



HAL
open science

Dominance, indicator and decomposition based search for multi-objective QAP: landscape analysis and automated algorithm selection

Arnaud Liefoghe, Sébastien Verel, Bilel Derbel, Hernan Aguirre, Kiyoshi
Tanaka

► **To cite this version:**

Arnaud Liefoghe, Sébastien Verel, Bilel Derbel, Hernan Aguirre, Kiyoshi Tanaka. Dominance, indicator and decomposition based search for multi-objective QAP: landscape analysis and automated algorithm selection. PPSN 2020 - Sixteenth International Conference on Parallel Problem Solving from Nature, Sep 2020, Leiden, Netherlands. pp.33-47, 10.1007/978-3-030-58112-1_3 . hal-02920085

HAL Id: hal-02920085

<https://hal.science/hal-02920085>

Submitted on 24 Aug 2020

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

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

Dominance, Indicator and Decomposition based Search for Multi-objective QAP: Landscape Analysis and Automated Algorithm Selection^{*}

Arnaud Liefooghe¹, Sébastien Verel², Bilel Derbel³,
Hernan Aguirre⁴, and Kiyoshi Tanaka⁴

¹ JFLI – CNRS IRL 3527, University of Tokyo, 113-0033 Tokyo, Japan
arnaud.liefooghe@univ-lille.fr

² Univ. Littoral Côte d’Opale, LISIC, F-62100 Calais, France

³ Univ. Lille, CNRS, Centrale, Inria, UMR 9189 - CRISTAL, F-59000 Lille, France

⁴ Shinshu University, Faculty of Engineering, Nagano, Japan

Abstract. We investigate the properties of large-scale multi-objective quadratic assignment problems (mQAP) and how they impact the performance of multi-objective evolutionary algorithms. The landscape of a diversified dataset of bi-, multi-, and many-objective mQAP instances is characterized by means of previously-identified features. These features measure complementary facets of problem difficulty based on a sample of solutions collected along random and adaptive walks over the landscape. The strengths and weaknesses of a dominance-based, an indicator-based, and a decomposition-based search algorithm are then highlighted by relating their expected approximation quality in view of landscape features. We also discriminate between algorithms by revealing the most suitable one for subsets of instances. At last, we investigate the performance of a feature-based automated algorithm selection approach. By relying on low-cost features, we show that our recommendation system performs best in more than 90% of the considered mQAP instances.

1 Introduction

The multi-objective quadratic assignment problem (mQAP) [12,13] appears to be one of the most challenging problem from multi-objective combinatorial optimization. This is probably due to its intrinsic difficulties and the variety of mQAP instances from the literature, having different structures and properties in terms of problem size and data distributions, but also with respect to the number of objectives to be optimized, and their degree of conflict. Evolutionary multi-objective optimization (EMO) algorithms and other population-based multi-objective search heuristics are natural candidates to solve them. They range from dominance-based approaches to indicator- and decomposition-based

^{*} This research was conducted in the scope of the French/Japanese MODŌ international lab, and was partially supported by the French national research agency under Project ANR-16-CE23-0013-01.

refinements [6,28,29]. However, they have only been partially investigated for the mQAP, and focused mainly on problems with few (mostly 2) objectives [8,16,19]. There is obviously no single method that is more suitable for all problems, and multi-objective problems are no exception. As such, in the panorama of EMO algorithms, it remains unclear if and how problem characteristics result in differences in the performance of multi-objective selection strategies, and what actually makes an algorithm efficient or not when solving a given problem.

Landscape analysis [17] has emerged as a valuable methodology for examining the properties of optimization problems and their effect on search performance. Based on high-level landscape features, it becomes possible to improve our understanding of problems and algorithms, and also to predict algorithm performance, eventually leading to automated algorithm selection [11,22]. There is a large body of literature on single-objective landscape analysis [23], including for the quadratic assignment problem [4,17,21,25]. However, the literature on multi-objective landscapes is more scarce. Interestingly, most papers deal with the mQAP, being about properties from the Pareto set [12,20] or from the solution space [8,9]. However, previous studies were once again mostly devoted to problems with few objectives (mostly 2, sometimes 3), and often require the solution space or the Pareto set to be exhaustively enumerated, making them impractical for prediction. At last, existing multi-objective features were not always related to search performance, and never used for automated algorithm selection.

In a recent paper [15], we revised landscape features for multi-objective combinatorial optimization by building upon those previous studies, and by deriving additional low-cost landscape features that were revealed as highly impactful for EMO algorithms. In this paper, we are interested in analyzing the impact of mQAP instance characteristics on higher-level landscape features, such as ruggedness and multimodality. We also aim at clarifying the impact of mQAP landscape features on problem difficulty and search performance, and at examining if a difference in feature values implies any difference in the performance of EMO algorithms. Our contributions can be summarized as follows:

- (1) We characterize the landscape of large-scale mQAP instances with different properties by means of local multi-objective features from [15];
- (2) We relate mQAP landscape features with the performance of a dominance-, an indicator-, and a decomposition-based EMO algorithm [6,29,28];
- (3) We investigate the performance of feature-based automated algorithm selection by measuring its ability to discriminate between EMO algorithms, and by carefully calibrating the budget allocated to features and search.

The paper is organized as follows. Section 2 gives the necessary background on multi-objective optimization and EMO algorithms. Section 3 presents the mQAP and the instance dataset considered in our analysis. Section 4 introduces multi-objective landscape features, studies how they correlate with one another and with algorithm performance, and highlights their importance to explain search difficulty. Section 5 investigates the prediction accuracy of a feature-based automated algorithm selection system by paying a particular attention to the cost of features. Section 6 concludes the paper and discusses further research.

2 Multi-objective Optimization

2.1 Definitions

Let us consider an objective function vector $f: X \mapsto Z$ to be minimized. Each solution from the solution space $x \in X$ maps to a vector in the objective space $z \in Z$, with $Z \subseteq \mathbb{R}^m$, such that $z = f(x)$. In multi-objective combinatorial optimization, the solution space X is a discrete set. Given two objective vectors $z, z' \in Z$, z is dominated by z' iff for all $i \in \{1, \dots, m\}$ $z'_i \leq z_i$, and there is a $j \in \{1, \dots, m\}$ such that $z'_j < z_j$. Similarly, given two solutions $x, x' \in X$, x is dominated by x' iff $f(x)$ is dominated by $f(x')$. An objective vector $z^* \in Z$ is non-dominated if there does not exist any $z \in Z$ such that z^* is dominated by z . A solution $x^* \in X$ is Pareto optimal (PO), or non-dominated, if $f(x)$ is non-dominated. The set of PO solutions is the Pareto set (PS); its mapping in the objective space is the Pareto front (PF). One of the main challenges in multi-objective optimization is to identify the PS, or a good approximation of it for large-size and complex problems. A number of EMO and other multi-objective heuristics have been designed to this end since the late eighties [3,5].

2.2 EMO Algorithms

We conduct our analysis on three EMO algorithms: NSGA-II, IBEA, MOEA/D. They were selected as representatives of the state-of-the-art in the EMO field, covering dominance-, indicator-, and decomposition-based approaches, respectively. They differ in their selection mechanism, which is described below.

NSGA-II [5] is an elitist *dominance*-based EMO algorithm using Pareto dominance for survival and parent selections. At a given iteration, the current population P_t is merged with its offspring Q_t , and is divided into non-dominated fronts $F = \{F_1, F_2, \dots\}$ based on the non-dominated sorting procedure [10]. The front in which a given solution belongs to gives its rank within the population. Crowding distance is also calculated within each front. Selection is based on ranking, and crowding distance is used as a tie breaker. Survival selection consists in filling the new population P_{t+1} with solutions having the best (smallest) ranks. In case a front F_i overfills the population size, the required number of solutions from F_i are chosen based on their crowding distance. Parent selection for reproduction consists of binary tournaments between random individuals, following the lexicographic order induced by ranks first, and crowding distance next.

IBEA [29] introduces a total order between solutions by means of a binary quality *indicator* I . Its selection mechanisms is based on a pairwise comparison of solutions from the current population P_t with respect to I . A fitness value is assigned to each individual $x \in P_t$, measuring the “loss in quality” if x was removed from the current population; i.e. $Fitness(x) = \sum_{x' \in P \setminus \{x\}} (-e^{-I(x',x)/\kappa})$, where $\kappa > 0$ is a user-defined scaling factor. The survival selection mechanism is based on an elitist strategy that combines the current population P_t with its offspring Q_t . It iteratively removes the worst solution until the required population size is reached, and assigns the resulting population into P_{t+1} . Each time a

solution is deleted, the fitness values of the remaining individuals are updated. Parent selection for reproduction consists of binary tournaments between randomly chosen individuals. Different indicators can be used within IBEA. We here consider the binary additive ε -indicator ($I_{\varepsilon+}$), as defined by the original authors [29]: $I_{\varepsilon+}(x, x') = \max_{i \in \{1, \dots, m\}} \{f_i(x) - f_i(x')\}$. It gives the minimum value by which a solution $x \in P_t$ has to, or can be, translated in the objective space in order to weakly dominate another solution $x' \in P_t$.

MOEA/D [28] is a *decomposition*-based EMO algorithm that seek a high-quality solution in multiple regions of the objective space by decomposing the original (multi-objective) problem into a number of scalarizing (single-objective) sub-problems. Let μ be the population size. A set $(\lambda^1, \dots, \lambda^i, \dots, \lambda^\mu)$ of uniformly-distributed weighting coefficient vectors defines the scalarizing sub-problems, and a population $P = (x^1, \dots, x^i, \dots, x^\mu)$ is maintained such that each individual x^i maps to the sub-problem defined by λ^i . Different scalarizing functions can be used within MOEA/D. We here consider the weighted Chebyshev scalarizing function: $g(x, \lambda) = \max_{i \in \{1, \dots, m\}} \lambda_i \cdot |z_i^* - f_i(x)|$, such that x is a solution, λ is a weighting coefficient vector and z^* is a reference point. In addition, a neighboring relation is defined among sub-problems, based on the assumption that a given sub-problem is likely to benefit from the solution maintained in neighboring sub-problems. The neighborhood $\mathcal{B}(i)$ is defined by considering the T closest weighting coefficient vectors for each sub-problem i . At each iteration, the population evolves with respect to a given sub-problem. Two solutions are selected at random from $\mathcal{B}(i)$ and an offspring is produced by means of variation operators. Then, for each sub-problem $j \in \mathcal{B}(i)$, the offspring is used to replace the current solution x^j if there is an improvement in terms of the scalarizing function. The algorithm iterates over sub-problems until a stopping condition is satisfied.

3 Multi-objective Quadratic Assignment Problem

3.1 Problem Definition

Let us assume a given set of n facilities with e_{ij} being the flow between facilities i and j , and a given set of n locations with d_{ij} being the distance between locations i and j . The Quadratic Assignment Problem (QAP) [24] aims at assigning facilities to locations such that the sum of the products between flows and distances is minimal, and such that each facility is assigned to exactly one location, which is NP-hard [24]. The multi-objective QAP (mQAP) [12,13] considers m flow matrices under the same distance matrix, and can be stated as follows:

$$\min_{x \in X} \sum_{i=1}^n \sum_{j=1}^n d_{x_i x_j} e_{ij}^k \quad k \in \{1, \dots, m\} \quad (1)$$

where x_i gives the location of facility i in the current solution $x \in X$, and the solution space X is the set of all possible permutations $\{1, \dots, n\}$ (such that $|X| = n!$). By increasing the number flow matrices m , we can define bi-objective ($m = 2$), multi-objective ($m = 3$) and many-objective ($m \geq 4$) mQAP instances.

3.2 Instance Dataset

Knowles and Corne [13] provide an instance generator⁵ that can produce mQAP instances with different characteristics in terms of the number of variables (\mathbf{n}), the number of objectives (\mathbf{m}), the correlation among flow matrices (ρ), and the structure of flow matrices (\mathbf{type}): uniformly random (\mathbf{uni}) or real-like ($\mathbf{r1}$) flow values. Assuming that the dynamics and performance of EMO algorithms are impacted by these parameters, we consider a dataset covering a wide range of problems. In particular, we generate 1 000 mQAP instances following a design of experiments based on random latin hypercube sampling [2]. We consider a problem size in the range $\mathbf{n} \in \{30, \dots, 100\}$, a number of objectives $\mathbf{m} \in \{2, \dots, 5\}$, an objective correlation $\rho \in [-1, 1]$, and two instance types (\mathbf{uni} and $\mathbf{r1}$). We notice that, although the problem size and number of objectives are given, the type and the objective correlation are *unknown* in practice for unseen instances.

3.3 Algorithms Setting and Search Performance

We rely on an out-of-the-box implementation of the considered EMO algorithms with default parameters, as provided in the `jMetal 4.5` framework [7]. In terms of parameters, NSGA-II, IBEA and MOEA/D all use a population of size of 100, an exchange mutation with a rate of 0.2, and a partially-mapped crossover [10] with a rate of 0.95. Preliminary experiments revealed that using the partially-mapped crossover allows the search process to reach better quality in more than 90% of the cases, compared against the 2-point crossover used in a previous setting [15]. All the algorithms stop after 1 000 000 evaluations. We measure algorithm performance in terms of hypervolume (\mathbf{hv}) [30], and more particularly in terms of hypervolume relative deviation: $\mathbf{hvr}d = (\mathbf{hv}^* - \mathbf{hv})/\mathbf{hv}^*$, where \mathbf{hv}^* is the best-known hypervolume for the instance under consideration. The hypervolume measures the multi-dimensional area of the objective space covered by an approximation set, and is the only known strictly Pareto-compliant indicator [31]. The hypervolume reference point is set to the upper bound of objective values. For a given instance, each algorithm is executed 20 times, and the obtained $\mathbf{hvr}d$ values are averaged to estimate its expected performance. Significant difference between algorithms is also investigated in terms of statistical test.

4 Feature-based Landscape Analysis

We start our analysis by characterizing mQAP instances with relevant features from the literature. We rely on the multi-objective landscape features introduced in [15], and particularly on *local* features, based on sampling, that do not require any prior knowledge about the solution space enumeration and/or the Pareto set. We start by recalling their definition. Then, we measure how they relate with each other, and how they individually relate with search performance. At last, we assess their joint effect on performance in an attempt to highlight the main difficulties encountered by EMO algorithms when solving a mQAP instance.

⁵ <http://www.cs.bham.ac.uk/~jdk/mQAP/>

Table 1. Considered multi-objective landscape features from [15].

description	random walk		adaptive walk
correlation among objectives	f_cor_rws		–
average length of walks	–		length_aws
	average	autocorrelation	average
prop. dominated neighbors	#inf_avg_rws	#inf_r1_rws	#inf_avg_aws
prop. dominating neighbors	#sup_avg_rws	#sup_r1_rws	#sup_avg_aws
prop. incomparable neighbors	#inc_avg_rws	#inc_r1_rws	#inc_avg_aws
prop. locally non-dominated neighbors	#lnd_avg_rws	#lnd_r1_rws	#lnd_avg_aws
prop. supported locally non-dom. neighbors	#lsupp_avg_rws	#lsupp_r1_rws	#lsupp_avg_aws
solution’s hypervolume	hv_avg_rws	hv_r1_rws	hv_avg_aws
solution’s hypervolume difference	hvd_avg_rws	hvd_r1_rws	hvd_avg_aws
neighborhoods hypervolume	nhv_avg_rws	nhv_r1_rws	nhv_avg_aws

4.1 Multi-objective Landscape Features

The considered multi-objective landscape features are listed in Table 1. When adding the mQAP benchmark parameters (i.e., `type`, `n`, `m`, and ρ), this sums to a total of 30 features. We define the multi-objective landscape for a given mQAP instance as a triplet (X, f, \mathcal{N}) , such that X is the solution space (i.e., the set of all possible permutations $\{1, \dots, n\}$), $f: X \mapsto Z$ is the objective function vector defined in Eq. (1), and $\mathcal{N}: X \mapsto 2^X$ is a neighborhood relation based on the exchange operator, that consists in exchanging the locations of two facilities. The considered features are based on different measures computed on a *sample* of solutions extracted from a walk over the multi-objective landscape [15]. A *walk* is an ordered sequence of solutions $(x_0, x_1, \dots, x_\ell)$ such that $x_0 \in X$, and $x_t \in \mathcal{N}(x_{t-1})$ for all $t \in \{1, \dots, \ell\}$. During a *random walk* [27], there is no particular criterion to pick the neighboring solution at each step, a random neighbor is selected. The length of the walk ℓ_{rws} is a parameter: the longer the length, the better the features estimation. By contrast, during an *adaptive walk* [26], a dominating neighbor is selected at each step. The length ℓ_{aws} corresponds the number of steps performed until no further improvement is possible, and the walk falls into a Pareto local optimal solution (PLO) [18]. Multiple adaptive walks are typically performed to improve the features estimation.

Given an ordered sequence of solutions collected along a walk, we consider the following measures. For each solution, we sample its neighborhood, and we measure the proportion of dominated (**#inf**), dominating (**#sup**), and incomparable (**#inc**) neighbors. We also consider the proportion of non-dominated neighbors (**#lnd**), as well as the proportion of supported solutions therein (**#lsupp**). In addition, we compute the average hypervolume covered by each neighbor (**hv**), the average difference with the hypervolume covered by the current solution (**hvd**), and the hypervolume covered by all neighbors (**nhv**). For samples collected by means of a random walk, we compute both an average over all solutions from the walk and the first autocorrelation coefficient of the measures reported above. We also use solutions from the random walk to estimate the degree of correlation among the objectives (**f_cor_rws**). For adaptive walks, we simply compute average values for each measure, as well as the walk length ℓ_{aws} (**length_aws**), which is known to be a good estimator for the number of PLO [26].

Given η_{rws} random walks of length ℓ_{rws} , and a neighborhood sample size η_{neig} , the computational complexity for random walk features in terms of calls to the objective function is: $\eta_{\text{rws}} (1 + (1 + \ell_{\text{rws}}) \cdot \eta_{\text{neig}})$. Similarly, the computational complexity for adaptive walk features is: $\eta_{\text{aws}} ((1 + \ell_{\text{aws}}) \cdot \eta_{\text{neig}} + e_{\text{aws}})$, where η_{aws} is the number of adaptive walks, ℓ_{aws} is the number of steps before the adaptive walk falls into a PLO, and e_{aws} is the total number of evaluations performed for the walk to progress. However, we remark that `length_aws` alone is cheaper to compute, as it does not require any neighborhood exploration. Its complexity is just: $\eta_{\text{aws}} \cdot e_{\text{aws}}$. Similarly, the complexity of `f_cor_rws` alone is: $\eta_{\text{rws}} (1 + \ell_{\text{rws}})$. We also remark that η_{rws} , ℓ_{rws} , η_{aws} and η_{neig} must be defined by the user for feature estimation. By contrast, the expected value for ℓ_{aws} and e_{aws} , observed in average over instances from our dataset is 45 and 10 845, respectively.

4.2 Correlation among Landscape Features

In this section, we consider an expensive budget of $\eta_{\text{rws}} = 1$ random walk of length $\ell_{\text{rws}} = 1\,000$, and of $\eta_{\text{aws}} = 100$ independent adaptive walks, both using a neighborhood sample of $\eta_{\text{neig}} = 400$. Fig. 1 reports the correlation matrix of all features, as measured on the instance dataset. The correlation is measured in terms of the non-parametric Spearman rank correlation coefficient. The matrix highlights the similarities between features from mQAP, and their association with benchmark parameters. Interestingly, we remark that the number of variables and the instance type are only slightly correlated with landscape features, apart from autocorrelation measures, and the length of adaptive walks for `n`: the larger the search space, the longer the length. By contrast, the number of objectives and their degree of conflict are correlated with average dominance measures, and `m` is also highly positively correlated with average hypervolume measures. Unsurprisingly, there is a high association among average dominance measures, and among average hypervolume measures. This suggests that performing both random and adaptive walks is redundant for those features, and that considering a single walk type might allow us to save computations. At last, we remark in the last column that the correlation between the length of adaptive walks and other features is quite high overall, and we already infer that `length_aws` will be informative for characterizing problem difficulty.

4.3 Correlation of Landscape Features with Search Performance

We now report in Fig. 2 the Spearman correlation of each feature with the expected performance of the three considered EMO algorithms, measured in terms of hypervolume relative deviation (`hvr_d`). The corresponding scatterplots (with locally estimated scatterplot smoothing) for a selected subset of features are shown in Fig. 3 (others are not reported due to space restriction). Firstly, the effect of features on search difficulty has a similar trend for NSGA-II and IBEA, while being quite different for MOEA/D. In particular, the absolute correlation of each feature with the performance of MOEA/D is always below 0.5.

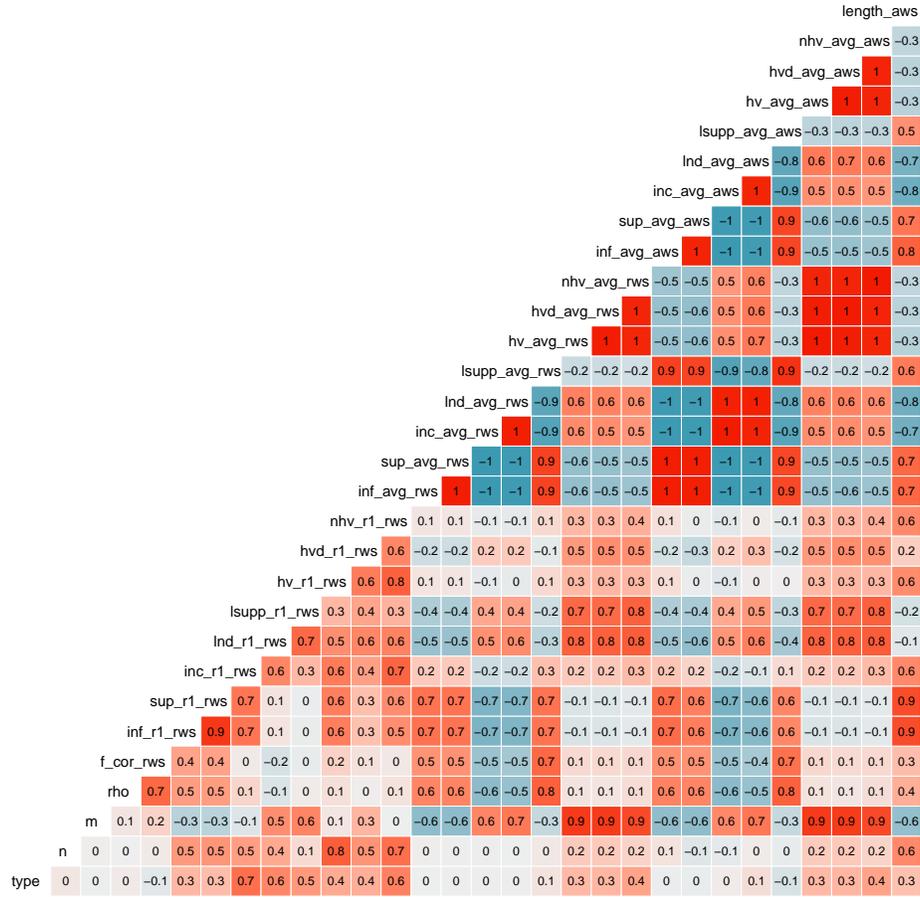


Fig. 1. Pairwise correlation among landscape features.

NSGA-II and IBEA are highly impacted by the number of objectives: their relative performance severely decreases with m , whereas MOEA/D performs almost constantly. Similarly, they perform better when average hypervolume measures are small, given that these are correlated with m , as pointed out earlier. IBEA is also impacted by average dominance measures: it performs best when there is not too few (nor too much) locally dominating points. Once again, it does not seem necessary to run both random and adaptive walks to measure average dominance and hypervolume values, given the similar impact of the corresponding features on search difficulty. By contrast, MOEA/D seems more impacted by autocorrelation measures, which quantify the ruggedness of the multi-objective landscape [15]: the rugger the landscape the less efficient MOEA/D. However, we argue that autocorrelation measures alter other algorithms as well; see, e.g., Fig. 3 (middle-left). Unfortunately, this effect is not captured by the correlation coefficients due to the particular trend of features against performance.

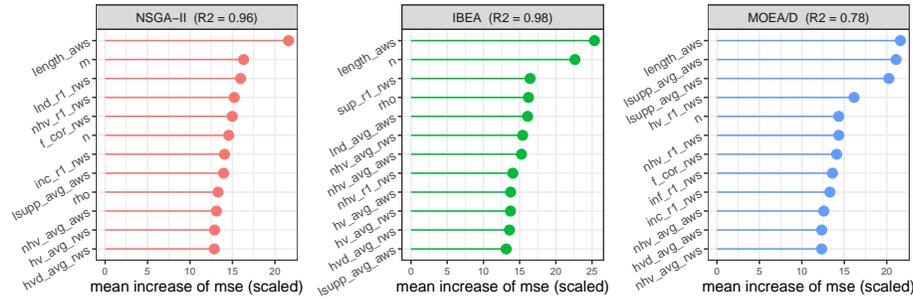


Fig. 4. Importance of landscape features for performance prediction of each algorithm.

arately for each algorithm. Given the non-linearity observed in the dataset, we employ a random forest model [1] from the `randomForest` R package [14] with default parameters. Due to the stochastic nature of random forests, we perform 100 independent trainings and report average values. The coefficient of determination of the models on training data is 0.96, 0.98, and 0.78 respectively, for NSGA-II, IBEA, and MOEA/D. This means that more than 75% of the variance in search performance between instances is explained by landscape features.

Beyond prediction accuracy, random forest models have the ability to render the relative importance of each feature for making accurate predictions. In particular, we consider the mean decrease of prediction accuracy after each split on a given predictor [1]: the larger the decrease, the more important the predictor. The importance scores are depicted in Fig. 4. For readability, only the 12 most important features are depicted for each algorithm, sorted in decreasing order of importance, from top to bottom. As conjectured in Section 4.2, `length_aws` turns out to be the most important feature for each algorithm. It relates to the multimodality of the landscape [15,26]: the longer the walk, the fewer the number of Pareto local optima, and the better search performance; see also Fig. 3 (bottom-left). For NSGA-II, the number of objectives `m` is the most important benchmark parameter, whereas it is the number of variables `n` for IBEA and MOEA/D. Some autocorrelation measures also appear for all algorithms, together with the degree of conflict among the objectives (whether `rho` or `f_cor_rws`). Interestingly, the proportion of supported non-dominated neighbors is particularly influential for the scalarization-based MOEA/D.

5 Feature-based Automated Algorithm Selection

5.1 Prediction Accuracy with Expensive Features

Let us examine the ability of landscape features to discriminate between the three algorithms. To do so, we now train a random forest *classification* model to predict whether NSGA-II, IBEA, or MOEA/D performs better, on average, for a given instance. The classification accuracy is reported in Table 2 for models based on different subset of features, corresponding to different costs. A feature-based classification model can predict the algorithm with the best average performance

Table 2. Classification error for different subset of features, measured on random sub-sampling cross-validation (100 repetitions, 80/20% split). Two values are reported: the error rate in predicting the algorithm with the best performance on average, and the error rate in predicting an algorithm that is not statistically outperformed by any other, according to a Mann-Whitney test at a significance level of 0.05 with Bonferroni correction. The dummy classifier always returns the most frequent algorithm.

subset of features	classification error	error predicting statistical best
{n, m}	.1962	.0332
{type, n, m, ρ }	.1197	.0072
{*_rws, n, m}	.1114	.0062
{*_aws, n, m}	.1125	.0065
{*_rws, length_aws, n, m}	.1089	.0056
{*_rws, *_aws, n, m}	.1077	.0063
{*_rws, *_aws, type, n, m, ρ }	.1078	.0063
random classifier	.6667	.3810
dummy classifier (MOEA/D)	.4200	.1040

in about 90% of the cases, and an algorithm which is not statistically outperformed by any other in more than 99% of the cases. This is significantly more accurate than a random classifier, a dummy classifier that always predicts the most-frequent best algorithm (here, MOEA/D), and a classifier based on benchmark parameters only. Interestingly, we did not find any significant difference in terms of prediction accuracy between a model using all features and a model using solely features based on random walk plus only the length of adaptive walks. This might actually reduce the computational cost of algorithm selection.

Features importance for algorithm selection is depicted in Fig. 5. The length of adaptive walks is, once again, the most important feature. The subsequent features have a very similar score, and cover complementary landscape characteristics, ranging from autocorrelation coefficients, to average dominance and hypervolume measures and benchmark parameters. Most notably, important adaptive walk features almost always have their random walk counterpart, whether it is for dominance or hypervolume measures.

5.2 Low-cost Features Subtracted from Search Budget

We conclude our analysis by investigating the performance of a feature-based automated EMO algorithm selection method (AUTO-EMOA for short), while taking the budget allocated to the feature computation into account. Given the results presented above, we focus on a classification model based on features from random walk sampling (**rws**), together with **length_aws** and problem parameters that are given in practical scenarios (dimensions **n** and **m**). In contrast to the previous setting, we now consider a *low-cost* budget for features computation: $\eta_{rws} = 1$ random walk of length $\ell_{rws} = 200$ using a sample of $\eta_{neig} = 100$ neighbors at each step, and $\eta_{aws} = 1$ adaptive walk for estimating **length_aws** only. By measuring the one-to-one correlation between expensive and low-cost features (not reported), we remark that it is always larger than 0.85, apart from locally supported and hypervolume autocorrelation features (between 0.58 and 0.75), that were not detected as important previously.

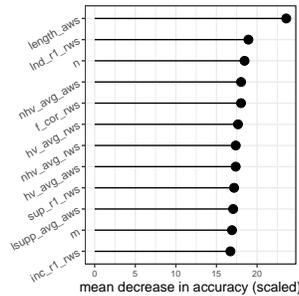


Fig. 5. Importance of features for algorithm selection.

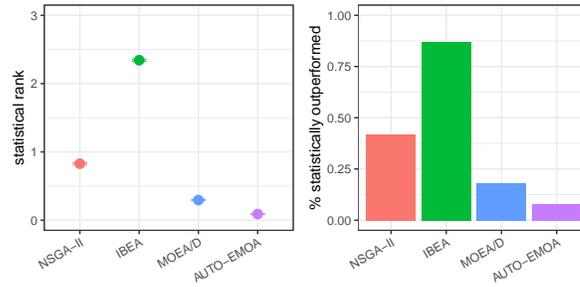


Fig. 6. Performance of AUTO-EMOA compared against other algorithms.

The total computation of the considered low-cost features sums up to 30 946 evaluations, in average, per instance. Consequently, we deduce 50 000 ($> 30\,946$) evaluations from the search process allocated to AUTO-EMOA. In other words, we compare AUTO-EMOA with a search budget of 950 000 evaluations against NSGA-II, IBEA, and MOEA/D with a search budget of 1 000 000 evaluations. Results from 100 repetitions of random sub-sampling cross-validation with a 80/20% split are presented in Fig. 6. The statistical rank of AUTO-EMOA is 0.09 on average, more than three times lower than the best standalone approach (MOEA/D, with 0.29). Among all instances seen during cross-validation, AUTO-EMOA was *not* significantly outperformed by any other approaches on 92% of the cases (82% for MOEA/D). As such, deducing a small part of the budget allocated to the search process for feature computation appears to be beneficial in order to gain knowledge about the tackled problem, and make better-informed decision on the appropriate multi-objective search strategy to apply for solving it.

6 Conclusions

In this paper, we analyzed the landscape of large-scale bi-, multi- and many-objective mQAP instances, and highlighted the relationship between landscape features and the performance of a dominance-, indicator-, and decomposition-based EMO algorithm. Our study highlights that algorithms are not only impacted by the number of objectives, but that the ruggedness and multimodality of the multi-objective landscape are also crucially important to properly explain search performance. An automated algorithm selection model also revealed the ability of multi-objective landscape features to discriminate between EMO algorithms. By simply allocating less than 1% of the budget to analyze the landscape of a given instance, our recommendation system was shown to perform best on more than 90% of instances under the scenario considered for validation.

Further research includes the investigation of other landscapes, including multi-objective continuous functions that require particular walks for sampling the solution space when computing the features. Additionally, we plan to consider additional EMO algorithms, and in particular highly-configurable frameworks for which we infer that feature-based algorithm configuration is essential.

References

1. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (2001)
2. Carnell, R.: lhs: Latin Hypercube Samples (2018), <https://CRAN.R-project.org/package=lhs>, r package version 0.16
3. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, second edn. (2007)
4. Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of the quadratic assignment problem. In: *Congress on Evolutionary Computation (CEC 2010)*. pp. 1–8. IEEE (2010)
5. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002)
7. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software* **42**, 760–771 (2011)
8. Garrett, D., Dasgupta, D.: Analyzing the performance of hybrid evolutionary algorithms for the multiobjective quadratic assignment problem. In: *IEEE Congress on Evolutionary Computation (CEC 2006)*. pp. 1710–1717 (2006)
9. Garrett, D., Dasgupta, D.: Multiobjective landscape analysis and the generalized assignment problem. In: *Learning and Intelligent OptimizatioN (LION 2)*. Lecture Notes in Computer Science, vol. 5313, pp. 110–124. Springer, Trento, Italy (2007)
10. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, MA, USA (1989)
11. Kerschke, P., Hoos, H., Neumann, F., Trautmann, H.: Automated algorithm selection: Survey and perspectives. *Evolutionary Computation* **27**(1), 3–45 (2019)
12. Knowles, J., Corne, D.: Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem. In: *Soft Computing Systems: Design, Management and Applications*. vol. 2002, pp. 271–279 (2002)
13. Knowles, J., Corne, D.: Instance generators and test suites for the multiobjective quadratic assignment problem. In: *Evolutionary Multi-Criterion Optimization (EMO 2003)*. Lecture Notes in Computer Science, vol. 2632, pp. 295–310. Springer, Faro, Portugal (2003)
14. Liaw, A., Wiener, M.: Classification and regression by randomforest. *R News* **2**(3), 18–22 (2002)
15. Liefvooghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., Tanaka, K.: Landscape-aware performance prediction for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* pp. 1–1 (2019), early access
16. López-Ibáñez, M., Paquete, L., Stützle, T.: Hybrid population-based algorithms for the bi-objective quadratic assignment problem. *Journal of Mathematical Modelling and Algorithms* **5**(1), 111–137 (2006)
17. Merz, P.: Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation* **12**(3), 303–325 (2004)
18. Paquete, L., Schiavinotto, T., Stützle, T.: On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* **156**(1), 83–97 (2007)
19. Paquete, L., Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research* **169**(3), 943–959 (2006)

20. Paquete, L., Stützle, T.: Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In: Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications. Lecture Notes in Economics and Mathematical Systems, vol. 618, pp. 69–77. Springer (2009)
21. Pitzer, E., Beham, A., Affenzeller, M.: Correlation of problem hardness and fitness landscapes in the quadratic assignment problem. In: Advanced Methods and Applications in Computational Intelligence, pp. 165–195. Springer (2014)
22. Rice, J.R.: The algorithm selection problem. *Advances in Computers* **15**, 65–118 (1976)
23. Richter, H., Engelbrecht, A. (eds.): Recent Advances in the Theory and Application of Fitness Landscapes. Emergence, Complexity and Computation, Springer (2014)
24. Sahni, S., Gonzalez, T.: P-complete approximation problems. *Journal of the ACM* **23**(3), 555–565 (1976)
25. Smith-Miles, K., Lopes, L.: Measuring instance difficulty for combinatorial optimization problems. *Computers & Operations Research* **39**(5), 875 – 889 (2012)
26. Verel, S., Liefvooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multi-objective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research* **227**(2), 331–342 (2013)
27. Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics* **63**(5), 325–336 (1990)
28. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* **11**(6), 712–731 (2007)
29. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Conference on Parallel Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science, vol. 3242, pp. 832–842. Springer, Birmingham, UK (2004)
30. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* **3**(4), 257–271 (1999)
31. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132 (2003)