.

5.5 Hierarchical Linear Models

Question #7: *Does the impact of features on algorithm performance change with landscape ruggedness? Can ruggedness be used to explain changes across instance groups?*

So far, we considered a multi-linear model to study the conditional effects of landscape features on algorithm runtime, assuming these effects to be *fixed* across instance classes or groups, but each group having a different *random* baseline. Random intercepts allow for variability in baseline algorithm performance across instance classes, i.e. at the level of instance groups, whereas fixed regression slopes estimate the impact of features on performance at the instance level. The next logical step is to allow for variability also in the effect of landscape features across instance groups, and to try to

explain these variations using class characteristics, such as the degree of epistasis K. To this purpose, we shall reformulate our model as a multi-level model.

For simplicity, let us focus on a single feature x_1. At the instance level (level 1), algorithm performance y_{ij} on problem instance i from group j can be rewritten as:

$$y_{ij} = \beta_{0j} + \beta_1 x_{1ij} + \varepsilon_{ij} \qquad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

(notice the group-dependent intercept β_{0j}). At the group level (level 2), we can write:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + u_{0j} \qquad u_{0j} \sim \mathcal{N}(0, \tau_{00}) \\ \beta_1 &= \gamma_{10} \end{aligned}$$

where u_{0j} (previously α_j) is the group-level random effect describing the deviation of the intercept of group j from the common mean intercept γ_{00} (previously, simply β_0). The random intercept variance τ_{00} accounts for the heterogeneity in the baseline performance due to the fact that instance groups are created from different combinations of benchmark parameter values ρ, M, and K.

However, we could also allow for slope variability across instance groups (i.e. variability in the effect of features on performance). In this case the level 1 model becomes:

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{1ij} + \varepsilon_{ij} \qquad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

(notice the group index j on both intercept and slope), and the level 2 becomes:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + u_{0j} \\ \beta_{1j} &= \gamma_{10} + u_{1j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim \mathcal{N}_2 \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{bmatrix} \right)$$

where the additional random effect u_{1j} describes the difference between the slope of instance-group j and the common mean slope γ_{10}. We assume this difference to be normally distributed with zero mean and variance τ_{11}. Indeed, group effects are treated here as random variables: the only inferences we can make about them, are relative to their variance and covariance. What can we say then about instance classes?

To answer such a question, we need to introduce predictors from the instance class level in a way that expresses slopes and intercepts as outcomes of benchmark parameters. In other words, we need to explain some of the variability in the impact of landscape features on algorithm performance as a function of instance class characteristics, such as the number of epistatic links K. Hence, we allow for a given landscape feature to have a varying effect on algorithm performance depending on the instance class, and at the same time we can predict this effect for each value of the class characteristic under study. If we indicate this group feature by w, the level 2 model becomes:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01} w_j + u_{0j} \\ \beta_{1j} &= \gamma_{10} + \gamma_{11} w_j + u_{1j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim \mathcal{N}_2 \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{bmatrix} \right)$$

where the random effects u_{0j} and u_{1j} have the same structure as in the previous model, but hopefully their residual variance will be reduced by taking the level 2 predictor w into account, since part of the heterogeneity in intercepts and slopes will be explained by the additional w_j terms. Essentially, bar the random terms, this model is expressed as a traditional regression with cross-level interaction between group-level and individual-level predictors, as it would be clear by substituting level 2 equations into the level 1 equation. The random terms, for their part, reflect the clustered structure from the data: the classical regression assumption of mutually independent observations would be violated otherwise. Moreover, the statistical literature suggests that

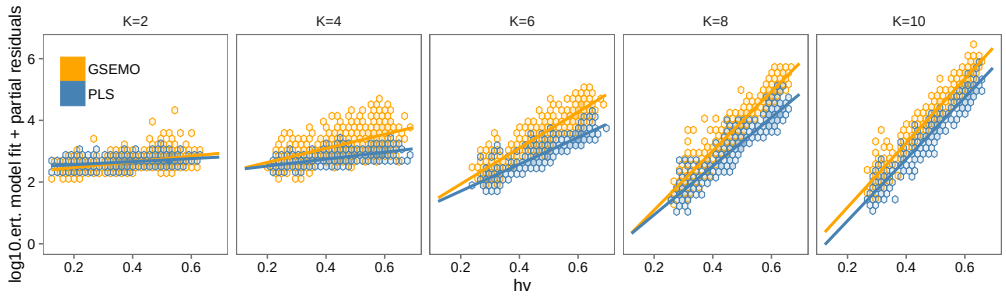

Figure 10: Cross-sectional plots visualizing the effects of hypervolume and problem non-linearity K on estimated runtime in a hierarchical linear mixed model. Regression lines and partial residuals are given for each K-value (see titles) and for both algorithms (see color legend).

mixed models with the maximal random structure justified by the experimental design, are the ones with the best potential to produce generalizable results (Barr et al., 2013).

5.6 Multi-level Analysis

More concretely, let us focus on problem non-linearity K as instance class characteristic, which we treat as a categorical variable with values in $\{2, 4, 6, 8, 10\}$. As a result of this choice, the level 2 predictor w is encoded as a series of dummy variables, one for each K-value in $\{4, 6, 8, 10\}$. Model fitting would then yield a series of $\gamma_{01'}$ and $\gamma_{11'}$ coefficients, two for each dummy variable, which we can interpret in the following way: γ_{01K4} and γ_{11K4} represent, respectively, the difference in average intercept and average slope between problem instances with $K = 4$ and problem instances with $K = 2$; γ_{01K6} and γ_{11K6} represent the difference in average intercept and average slope between problem instances with $K = 6$ and problem instances with $K = 2$, and so forth. These average slopes and average intercepts, i.e. the partial regression lines for a given value of problem non-linearity K, can be visualized through so-called cross-sectional plots, as in any regression that includes predictor interactions. Such plots allow us to see how the relationship between runtime and the feature of interest changes depending on the degree of epistasis of the problem.

In Sections 5.3 and 5.4, we identified the hypervolume (`hv`) and its autocorrelation measures (`corhv` and `corlhv`) to have the highest impact on the estimated runtime of both GSEMO and PLS. We also showed how both algorithms are impacted differently by the multi-modality of a problem instance, as measured by the number of Pareto local optima (`nplo`) or by the length of a Pareto-based adaptive walk (`ladapt`). In the following, we want to see if, and how, the effect of those features changes depending on K. Indeed, the ruggedness of the objective functions is often overlooked in the multi-objective optimization literature, but here we are able to exploit the tunable nature of ρMNK-landscapes.

First, Figure 10 displays the effect of hypervolume on runtime depending on K, as captured by a multi-level model: a simple regression of runtime on hypervolume with hypervolume-epistasis interaction and random deviations in slope and intercept for each instance group. The general trend is the same as in the multi-linear model of Section 5.3: the larger the hypervolume of the (exact) Pareto set to be approximated, the longer the runtime required to find a $(1+\varepsilon)$-approximation of it. However, we now

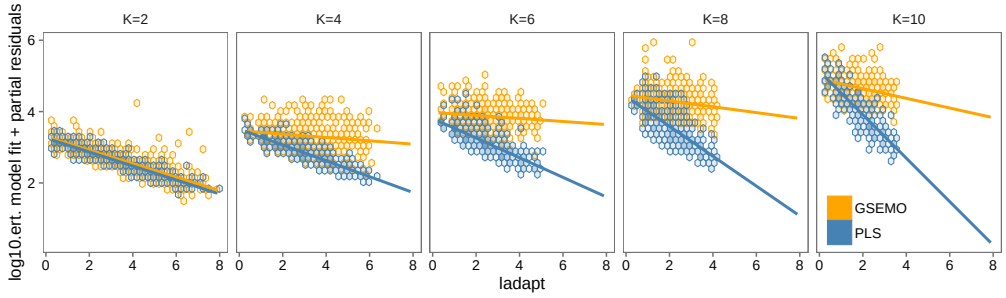

Figure 11: Cross-sectional plots visualizing the effects of adaptive walks' length and problem non-linearity K on estimated runtime in a hierarchical linear mixed model. Regression lines and partial residuals are given for each K-value (see titles) and for both algorithms (see color legend).

clearly see that this effect is milder on smoother landscapes, while it becomes more pronounced as the degree of epistasis increases, i.e. as the problem instance gets more rugged. This trend is similar for both GSEMO and PLS.

Next, Figure 11 displays the effect of the length of a Pareto-based adaptive walk on the algorithm runtime depending on K. In this case, the two EMO algorithms show a divergent behavior in the way the ruggedness of the objective functions interacts with the multi-modality of the multi-objective optimization problem. For both GSEMO and PLS, the common trend is as follows: the shorter `ladapt`, i.e. the larger the expected number of Pareto local optima, the longer the estimated runtime to find a good approximation of the exact Pareto set. But while the impact of multi-modality on GSEMO is rather small and does not seem to depend on K, PLS is comparatively much more affected by the number of Pareto local optima, as the impact of multi-modality on runtime increases with the ruggedness parameter K. For both algorithms, the larger the degree of epistasis, the longer the runtime. However, on more rugged landscapes, the local EMO algorithm (PLS) appears to be increasingly more effective than its global counterpart (GSEMO) when the problem under consideration has few Pareto local optima.

At last, let us notice that the data does not support a multi-level analysis considering K and the autocorrelation measures of hypervolume `corhv` and `corlhv`. Indeed, the values of those predictors are too tightly tied to the ruggedness parameter K and thus, once we control for K, the residual variance is too low to allow for any meaningful analysis. For this reason, related figures are not shown in the paper. However, the observations obtained from our multi-level analysis surely complement those from Section 5.3, where the focus was on instance features alone, and where reasons of model-fitting convergence did not allow us to have random slopes and a cross-level interaction term for each feature.

6 Conclusions

In this paper, we proposed a general-purpose methodology to understand the impact of problem characteristics and fitness landscape features on the performance of EMO algorithms. To the best of our knowledge, this is novel in the multi-objective optimization literature. Our statistical investigation, based on correlation and regression analysis, does stem from the *intuitions* we may have about problem features and how

they might relate to algorithm performance. But our conclusions are quantitatively supported by *empirical data*, coming from a large set of experiments (albeit on a single testbed of enumerable instances), and covering a wide range of the structural properties that EMO algorithms might encounter. In addition, the use of mixed models allows us to respect the clustered structure of the particular dataset (different instance groups, defined by the combinations of benchmark parameters), while still aiming to produce generalizable inferences: instance groups are modeled as *random* effects, whereas the interest is on the *fixed* effects of problem features on algorithm performance at the instance level. Through these mixed models, we assess and contrast the impact of landscape characteristics on the runtime of two prototypical EMO algorithms.

In particular, our analysis on ρMNK-landscapes emphasizes the importance of *ruggedness* and *multi-modality* as the more impactful characteristics for both local and global dominance-based EMO algorithms. Although the significance of those features is certainly recognized in single-objective optimization, this is in our opinion often overlooked in the multi-objective optimization literature. Indeed, those features interact differently on the runtime of the considered EMO algorithms, with the Pareto local search algorithm showing a competitive advantage when the landscape is rugged but Pareto local optima are few. In addition, the *hypervolume* covered by the optimal Pareto set is also reported as a key feature to explain the runtime of both EMO algorithms under consideration. We could attribute this to the chosen stopping criterion, a quality threshold on the approximation set measured in terms of epsilon distance to the optimal Pareto front.

As for the choice of problem instances, we reckon the family of ρMNK-landscapes as a synthetic benchmark that can generalize other multi-objective combinatorial optimization problems, as NK-landscapes do in the single-objective case (Heckendorn and Whitley, 1997). However, we must acknowledge that the obvious next step is to consider additional (large-size) problem and algorithm classes, and more importantly additional (even multiple) stopping conditions and quality assessment indicators, in order to further generalize the current findings. For instance, considering a $(1+\varepsilon)-$approximation of the Pareto set as a target is indeed just one way of measuring performance. In any case, it is our hope that the proposed methodology will be helpful, not only to the practitioner who wants to gain insights about his/her problem classes, but also to the algorithm designer. In fact, understanding the performance of EMO algorithms is a necessary step before improving them, for example by comparing the impact on algorithm performance of different operators (i.e. different landscapes) or of different selection mechanisms, depending on the problem characteristics. This might not only help us to have a better understanding of the respective strengths and weaknesses of multi-objective optimization algorithms, but it might also lead us to predict their expected performance on a black-box problem instance.

Acknowledgments. The authors are grateful to the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by the JSPS project "Global Research on the Framework of Evolutionary Solution Search to Accelerate Innovation" (2013-2016), and by the JSPS-Inria project "Threefold Scalability in Any-objective Black-Box Optimization" (2015-2017).

References

Aguirre, H. E. and Tanaka, K. (2007). Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research*, 181(3):1670–1690.

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, Budapest, Hungary.

Andersen, K. A., Jörnsten, K., and Lind, M. (1996). On bicriterion minimal spanning trees: An approximation. *Computers & Operations Research*, 23(12):1171–1182.

Auger, A. and Hansen, N. (2005). Performance evaluation of an advanced local search evolutionary algorithm. In *Congress on Evolutionary Computation (CEC 2005)*, pages 1777–1784, Piscataway, NJ, USA. IEEE Press.

Barr, D. J., Levy, R., Scheepers, C., and Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

Bartoń, K. (2014). *MuMIn: Multi-Model Inference*. R package version 1.12.1.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2014). Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*.

Bischl, B., Mersmann, O., Trautmann, H., and Preuß, M. (2012). Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 313–320, Philadelphia, Pennsylvania, USA. ACM.

Borges, P. and Hansen, M. (1998). A basis for future successes in multiobjective combinatorial optimization. Technical Report IMM-REP-1998-8, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Breheny, P. and Burchett, W. (2015). *visreg: Visualization of Regression Models*. R package version 2.2-0.

Burnham, K. P. and Anderson, D. R. (2002). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media.

Burnham, K. P. and Anderson, D. R. (2004). Multimodel inference understanding AIC and BIC in model selection. *Sociological methods & research*, 33(2):261–304.

Chiarandini, M. and Goegebeur, Y. (2010). Mixed models for the analysis of optimization algorithms. In *Experimental Methods for the Analysis of Optimization Algorithms*, pages 225–264. Springer.

Coello, C. A. and Lamont, G. B., editors (2004). *Applications of Multi-Objective Evolutionary Algorithms*. Advances in Natural Computation. World Scientific.

Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, second edition.

Daolio, F., Liefooghe, A., Verel, S., Aguirre, H., and Tanaka, K. (2015). Global vs local search on multi-objective nk-landscapes: Contrasting the impact of problem features. In *Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 369–376, Madrid, Spain. ACM.

Daolio, F., Verel, S., Ochoa, G., and Tomassini, M. (2012). Local optima networks and the performance of iterated local search. In *Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 369–376, Vancouver, Canada. ACM.

Davis, C., Hyde, J., Bangdiwala, S., and Nelson, J. (1986). An example of dependencies among variables in a conditional logistic regression. *Modern statistical methods in chronic disease epidemiology*, pages 140–147.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

Ehrgott, M. and Klamroth, K. (1997). Connectedness of efficient solutions in multiple criteria combinatorial optimization. *European Journal of Operational Research*, 97(1):159–166.

Faraway, J. J. (2005). *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. Texts in Statistical Science. Chapman & Hall/CRC.

Fox, J. and Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, 87(417):178–183.

Garrett, D. and Dasgupta, D. (2007). Multiobjective landscape analysis and the generalized assignment problem. In *Learning and Intelligent OptimizatioN (LION 2)*, volume 5313 of *Lecture Notes in Computer Science*, pages 110–124, Trento, Italy. Springer.

Garrett, D. and Dasgupta, D. (2009). Plateau connection structure and multiobjective metaheuristic performance. In *Congress on Evolutionary Computation (CEC 2009)*, pages 1281–1288, Trondheim, Norway. IEEE Press.

Gorski, J., Klamroth, K., and Ruzika, S. (2011). Connectedness of efficient solutions in multiple objective combinatorial optimization. *Journal of Optimization Theory and Applications*, 150(3):475–497.

Harrell Jr, F. E. (2016). *rms: Regression Modeling Strategies*. R package version 4.5-0.

Heckendorn, R. B. and Whitley, D. (1997). A walsh analysis of NK-landscapes. In *International Conference on Genetic Algorithms*, pages 41–48. Morgan Kaufmann.

Hutter, F., Xu, L., Hoos, H. H., and Leyton-Brown, K. (2014). Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206:79–111.

Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.

Knowles, J. and Corne, D. (2003). Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 295–310, Faro, Portugal. Springer.

Larsen, W. A. and McCleary, S. J. (1972). The use of partial residual plots in regression analysis. *Technometrics*, 14(3):781–790.

Laumanns, M., Thiele, L., and Zitzler, E. (2004). Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing*, 3(1):37–51.

Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., and Deb, K. (2002). Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Conference on Parallel Problem Solving From Nature (PPSN VII)*, volume 2439 of *Lecture Notes in Computer Science*, pages 44–53, Granada, Spain. Springer.

Liefooghe, A., Jourdan, L., and Talbi, E.-G. (2011). A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO. *European Journal of Operational Research*, 209(2):104–112.

Liefooghe, A., Paquete, L., and Figueira, J. R. (2013a). On local search for bi-objective knapsack problems. *Evolutionary Computation*, 21(1):179–196.

Liefooghe, A., Verel, S., Aguirre, H., and Tanaka, K. (2013b). What makes an instance difficult for black-box 0–1 evolutionary multiobjective optimizers? In *International Conference on Artificial Evolution (EA 2013)*, volume 8752 of *LNCS*, pages 3–15.

McLeod, A. (2011). *Kendall: Kendall rank correlation and Mann-Kendall trend test*. R package version 2.2.

Mersmann, O., Bischl, B., Bossek, J., Trautmann, H., Wagner, M., and Neumann, F. (2012). Local search and the traveling salesman problem: A feature-based characterization of problem hardness. In *Learning and Intelligent OptimizatioN Conference (LION 6)*, volume 7219 of *Lecture Notes in Computer Science*, pages 115–129, Paris, France. Springer.

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., and Rudolph, G. (2011). Exploratory landscape analysis. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 829–836, Dublin, Ireland. ACM.

Merz, P. (2004). Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–325.

Mote, J., Murthy, I., and Olson, D. L. (1991). A parametric approach to solving bicriterion shortest path problems. *European Journal of Operational Research*, 53(1):81–92.

Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142.

O'brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, 41(5):673–690.

Papadimitriou, C. H. and Yannakakis, M. (2000). On the approximability of trade-offs and optimal access of web sources. In *Symposium on Foundations of Computer Science (FOCS 2000)*, pages 86–92.

Paquete, L., Camacho, C., and Figueira, J. R. (2008). A two-phase heuristic for the biobjective 0/1 knapsack problem. Technical report, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal.

Paquete, L., Chiarandini, M., and Stützle, T. (2004). Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer.

Paquete, L., Schiavinotto, T., and Stützle, T. (2007). On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research*, 156(1):83–97.

Paquete, L. and Stützle, T. (2006). A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research*, 169(3):943–959.

Paquete, L. and Stützle, T. (2009). Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In *Multiobjective Programming and Goal Programming*, volume 618 of *Lecture Notes in Economics and Mathematical Systems*, pages 69–77. Springer.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Richter, H. and Engelbrecht, A., editors (2014). *Recent Advances in the Theory and Application of Fitness Landscapes*. Emergence, Complexity and Computation. Springer.

Verel, S., Liefooghe, A., Jourdan, L., and Dhaenens, C. (2011). Analyzing the effect of objective correlation on the efficient set of MNK-landscapes. In *Learning and Intelligent OptimizatioN (LION 5)*, volume 6683 of *Lecture Notes in Computer Science*, pages 238–252. Springer, Rome, Italy.

Verel, S., Liefooghe, A., Jourdan, L., and Dhaenens, C. (2013). On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research*, 227(2):331–342.

Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *Lecture Notes in Computer Science*, pages 742–756, Matsushima, Japan. Springer.

Weinberger, E. D. (1990). Correlated and uncorrelatated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336.

Weinberger, E. D. (1991). Local properties of Kauffman's N-k model: A tunably rugged energy landscape. *Physical Review A*, 44(10):6399.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer.

Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C. M., and Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.