What Makes an Instance Difficult for Black-Box 0–1 Evolutionary Multiobjective Optimizers?

Arnaud Liefooghe[1,2(✉)], Sébastien Verel[3], Hernán Aguirre[4], and Kiyoshi Tanaka[4]

[1] LIFL (UMR CNRS 8022), Université Lille 1, Villeneuve d'Ascq, France
[2] Dolphin, Inria Lille-Nord Europe, Villeneuve d'Ascq, France
arnaud.liefooghe@univ-lille1.fr
[3] LISIC, Université du Littoral Côte d'Opale, Calais, France
verel@lisic.univ-littoral.fr
[4] Faculty of Engineering, Shinshu University, Nagano, Japan
{ahernan,ktanaka}@shinshu-u.ac.jp

Abstract. This paper investigates the correlation between the characteristics extracted from the problem instance and the performance of a simple evolutionary multiobjective optimization algorithm. First, a number of features are identified and measured on a large set of enumerable multiobjective NK-landscapes with objective correlation. A correlation analysis is conducted between those attributes, including low-level features extracted from the problem input data as well as high-level features extracted from the Pareto set, the Pareto graph and the fitness landscape. Second, we experimentally analyze the (estimated) running time of the global SEMO algorithm to identify a $(1+\varepsilon)$-approximation of the Pareto set. By putting this performance measure in relation with problem instance features, we are able to explain the difficulties encountered by the algorithm with respect to the main instance characteristics.

1 Introduction

In single-objective black-box combinatorial optimization, fitness landscape analysis aims at apprehending the relation between the geometry of a problem instance and the dynamics of randomized search algorithms. Understanding the main problem-related features allows to explain the behavior and the performance of such algorithms, the ultimate goal being to predict this performance and adapt the algorithm setting to the instance being solved. Recently, the performance of single-objective randomized search algorithms has been correlated to fitness landscape features [2]. In this paper, we propose a general methodology to analyze the correlation between problem features and algorithm performance in black-box 0–1 evolutionary multiobjective optimization. To the best of our knowledge, this is the first time that such an analysis is conducted in multiobjective optimization.

P. Legrand et al. (Eds.): EA 2013, LNCS 8752, pp. 3–15, 2014.
DOI: 10.1007/978-3-319-11683-9_1

We first identify a number of existing and original multiobjective problem features. They include low-level features extracted from the problem input data like variable correlation, objective correlation, and objective space dimension, as well as high-level features from the Pareto set, the Pareto graph and the ruggedness and multimodality of the fitness landscape. Some of them are here proposed for the first time. They consist of a simple autocorrelation function, based on a local hypervolume measure, and allowing to estimate the ruggedness of the fitness landscape. We report all these measures on a large number of enumerable multiobjective NK-landscapes with objective correlation (ρMNK-landscapes), together with a correlation analysis between them.

Next, we conduct an experimental analysis on the correlation between instance features and algorithm performance. To do so, we investigate the estimated running time of a simple evolutionary multiobjective optimization algorithm, namely global SEMO [7], to identify a $(1+\varepsilon)$-approximation of the Pareto set. In particular, the original hypervolume-based autocorrelation functions appear to be the features with the highest correlation with the algorithm performance. Overall, the running time of the algorithm is impacted by each of the identified multiobjective problem feature. Our analysis shows their relative importance on the algorithm efficiency. Moreover, taking the features all together allows to better explain the dynamics of randomized search algorithms.

The paper is organized as follows. Section 2 details the background information related to fitness landscape analysis, multiobjective optimization and ρMNK-landscapes. In Sect. 3, low-level and high-level instance features are identified, and quantitative results, together with a correlation analysis, are reported for ρMNK-landscapes. Section 4 presents the experimental setup of global SEMO and discusses the correlation between the problem features and the estimated running time of global SEMO. Section 5 concludes the paper and discusses further research.

2 Preliminaries

2.1 Fitness Landscape Analysis

In single-objective optimization, fitness landscape analysis allows to study the topology of a combinatorial optimization problem [13], by gathering important information such as ruggedness or multimodality. A fitness landscape is defined by a triplet $(X, \mathcal{N}, \phi)$, where X is a set of admissible solutions (the search space), $\mathcal{N} : X \rightarrow 2^X$ is a neighborhood relation, and $\phi : X \rightarrow \mathbb{R}$ is a (scalar) 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 search 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 is the length of the adaptive walk. This length is a good estimator of the average diameter of the local optima basins of attraction, characterizing a problem instance multimodality. The larger the

length, the larger the basin diameter. This allows to estimate the number of local optima when the whole search space cannot be enumerated exhaustively.

Let $\langle x_0, x_1, \ldots \rangle$ be an infinite *random walk* over the search space. The autocorrelation function and the correlation length of such a random walk allow to measure the ruggedness of a fitness landscape [13]. The random walk autocorrelation function $r : \mathbb{N} \to \mathbb{R}$ of a (scalar) fitness function ϕ is defined as follows.

$$r(k) = \frac{\mathbb{E}[\phi(x_t) \cdot \phi(x_{t+k})] - \mathbb{E}[\phi(x_t)] \cdot \mathbb{E}[\phi(x_{t+k})]}{\mathrm{Var}(\phi(x_t))} \tag{1}$$

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.

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

where $\bar{\phi} = \frac{1}{\ell} \sum_{t=1}^{\ell} \phi(x_t)$, and $\ell \gg 0$. The estimation error diminishes with the walk length ℓ. 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 [13], we define the correlation length by $\tau = -\frac{1}{\ln(r(1))}$, making the assumption that the autocorrelation function decreases exponentially.

2.2 Multiobjective Optimization

A *multiobjective optimization problem* can be defined by an objective vector function $f = (f_1, \ldots, f_M)$ with $M \geqslant 2$ objective functions, and a set X of feasible solutions in the *decision space*. In the combinatorial case, X is a discrete set. Let $Z = f(X) \subseteq \mathbb{R}^M$ be the set of feasible outcome vectors in the *objective space*. To each solution $x \in X$ is assigned an objective vector $z \in Z$ on the basis of the vector function $f : X \to Z$ with $z = f(x)$. The conventional Pareto dominance relation is defined as follows. In a maximization context, an objective vector $z \in Z$ is dominated by an objective vector $z' \in Z$, denoted by $z \prec z'$, if and only if $\forall m \in \{1, \ldots, M\}$, $z_m \leqslant z'_m$ and $\exists m \in \{1, \ldots, M\}$ such that $z_m < z'_m$. By extension, a solution $x \in X$ is dominated by a solution $x' \in X$, denoted by $x \prec x'$, if and only if $f(x) \prec f(x')$. A solution $x^\star \in X$ is said to be *Pareto optimal* (or efficient, non-dominated), if and only if there does not exist any other solution $x \in X$ such that $x^\star \prec x$. The set of all Pareto optimal solutions is called the *Pareto set* $X^\star \subseteq X$. Its mapping in the objective space is called the *Pareto front* $Z^\star \subseteq Z$. One of the most challenging task in multiobjective optimization is to identify a minimal complete Pareto set [3], *i.e.* a Pareto set of minimal size, that is one Pareto optimal solution for each point from the Pareto front.

However, in the combinatorial case, generating a complete Pareto set is often infeasible for two main reasons [3]: (*i*) the number of Pareto optimal solutions is

typically exponential in the size of the problem instance, and (*ii*) deciding if a feasible solution belongs to the Pareto set may be NP-complete. Therefore, the overall goal is often to identify a good *Pareto set approximation*. To this end, heuristics in general, and evolutionary algorithms in particular, have received a growing interest since the late eighties.

2.3 ρMNK-Landscapes

The family of ρMNK-landscapes constitutes a problem-independent model used for constructing multiobjective multimodal landscapes with objective correlation [12]. It extends single-objective NK-landscapes [6] and multiobjective NK-landscapes with independent objective functions [1]. Feasible solutions are binary strings of size N, *i.e.* the decision space is $X = \{0,1\}^N$. The parameter N refers to the problem size (the bit-string length), and the parameter K to the number of variables that influence a particular position from the bit-string (the epistatic interactions). The objective vector function $f = (f_1, \ldots, f_m, \ldots, f_M)$ is defined as $f : \{0,1\}^N \rightarrow [0,1)^M$. Each objective function f_m is to be maximized and can be formalized as follows.

$$f_m(x) = \frac{1}{N} \sum_{i=1}^{N} c_i^m(x_i, x_{i_1}, \ldots, x_{i_K}), m \in \{1, \ldots, M\} \tag{3}$$

where $c_i^m : \{0,1\}^{K+1} \rightarrow [0,1)$ defines the multidimensional component function associated with each variable x_i, $i \in \{1, \ldots, N\}$, and where $K < N$. By increasing the number of variable interactions K from 0 to $(N-1)$, ρMNK-landscapes can be gradually tuned from smooth to rugged. In this work, we set the position of these epistatic interactions uniformly at random. The same epistatic degree $K_m = K$ and the same epistatic interactions are used for all objectives $m \in \{1, \ldots, M\}$. Component values are uniformly distributed in the range $[0,1)$, and follow a multivariate uniform distribution of dimension M, defined by a correlation coefficient $\rho > \frac{-1}{M-1}$, *i.e.* the same correlation ρ is defined between all pairs of objective functions. As a consequence, it is very unlikely that the same objective vector is assigned to two different solutions. The positive (respectively negative) data correlation allows to decrease (respectively increases) the degree of conflict between the objective function values very precisely [12]. An instance generator and the problem instances under study in this paper can be found at the following URL: http://mocobench.sf.net/.

In the following, we investigate 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 $\rho > \frac{-1}{M-1}$. The problem size is set to $N = 18$ in order to enumerate the search space exhaustively. The search space size is then $|X| = 2^{18}$. 30 different landscapes, independently generated at random, are considered for each parameter combination: ρ, M, and K. This leads to a total of 3300 problem instances.

3 Problem Features and Correlation Analysis

In this section, we identify a number of general-purpose features, either directly extracted from the problem instance itself (low-level features), or computed from the enumerated Pareto set and from the fitness landscape (high-level features). Then, a correlation analysis is conducted on those features in order to highlight the main similarities in characterizing the difficulties of a problem instance.

3.1 Low-Level Features from Problem Input Data

First, we consider some features related to the definition of ρMNK-landscapes.

Number of epistatic interactions (K): This gives the number of variable correlations in the construction of ρMNK-landscapes. As will be detailed later, despite the K-value can generally not be retrieved directly from an unknown instance, it can be precisely estimated within some high-level fitness landscape metrics described below.

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

Objective correlation (ρ): This parameter allows 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.

3.2 High-Level Features from the Pareto Set

The high-level fitness landscape metrics 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.* $\texttt{npo} = |X^\star|$. 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 [12].

Hypervolume (`hv`): The hypervolume value of a the Pareto set $X^\star$ gives the portion of the objective space that is dominated by $X^\star$ [14]. 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.

3.3 High-Level Features from the Pareto Graph

In the following, we describe some high-level features related to the *connectedness* of the Pareto set [4]. 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 many non-dominated solutions by starting with at least one Pareto optimal solution; see *e.g.* [9,10]. We follow the definition of *k-Pareto graph* from [9]. The k-Pareto graph is defined as a graph $PG_k = (V, E)$, 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 k, *i.e.* $d(x_i, x_j) \leqslant k$. The distance $d(x_i, x_j)$ is taken as the Hamming distance for ρMNK-landscapes. This corresponds to the *bit-flip* neighborhood operator. Some connectedness-related high-level features under investigation are given below.

Number of connected components (`nconnec`): This metric gives the number of connected components in the 1-Pareto graph, *i.e.* in PG_k with $k = 1$.

Size of the largest connected component (`lconnec`): This corresponds to the size of the largest connected component in the 1-Pareto graph PG_1.

Minimum distance to be connected (`kconnec`): This measure corresponds to the smallest distance k such that the k-Pareto graph is connected, *i.e.* for all pairs of vertices $x_i, x_j \in V$ in PG_k, there exists and edge $e_{ij} \in E$.

3.4 High-Level Features from the Fitness Landscape

At last, we give some high-level metrics 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.* [11]. For ρMNK-landscapes, the neighborhood structure is taken as the *1-bit-flip*, which is directly related to a Hamming distance 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 here compute 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 [12].

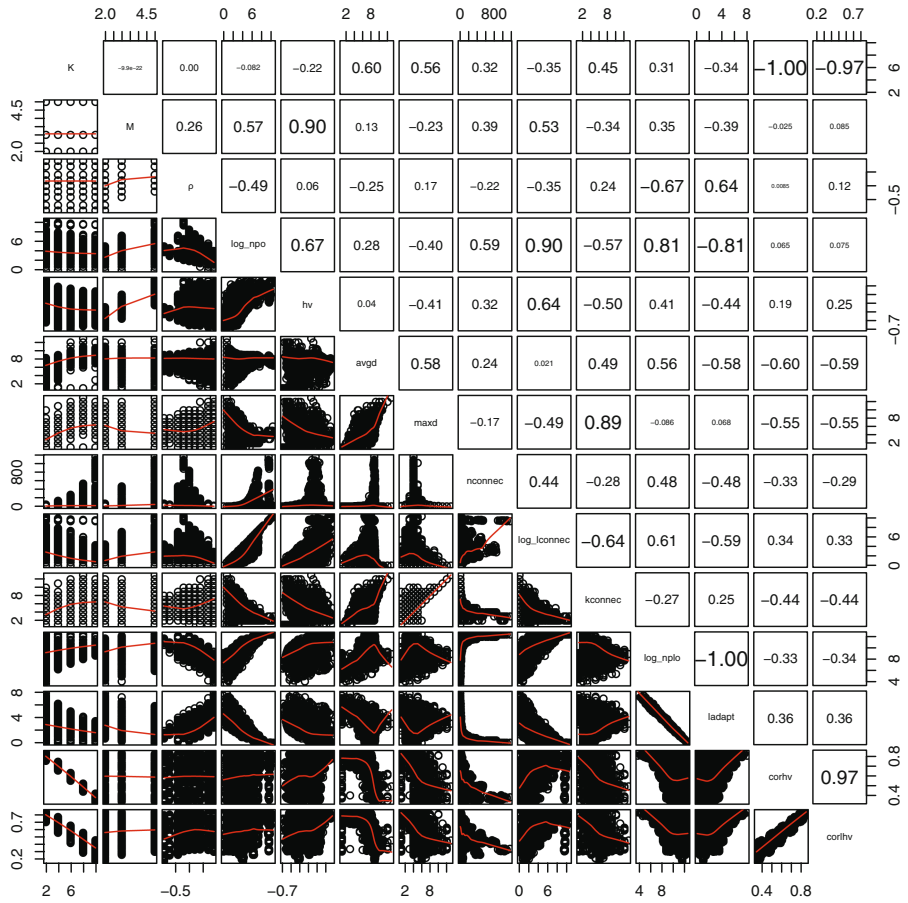

Fig. 1. Correlation matrix between all pairs of features. The feature names are reported on the diagonal. For each pair of features, scatter plots and smoothing splines are displayed below the diagonal, and the corresponding correlation coefficients are reported above the diagonal. The smoothing spline is a smoothing method that fits a smooth curve to a set of noisy observations using a spline function. The correlation coefficient is based on a Pearson product-moment correlation coefficient measuring the linear correlation (dependence) between both features. Correlation coefficient values lie between -1 (total negative correlation) and $+1$ (total positive correlation), while 0 means *no* correlation.

Correlation length of solution hypervolume (`corhv`): The ruggedness is here measured in terms of the autocorrelation of the hypervolume along a random walk. As explained in Sect. 2.1, the correlation length τ measures how the autocorrelation function, estimated with a random walk, decreases. The autocorrelation coefficients are here computed with the following scalar fitness function $\phi : X \to \mathbb{R}$: $\phi(x) = \texttt{hv}(\{x\})$, where $\texttt{hv}(\{x\})$ is the hyper-

volume 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 is the *1-bit-flip*.

Correlation length of local hypervolume (`corlhv`): This metric is similar to the previous one, except that the fitness function is here based 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*.

3.5 Correlation Analysis

The correlation matrix between each pair of features is reported in Fig. 1. First of all, when taken independently, the number of objective functions M and the objective correlation ρ are both moderately correlated to the cardinality of the Pareto set `npo` (the absolute correlation coefficient is around 0.5 in both cases). Surprisingly, the objective space dimension does not explain by itself the large amount of non-dominated solutions found in many-objective optimization. As pointed out in [12], this should be put in relation with the degree of conflicts between the objective function values. Indeed, as shown in Fig. 2, it is easy to build a simple multi-linear regression model based on M and ρ to predict the value of `npo` with a very high precision (resulting in a correlation coefficient of 0.87, and explaining 76 % of the variance). This highlights that the impact of many-objective fitness landscapes on the search process cannot be analyzed properly without taking the objective correlation into account.

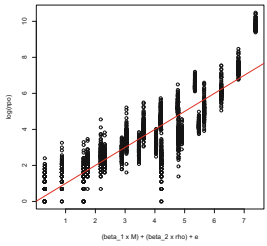

Fig. 2. Scatter plot of the linear regression model $\log(\texttt{npo}) = \beta_1 M + \beta_2 \rho + e$, with $\beta_1 = 1.30567$, $\beta_2 = -2.87688$, and $e = 0.27735$. Residual standard error: 1.037 on 3297 degrees of freedom, multiple R-squared: 0.7629, adjusted R-squared: 0.7627, F-statistic: 5303 on 2 and 3297 DF, p-value: $< 2.2e-16$.

Interestingly, other important remarks can be extracted from the figure. With respect to the Pareto set, the hypervolume value increases with the objective space dimension. Moreover, and unsurprisingly, the Pareto set size and the size of the largest connected component from the Pareto graph are highly correlated. So are the maximum distance between Pareto optimal solutions and the minimum distance for the Pareto set to be connected. As also reported in [12], there is a high correlation between the number of Pareto optimal solutions `npo` and of Pareto local optima `nplo`. More importantly, the number of Pareto local optima `nplo` can be precisely estimated with the length of a Pareto-based adaptive walk `ladapt` (the absolute correlation coefficient between log(`nplo`) and `ladapt` is 1). As a consequence, this allows to estimate the size of the Pareto set as well. At last, the number of epistatic interactions (decision variable correlations) K can be estimated with hypervolume-based autocorrelation functions along a random walk `corhv` and `corlhv`. Since there is not much difference

Algorithm 1. Pseudo-code of G-SEMO

Input: $x^0 \in X$
Output: Archive A
1: $A \leftarrow x^0$
2: **loop**
3: select x from A at random
4: create x' by flipping each bit of x with a probability $1/N$
5: $A \leftarrow$ non-dominated solutions from $A \cup \{x'\}$
6: **end loop**

between the correlations coefficients of both functions, the first one `corhv` should preferably be considered due to its simplicity. Notice that a similar analysis involving instances with the same number of objectives resulted in comparable results.

4 Problem Features *vs.* Algorithm Performance

4.1 Experimental Setup

Global SEMO. Global SEMO (G-SEMO for short) [7] is a simple elitist steady-state EMO algorithm for black-box 0–1 optimization problems dealing with an arbitrary objective vector function defined as $f : \{0,1\}^N \rightarrow Z$ such that $Z \subseteq \mathbb{R}^M$, like ρMNK-landscapes. A pseudo-code is given in Algorithm 1. It maintains an unbounded archive A of non-dominated solutions found so far. The archive is initialized with one random solution from the search space. At each iteration, one solution is chosen at random from the archive. Each bit of this solution is independently flipped with a rate $r = 1/N$, and the obtained solution is checked for insertion in the archive. Within such an independent bit-flip mutation, any solution from the search space can potentially be reached by applying the mutation operator to any arbitrary solution. In its general form, the G-SEMO algorithm does not have any explicit stopping rule [7]. In this paper, we are interested in its running time, in terms of a number of function evaluations, until an $(1+\varepsilon)$-approximation of the Pareto set has been identified and is contained in the internal memory A of the algorithm, subject to a maximum number of function evaluations.

Performance Evaluation. For any constant value $\varepsilon \geqslant 0$, the (multiplicative) ε-dominance relation $\preceq_\varepsilon$ can be defined as follows. For all $z, z' \in Z$, $z \preceq_\varepsilon z'$ if and only if $z_m \cdot (1+\varepsilon) \leqslant z'_m$, $\forall m \in \{1, \ldots, M\}$. Similarly, for all $x, x' \in X$, $x \preceq_\varepsilon x'$ if and only if $f(x) \preceq_\varepsilon f(x')$. Let $\varepsilon \geqslant 0$. A set $X^\varepsilon \subseteq X$ is an $(1+\varepsilon)$-approximation of the Pareto set if and only if, for any solution $x \in X$, there is one solution $x' \in X^\varepsilon$ such that $x \preceq_\varepsilon x'$. This is equivalent of finding a Pareto set approximation whose multiplicative epsilon quality indicator value with respect to the exact Pareto set is $(1+\varepsilon)$, see *e.g.* [14]. Interestingly, under some general

assumptions, there always exists an $(1+\varepsilon)$-approximation, for any given $\varepsilon \geqslant 0$, whose cardinality is both polynomial in the problem size and in $1/\varepsilon$ [8].

Following a conventional methodology from single-objective continuous black-box optimization benchmarking [5], the expected number of function evaluations to identify an $(1+\varepsilon)$-approximation is chosen as a performance measure. However, as any EMO algorithm, G-SEMO can either succeed or fail to reach an accuracy of ε in a single simulation run. In case of a success, the runtime is the number of function evaluations until an $(1+\varepsilon)$-approximation was found. In case of a failure, we simply restart the algorithm at random. We then obtain a "*simulated runtime*" [5] from a set of given trials of G-SEMO on a given instance. Such a performance measure allows to take into account both the success rate $p_s \in (0,1]$ and the convergence speed of the G-SEMO algorithm. Indeed, after $(n-1)$ failures, each one requiring T_f evaluations, and the final successful run with T_s evaluations, the total runtime is $T = \sum_{i=1}^{n-1} T_f + T_s$. By taking the expectation value and by considering that the probability of success after $(n-1)$ failures follows a Bernoulli distribution of parameter p_s, we have:

$$\mathbb{E}[T] = \left(\frac{1-p_s}{p_s}\right)\mathbb{E}[T_f] + \mathbb{E}[T_s] \tag{4}$$

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 to a constant function evaluation limit 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.

$$\texttt{ert} = \left(\frac{1-\hat{p}_s}{\hat{p}_s}\right) T_{max} + \frac{1}{N_s}\sum_{i=1}^{N_s} T_i \tag{5}$$

where N_s is the number of successful runs, and T_i is the number of evaluations required for successful run i. For more details, we refer to [5].

Parameter Setting. In our analysis, we set $\varepsilon = 0.1$. The time limit is set to $T_{max} = 2^N/10 < 26215$ function evaluations without identifying an $(1+\varepsilon)$-approximation. The G-SEMO algorithm is executed 100 times *per* instance. For a given instance, the success rate and the expected number of evaluations for successful runs are estimated from those 100 executions. However, let us note that G-SEMO was not able to identify a $(1+\varepsilon)$-approximation set for any of the runs on one instance with $M = 3$, $\rho = 0.2$ and $K = 10$, one instance with $M = 3$, $\rho = 0.4$ and $K = 10$, ten instances with $M = 5$, $\rho = 0.2$ and $K = 10$, six instances with $M = 5$, $\rho = 0.4$ and $K = 10$, as well as two instances with $M = 5$, $\rho = 0.7$ and $K = 10$. Moreover, G-SEMO was not able to solve the following instances due to an overload CPU resources available: $M = 5$ and $\rho \in \{-0.2, 0.0\}$. Overall, this represents a total amount of 2980 instances times 100 executions, that is 298000 simulation runs.

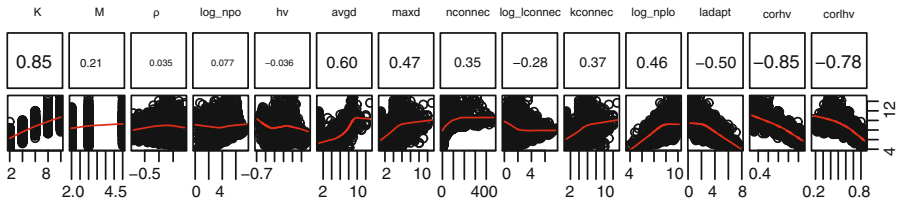

Fig. 3. Correlation between log(`ert`) and each feature. The feature names are reported on the first line, correlation coefficients are reported on the second line, and scatter plots as well as smoothing splines are displayed on the third line.

4.2 Computational Results

The correlation between each feature and the running time of G-SEMO is reported in Fig. 3. First, with respect to low-level features, there exists a high correlation between log(`ert`) and K, which is the highest absolute correlation observed on our data. However, surprisingly, the correlation of the performance measure with M and ρ is not significant. Second, with respect to high-level features from the Pareto set, the size of the Pareto set and its hypervolume does not explain the variance of log(`ert`). Nevertheless, the larger the distance between Pareto optimal solutions in the decision space, the larger the running time of G-SEMO. Similarly, when the Pareto graph is close to a fully connected graph, G-SEMO is likely to take less time to identify a $(1+\varepsilon)$-approximation (the absolute correlation value is around 0.3). As a consequence, the number of Pareto optimal solutions has a smaller impact on the performance of G-SEMO than the structure existing between those solutions in the decision space.

With respect to high-level fitness landscape features, the number of Pareto local optima `nplo` and its estimator `ladapt` both present a significant correlation with the estimated running time of G-SEMO. Indeed, the more Pareto local optima, the longer the running time (the absolute correlation value is close to 0.5). At last, the hypervolume-based autocorrelation functions highly explain the variance of the G-SEMO performance. For both `corhv` and `corlhv`, the absolute correlation value is around 0.8. Overall, this correlation analysis gives a "big picture" of a well-suited multiobjective fitness landscape for G-SEMO. This corroborates the impact of the problem instance properties identified in the previous section on the performance of multiobjective evolutionary algorithms.

5 Discussion

In this paper, we attempted to give a first step towards a better understanding of the evolutionary multiobjective optimization algorithm performance according to the main characteristics of the problem instance. We first presented a number of general problem features, together with a correlation analysis between those features on a large set of enumerable multiobjective NK-landscapes. Then, we

put in relation the running time of a simple evolutionary multiobjective optimization algorithm with those features. Our analysis clearly shows the high impact of theses problem-related properties on the performance of the algorithm. In particular, two relevant hypervolume-based autocorrelation functions have been proposed for the first time, allowing to precisely estimate the ruggedness of the instance under consideration, as well as the algorithm running time.

Using the general methodology introduced in the paper applied to larger problem instances would allow to appreciate the impact of the multiobjective features on the performance of evolutionary multiobjective optimizations when tackling large-size instances. This should be possible with features that do not require the complete enumeration of the decision space, including the problem size, the number of objectives, the objective correlation, the length of a Pareto-based adaptive walk, and the hypervolume-based autocorrelation functions proposed in this paper. As well, the impact of the stopping condition, and in particular the approximation quality (the ε-value) should be carefully investigated. At last, a similar study would allow to better understand the structure of the landscape for real-world multiobjective combinatorial optimization problems. This work pushes towards the design of a *meta-algorithm* able to select the most efficient evolutionary multiobjective algorithm or parameter setting according to a prediction model based on the main problem instance features.

References

1. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. Eur. J. Oper. Res. **181**(3), 1670–1690 (2007)
2. Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local optima networks and the performance of iterated local search. In: Genetic and Evolutionary Computation Conference (GECCO 2012), pp. 369–376 (2012)
3. Ehrgott, M.: Multicriteria Optimization, 2nd edn. Springer, Heidelberg (2005)
4. Gorski, J., Klamroth, K., Ruzika, S.: Connectedness of efficient solutions in multiple objective combinatorial optimization. J. Optim. Theory Appl. **150**(3), 475–497 (2011)
5. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Conference on Genetic and Evolutionary Computation (GECCO 2010), pp. 1689–1696 (2010)
6. Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)
7. Laumanns, M., Thiele, L., Zitzler, E.: Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. Nat. Comput. **3**(1), 37–51 (2004)
8. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: Symposium on Foundations of Computer Science (FOCS 2000), pp. 86–92. IEEE (2000)
9. Paquete, L., Camacho, C., Figueira, J.R.: A two-phase heuristic for the biobjective 0/1 knapsack problem. Unpublished (2008)
10. Paquete, L., Stützle, T.: Clusters of non-dominated solutions in multiobjective combinatorial optimization: an experimental analysis. In: Barichard, V., et al. (eds.) Multiobjective Programming and Goal Programming. Lecture Notes in Economics and Mathematical Systems, vol. 618, pp. 69–77. Springer, Heidelberg (2009)

11. Paquete, L., Schiavinotto, T., Stützle, T.: On local optima in multiobjective combinatorial optimization problems. Ann. Oper. Res. **156**(1), 83–97 (2007)
12. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. Eur. J. Oper. Res. **227**(2), 331–342 (2013)
13. Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern. **63**(5), 325–336 (1990)
14. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)