

An investigation of clustering strategies in many-objective optimization: I-Multi as a case study

Olacir R. Castro Jr · Aurora Pozo · Jose A. Lozano · Roberto Santana

Received: date / Accepted: date

Abstract A variety of general strategies have been applied to enhance the performance of multi-objective optimization algorithms for many-objective optimization problems (those with more than three objectives). One of these strategies is to split the solutions to cover different regions (clusters) and apply an optimizer to each region with the aim of producing more diverse solutions and achieving a better distributed approximation of the Pareto front. However, the effectiveness of clustering in this context depends on a number of issues, including the characteristics of the objective functions. In this paper we show how the choice of the clustering strategy can greatly influence the behavior an optimizer. We try to find a relation between the characteristics a of multi-objective optimization problem (MOP) and the efficiency of the use of a clustering type in its resolution. Using as a case study, the Iterated Multi-swarm (I-Multi), a recently introduced multi-objective particle swarm optimization (MOPSO) algorithm, we scrutinize the impact that clustering in different spaces (of variables, objectives, and a combination of both) can have on the approximations of the Pareto front. Furthermore, using two difficult function benchmarks of problems of up to 20 objectives, we evaluate the effect of using different metrics for determining the similarity between the solutions during the clustering process. Our results confirm the important effect of the clustering strategy on the behavior of multi-objective optimizers. Moreover, we present evidence that some problem characteristics can be used to select the

This work was supported by CNPq, National Council for Scientific and Technological Development - Brazil (Productivity Grant Nos. 305986/2012-0 and Program Science Without Borders Nos. 200040/2015-4) and by IT-609-13 program (Basque Government) and TIN2013-41272P (Spanish Ministry of Science and Innovation).

Olacir R. Castro Jr · Aurora Pozo
Computer Science Department, Federal University of Paraná, Avenida Coronel Francisco
Heráclito dos Santos, 210, 81531-970, Curitiba, Brazil.
E-mail: olacirjr@gmail.com, aurora@inf.ufpr.br

Jose A. Lozano · Roberto Santana
Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, Uni-
versity of the Basque Country (UPV/EHU), Paseo Manuel de Lardizábal 1, 20080, San Se-
bastián-Donostia, Spain
E-mail: {ja.lozano, roberto.santana}@ehu.eus

most effective clustering strategy, significantly improving the quality of the Pareto front approximations produced by I-Multi.

Keywords Many-objective · clustering · MOPSO · I-Multi · distance metrics · PSO

1 Introduction

The design of efficient approaches for many objective optimization problems (MaOPs) has become an active research area in meta-heuristics. A great number of approaches (Li et al 2015a) and algorithms have been proposed recently to treat these problems. One of these approaches is the extension of particle swarm optimization (PSO) (Kennedy and Eberhart 1995) algorithms to deal with these problems. Multi-objective particle swarm optimization (MOPSO) algorithms have been successfully applied to continuous MaOPs (Britto et al 2013; Britto and Pozo 2012; Castro Jr. et al 2015). Among other characteristics, MOPSOs can be very fast, are particularly good at finding diverse sets of solutions, and are relatively simple to understand and implement.

One of the recent developments in MOPSOs is the application of clustering strategies that search more efficiently for non-dominated solutions in different regions of the search space (Britto et al 2013; Mostaghim and Teich 2004; Pulido and Coello Coello 2004; Zhang et al 2011). The rationale of applying clustering strategies is similar to that behind the application of niching methods in single-objective evolutionary algorithms (EAs) (Mahfoud 1995). However, the problem is more difficult due to several characteristics, such as the existence of conflicting objectives, the explosion in the number of non-dominated solutions for MaOPs (Ishibuchi et al 2008) and the discontinuities and deceptive nature of some Pareto fronts. Therefore, the application of clustering strategies in MaOPs is itself a research problem. Nevertheless, even if the application of clustering in populational meta-heuristics is widespread, and the benefits of these methods for the search can be significant, the question of how the choice of the clustering strategies should be made has not received due attention. We investigate this issue in the paper, showing that the specific choice of the clustering strategy used by a MOPSO can have an important impact in the behavior of the algorithm and in the quality of the Pareto fronts generated by the MOPSOs.

We focus on finding answers for the following questions: 1) What is the most efficient clustering strategy for MaOPs: clustering in the space of decision variables, objectives, or both? 2) Is it possible to characterize the type of problems (benchmark functions) for which one type of clustering is better than the others? 3) What is the influence of the similarity metric used for clustering in the behavior of the algorithms? 4) Among the metrics compared, which one contributes the most to obtaining good solutions? What is the best metric among the compared?

We investigate these issues using I-Multi (Britto et al 2013), a recently introduced MOPSO that has been successfully applied to MaOPs and includes as one of its distinguished features the application of a clustering step. I-Multi is a paradigmatic example of a class of MOPSOs specifically conceived to deal with MaOPs. I-Multi's design, in particular the incorporation of multiple swarms to cover different areas of the search space, allows the algorithm to naturally extend

the strength of PSO approaches to multi-objective domains. Although we focus on I-Multi, our study is also suitable for other MOPSOs that incorporate clustering of the solutions as a component of the search. Moreover, there are two other reasons that make our research relevant to the field.

The first reason is that some of our findings, in particular those related to the influence of the modalities of difficulty of the functions on the behavior of the different clustering strategies, can be useful to understand the behavior of other optimizers that incorporate clustering (Bosman and Thierens 2002; Pelikan et al 2005; Pulido and Coello Coello 2004).

The second reason that makes our research valuable beyond the scope of MOPSOs, is that clustering in the different spaces can serve as a source of information about the type of relationships between variables that arise in the problem, and between objectives and variables. Recent research in MOPs (Fritsche et al 2015; Karshenas et al 2014; Ma et al 2015) has emphasized the importance of capturing, modeling, and using the different types or relationships between the variables and objectives of a problem. Clustering is commonly applied as an essential tool in exploratory data mining¹ and it can be useful to identify patterns between the grouped solutions and extract general rules describing these solutions. When clustering Pareto-optimal solutions, we can expect to extract important patterns as well. Hence, determining the characteristics of MOPs that make a particular type of clustering strategy more effective for optimization could also lead to a better understanding of the patterns that arise in the (clustered) optimal solutions for these MOPs.

After conducting extensive experiments using two families of difficult benchmark functions of up to 20 objectives, we clearly identified clustering in the objective space as the most efficient clustering strategy in most of the cases. However, we also identify functions for which clustering in the decision space leads to better approximations of the Pareto front. Further examination of these cases allows us to identify two modalities of difficulty that are particularly suited to be treated using clustering in the decision space. These are: bias and deception. Our study of the influence of the clustering metrics reveals a clearer scenario in which the use of other metrics, different to the commonly applied Euclidean distance, does not produce significant improvements in the general case. The authors of this paper have not found any previous work where the performance of different clustering strategies for MOPs is linked to the modalities of difficulty of the functions. Similarly, we did not find any previous study on the impact of the similarity metrics on the behaviors of the algorithms.

The remainder of this paper is organized as follows: Section 2 presents some preliminary concepts and introduces the notation used throughout this work. The I-Multi algorithm used here as a case study is explained in Section 3 and some representative clustering algorithms related to ours are presented in Section 4. Section 5 describes the clustering strategies investigated, and Section 6 shows the experimental study conducted to compare these strategies. Finally, Section 7 presents our conclusions.

¹ "give a reference"

2 Elementary concepts

In this section, we introduce the notation used and present the concepts of multi and many-objective optimization. Moreover, we explain what the main characteristics of a good Pareto front are, and discuss some challenges and alternatives to achieve these characteristics.

2.1 Many-objective optimization

Multi-objective optimization problems (MOPs) require the simultaneous optimization (maximization or minimization) of two or more objective functions. These objectives are usually in conflict, so these problems have a set of optimal solutions, instead of just one as in single objective optimization problems. This optimal set of solutions is usually found using the non-dominance relation.

A general MOP without constraints can be defined as optimizing $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, where $\mathbf{x} \in \Omega$ is an n -dimensional decision variable vector $\mathbf{x} = (x_1, \dots, x_n)$ from a universe Ω , and m is the number of objective functions.

An objective vector $\mathbf{u} = \mathbf{f}(\mathbf{x})$ dominates a vector $\mathbf{v} = \mathbf{f}(\mathbf{y})$, denoted by $\mathbf{u} \preceq \mathbf{v}$ (in case of minimization) if \mathbf{u} is partially less than \mathbf{v} i.e., $\forall i \in \{1, \dots, m\}, u_i \leq v_i \wedge \exists j \in \{1, \dots, m\} : u_j < v_j$.

A vector \mathbf{u} is non-dominated if there is no \mathbf{v} that dominates \mathbf{u} . Given that $\mathbf{u} = \mathbf{f}(\mathbf{x})$, if \mathbf{u} is non-dominated, then \mathbf{x} is Pareto optimal. The set of Pareto optimal solutions is called the Pareto optimal set, and the image of these solutions in the objective space is called the Pareto front (Coello et al 2006).

Many-objective optimization problems (MaOPs) are a type of MOPs that present more than three objective functions to be optimized simultaneously. Several studies have indicated that Pareto based algorithms scale poorly in MaOPs (Britto et al 2013; Britto and Pozo 2012; Ishibuchi et al 2011). The main reason for this is the number of non-dominated solutions which increases greatly with the number of objectives. Consequently, the search ability is deteriorated because it is not possible to impose preferences for selection purposes.

Other issues faced when dealing with MaOPs are: The crowdingness of solutions becomes difficult to gauge, since estimation operators such as the Crowding Distance (Deb et al 2000) become ineffective as the number of objectives increase (Kukkonen and Deb 2006); Dominance-resistant solutions may potentially degrade the search (Ikeda et al 2001); Performance metrics such as hypervolume become computationally expensive to calculate; Visualization of the objective space becomes extremely challenging in comparison to 2 and 3-objective problems.

2.2 Well distributed Pareto fronts

In multi-objective optimization, the two most important requirements to a Pareto set generated by an optimization algorithm are *convergence* and *diversity* (Adra 2007). Convergence means that the approximation of the Pareto front generated for a MOP is as close as possible to the true Pareto front. Diversity means that, since a single ideal solution in a MOP does not exist, and a trade-off surface can

potentially present an infinite number of solutions, a good approximation needs to be well spread and to uniformly cover wide areas of the Pareto front.

However, with an increase in the number of objectives, the size of the objective space and the surface of the Pareto front can increase greatly, making it much harder for an optimizer to achieve good convergence and diversity.

A possible alternative to deal with such problems is to use multiple populations (or swarms). These populations should be well spread over the entire search space in order to increase the diversity of solutions. Moreover, when each single population concentrates in a small portion of the search space, the individuals are able to specialize, leading to better convergence. Thus, the key question is how to make a partition of the search space that allows an efficient spread of the populations.

3 I-Multi algorithm

The PSO (Kennedy and Eberhart 1995) algorithm is an optimization technique inspired by bird flocking behavior. In PSO, the movement of each individual (or particle) is affected by its own experience and that of its neighbors (we use a fully connected topology). The approach uses the concept of population (or swarm) and a measure of performance similar to the fitness value used in evolutionary algorithms (Coello et al 2006).

Different algorithms have been proposed to extend the PSO to solve multi-objective problems, thus creating a MOPSO (Britto and Pozo 2012; Nebro et al 2009). Most approaches differ from the single-objective PSO in the use of an external archive (repository) to store the best (non-dominated) solutions found so far. Another common difference is the leader selection scheme, which has to be chosen from a set of equally good leaders according to some criterion. These questions become more difficult since, as the number of non-dominated solutions increases, an archiving method will need to prune the repository (according to a predefined criterion) and keep only a bounded number of solutions.

I-Multi (Britto et al 2013) is a recently introduced MOPSO designed to deal with many-objective problems. A distinguished characteristic of I-Multi is that it uses multiple swarms to cover different areas of the objective space. Its search procedure can be divided in two phases: diversity and multi-swarm searches. A pseudocode of the I-Multi is presented in Algorithm 1.

The first phase of I-Multi is called diversity search. In this step, a traditional single-swarm MOPSO is executed by a predefined number of iterations in order to obtain a set of well-distributed (diversified) non-dominated solutions (basis front) for multi-swarm initialization. In this phase we use the the Speed-constrained Multi-objective PSO (SMPSO) (Nebro et al 2009) algorithm with the Multi-level Grid Archiving (MGA) (Laumanns and Zenklusen 2011) as archiver. SMPSO is an efficient MOPSO that presents a velocity constriction mechanism as presented in (Clerc and Kennedy 2002). The mechanism is based on a factor χ that varies according to the values of the influence coefficients of personal and global leaders (C_1 and C_2 respectively). The global leader selection method of SMPSO uses a binary tournament based on the crowding distance metric from (Deb et al 2000), and its original archiving strategy also uses the crowding distance. The parameters used in the I-Multi algorithm can be found in Table 2.

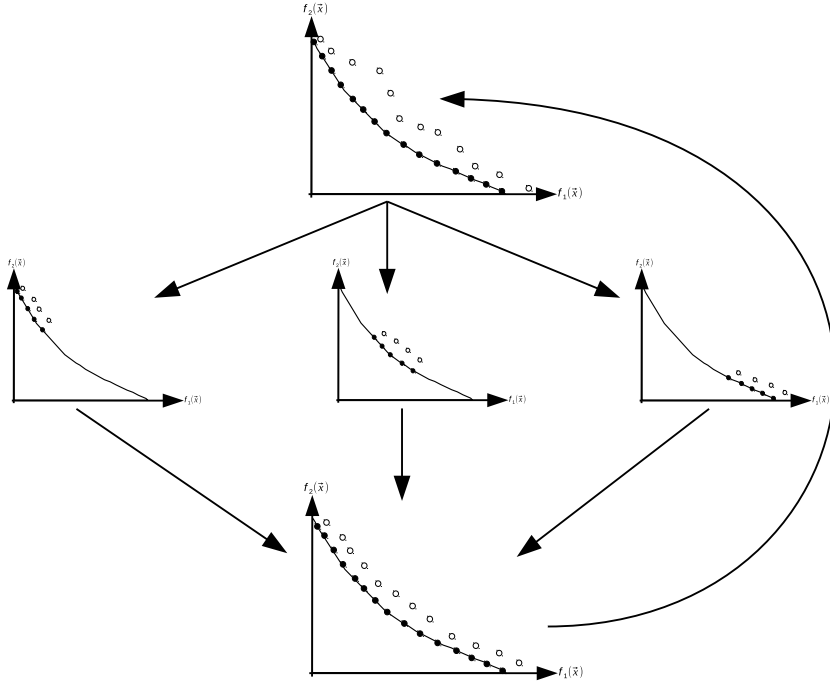


Fig. 1 Representation of the I-Multi algorithm.

The multi-swarm phase of I-Multi, begins by using the K-Means algorithm (Hartigan and Wong 1979) for clustering the solutions contained in the basis front (F_b) to generate a predefined number of sub-swarms (NS). The solutions from each cluster compose the initial repository of each sub-swarm (F_k) and the centroid of the cluster is used as seed (S_k) for the swarm. Around each seed (within a specified search region (\mathcal{V})), a set of solutions is randomly generated as particles of the sub-swarm. For a predefined number of iterations, each sub-swarm runs independently, using the SMPSO with the Ideal archiver (Britto and Pozo 2012) to enhance its convergence. After that, the repository of each sub-swarm is integrated to the basis front, so only the non-dominated solutions regarding all repositories are kept. At the end of this process, the basis front is split into sub-swarms as before. This process of joining and splitting the fronts is called split iteration, and it is repeated a predefined number of times (SI). This process enables an indirect communication between the sub-swarms.

The process of a split iteration is depicted in Figure 1, where at first there is a single swarm whose solutions in the repository are represented as black circles and the particles are presented as white circles. Next repository (basis front) of the single swarm is split into a predefined number of clusters, where the solutions clustered together in each cluster becomes the initial repository of a sub-swarm and the centroid of this cluster is used as seed for this sub-swarm. To complete each sub-swarm, a set of particles is randomly generated around the seed. After a predefined number of runs, all the non-dominated solutions regarding all clusters are combined again to form a new basis front and start a new split iteration.

Algorithm 1: I-Multi

```

// Phase1: Diversity search
Fb=Run-MGA-SMPSO()
// Phase 2: Multi-swarm search
for s=1 to SI do
    S,F=SplitFront(Fb)
    V=calculateInterval()
    for k=1 to NS do
        Pk=initializePop(Sk,V)
        Fk=Run-Ideal-SMPSO(Pk,Sk,V,Fk)
    end for
    Fb=Non-dominated(F)
end for
return Fb

```

4 Related work

Clustering is a useful mechanism to maintain diversity. It can be used to keep groups of solutions in different regions or to aid in archiving for multi-objective optimizers. Moreover it can be used to ensure convergence, since similar solutions grouped together can specialize in smaller areas of the search space. Both single-objective and multi-objective optimization algorithms are known for taking advantage of clustering. In this section we review some related work in which clustering plays an important role in improving the results obtained by the algorithms. Our goal here is not to make an extensive literature review, but to discuss a number of representative algorithms that cover different facets of the impact of clustering in optimization algorithms. Table 11 shows a summary of the algorithms reviewed. Each column of the table displays the main elements considered for our classification of the algorithms: 1) Space in which clustering is conducted, 2) Clustering metric, 3) Number of objectives of the optimization problems addressed, and 4) Clustering method.

4.1 Clustering in single objective optimization

Clustering is usually employed in single-objective optimization algorithms as a way of splitting the population (or swarm) to obtain more diversity through exploring larger areas of the decision space. Moreover, convergence can be achieved through specialization of subsets of solutions in smaller areas of the search space. Examples of such strategy are presented in (Liang et al 2015; Yen and Daneshyari 2006), where two multi-swarm PSOs that use clustering to split the swarms are presented.

Another popular use of clustering in optimization algorithms is to promote diversity and help identifying interactions between variables of the problems. Estimation of Distribution Algorithms (EDAs) (Larrañaga and Lozano 2002) that employ this technique are (Emmendorfer and Pozo 2009; Pelikan and Goldberg 2000; Tsuji et al 2006; Bosman and Thierens 2002).

4.2 Clustering in multi-objective optimization

Multi-objective optimizers take advantage of clustering the solutions as well. In this case, diversity maintenance can be even more important than in single-objective, since the algorithms have to ensure well-spread solutions in the objective space to achieve a good coverage of the Pareto front, and in the decision space to avoid local optimal regions.

As in single-objective, a popular use for clustering in multi-objective algorithms is to split the population (or swarm) in several subgroups, examples of these approaches can be found in (Pulido and Coello Coello 2004; Zhang and Xue 2007; Benameur et al 2009).

Multi-objective EDAs that employ clustering are available in the literature as well. In this case, usually a separate probabilistic model is built for each cluster, since it is assumed that the solutions inside each cluster share important characteristics. Examples of these algorithms can be found in (Pelikan 2005; Okabe et al 2004)

Another promising use for clustering in multi-objective optimization is to aid in the diversity preservation of archiving methods. The most notable example of this class is the well-known Strength Pareto Evolutionary Algorithm (SPEA), proposed by Zitzler and Thiele (1999). It clusters the solutions into a predefined number of clusters and only one representative solution per cluster is kept.

4.3 Clustering spaces, similarity measures and quality indicators

While doing this review, we were not able to find any work that investigates the relation between the performance of the optimization algorithm and the space used for clustering the solutions (decision or objective). Furthermore, we did not find any previous works investigating the impact of using different similarity measures for the clustering of solutions in an optimization algorithm.

Regarding the use of clustering quality indicators, Jin and Sendhoff (2004) use silhouette analysis to verify the quality of the clusters, but they do not use the clustering directly in the Evolution Strategy (ES). Instead, they use the clustering to determine if a solution should be evaluated by the original fitness function or by a surrogate model. In (Sindhya et al 2013), a clustering mechanism is employed and a clustering quality measure is calculated as an approximate quality measure of the diversity of the population. If the diversity of the population is considered too low, a diversity enhancement module is activated. Despite using a clustering quality measure, in this paper the authors are only interested in estimating the diversity of solutions in the population, and not to evaluate the clustering quality itself for other purposes. Benameur et al (2009) used a measure called the normalized partition entropy to compare the quality of different clustering runs to choose the best of these runs.

These works apply clustering quality measures to optimization algorithms in different ways. Nevertheless, none of them correlates the quality of the clustering with the quality of the solutions generated by the algorithms as is done in our work.

5 Clustering strategies for multi-objective problems

In this section we present the different clustering strategies that have been implemented as part of I-Multi, explaining the rationale behind their choice.

5.1 Components of the clustering algorithms

The following elements influence the behavior of the clustering strategies when used within many-objective optimizers:

1. Clustering space explored: decision space, objective space or a combination of both spaces.
2. Similarity measure employed to compute the clusters.
3. Number of clusters.
4. Type of clustering algorithm.

In this paper we focus on the examination of the first two strategies. As illustrated in the review of related work, clustering in the decision and objective spaces is extensively applied, but the impact of the choice on the behavior of the EA is usually not addressed. Similarly, the issue of the similarity measure applied is commonly overlooked. The influence of the number of clusters has been investigated in previous works (Britto et al 2013; Castro Jr. et al 2015) and the effect that the type of clustering algorithm may have in the search is left for future work.

In an ideal optimization problem, a high correlation between decision and objective spaces is expected. In other words, solutions close in the objective space are close in the decision space as well. However, there are problems where this correlation is not high, like problems where some of the variables are redundant or marginally influence the optimization function. For such problems the space where the solutions are clustered plays an important role in the optimization process.

In the original I-Multi paper (Britto et al 2013), the objective of clustering in the decision space is mostly to increase the convergence of the algorithm by means of specialization. Since each cluster has very similar solutions, the interactions among their decision vectors produce small perturbations, and consequently the exploitation of a small part of the decision space is increased.

Here we propose to change the space of clustering from decision to objective. By making this change, we expect to achieve more diversity in both spaces, since each cluster will be concentrated in a different region of the Pareto front. However, their decision variables will not necessarily be very similar, hence the interactions among these solutions are more likely to generate greater perturbations, leading to the exploration of a larger part of the decision space. By combining decision and objective spaces, both contribute to the distance calculation. Then it is expected that the solutions in the cluster have a good balance between both distances. Consequently a good trade-off between convergence and diversity is expected.

Besides the correlation between the different spaces, the similarity of two solutions can be interpreted differently depending on the indicator used to measure this similarity. By changing the similarity metric, the shape and location of the clusters are changed as well, and consequently the metric used has an impact on the behavior of the search algorithm.

5.2 Clustering space

In this work we investigate two different alternative spaces to perform the clustering procedure in I-Multi: clustering in the objective space and in an alternative space that we called *both*, and is composed by a combination of objective and decision spaces.

In the traditional (decision space clustering) approach used by I-Multi, the centroids of the clusters are used as “seeds”, i.e., solutions around which the search region is defined. However, when doing clustering in the objective space, we cannot use the centroids of the clusters found by K-Means as seeds. Instead, in this approach we set the seed as the average of the decision variables of the solutions whose images (objective vectors) have been grouped in the cluster.

Our other proposed approach (*both*) uses a combination of both spaces in order to cluster the solutions. In this case, the K-Means algorithm is executed in the space defined by the concatenation of decision variables and objectives $\mathbf{c} = (u_1, \dots, u_m, x_1, \dots, x_n)$, where $\mathbf{u} = \mathbf{f}(\mathbf{x})$. m is the number of objectives and n is the number of decision variables.

Since this approach uses the decision variables as part of the clustering space, we use the last n elements of \mathbf{c} as the centroid of the swarm.

5.3 Measures of similarity

The metric used to evaluate the similarity between the solutions have an influence on the results of the clustering algorithm. The clustering metrics define different ways to look at the similarity relationships between the solutions. Previous work in the areas of machine learning and pattern recognition (Aggarwal et al 2001; Deborah et al 2015; Howarth and Rüger 2005) have acknowledged the impact that the choice of the distance metric have on different algorithms where computing the similarity between solutions is required. For instance, the so called *fractional distance* metrics have shown to significantly improve the effectiveness of clustering algorithms for high dimensional problems (Aggarwal et al 2001). As part of our study, we have selected a set of representative similarity metrics that include Euclidean distance (Bandyopadhyay and Saha 2012), the most commonly applied distance in optimization algorithms, and a set of other metrics extensively applied in other areas but rarely investigated in the context of optimization algorithms. One of the implicit questions we address is whether clustering algorithms that use such metrics can promote better results than those that employ the Euclidean metric. Therefore, as part of our study, we investigated a set of representative similarity metrics and the sensitivity of I-Multi to this choice.

The Minkowski distance (Bandyopadhyay and Saha 2012) between two vectors $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ is defined as:

$$\left(\sum_{i=1}^n |x_i - y_i|^k \right)^{\frac{1}{k}} \quad (1)$$

where k is a parameter of the Minkowski metric that defines a family of metrics. The effect of changing the value of k is displayed in Figure 2. In our experiments, we considered $k \in \{0.5, 2, 4, \infty\}$ as parameters of the Minkowski metric because

these values represent the most commonly applied distance (Euclidean) and a number of other metrics extensively applied in other areas, but rarely investigated in the context of optimization algorithms, moreover these values allow us to explore a variety of scenarios in terms of how to compute the similarity between the solutions, and in particular the weight given to the difference in the components of the vectors.

For $k = 2$, the Euclidean distance is obtained from Equation (1), and for $k = \infty$ we obtain the Tchebycheff (Bandyopadhyay and Saha 2012) distance. For $k = 0.5$, the distance is not a metric, since the triangle inequality is violated, however $k = 0.5$ exhibit properties that are midway between the properties of the Euclidean distance ($k = 2$) and ($k = 0$), a fact that makes it worth of investigation.

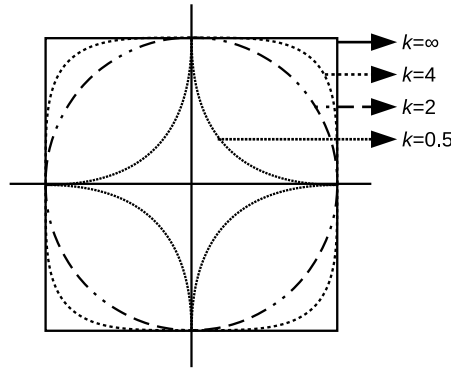


Fig. 2 Unit circles of the Minkowski metric with various values of k .

6 Experiments

The primary objectives of the experiments are the following: Q1) To find which of the clustering strategies produces the best results when used within I-Multi. Q2) To determine the impact that the choice of the distance metric has on the results of I-Multi. Q3) To identify or unveil any type of casual relationship between the characteristics of the optimization problem (number of objectives, deception, multi-modality or bias) and the behavior of I-Multi when the different clustering strategies are applied.

Finding an answer to question Q1 will contribute to a better understanding of the behavior of the I-Multi algorithm and MOPSOs in general. Similarly, investigating question Q2 will help to ascertain if the popular assumption of using the Euclidean distance between solutions is the right choice, or if the results of MOPSOs that apply clustering techniques could be further improved by using other metrics.

We will empirically address questions Q1 and Q2 by employing the I-Multi algorithm using different methods on a representative set of difficult multi-objective functions for which a characterization of their domains of difficulty exists. We will

evaluate the quality of the Pareto fronts obtained using the different clustering strategies.

Question Q3 helps us to get some insight about how the characteristics of the functions being optimized make the application of the different clustering strategies particularly suitable for each characteristic. This potential mapping between the characteristics of the functions and the “most promising” strategy for each characteristic is important, since it allows the user to have at least some heuristic criteria to decide in which situations a clustering strategy is expected to behave better than the others. We address question Q3 by first detecting characteristic patterns of behavior of the algorithms for each of the functions, and conceiving additional experiments to test alternative hypotheses that explain this behavior from the characteristics of the functions.

6.1 Function benchmarks and performance metrics

Selecting an appropriate testbed of functions is an important element of our analysis. In this paper we use the DTLZ (Deb et al 2005) and WFG (Huband et al 2006) benchmarks. These benchmark problems have been characterized in terms of their different domains of difficulty (shape of the Pareto front, multimodality, bias, separability and deception). In addition, these benchmark functions can scale both in the number of objectives and in the number of decision variables. Finally, the true Pareto optimal front is known for these functions.

A potential drawback of these benchmarks functions (except for DTLZ7 and WFG8) is that the decision variables are split into two main groups according to their relationship with the fitness landscape: distance variables and position variables. *Distance variables* are related to the convergence characteristic of sets of solutions for the problem. By changing a single distance variable of a solution, we generate a new solution that dominates, is equal to or is dominated in relation to the previous. These solutions will never be strictly non-dominated in relation to each other. *Position variables* are related to the spread of solutions, and by modifying an individual position variable of a solution, we only generate a new solution that is incomparable (non-dominated) or equal to the original solution (Huband et al 2006). Although this division allows to separately evaluate the behavior of optimizers in terms of spread and convergence, in real-world problems a more refined classification of variables is often required in order to model the characteristics of the problems Brownlee and Wright (2012)

A summary of the characteristics of the DTLZ and WFG problems used is shown in Table 1, adapted from (Li et al 2015b).

Two particularly relevant domains of difficulty for our work are *Bias* and *Deception*. Bias means that there is a significant variation in distribution between vectors in the decision space and the objective vectors (Huband et al 2006). Deception means that besides the true Pareto front, a problem has at least one local optimum Pareto front and most of the search space favors it (Coello et al 2006).

6.1.1 Performance metrics and statistical tests

Inverted Generational Distance (IGD) (Coello and Cortés 2005), is a non Pareto compliant indicator that measures the smallest distance between each point in

Table 1 Characteristics of the functions included in the DTLZ and WFG benchmarks

Problem	Pareto front shape	Challenges of decision space
DTLZ1	linear	multi-modal
DTLZ2	concave	none
DTLZ3	concave	multi-modal
DTLZ4	concave	biased
DTLZ5	concave, degenerate	none
DTLZ6	concave, degenerate	multi-modal
DTLZ7	mixed, disconnected	multi-modal
WFG1	mixed	biased
WFG2	convex, disconnected	multi-modal, non-separable
WFG3	linear, degenerate	non-separable
WFG4	concave	multi-modal
WFG5	concave	deceptive
WFG6	concave	non-separable
WFG7	concave	biased
WFG8	concave	biased, non-separable
WFG9	concave	biased, multi-modal, deceptive, non-separable

the true discretized Pareto front (F^t) and the points in a Pareto front found by an optimizer (F^k). IGD is a widely used metric, especially in the many-objective community due to its low computational cost and its ability to measure the convergence and diversity of a Pareto front approximation at the same time. It is defined by Equation 2.

$$IGD(F^k, F^t) := \frac{1}{|F^t|} \left(\sum_{i=1}^{|F^t|} dist(F_i^t, F^k)^p \right)^{\frac{1}{p}} \quad (2)$$

where $dist(F_i^t, F^k)$ means the minimal Euclidean distance from F_i^t to F^k and $p = 2$.

In this work we use a modified version of the IGD known as IGD_p (Schutze et al 2012) as main quality indicator. This modification makes the indicator fairer by allowing it to be insensitive to the number of points in the discretized Pareto front, but does not alter its main properties. IGD_p is defined in Equation 3.

$$IGD(F^k, F^t) := \left(\frac{1}{|F^t|} \sum_{i=1}^{|F^t|} dist(F_i^t, F^k)^p \right)^{\frac{1}{p}} \quad (3)$$

Although we focus our analysis on the IGD_p metric, we used as second metric another extensively used indicator called hypervolume (While et al 2012). The hypervolume was used in summarized analyses to support additional information about the overall differences obtained with each algorithm variant. Notice, that the application of the exact hypervolume computation (While et al 2012) becomes unfeasible as the number of objectives is increased. Therefore, we used an approximated version (Bader et al 2010) for ten objectives or more. The hypervolume is defined as follows (Bringmann et al 2013):

$$HV(F^k) := VOL \left(\bigcup_{(u_1, \dots, u_m) \in F^k} [r_1, u_1] \times \dots \times [r_m, u_m] \right) \quad (4)$$

where $VOL(.)$ is the usual Lebesgue measure and the reference point \mathbf{r} is the nadir (anti-optimal or “worst possible”) point in space. The greater the hypervolume value of a set is, the better that set is taken to be.

The results of the IGD_p in 30 independent runs of the algorithms are submitted to the Kruskal-Wallis (Kruskal and Wallis 1952) statistical test at a 5% significance level. When significant differences were found, we conducted a post-hoc analysis using the Nemenyi (Nemenyi 1963) test to identify particular differences between samples (Demsar 2006).

Since there are many results being compared in different situations, sometimes it is hard to draw general conclusions. To ease this visualization, we present a summarized table disregarding specific functions and objective numbers and focusing the analysis on the differences between the algorithms. This table is generated for both indicators (IGD_p and Hypervolume) by using the Friedman (Friedman 1937) statistical test also at a 5% significance level, on the averages of the 30 runs of each subproblem (problem/objective number).

The comparison between the different algorithm components using both statistical tests is presented in tables containing the mean ranks of the results obtained on 30 runs of the algorithm using each component. We assign final ranks to the algorithms (presented in parenthesis) according to their mean ranks. In case of a statistical tie (algorithms presenting no statistically significant difference), the final rank of each of the tied algorithms is equal to the average of the ranks that would be assigned to them. The algorithm(s) with the smallest final ranks are highlighted.

6.2 Parameters of the algorithms

The parameters of the algorithms used in our experiments are summarized in Table 2.

Table 2 Parameters of the functions and I-Multi

C_1, C_2	varies randomly in $[1.5, 2.5]$
Objectives (m)	3, 5, 8, 10, 15 and 20
Decision variables for WFG (n)	$k + l$ where $k = 2 \times (m - 1)$ and $l = 20$
Decision variables for DTLZ (n)	$m + k - 1$ where $k = 5$ for DTLZ1, $k = 20$ for DTLZ7 and $k = 10$ otherwise
Initial phase duration	100 iterations
Initial phase population	100 particles
Multi-swarm phase duration	100 iterations
Number of swarms & clusters	50
Multi-swarm phase population	(750/number of swarms) particles
Repository maximum size	200 solutions
Multi-swarm region size	decrease from 0.5 to 0.1
Split iterations	5

The parameters C_1 and C_2 respectively control the effect of the personal and global best particles in the velocity. They are set according to the recommendation given in the original SMPSO paper (Nebro et al 2009). The number of decision variables was set according to the recommendation given for the problems in (Deb et al 2005) and (Huband et al 2006). The number of split iterations, as well as

the multi-swarm region size were calibrated in (Britto et al 2013) and we use the best values found. The number of iterations (initial and total), the initial size of the population, the maximum size of the repositories and the total number of particles were also set as proposed in (Britto et al 2013). The number of sub-swarms and clusters was investigated in a previous work (Castro Jr. et al 2015) that used a variant of I-Multi, but we believe the results hold. Regarding the number of particles per swarm, if the total number is not divisible by the number of sub-swarms, the remaining particles are distributed among the sub-swarms. As in (Castro Jr. et al 2015), we used the Crowding Distance (CD) (Deb et al 2000) archiver in the multi-swarm phase.

6.3 Comparison between the spaces of clustering

This section presents the results of the different strategies used by I-Multi to cluster the solutions. Each of these strategies is defined by the space in which clustering is accomplished.

6.3.1 DTLZ benchmark

Table 12 shows the IGD_p results. In this table, the first column represents the number of objectives, the second column represents the different strategies: *Objective* indicates that clustering is made in the objective space. *Both* indicates that the clustering is made in a combination of both, the objectives and decision spaces. *Decision* means that the clustering is made in the decision variables space. The other columns show the mean ranks of the results obtained on 30 independent runs of the algorithm using each clustering space.

From the analysis of Table 12, we can group the behavior of the three algorithms on all the functions in three classes. I) No statistical differences between the three algorithms across different number of objectives (Functions DTLZ3 and to a lesser extent functions DTLZ1 and DTLZ7). II) Statistical differences indicate that objective clustering is the best choice (functions DTLZ2, DTLZ5, and DTLZ6). III) Statistical differences indicate that objective clustering is the worst (function DTLZ4), while decision is the best.

In the previous classification, we have extracted global behavior patterns regarding the clustering space according to specific problems and objective numbers. We considered the general rankings as a secondary measure of difference between the algorithms, even if statistical differences were not found. A summarized analysis, considering all the combinations of problems and numbers of objectives (seven problems and six numbers of objectives, on a total of 42 subproblems) for each algorithm is presented in Table 3 for the IGD_p indicator and in Table 4 for the Hypervolume indicator.

In these tables we can see that the summarized analysis erases the individual differences detected for each function. However, it can be appreciated that clustering in the objective space has a lower value of the average ranking, i.e. its global results (considering the 42 subproblems) are slightly better when considering both indicators.

In the next step, we focus on unveiling the characteristics of the functions that influence the behavior of the clustering strategies. To do so, we will present a

Table 3 Overall ranks of the IGD_p as used in the Friedman test for all the DTLZ problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Objectives	Both	Decision
75.0 (2.0)	96.0 (2.0)	81.0 (2.0)

Table 4 Overall ranks of the Hypervolume as used in the Friedman test for all the DTLZ problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Objectives	Both	Decision
71.5 (2.0)	92.5 (2.0)	88.0 (2.0)

comparative analysis of the behavior of the algorithms on functions DTLZ2 and DTLZ4. These functions are particularly interesting, because despite of presenting similar expression (as shown in Equations (5) and (6)), they are exemplars of the classes II and III previously described, i.e. the clustering strategies present an opposite behavior when optimizing these functions.

$$\begin{aligned}
& \text{Min } f_1(\mathbf{x}) = (1 + g(x_m)) \cos(x_1 \frac{\pi}{2}) \cdots \cos(x_{m-1} \frac{\pi}{2}), \\
& \text{Min } f_2(\mathbf{x}) = (1 + g(x_m)) \cos(x_1 \frac{\pi}{2}) \cdots \sin(x_{m-1} \frac{\pi}{2}), \\
& \quad \vdots \\
& \text{Min } f_{m-1}(\mathbf{x}) = (1 + g(x_m)) \cos(x_1 \frac{\pi}{2}) \sin(x_2 \frac{\pi}{2}), \\
& \text{Min } f_m(\mathbf{x}) = (1 + g(x_m)) \sin(x_1 \frac{\pi}{2}), \\
& \quad 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n, \\
& \text{where } g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2.
\end{aligned} \tag{5}$$

$$\begin{aligned}
& \text{Min } f_1(\mathbf{x}) = (1 + g(x_m)) \cos(x_1^\alpha \frac{\pi}{2}) \cdots \cos(x_{m-1}^\alpha \frac{\pi}{2}), \\
& \text{Min } f_2(\mathbf{x}) = (1 + g(x_m)) \cos(x_1^\alpha \frac{\pi}{2}) \cdots \sin(x_{m-1}^\alpha \frac{\pi}{2}), \\
& \quad \vdots \\
& \text{Min } f_{m-1}(\mathbf{x}) = (1 + g(x_m)) \cos(x_1^\alpha \frac{\pi}{2}) \sin(x_2^\alpha \frac{\pi}{2}), \\
& \text{Min } f_m(\mathbf{x}) = (1 + g(x_m)) \sin(x_1^\alpha \frac{\pi}{2}), \\
& \quad 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n, \\
& \text{where } g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2.
\end{aligned} \tag{6}$$

Since the only difference between the functions is the bias, represented by the parameter α in Equation (6), this source of difficulty seems to be the one that determines the opposite behavior of the clustering strategies. We conducted a detailed analysis of the behavior of the clustering strategies for different values of α in Equation (6).

Figure 3 shows the average of the IGD_p results when optimizing the problem DTLZ4 for three, five and eight objectives with different values of α , remembering that smaller IGD_p values are better. As can be seen, along with the increase of the

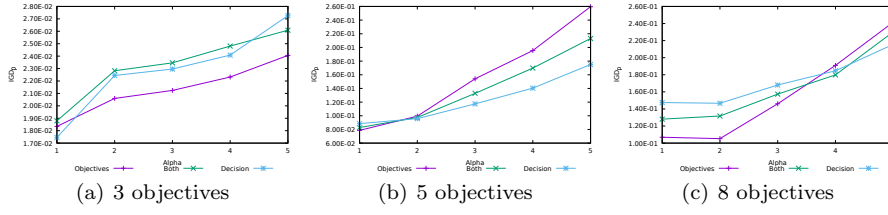


Fig. 3 Function DTLZ4 with different α values for 3,5 and 8 objectives. Average indicator results for the Pareto approximations obtained by I-Multi with different clustering strategies.

Table 5 Overall ranks of the IGD_p as used in the Friedman test for all the WFG problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Objectives	Both	Decision
75.0 (1.0)	110.0 (2.0)	139.0 (3.0)

Table 6 Overall ranks of the Hypervolume as used in the Friedman test for all the WFG problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Objectives	Both	Decision
97.0 (1.5)	97.0 (1.5)	130.0 (3.0)

α value (consequently the bias), the results obtained using each clustering space become closer to those obtained using the DTLZ4 function and farther from those using the DTLZ2 function, where $\alpha = 1$, as can be seen in Table 12. Moreover, as the number of objectives increase, the sensibility of the algorithm to the increase of the bias seems to become smaller, where a change in the best clustering strategy takes higher α value to happen. Another interesting finding of the analysis is that increasing bias worsens the quality of the Pareto approximations found by all versions of the algorithm.

6.3.2 WFG benchmark

Table 13 shows the results of the clustering strategies on the WFG benchmark. From the analysis of Table 13, it is possible to conclude that only classes II and III arise for this benchmark. Clearly, in most problems, it is preferable to cluster the solutions in the objective space (Class II). The most notable exception is problem WFG5, in which clustering in the objective space is the worst for all numbers of objectives and in general it is better to cluster in both spaces (Class III).

The differences presented in Table 13 are quite visible, however we made a global analysis of the results as in the previous benchmark. Tables 5 and 6 show the global analysis considering the IGD_p and Hypervolume indicators respectively. When considering the IGD_p it is more advantageous to cluster in the objective space, followed by both spaces, while according to the Hypervolume indicator clustering in the objective space or in both is equally good. IGD_p and Hypervolume indicate that clustering in the decision space, as done in (Britto et al 2013), produces the worst results. Notice that there are significant statistical differences between the three strategies. Since the ranks of the results obtained with each

space can be ordered as *objectives* < *both* < *decision*, we can assume that the only reason why using the combined space (both) is better than the decision space is because it encodes the objective space.

The fact that for most of the functions the objective clustering strategy is the best, makes the case of function WFG5 more intriguing. In addition to being the only function for which clustering the decision variables is better, WFG5 shares some of the characteristics of function WFG4. Therefore, we conducted a similar analysis to that presented in the previous section for functions DTLZ2 and DTLZ4. Our goal is to identify the characteristics of WFG5 that make decision clustering a more efficient algorithm for this function. Function WFG5 defined in Equation 7, where $|\mathbf{z}| = n = k + l$ and $\mathbf{y} = \mathbf{z}_{[0,1]} = (\frac{z_1}{2}, \dots, \frac{z_n}{2n})$.

$$\begin{aligned}
 \text{Given } \mathbf{z} &= (z_1, \dots, z_k, z_{k+1}, \dots, z_n) \\
 \text{Min } f_1(\mathbf{x}) &= x_m + 2 \prod_{j=1}^{m-1} \sin(x_j \frac{\pi}{2}) \\
 f_{i=2:m-1}(\mathbf{x}) &= x_m + 2i(\prod_{j=1}^{m-i} \sin(x_j \frac{\pi}{2})) \cos(x_{m-i+1} \frac{\pi}{2}) \\
 f_m(\mathbf{x}) &= x_m + 2m \cos(x_1 \frac{\pi}{2}) \\
 \text{Where } x_{i=1:m-1} &= r_sum((y_{\frac{(i-1)k}{(m-1)+1}}, \dots, y_{\frac{ik}{(m-1)}}), (1, \dots, 1)) \\
 x_m &= r_sum((y_{k+1}, \dots, y_n), (1, \dots, 1)) \\
 y_{i=1:n} &= s_deceptive(y_i, 0.35, 0.001, 0.05)
 \end{aligned} \tag{7}$$

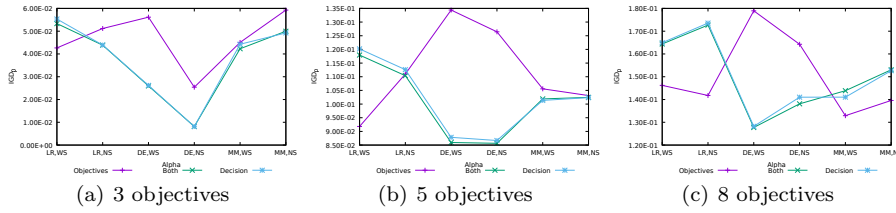


Fig. 4 WFG functions with different combinations of transformations for 3, 5 and 8 objectives. Average indicator results for the Pareto approximations obtained by I-Multi with different clustering strategies.

Figure 4 shows the average of the IGD_p results when optimizing instances of three, five and eight objectives of custom WFG problems with different combinations of transformation functions. Here, we combine the three shift functions available (linear (LR), deceptive (DE) and multi-modal (MM)), with the two reduction functions (weighted sum (WS) and non-separable (NS)). The bias functions were not used because neither WFG4 nor WFG5 are biased but display different results, hence the bias is not determinant in this case.

We also want to highlight that some combinations tested are already part of the WFG family: WFG4 is multi-modal, weighted sum; WFG5 is deceptive, weighted sum; WFG6 is linear, non-separable. The combination linear, weighted sum can be considered an unbiased version of WFG7 and WFG8, since the only difference between these function is if the bias is applied in the position-related or distance-related decision variables. Other combinations are found in part of the decision

variables of WFG9: deceptive, non-separable (position-related), and multi-modal, non-separable (distance-related).

As can be seen from Figure 4, in the linear and multi-modal problems it is harder to recommend a clustering space, since the best space changes according to the objective number. On the other hand, in all the numbers of objectives studied, clustering in the objective space is the worst when the problem has a deceptive component, hence in these cases, clustering in decision space or in both spaces is recommended. This similar behavior of using both spaces or only the decision space can be explained by the fact that we have 24, 28 and 34 decision variables for only 3, 5 and 8 objectives respectively. Therefore, the decision space influences more than the objective space when using both spaces. From these results, we conclude that the deceptive character of the function has the main effect in the deterioration of the results of the objective clustering strategy. The relevant finding here is that clustering in the decision space is considerably less sensitive to this effect.

6.3.3 Correlation between quality of solutions and quality of clustering

In this section we want to study the relationship between the quality of the clustering and the quality of the solutions obtained during the search. We understand the quality of the clustering, as how good it is grouping similar solutions and putting different solutions in different clusters. As a measure of clustering quality we use the Davies-Bouldin Index (DB) (Davies and Bouldin 1979). Similarly, we evaluate the quality of the solutions using the IGD_p metric. We then investigate the correlation between DB and IGD_p .

In order to calculate this correlation, we calculated the DB and IGD_p at each iteration of the multi-swarm phase (last 100 iterations) of I-Multi. Then we averaged these 100 values over 30 independent runs and used these averaged values to calculate the correlation. To calculate the DB, first the scatter within each cluster is calculated by:

$$T_i = \left\{ \frac{1}{|F_i|} \sum_{j=1}^{|F_i|} |x_j - S_i|^q \right\}^{\frac{1}{q}} \quad (8)$$

where x_j is one point assigned to cluster F_i , S_i is the centroid of the cluster and $|F_i|$ is its size. Usually, the value of q is 1, so T_i becomes the average Euclidean distance between the points in the cluster and its centroid. Hence the separation between the clusters is measured by:

$$M_{ij} = |S_i - S_j|^p \quad (9)$$

When $p = 2$, this measure corresponds to the Euclidean distance between the centroids of F_i and F_j . Next, a measure of quality of each cluster is calculated by:

$$R_{ij} \equiv \frac{T_i + T_j}{M_{ij}} \quad (10)$$

and this is used to calculate $R_i \equiv \max_{i \neq j} R_{ij}$. Finally, the DB index is calculated as:

$$DB \equiv \frac{1}{NS} \sum_{i=1}^{NS} R_i \quad (11)$$

where NS is the number of clusters.

DB is the system-wide average of the similarity measures of each cluster with its most similar cluster. The “best” choice of cluster will be that which minimizes this average similarity.

To investigate the relationship between IGD_p and DB, we selected four representative optimization functions: two problems from each set of benchmark functions and from these, we selected problems where the best results were achieved by clustering in the decision and objective spaces. These four problems can be considered representative for the entire set of benchmark functions used. Moreover, if we add all problems, we could bias the classifier to a determined outcome, for instance to clustering in the objective space since most of the problems present good results in this clustering space. The selected functions were: DTLZ2, DTLZ4, WFG4 and WFG5.

We intend to determine if the quality of the clustering is correlated with the quality of the Pareto front, and how this correlation is influenced by the choice of the clustering strategy. One possible approach to investigate this question would be to compute the correlation between IGD_p and DB for each possible combination of factors involved in the performance of the algorithm (e.g. clustering strategy; number of objectives; optimization problem) but we would like to obtain a more structured and informative representation of how the correlation depends on these factors. One way to obtain such type of informative representation is learning a decision tree (Breiman 1984).

A decision tree can be used for classification or regression. In the case of classification, a set of attributes is classified based on previously observed instances of these attributes. In the tree structure, the leaves correspond to the class given for the target variable and branches represent conjunctions of features that determine the classes of the target variable. We define three attributes: clustering strategy; number of objectives; optimization problem, and one class variable: correlation between IGD_p and DB. The first attribute serves as a descriptor of the algorithm, relevant for our analysis. The other two attributes characterize the difficulty of the problem. Basically, we want the decision tree to predict, based on the characteristics of the problem and the clustering strategy applied, how good will be the correlation between the IGD_p and DB. That way we can learn under which conditions will a good clustering likely produce good Pareto front approximations. The decision tree serves as a representation of how the correlation between IGD_p and DB depends on the clustering strategy and the characteristics of the problem.

We employed a Reduced Error Pruning Tree (REPTree) (Elomaa and Kääriäinen 2001) on our dataset containing all 72 instances (combination of six numbers of objectives, three clustering spaces and four problems). As an example, one line of the dataset used to learn the tree is: “Objectives, 3, DTLZ2, 0.9074”, meaning that for the three objective version of problem DTLZ2 with clustering done in objective space, we have a correlation of 0.9074. Therefore, given an instance of the input

variables, the tree will output what is the expected correlation between IGD_p and DB. Figure 5 presents the generated decision tree.

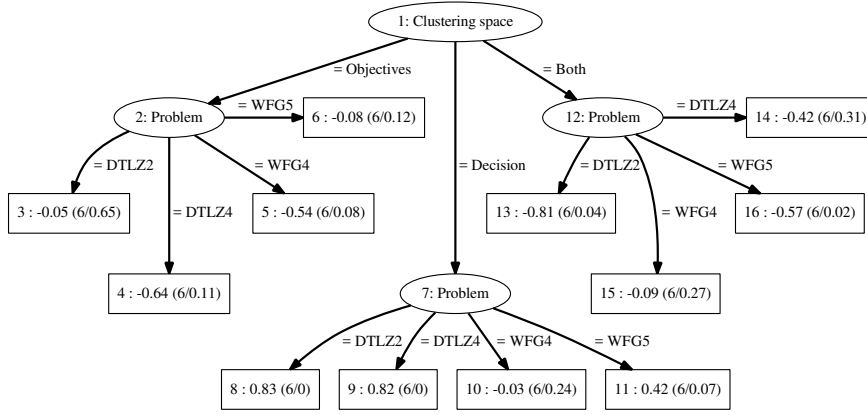


Fig. 5 Decision tree of the correlation between the IGD_p and the DB.

In the leaves of the output tree, besides the node index and the class value, we can appreciate in parentheses the number of instances matching the rule and the percentage of instances misclassified by that rule. The mean absolute error of the model is 0.283.

In this tree, we can clearly identify three subtrees: on the left, a subtree corresponding to the clustering in the objective space; in the center, a subtree corresponding to the clustering in the decision space; and in the right, a subtree corresponding to the clustering in the combination of both spaces. Moreover, each of these subtrees has the same number of samples in it: 24.

This clear split of the tree in three symmetrical parts corresponding to each clustering strategy, points to a strong relationship between the quality of the clustering and the quality of the solutions generated by I-Multi. Other than that, we can not identify other influences from the clustering strategy in the correlations.

6.3.4 Discussion

In general, clustering in the objective space is recommended in both, the DTLZ and the WFG problems, since it achieves the best performance in most of the cases. However this clustering space is more sensitive to properties of the problem, such as bias or deception and can yield poor results in such conditions.

In the investigation of the correlation between the quality of the clustering and the quality of the solutions generated by the algorithm at each generation, we have seen that the space in which the clustering is done is an important factor to determine the strength of this correlation. Hence, it is expected that the choice of the clustering space affects the final results obtained by the algorithm.

Table 7 Overall ranks of the IGD_p as used in the Friedman test for all the DTLZ problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Euclidean	Tchebycheff	Minkowski (0.5)	Minkowski (4)
99.0 (2.5)	106.0 (2.5)	103.0 (2.5)	112.0 (2.5)

Table 8 Overall ranks of the Hypervolume as used in the Friedman test for all the DTLZ problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Euclidean	Tchebycheff	Minkowski (0.5)	Minkowski (4)
95.0 (2.5)	100.5 (2.5)	123.0 (2.5)	101.5 (2.5)

6.4 Comparison between distance metrics

In this section we investigate another significant question, i.e., whether and how the choice of the similarity metric influences the results of the clustering strategy. Building on the results shown in previous sections in which we found that, in general, better results can be obtained using the clustering in the objective space, we constrain the study of the metrics to I-Multi using this clustering space.

In the previous section, we used the K-Means algorithm. This algorithm was designed to work with the Euclidean distance and uses this metric in two places: explicitly in the distance calculation to allocate the closest points to a cluster and implicitly in the calculation of the new centroids that are defined as the mean of the points in a cluster. However, when we change the similarity metric employed, the mean may no longer represent the center of a cluster in the point of view of the new metric.

One alternative to solve this issue is to change the method used to define the center of the cluster. In this section we used the medoid as center, which is the representative point of a cluster for which the dissimilarity to all the other points of the cluster is minimal (Kaufman and Rousseeuw 1987). By using the medoid instead of the mean, we change the algorithm from K-Means to K-Medoids (Kaufman and Rousseeuw 1987). In K-Medoids, we can use any similarity measure, since the same distance will be used to calculate the medoids and to allocate all the points to their closest medoid to form a cluster.

Tables 14 and 15 show the results of I-Multi using different similarity measures for function benchmarks DTLZ and WFG, respectively. Similar to the analysis conducted in previous sections, the results of the global analysis are shown in Tables 7 and 8 for the IGD_p and Hypervolume of the DTLZ problems respectively and in Tables 9 and 10 for the IGD_p and Hypervolume of the WFG problems respectively.

6.4.1 DTLZ benchmark

When considering the DTLZ problems, in general we can see few statistically significant differences, and when they occur it is very hard to find a pattern. The only general pattern that emerged is the increase in the statistically significant differences with the number of objectives. The exception of this behavior is the problem DTLZ2, where we can clearly see that Minkowski with $k = 0.5$ performs

Table 9 Overall ranks of the IGD_p as used in the Friedman test for all the WFG problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Euclidean	Tchebycheff	Minkowski (0.5)	Minkowski (4)
110.0 (1.5)	162.0 (3.5)	148.0 (3.0)	120.0 (2.0)

Table 10 Overall ranks of the Hypervolume as used in the Friedman test for all the WFG problems and numbers of objectives. Final ranks, presented in parenthesis assigned according to the ranks.

Euclidean	Tchebycheff	Minkowski (0.5)	Minkowski (4)
111.0 (1.5)	148.0 (3.0)	151.0 (3.0)	130.0 (2.5)

the best for all numbers of objectives. Since the DTLZ2 is a concave shaped function that does not pose any further challenge to the algorithms, the number of non-dominated solutions found for it is usually very high, which can increase the influence of clustering mechanisms in the final result of the algorithm.

In an overall view of the results from the summarized Tables 7 and 8 considering the IGD_p and Hypervolume results respectively, there is no significant difference between any of the algorithms, although the overall ranking attributed to the Euclidean distance is slightly smaller than the others.

6.4.2 WFG benchmark

Considering the results for the WFG benchmark functions, we can identify more significant differences than for the DTLZ. As in the previous case, in general the differences increase with the number of objectives, but besides that behavior it is very hard to find patterns. Exceptions are the functions WFG3 and WFG4, where for $m > 8$, Minkowski with $k = 0.5$ achieves the best results in all cases.

In an overall analysis of Tables 9 and 10 considering the IGD_p and Hypervolume indicators respectively, the Euclidean distance achieved the best results. This can be explained by the fact that it is usually among the best measures in most of the cases, and even when it is not among the best, it usually is not among the worst, so in a general analysis it performs best and can be considered a stable metric.

6.4.3 Discussion

For the distance measures considered, we can state that in general I-Multi is not very sensitive to the choice of a similarity metric for clustering. However, this difference increases with the number of objectives. This behavior can happen for two reasons: first the size of the Pareto front increases greatly with the number of objectives (except for functions DTLZ5, DTLZ6 and WFG3 which are degenerate), secondly because of the number of non-dominated solutions to be clustered also increases greatly with the number of objectives.

Although most of the functions do not exhibit a clear pattern suggesting which measure is better, for DTLZ2, WFG3 and WFG4, the measure Minkowski with $k = 0.5$ achieved the best results, which indicates that the selection of an appropriate similarity measure can be problem-dependent.

Finally, from the two summarized tables for both problems, we can recommend using the Euclidean distance to cluster the solutions, since the results indicate that it is a robust metric which in most of the cases appears ranked among the best solutions and in very few cases was ranked among the worst metrics.

7 Conclusion

In this paper we investigated two important characteristics of the solutions clustering in optimization algorithms for multi and many-objective problems by taking I-Multi as an example. These characteristics were the space in which the clustering is performed and the similarity metric used to compare the solutions.

Understanding the influence of these characteristics is an important aspect for clustering-based evolutionary algorithms, since one can exploit this information when designing new algorithms as well as when applying them to problems with known properties, such as deception or bias.

As far as we know, this is the first study covering the impact of using different clustering spaces and similarity metrics on the performance of many-objective optimization algorithms. Since we investigated two different characteristics of the clustering phase of I-Multi, we conducted two separate experimental studies.

In the first study, we compared the results obtained by clustering the solutions in three different spaces: objectives, decision and both, which is a combination of the two aforementioned spaces. From the results obtained, we could identify that the space in which the clustering is done has an important impact in the performance of the algorithm. Moreover, the best choice of clustering space can be problem-dependent, and is impacted by specific properties of the problem, especially deception and bias. In an overall analysis, we can recommend clustering the solutions in the objective space, since it presented good results in most of the problems.

In the second study, we compared the results obtained when using four different clustering metrics: Euclidean, Tchebycheff, Minkowski with $k = 0.5$ and Minkowski with $k = 4$. The results indicated that, in general, the algorithm is not sensitive to the choice of the metric used, however, this sensitivity increases with the number of objectives. This choice can also be problem-dependent, but in an overall view we can recommend the use of the Euclidean distance, since it was the most robust metric in our comparison.

7.1 Further work

In this study we have focused on the analysis of the K-Means algorithm and its modification to K-Medoids in the second experimental study. However, an important question is to determine if further improvement to MOPSOs could be achieved by using other clustering methods, especially those that do not require defining in advance the number of clusters. These clustering methods can generate clusters of better quality and our results indicate that the clustering quality influences the performance of the algorithm. One possible direction is to evaluate the behavior of other clustering methods that have shown good results for single-

objective EAs such as hierarchical clustering (Lozano and Larrañaga 1999) and affinity propagation (Santana et al 2010) algorithms.

An interesting issue is whether there exist any type of interactions between the distance metrics and the type of clustering. Is one distance metric better than the others for some clustering strategy? We have not addressed this question but it is worth considering this problem in further work.

While I-Multi is a good example of other MOPSOs, the investigation of the effect of clustering could be extended to algorithms that use probabilistic modeling of the solutions contained in each cluster. One representative example of this type of algorithms is C-Multi (Castro Jr. et al 2015).

References

- Adra SF (2007) Improving convergence, diversity and pertinency in multiobjective optimisation. PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK
- Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: International Conference on Database Theory, Springer, pp 420–434
- Bader J, Deb K, Zitzler E (2010) Faster hypervolume-based search using monte carlo sampling. In: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Lecture Notes in Economics and Mathematical Systems, vol 634, Springer Berlin Heidelberg, pp 313–326
- Bandyopadhyay S, Saha S (2012) Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications. Springer, DOI 10.1007/978-3-642-32451-2
- Benamer L, Alami J, El Imrani A (2009) A new hybrid particle swarm optimization algorithm for handling multiobjective problem using fuzzy clustering technique. In: International Conference on Computational Intelligence, Modelling and Simulation, pp 48–53, DOI 10.1109/CSSim.2009.42
- Bosman PA, Thierens D (2002) Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. International Journal of Approximate Reasoning 31(3):259–289
- Breiman L (1984) Classification and regression trees. Chapman & Hall/CRC
- Bringmann K, Friedrich T, Igel C, Voß T (2013) Speeding up many-objective optimization by monte carlo approximations. Artificial Intelligence 204:22–29, DOI <http://dx.doi.org/10.1016/j.artint.2013.08.001>, URL <http://www.sciencedirect.com/science/article/pii/S0004370213000738>
- Britto A, Pozo A (2012) Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization. In: IEEE Congress on Evolutionary Computation, pp 1–8, DOI 10.1109/CEC.2012.6256149
- Britto A, Mostaghim S, Pozo A (2013) Iterated multi-swarm: A multi-swarm algorithm based on archiving methods. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, GECCO '13, pp 583–590

- Brownlee AE, Wright JA (2012) Solution analysis in multi-objective optimization. *Building Simulation and Optimization* pp 317–324, URL <https://dspace.lboro.ac.uk/2134/10275>
- Castro Jr OR, Santana R, Pozo A (2015) C-Multi: a competent multi-swarm approach for many-objective problems. *Neurocomputing* DOI <http://dx.doi.org/10.1016/j.neucom.2015.06.097>, URL <http://www.sciencedirect.com/science/article/pii/S0925231215016215>, <http://dx.doi.org/10.1016/j.neucom.2015.06.097>
- Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1):58–73
- Coello CAC, Cortés NC (2005) Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines* 6(2):163–190, DOI 10.1007/s10710-005-6164-x
- Coello CAC, Lamont GB, Veldhuizen DAV (2006) *Evolutionary Algorithms for Solving Multi-Objective Problems* (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus, NJ, USA
- Davies DL, Bouldin DW (1979) A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* 1(2):224–227, DOI 10.1109/TPAMI.1979.4766909, URL <http://dx.doi.org/10.1109/TPAMI.1979.4766909>
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In: *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, London, UK, PPSN VI, pp 849–858
- Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, Springer London, pp 105–145
- Deborah H, Richard N, Hardeberg JY (2015) A comprehensive evaluation of spectral distance functions and metrics for hyperspectral image processing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8(6):3224–3234
- Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7:1–30
- Elomaa T, Kääriäinen M (2001) An analysis of reduced error pruning. *J Artif Int Res* 15(1):163–187, URL <http://dl.acm.org/citation.cfm?id=1622845.1622850>
- Emmendorfer LR, Pozo A (2009) Effective linkage learning using low-order statistics and clustering. *Evolutionary Computation*, *IEEE Transactions on* 13(6):1233–1246
- Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32(200):675–701
- Fritsche G, Strickler A, Pozo A, Santana R (2015) Capturing relationships in multi-objective optimization. In: *Proceedings of the Brazilian Conference on Intelligent Systems (BRACIS 2015)*, Natal, Brazil, accepted for publication
- Hartigan J, Wong M (1979) Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics* 28(1):100–108
- Howarth P, Rüger S (2005) Fractional distance measures for content-based image retrieval. In: *European Conference on Information Retrieval*, Springer, pp 447–

456

- Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10(5):477–506, DOI 10.1109/TEVC.2005.861417
- Ikeda K, Kita H, Kobayashi S (2001) Failure of pareto-based moeas: does non-dominated really mean near to optimal? In: *Proceedings of the 2001 Congress on Evolutionary Computation*, vol 2, pp 957–962 vol. 2, DOI 10.1109/CEC.2001.934293
- Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: A short review. In: *IEEE Congress on Evolutionary Computation*, pp 2419–2426
- Ishibuchi H, Akedo N, Ohyanagi H, Nojima Y (2011) Behavior of EMO algorithms on many-objective optimization problems with correlated objectives. In: *IEEE Congress on Evolutionary Computation*, pp 1465–1472
- Jin Y, Sendhoff B (2004) Reducing fitness evaluations using clustering techniques and neural network ensembles. In: Deb K (ed) *Genetic and Evolutionary Computation – GECCO 2004*, *Lecture Notes in Computer Science*, vol 3102, Springer Berlin Heidelberg, pp 688–699, DOI 10.1007/978-3-540-24854-5_71
- Karshenas H, Santana R, Bielza C, Larrañaga P (2014) Multi-objective optimization based on joint probabilistic modeling of objectives and variables. *IEEE Transactions on Evolutionary Computation* 18(4):519–542
- Kaufman L, Rousseeuw P (1987) Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods* pp North-Holland
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, pp 1942–1948
- Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* 47(260):583–621
- Kukkonen S, Deb K (2006) Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In: *2006 IEEE International Conference on Evolutionary Computation*, pp 1179–1186, DOI 10.1109/CEC.2006.1688443
- Larrañaga P, Lozano JA (eds) (2002) *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London
- Laumanns M, Zenklusen R (2011) Stochastic convergence of random search methods to fixed size Pareto front approximations. *European Journal of Operational Research* 213(2):414–421, DOI <http://dx.doi.org/10.1016/j.ejor.2011.03.039>
- Li H, Deb K, Zhang Q, Kwong S (2015a) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation* Accepted for publication
- Li K, Deb K, Zhang Q, Kwong S (2015b) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation* 19(5):694–716, DOI 10.1109/TEVC.2014.2373386
- Liang X, Li W, Zhang Y, Zhou M (2015) An adaptive particle swarm optimization method based on clustering. *Soft Computing* 19(2):431–448, DOI 10.1007/s00500-014-1262-4, URL <http://dx.doi.org/10.1007/s00500-014-1262-4>
- Lozano J, Larrañaga P (1999) Applying genetic algorithms to search for the best hierarchical clustering of a dataset. *Pattern Recognition Letters* 20(9):911 – 918, DOI [http://dx.doi.org/10.1016/S0167-8655\(99\)00057-4](http://dx.doi.org/10.1016/S0167-8655(99)00057-4), URL

- <http://www.sciencedirect.com/science/article/pii/S0167865599000574>
- Ma X, Liu F, Qi Y, Wang X, Li L, Jiao L, Yin M, Gong M (2015) A multiobjective evolutionary algorithm based on decision variable analyses for multi-objective optimization problems with large scale variables. *IEEE Transactions on Evolutionary Computation* (99), DOI 10.1109/TEVC.2015.2455812
- Mahfoud SW (1995) Niching methods for genetic algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, also IlliGAL Report No. 95001
- Mostaghim S, Teich J (2004) Covering pareto-optimal fronts by subswarms in multi-objective particle swarm optimization. In: *Congress on Evolutionary Computation.*, vol 2, pp 1404–1411, DOI 10.1109/CEC.2004.1331061
- Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CAC, Luna F, Alba E (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: *Computational intelligence in multi-criteria decision-making.*, IEEE, pp 66–73
- Nemenyi P (1963) Distribution-free multiple comparisons. PhD thesis, Princeton University, URL <https://books.google.com.sg/books?id=nhDMtgAACAAJ>
- Okabe T, Jin Y, Sendhoff B, Olhofer M (2004) Voronoi-based estimation of distribution algorithm for multi-objective optimization. In: *Proceedings of the 2004 Congress on Evolutionary Computation CEC-2004*, IEEE Press, Portland, Oregon, pp 1594–1601
- Pelikan M (2005) Hierarchical Bayesian Optimization Algorithm. *Toward a New Generation of Evolutionary Algorithms*, Studies in Fuzziness and Soft Computing, vol 170. Springer
- Pelikan M, Goldberg DE (2000) Genetic algorithms, clustering, and the breaking of symmetry. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel HP (eds) *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Springer, Paris, France, pp 385–394, *lecture Notes in Computer Science* 1917
- Pelikan M, Sastry K, Goldberg DE (2005) Multiobjective hBOA, clustering and scalability. IlliGAL Report No. 2005005, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL
- Pulido GT, Coello Coello CA (2004) Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. In: Deb K (ed) *Genetic and Evolutionary Computation*, *Lecture Notes in Computer Science*, vol 3102, Springer Berlin Heidelberg, pp 225–237, DOI 10.1007/978-3-540-24854-5_20
- Santana R, Larrañaga P, Lozano JA (2010) Learning factorizations in estimation of distribution algorithms using affinity propagation. *Evolutionary Computation* 18(4):515–546, URL http://www.mitpressjournals.org/doi/abs/10.1162/EVCO_a.00002
- Schutze O, Esquivel X, Lara A, Coello CAC (2012) Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 16(4):504–522, DOI 10.1109/TEVC.2011.2161872
- Sindhya K, Miettinen K, Deb K (2013) A hybrid framework for evolutionary multi-objective optimization. *Evolutionary Computation*, *IEEE Transactions on* 17(4):495–511, DOI 10.1109/TEVC.2012.2204403
- Tsou CS, Fang HH, Chang HH, Kao CH (2006) An improved particle swarm pareto optimizer with local search and clustering. In: Wang TD, Li X,

- Chen SH, Wang X, Abbass H, Iba H, Chen GL, Yao X (eds) Simulated Evolution and Learning, Lecture Notes in Computer Science, vol 4247, Springer Berlin Heidelberg, pp 400–407, DOI 10.1007/11903697_51, URL http://dx.doi.org/10.1007/11903697_51
- Tsuji M, Munetomo M, Akama K (2006) Linkage identification by fitness difference clustering. *Evolutionary Computation* 14(4):383–409
- While L, Bradstreet L, Barone L (2012) A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation* 16(1):86–95
- Yen G, Daneshyari M (2006) Diversity-based information exchange among multiple swarms in particle swarm optimization. In: *IEEE Congress on Evolutionary Computation*, pp 1686–1693, DOI 10.1109/CEC.2006.1688511
- Zhang Q, Xue S (2007) An improved multi-objective particle swarm optimization algorithm. In: Kang L, Liu Y, Zeng S (eds) *Advances in Computation and Intelligence*, Lecture Notes in Computer Science, vol 4683, Springer Berlin Heidelberg, pp 372–381, DOI 10.1007/978-3-540-74581-5_41, URL http://dx.doi.org/10.1007/978-3-540-74581-5_41
- Zhang Y, wei Gong D, hai Ding Z (2011) Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. *Expert Systems with Applications* 38(11):13,933–13,941
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4):257–271

Table 11 Relevant characteristics of the algorithms considered in the review of related work.

Base Alg.	Space	Metric	Obj. Number	Method	Ref.
PSO	Decision	Euclidean	1	K-Means	(Liang et al 2015) (Yen and Daneshyari 2006)
PSO	Decision	Euclidean	2	Hierarchical clustering	(Pulido and Coello Coello 2004)
PSO	Objective	Euclidean	2	HC Average linkage	(Zhang and Xue 2007)
PSO	Objective	Euclidean	3	HC Average linkage	(Tsu et al 2006)
PSO	Decision	Euclidean	2,4	Fuzzy clustering	(Benamer et al 2009)
PSO	Decision	Euclidean	> 3	K-Means	(Britto et al 2013)
SPEA	Objective	Euclidean	2,3,4	HC Average linkage	(Zitzler and Thiele 1999)
PM, EDA	Fitness differences	Fitness differences	1	Centroid	(Tsuji et al 2006)
EDA	Decision	Euclidean	1	K-Means	(Emmendorfer and Pozo 2009) (Pelikan and Goldberg 2000)
EDA	Objective	Euclidean	2	K-Means	(Pelikan et al 2005)
EDA	Objective	Euclidean	2	Randomized leader	(Bosman and Thierens 2002)
EDA	Decision/reduced	Euclidean/grid ranks	2	K-Means/Voronoi	(Okabe et al 2004)
ES	Decision	Euclidean	1	K-Means	(Jin and Sendhoff 2004)
E/A	Objective	Euclidean	2,3,4	Orthogonal projection	(Sindhya et al 2013)

Table 12 Mean ranks of the IGD_p as used in the Kruskal-Wallis test for the DTLZ problems. Final ranks, presented in parenthesis assigned according to mean ranks.

Obj.	Spaces	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
3	Objective	51.33 (2.00)	57.87 (2.50)	42.87 (2.00)	45.87 (2.00)	74.30 (3.00)	19.73 (1.00)	40.53 (2.00)
	Both	47.13 (2.00)	56.33 (2.50)	50.77 (2.00)	47.60 (2.00)	34.60 (1.50)	56.00 (2.50)	46.30 (2.00)
	Decision	38.03 (2.00)	22.30 (1.00)	42.87 (2.00)	43.03 (2.00)	27.60 (1.50)	60.77 (2.50)	49.67 (2.00)
5	Objective	51.40 (2.00)	20.40 (1.00)	42.53 (2.00)	54.07 (2.50)	29.50 (1.00)	28.43 (1.00)	46.23 (2.00)
	Both	46.43 (2.00)	46.63 (2.00)	49.27 (2.00)	46.07 (2.00)	45.83 (2.50)	58.43 (2.50)	48.17 (2.00)
	Decision	38.67 (2.00)	69.47 (3.00)	44.70 (2.00)	36.37 (1.50)	61.17 (2.50)	49.63 (2.50)	42.10 (2.00)
8	Objective	50.63 (2.00)	15.50 (1.00)	42.70 (2.00)	62.67 (3.00)	21.30 (1.00)	34.13 (1.50)	42.80 (2.00)
	Both	46.03 (2.00)	45.50 (2.00)	48.00 (2.00)	42.10 (1.50)	54.87 (2.50)	49.97 (2.00)	48.70 (2.00)
	Decision	39.83 (2.00)	75.50 (3.00)	45.80 (2.00)	31.73 (1.50)	60.33 (2.50)	52.40 (2.50)	45.00 (2.00)
10	Objective	54.57 (2.50)	15.50 (1.00)	39.43 (2.00)	69.07 (3.00)	19.10 (1.00)	34.03 (1.50)	45.50 (2.00)
	Both	47.90 (2.00)	47.53 (2.00)	45.93 (2.00)	41.43 (1.50)	53.17 (2.50)	56.82 (2.50)	42.27 (2.00)
	Decision	34.03 (1.50)	73.47 (3.00)	51.13 (2.00)	26.00 (1.50)	64.23 (2.50)	45.65 (2.00)	48.73 (2.00)
15	Objective	58.53 (2.50)	15.50 (1.00)	49.80 (2.00)	75.00 (3.00)	21.27 (1.00)	36.70 (2.00)	30.13 (1.00)
	Both	44.33 (2.00)	56.53 (2.50)	42.70 (2.00)	38.50 (1.50)	57.07 (2.50)	50.83 (2.00)	55.40 (2.50)
	Decision	33.63 (1.50)	64.47 (2.50)	44.00 (2.00)	23.00 (1.50)	58.17 (2.50)	48.97 (2.00)	50.97 (2.50)
20	Objective	46.43 (2.00)	15.50 (1.00)	43.60 (2.00)	75.47 (3.00)	17.33 (1.00)	31.80 (1.00)	45.17 (2.00)
	Both	44.93 (2.00)	56.03 (2.50)	49.57 (2.00)	38.33 (1.50)	57.70 (2.50)	53.87 (2.50)	54.47 (2.50)
	Decision	45.13 (2.00)	64.97 (2.50)	43.33 (2.00)	22.70 (1.50)	61.47 (2.50)	50.83 (2.50)	36.87 (1.50)

Table 13 Mean ranks of the IGD_p as used in the Kruskal-Wallis test for the WFG problems. Final ranks, presented in parenthesis assigned according to mean ranks.

Obj.	Spaces	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
3	Objective	20.90 (1.00)	46.03 (2.00)	15.50 (1.00)	21.27 (1.00)	75.50 (3.00)	75.20 (3.00)	15.50 (1.00)	15.50 (1.00)	75.47 (3.00)
	Both Decision	52.87 (2.50) 62.73 (2.50)	43.67 (2.00) 46.80 (2.00)	51.37 (2.00) 69.63 (3.00)	41.47 (2.00) 73.77 (3.00)	26.97 (1.50) 34.03 (1.50)	31.30 (1.50) 30.00 (1.50)	50.13 (2.00) 70.87 (3.00)	50.40 (2.00) 70.60 (3.00)	28.13 (1.50) 32.90 (1.50)
5	Objective	16.93 (1.00)	25.93 (1.00)	15.50 (1.00)	15.50 (1.00)	75.50 (3.00)	75.03 (3.00)	15.50 (1.00)	15.50 (1.00)	29.57 (1.00)
	Both Decision	59.40 (2.50) 60.17 (2.50)	52.57 (2.50) 58.00 (2.50)	54.50 (2.50) 66.50 (2.50)	68.23 (2.50) 52.77 (2.50)	26.43 (1.50) 34.57 (1.50)	28.73 (1.50) 32.73 (1.50)	53.10 (2.50) 67.90 (2.50)	53.83 (2.50) 67.17 (2.50)	50.27 (2.50) 56.67 (2.50)
8	Objective	34.60 (1.50)	15.70 (1.00)	15.50 (1.00)	15.50 (1.00)	75.50 (3.00)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	16.73 (1.00)
	Both Decision	49.00 (2.00) 52.90 (2.50)	54.70 (2.50) 66.10 (2.50)	53.47 (2.50) 67.53 (2.50)	67.33 (2.50) 53.67 (2.50)	22.17 (1.00) 38.83 (2.00)	62.00 (2.50) 59.00 (2.50)	61.70 (2.50) 59.30 (2.50)	52.43 (2.00) 68.57 (3.00)	55.87 (2.50) 63.90 (2.50)
10	Objective	35.37 (1.50)	15.60 (1.00)	15.50 (1.00)	15.50 (1.00)	75.50 (3.00)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	17.03 (1.00)
	Both Decision	53.87 (2.50) 47.27 (2.00)	51.10 (2.00) 69.80 (3.00)	56.33 (2.50) 64.67 (2.50)	71.53 (3.00) 49.47 (2.00)	28.63 (1.50) 32.37 (1.50)	62.60 (2.50) 58.40 (2.50)	59.97 (2.50) 61.03 (2.50)	54.27 (2.50) 66.73 (2.50)	57.83 (2.50) 61.63 (2.50)
15	Objective	51.47 (2.00)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	75.50 (3.00)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	18.63 (1.00)
	Both Decision	43.63 (2.00) 41.40 (2.00)	52.10 (2.00) 68.90 (3.00)	56.80 (2.50) 64.20 (2.50)	72.03 (3.00) 48.97 (2.00)	30.50 (1.50) 30.50 (1.50)	61.63 (2.50) 59.37 (2.50)	58.53 (2.50) 62.47 (2.50)	52.57 (2.00) 68.43 (3.00)	52.53 (2.50) 65.33 (2.50)
20	Objective	56.23 (2.50)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	75.50 (3.00)	15.50 (1.00)	15.50 (1.00)	15.50 (1.00)	19.23 (1.00)
	Both Decision	39.63 (1.50) 40.63 (2.00)	58.70 (2.50) 62.30 (2.50)	54.57 (2.50) 66.43 (2.50)	69.70 (3.00) 51.30 (2.00)	31.33 (1.50) 29.67 (1.50)	63.53 (2.50) 57.47 (2.50)	56.27 (2.50) 64.73 (2.50)	55.13 (2.50) 65.87 (2.50)	52.50 (2.50) 64.77 (2.50)

Table 14 Mean ranks of the IGD_p as used in the Kruskal-Wallis test for the DTLZ problems. Final ranks, presented in parenthesis assigned according to mean ranks.

Obj.	Metrics	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
3	Euclidean	61.57 (2.50)	60.87 (2.50)	69.23 (2.50)	56.57 (2.50)	61.83 (2.50)	61.17 (2.50)	67.53 (2.50)
	Tchebycheff	57.13 (2.50)	65.33 (3.00)	48.77 (2.50)	62.80 (2.50)	58.67 (2.50)	60.90 (2.50)	53.67 (2.50)
	Minkowski (0.5)	57.17 (2.50)	39.67 (1.50)	62.90 (2.50)	58.47 (2.50)	56.23 (2.50)	56.03 (2.50)	55.00 (2.50)
	Minkowski (4)	66.13 (2.50)	76.13 (3.00)	61.10 (2.50)	64.17 (2.50)	65.27 (2.50)	63.90 (2.50)	65.80 (2.50)
5	Euclidean	56.23 (2.50)	69.90 (3.00)	59.70 (2.50)	55.60 (2.50)	57.63 (2.50)	60.23 (2.50)	62.07 (2.50)
	Tchebycheff	61.50 (2.50)	65.40 (3.00)	62.27 (2.50)	51.17 (2.50)	63.83 (2.50)	52.70 (2.50)	59.33 (2.50)
	Minkowski (0.5)	69.20 (2.50)	35.40 (1.00)	66.33 (2.50)	68.90 (2.50)	52.23 (2.50)	64.93 (2.50)	65.10 (2.50)
	Minkowski (4)	55.07 (2.50)	71.30 (3.00)	53.70 (2.50)	66.33 (2.50)	68.30 (2.50)	64.13 (2.50)	55.50 (2.50)
8	Euclidean	65.10 (2.50)	60.87 (3.00)	66.90 (2.50)	62.23 (2.50)	58.10 (2.00)	61.80 (2.50)	64.57 (2.50)
	Tchebycheff	42.77 (2.00)	80.13 (3.00)	52.23 (2.50)	61.13 (2.50)	47.60 (2.00)	67.43 (2.50)	60.33 (2.50)
	Minkowski (0.5)	69.87 (3.00)	17.87 (1.00)	64.27 (2.50)	58.67 (2.50)	83.73 (4.00)	52.17 (2.50)	56.20 (2.50)
	Minkowski (4)	64.27 (2.50)	83.13 (3.00)	58.60 (2.50)	59.97 (2.50)	52.57 (2.00)	60.60 (2.50)	60.90 (2.50)
10	Euclidean	66.07 (2.50)	61.80 (3.00)	53.00 (2.50)	55.57 (2.50)	51.90 (2.00)	55.63 (2.50)	56.07 (2.50)
	Tchebycheff	52.80 (2.50)	77.00 (3.00)	64.87 (2.50)	70.83 (2.50)	49.67 (2.00)	59.53 (2.50)	57.47 (2.50)
	Minkowski (0.5)	69.17 (2.50)	37.60 (1.00)	64.37 (2.50)	58.20 (2.50)	83.93 (4.00)	78.67 (3.00)	64.40 (2.50)
	Minkowski (4)	53.97 (2.50)	65.60 (3.00)	59.77 (2.50)	57.40 (2.50)	56.50 (2.00)	48.17 (2.00)	64.07 (2.50)
15	Euclidean	63.47 (2.50)	61.60 (3.00)	48.43 (2.50)	58.50 (2.50)	53.03 (2.00)	56.13 (2.50)	48.13 (2.00)
	Tchebycheff	59.03 (2.50)	69.10 (3.00)	71.20 (2.50)	68.00 (2.50)	51.30 (2.00)	59.57 (2.50)	97.17 (3.50)
	Minkowski (0.5)	72.70 (3.00)	35.13 (1.00)	60.97 (2.50)	55.60 (2.50)	90.93 (4.00)	73.10 (2.50)	20.10 (1.00)
	Minkowski (4)	46.80 (2.00)	76.17 (3.00)	61.40 (2.50)	59.90 (2.50)	46.73 (2.00)	53.20 (2.50)	76.60 (3.50)
20	Euclidean	67.33 (2.50)	57.40 (2.50)	60.60 (2.50)	50.80 (2.00)	44.57 (2.00)	63.63 (2.50)	47.07 (2.00)
	Tchebycheff	52.57 (2.50)	77.17 (3.00)	58.53 (2.50)	62.03 (2.50)	45.80 (2.00)	63.73 (2.50)	69.53 (3.00)
	Minkowski (0.5)	69.70 (2.50)	34.87 (1.50)	64.67 (2.50)	50.00 (2.00)	99.07 (4.00)	59.03 (2.50)	34.87 (1.50)
	Minkowski (4)	52.40 (2.50)	72.57 (3.00)	58.20 (2.50)	79.17 (3.50)	52.57 (2.00)	55.60 (2.50)	90.53 (3.50)

Table 15 Mean ranks of the IGD_p as used in the Kruskal-Wallis test for the WFG problems. Final ranks, presented in parenthesis assigned according to mean ranks.

Obj.	Metrics	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
3	Euclidean	62.50 (2.50)	61.70 (2.50)	59.47 (2.50)	48.30 (2.50)	69.90 (2.50)	53.10 (2.50)	60.60 (2.50)	48.03 (2.50)	64.77 (2.50)
	Tchebycheff	60.23 (2.50)	66.37 (2.50)	63.27 (2.50)	64.20 (2.50)	59.30 (2.50)	64.90 (2.50)	65.93 (2.50)	67.70 (2.50)	60.57 (2.50)
	Minkowski (0.5)	62.23 (2.50)	60.17 (2.50)	59.87 (2.50)	61.73 (2.50)	49.23 (2.50)	63.17 (2.50)	59.37 (2.50)	64.77 (2.50)	50.60 (2.50)
5	Minkowski (4)	57.03 (2.50)	53.77 (2.50)	59.40 (2.50)	67.77 (2.50)	63.57 (2.50)	60.83 (2.50)	56.20 (2.50)	61.50 (2.50)	66.07 (2.50)
	Euclidean	61.53 (2.50)	58.03 (2.50)	54.50 (2.50)	60.63 (2.50)	51.17 (2.50)	46.90 (2.50)	55.47 (2.00)	61.67 (2.50)	55.57 (2.50)
	Tchebycheff	52.63 (2.50)	66.97 (2.50)	74.67 (2.50)	43.97 (2.00)	74.53 (3.50)	69.27 (2.50)	57.83 (2.00)	39.37 (2.00)	64.33 (2.50)
8	Minkowski (0.5)	64.70 (2.50)	51.50 (2.50)	52.73 (2.50)	70.43 (3.00)	51.00 (2.00)	65.63 (2.50)	85.83 (4.00)	83.60 (3.50)	65.53 (2.50)
	Minkowski (4)	63.13 (2.50)	65.50 (2.50)	60.10 (2.50)	66.97 (2.50)	65.30 (2.50)	60.20 (2.50)	42.77 (2.00)	57.37 (2.00)	56.57 (2.50)
	Euclidean	51.83 (2.50)	50.47 (2.50)	55.63 (2.50)	66.77 (2.50)	60.83 (2.50)	42.67 (2.00)	44.60 (2.00)	49.80 (2.50)	43.63 (1.50)
10	Tchebycheff	66.90 (2.50)	72.00 (2.50)	71.30 (3.00)	56.47 (2.50)	50.03 (2.00)	61.90 (2.50)	61.97 (2.00)	69.90 (2.50)	67.10 (3.00)
	Minkowski (0.5)	61.00 (2.50)	55.17 (2.50)	42.40 (1.50)	49.47 (2.50)	74.60 (3.00)	84.30 (3.50)	87.77 (4.00)	70.37 (2.50)	73.50 (3.00)
	Minkowski (4)	62.27 (2.50)	64.37 (2.50)	72.67 (3.00)	69.30 (2.50)	56.53 (2.50)	53.13 (2.00)	47.67 (2.00)	51.93 (2.50)	57.77 (2.50)
15	Euclidean	65.50 (2.50)	60.43 (2.50)	70.00 (3.00)	54.63 (2.00)	58.03 (2.50)	50.23 (2.00)	38.13 (1.50)	45.80 (2.00)	55.30 (2.50)
	Tchebycheff	58.83 (2.50)	52.73 (2.50)	66.53 (3.00)	78.67 (3.50)	54.83 (2.00)	55.03 (2.00)	61.50 (2.50)	83.03 (3.50)	62.20 (2.50)
	Minkowski (0.5)	58.13 (2.50)	65.53 (2.50)	38.33 (1.00)	39.10 (1.50)	80.97 (3.50)	92.53 (4.00)	102.53 (4.00)	60.27 (2.50)	59.13 (2.50)
20	Minkowski (4)	59.53 (2.50)	63.30 (2.50)	67.13 (3.00)	69.60 (3.00)	48.17 (2.00)	44.20 (2.00)	39.83 (2.00)	52.90 (2.00)	65.37 (2.50)
	Euclidean	51.70 (2.50)	48.40 (2.00)	50.50 (2.00)	64.47 (2.50)	68.17 (3.00)	52.67 (2.00)	42.87 (2.00)	46.00 (2.00)	52.27 (2.00)
	Tchebycheff	67.53 (2.50)	74.83 (3.00)	86.33 (3.50)	88.03 (3.50)	34.97 (1.50)	55.17 (2.00)	59.27 (2.00)	65.10 (2.50)	58.00 (2.50)
20	Minkowski (0.5)	52.03 (2.50)	54.33 (2.50)	38.33 (1.50)	18.50 (1.00)	90.00 (3.50)	84.93 (4.00)	97.23 (4.00)	69.83 (3.00)	78.67 (3.50)
	Minkowski (4)	70.73 (2.50)	64.43 (2.50)	66.83 (3.00)	71.00 (3.00)	48.87 (2.00)	49.23 (2.00)	42.63 (2.00)	61.07 (2.50)	53.07 (2.00)
	Euclidean	71.60 (2.50)	56.10 (2.00)	44.17 (2.00)	68.03 (3.00)	74.67 (3.00)	56.27 (2.00)	33.10 (1.50)	54.43 (2.50)	63.63 (2.50)
20	Tchebycheff	58.40 (2.50)	87.03 (4.00)	93.20 (4.00)	80.70 (3.00)	29.00 (1.50)	50.63 (2.00)	66.83 (2.50)	64.97 (2.50)	51.93 (2.50)
	Minkowski (0.5)	51.40 (2.50)	39.93 (2.00)	39.20 (1.50)	16.80 (1.00)	97.77 (4.00)	92.60 (4.00)	98.10 (4.00)	70.23 (2.50)	73.80 (2.50)
	Minkowski (4)	60.60 (2.50)	58.93 (2.00)	65.43 (2.50)	76.47 (3.00)	40.57 (1.50)	42.50 (2.00)	43.97 (2.00)	52.37 (2.50)	52.63 (2.50)