

AN IMPROVEMENT OF CLUSTERING-BASED OBJECTIVE REDUCTION METHOD FOR MANY-OBJECTIVE OPTIMIZATION PROBLEMS

Nguyen Xuan Hung^{1,2}, Bui Thu Lam², Tran Cao Truong²

Abstract

Multi-Objective Evolutionary Algorithms (MOEAs) have been widely used for solving multi-objective optimization problems (MOPs). As the number of objectives is more than three, MOPs are regarded as many-objective optimization problems (MaOPs). They bring difficulties to existing MOEAs in terms of deterioration of the search ability, requirement of exponential population size, visualization of results, and especially computation cost. Although in real world, numerous optimization problems are considered as MaOPs but they contain redundant objectives. With these problems, removing the redundant objectives can alleviate these difficulties of MaOPs. Recently, clustering has been applied to remove redundant objectives such as Mutual Information and Clustering Algorithm (MICA-NORMOE) and Objective Clustering-Based Objective Reduction Algorithm (OC-ORA). However, these clustering-based algorithms are computationally complex and they also manually tune a set of parameters. This paper proposes a clustering-based objective reduction method (COR) for MaOPs. The proposed method is more efficient than existing ones and it can automatically determine these parameters. Moreover, the proposed method still achieves comparable results with existing methods.

Index terms

Many-objective optimization problem, PAM, Silhouette, objective reduction.

1. Introduction

IN many areas of social life, there often exist optimization problems with more than one objective which conflicts to each other. These problems are referred as MOPs [1]. One of the most common method to deal with these problem is to use evolutionary computation (EC) techniques such as genetic algorithm and particle swarm optimization [2]. EC has advantages including the simplicity of the approach, broad applicability, outperforming traditional methods on real problems, and capability for self-optimization [3]. MOEAs refer to algorithms using EC technique to evolve solutions for MOPs. MOEAs can evolve multiple solutions in a single run. Moreover, they can

¹ Military Science Academy

² Le Quy Don Technical University

achieve better solutions than traditional methods. For example, NSGA-II [4] is one of the most well-known MOEAs.

MOPs with more than three objectives are usually considered as many-objective problems (MaOPs). When dealing with these MaOPs, MOEAs encounter a number of obstacles. First, if applying Pareto-based MOEAs, a majority of population becomes “good” equally, so it is difficult to select candidates for next generation. Second, the size of population needs to increase exponentially to describe possible solutions. Third, it is difficult to envision solutions for decision makers to select a final solution [5].

The approaches for solving MaOPs can be categorized into 5 groups [6]: preference ordering relation-based, preference incorporation-based, indicator-based, decomposition-based, and objective reduction. The first four groups suppose that there is no redundant objectives in a given problem and try to directly alleviate or eliminate the difficulties encountered. For instance, NSGA-III [7] belongs to these groups. In contrast, the last group which is called objective reduction supposes that there remain redundant objectives in the given problem, and try to find a minimum subset of the original objectives which can generate the same Pareto Front as whole original objectives do [8, 9, 10].

By reducing the redundant objectives, the objective reduction approach has three main benefits. First, it can reduce the computational load of an MaOEA, i.e. it makes less time to operate and less space to store. Second, the problem with selected objectives can be solved by other MOEAs. Finally, thank to pointing out the redundant objectives, it can help decision makers understand the MaOP better. It is consistent with other approaches and is easy to be combined with other approaches [8, 11].

There exist two main objective reduction approaches. Those are dominance structure-based approach and correlation-based one. The first approach tries to retain the dominance relations as much as possible when removing redundant objectives in the given nondominated solutions [12, 13]. That is synonymous with solving the NP-hard problem which determines a minimum subset of objectives without lossing information (MOSS) [14]. The second approach aims to keep the most conflict objectives and remove the objectives that are low conflict, or non-conflict to other(s). This approach uses the correlation or mutual information of non-dominated solutions, which are used to measure the conflict between objectives [15, 16, 17].

Clustering involves grouping a set of objects in such a way that objects in the same group are more “close” to each other, and objects in the different group are “far” to each other. Recently, clustering has been used as a correlation based approach to remove redundant objectives when solving MaOPs [18, 19, 20]. In [18], objectives are divided into homogeneous neighborhoods around each objective. After that the most compact neighborhood is selected, the center of the neighborhood is then retained while other of that are discarded. Mutual Information and Clustering Algorithm (MICA-NORMOEA) [19] and Objective Clustering-Based Objective Reduction Algorithm (OC-ORA) [20] use partitioning around medoids (or PAM in short) clustering algorithm to cluster objective set and remove redundant objective(s). However, these methods are

computationally expensive and they need to provide a set a parameter in advanced.

This paper proposes a clustering-based objective reduction method (COR) for removing the redundant objectives in MaOPs. Instead of requiring a number of clusters of objectives in advanced, COR can automatically determine the number by itself, and it does not need to set up parameter to determine the redundant objectives. Therefore, it can reduce the computation time, and still attains as good results as existing methods.

The rest of this paper is organized as follows. Section 2 shows an overview of related work. Proposed algorithm is given in Section 3. Section 4 presents experimental setup. The result and discussion are showed in Section 5. Conclusion and future work are given in Section 6.

2. Related Works

This section presents related works including multi-objective optimization, many-objective optimization, and objective reduction.

2.1. Multi-Objective Optimization

Multi-objective optimization problem as defined as follows [1]:

$$\begin{aligned} & \text{minimize } \mathbf{f} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{subject to } \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

where there are k (≥ 2) objective function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. The decision vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ belong to (nonempty) feasible region Ω , which is a subset of decision variable space $\mathbb{R}^n$. All of k objective functions need to be minimized simultaneously. It is supposed that there does not remain a single optimal solution. The image of region by $\mathbb{Z} = f(\Omega)$, which is a subset of the objective space $\mathbb{R}^k$ is called the feasible objective region. The elements of $\mathbb{Z}$ are called objective vectors and denoted by $\mathbf{f}(\mathbf{x})$ or $\mathbf{z} = (z_1, z_2, \dots, z_k)^T$, where $z_i = f_i(\mathbf{x})$ for all $i = 1, \dots, k$ are objective values.

In order to solve MOPs, there exist two main techniques that are weighted sum technique and evolutionary computation based technique. The weighted sum technique solves MOPs by transforming the problem into a single objective optimization problem. After transformation, the new problem has a single objective function, then it can be solved by using existing methods and theories for single objective optimization. The advantages of weighted sum techniques are easy to understand, implement, and computationally efficient. The main disadvantage of weighted sum techniques is that the results depend on determination of weighting coefficients which are not easy to predetermine [1].

The evolutionary computation-based technique solves the MOP by using evolutionary algorithms to approximate optimal solutions for the problem. By evolving a population of solutions, MOEAs are able to approximate a set of optimal solutions in a single run and can be applied to problems that can be formulated as a function optimization task.

In the past few decades, researchers have proposed plenty of MOEAs. Some well-known MOEAs are nondominated sorting genetic algorithm II (NSGA-II) [4], strength Pareto evolutionary algorithm (SPEA2) [21], Pareto archived evolution strategy (PAES) [22], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [23].

2.2. Many-Objective Optimization

The MOEAs have been studied in solving effectively the MOPs of 2 or 3 objectives. However, in real-world, there exist MOPs with more than three objectives which are considered as many-objective optimization problems (MaOPs). When tackling these MaOPs, MOEAs encounter a number of difficulties. The first difficulty is that when applying a well-known and frequently-used Pareto-dominance based MOEAs such as NSGAII [4] to solve MaOPs, a large portion of population becomes non-dominated, so we cannot determine which solutions are better for next generation. When using none-Pareto-based MOEAs such as aggregation-based and indicator-based approaches such as MSOPS [24] or IBEA [25], they still have to search simultaneously in an exponentially increasing number of directions. The second difficulty is that the size of population has to increase exponentially to describe the front result [5]. The third difficulty is visualization the solution set in order to help decision makers to choose the final solution [5].

Many MaOEAs have been proposed to solve MaOPs. They can be categorized into 5 groups [6]: preference ordering relation-based, preference incorporation-based, indicator-based, decomposition-based, and objective reduction. The first four groups suppose that there are not any redundant objectives in these problems, and try to directly alleviate or eliminate the difficulties encountered. MaOEAs such as reference-point based non-dominated sorting (NSGA-III) [7], grid-based evolutionary algorithm (GrEA) [26], and knee point driven evolutionary algorithm (KnEA) [27] belong to these classes. In contrast, the last group or objective reduction group supposes that there exist redundant objectives in the given problem, and tries to find a minimum subset of the original objectives which can generate the same Pareto Front as whole objectives do.

2.3. Objective Reduction

Since in many fields of data mining or statistics, dimensionality reduction methods are being used to tackle explosion of data available from experimental life sciences. In general, dimensionality reduction methods is used to reduce a large feature space to a smaller feature one. There exist two approaches in dimensionality reduction: feature extraction and feature selection. Using feature extraction to extract a set of features to explain data. Feature extraction formulates the reduced features as a linear combination of the original features. Feature selection is utilized to determine a subset of the given features as small as possible in order to represent the given data best. There exist several benefits of dimensionality reduction. First, reducing the storage space required. Second, the time required for performing the same computation is less. Thank to less dimensions and less computing; the algorithms, which original unfit for a large number of dimensions, can be usage [28].

In evolutionary multiobjective optimization, objectives are called features. There also exist 2 approaches in objective reduction: objective feature extraction and objective feature selection. *Objective feature extraction* aims at creating novel features from the original features to explain data. For example, Cheung and Gu in [29] formulated the essential objective as a linear combination of the original objectives with the weight vectors determined based on the correlations of each pair of the original objectives; in [30], Cheung et al proposed an objective extraction which formulates the reduced objective as a linear combination of the original objectives. The correlation between each pair of reduced objectives need to be minimized.

Objective feature selection aims at finding the smallest subset of the given objectives in order to generate the same Pareto Front as the set of original objectives do. This approach can be classified into 2 sub-classes: dominance structure based and correlation based. The former is based on preserving the dominance relations in the given non-dominated solutions. That is, dominance structure is retained as much as possible after removing redundant objectives [31, 32]. The latter bases on the correlation between each of pairs of objectives aiming at keeping the most conflict objectives and remove the objectives that are low conflict, or non-conflict each other. This sub-class uses the correlation or mutual information of objective value of non-dominated solutions to measure the conflict between objectives [8, 19, 33].

In [18] two objective reduction algorithms are proposed by using $(1 - \rho) \in [0, 2]$ to measure the degree of correlation¹ between two objectives. Both algorithms first divide the objective set into homogeneous neighborhoods. Thereafter, the most compact neighborhood is chosen, and all the objectives in it except the center one are removed. While the first algorithm finds the minimum set of objectives with the minimum error possible, the second one tries to finds a subset of objectives of a given size with minimum error possible.

Continuing the above idea, Guo et al proposed MICA-NORMOEA [19] and OC-ORA [20] algorithms. In order to measure the relationship between pairs of objectives, they adopted a new criterion by employing the concept of mutual information. In these algorithms, PAM clustering algorithm (presented in subsection 2-D1) and NSGA-II [4] are invoked iteratively to reduce one redundant objective in each step until criterion is satisfied. To cluster the objective set, the algorithms required entering number of clusters in advance. Moreover, threshold θ which is used for comparing the dissimilarity between two objectives in order to discard one objective, also must be set before the algorithms start.

Beside all algorithms mentioned as an evolutionary single objective approach, Yuan et al in [34] exploited both the global search ability of EC and the decision support characteristic of multiobjective optimization, then proposed a study on evolutionary mul-

¹where $\rho(x, y)$ is the correlation coefficient between random variables x and y , the range of ρ is from -1 to 1, the lower ρ value is, the higher two variables negative correlated, means that one objective increases while the other decreases; and vice versa, the higher ρ is, the higher two variables positive correlated, means that both objectives increase or decrease at the same time

tiobjective approach, namely bi-objective between error level (or dominance structure change, or the change of correlation structure) and the size of the objective set.

2.4. Partitioning Around Medoids and Determination The Number of Clusters

2.4.1. Partitioning Around Medoid (PAM): PAM [35], which is a clustering algorithm, is proposed by Kaufman and Rousseeuw, divides a set of objects into k clusters. The objects which are in the same cluster show a high degree of similarity, while objects belonging to different clusters show a high degree of dissimilarity. A medoid of a cluster is an object whose average dissimilarity to all others in the same cluster is minimal. The PAM has two steps. Step one: initialization k medoids by selects k "centrally located" objects. Step two: if a selected object with an unselected object are interchanged and objective function is smaller then the swap is carried out. Step two is repeated until the objective function can no longer be reduced.

2.4.2. Determination The Number of Clusters: With user's interpretation, depending on the shape, size, and resolution of distribution of objects in space, the correct value of k is often ambiguous. In addition, increasing k without penalty will always reduce the number of errors in the resulting cluster, to the extreme case of zero error if each object is considered its own cluster. Intuitively, the optimal choice of k balances the maximum compression of data using a single cluster and maximum precision by considering each object as one cluster separately. If there is no prior knowledge about value of k then it must be chosen by some way.

One method to determine value of k is use of silhouette index [36]. This method measures how close each object in a cluster is to the objects in its neighboring clusters. The value of each object in data set is computed as follow:

$a(i)$ is average dissimilarity of object i to others in the cluster contains them.

$b(i)$ is minimum of all averages dissimilarity of object i to all objects in each cluster which does not contain object i .

The *silhouette value* of object i is formulated as equation 2.

$$s(i) = \frac{(b(i) - a(i))}{\max \{a(i), b(i)\}} \quad (2)$$

The range of *silhouette value* is in $[-1, 1]$. A value of +1 shows that the object is far away from its neighboring cluster and very close to the cluster to which it is assigned. On the contrary, value of -1 means that the object is close to its neighboring cluster than to the cluster to which it is assigned. And, a value of 0 indicates that object is at the boundary of the distance between the two clusters. After clustering, value of +1 is ideal and -1 is least preferred. Hence, the higher the silhouette value is, the better the cluster configuration is.

3. Proposed Algorithm

Two algorithms MICA-NORMOEa [19] and OC-ORA [20] have a number of limitations. Firstly, they are computational complexity expensive, and they can remove only one objective in each iteration. Secondly, a threshold θ need to be provided for determining which objective should be remove. Furthermore, a number of clusters k needs to be set up manually in advance. The value of k is suggested between 2 and $\lfloor \sqrt{M} \rfloor$, where M is number of objectives. However, with MaOPs have a number of relevant objectives is greater than $\lfloor \sqrt{M} \rfloor$ then the algorithms cannot be applicable [20] or they may give incorrect results.

We propose an improvement for MICA-NORMOEa and OC-ORA algorithms. The algorithm has two key ideas. The first idea, the proposed algorithm can self-determine the number of clusters when it using PAM to partition objects (set of objectives in this case) instead of requiring a specific number of clusters before running the PAM. The second idea is that the proposed algorithm does not need to use a threshold parameter while determining and removing redundant objectives. Algorithm 1 shows the proposed method.

The process of the algorithm includes performing MaOEAs, calculating coefficient matrix, clustering objective set and retain one objective in each cluster. First, a non-dominated set is obtained after performing a MaOEa. Second, based on this non-dominated set, a matrix of distance between objectives is calculated. Third, PAM is executed to cluster the objective and the best number of clusters is determined by using Silhouette index. Final, in each cluster, one objective is retained while others are discarded. The process stops until no more objective is removed.

The following paragraphs show two main components of method:

3.1. Calculation the distance coefficient matrix

In evolutionary multi-objective optimization, we get a set of non-dominated solutions after running multi/many-objective evolutionary algorithms. Based on objectives in solution set, a distance coefficient matrix is computed. This proposed algorithm uses two methods for calculation distance coefficient matrix. The first distance coefficient matrix is interdependence coefficient as proposed in [19, 20]. We call COR using this method COR_i . The second distance coefficient matrix is correlation (the idea from [18]). It equals $(1 - \rho_{i,j})$, where $\rho_{i,j}$ is correlation between objective i and objective j . We call COR using this method COR_c .

3.2. Determination the number of clusters

Pseudo-code between line 4 and line 14 in the proposed algorithm (Algorithm 1) determines the number of cluster. It partitions the objectives with number of clusters from 2 to $sizeof(F_t)$, where $sizeof(F_t)$ is the total number of objectives in current loop. In each way, we calculate the silhouettes for each point (objective), then we compute

Algorithm 1: . COR algorithm

Input: iteration counter $t \leftarrow 0$; original objectives $F_t \leftarrow \{f_1, f_2, \dots, f_M\}$

Output: reduced objective set F_s

```

1 repeat
2   Run an MaOEA algorithm to obtain a non-dominated set  $A_t$ , corresponding
     to  $F_t$ .
   /* Calculate the distance coefficient matrix based on the
     non-dominated set  $A_t$ , PAM and silhouette are used to cluster the
     objectives  $F_t$  into  $k$  clusters */
3   Base on a non-dominated set  $A_t$ , calculate distance coefficient matrix
   /* Determine the number of clusters using silhouette index */
4    $Silhouette_{max} \leftarrow -\infty$ ;
5    $i \leftarrow 2$ ;
6   while  $i \leq sizeof(F_t)$  do
7     Execute PAM for  $A_t$  with  $i$  clusters, calculate  $Silhouette_i$ ;
8     if  $Silhouette_i > Silhouette_{max}$  then
9        $k \leftarrow i$ ;
10       $Silhouette_{max} \leftarrow Silhouette_i$ ;
11      store  $k$  clusters (with replacement);
12    end
13     $i \leftarrow i + 1$ ;
14  end
15   $F_s \leftarrow \emptyset$ ;
   /* for each cluster: one is retained, discard the others */
16  for  $i = 1$  to  $k$  do
17     $F_s \leftarrow F_s \cup (\text{an objective in cluster } i^{th})$ 
18  end
19  if  $F_t = F_s$  then
20     $stop \leftarrow true$ 
21  else
22     $t \leftarrow t + 1$ ;
23     $F_t \leftarrow F_s$ ;
24     $stop \leftarrow false$ ;
25  end
26 until  $stop$ ;

```

the mean of all points $Silhouette_i$. We store the result with $Silhouette_i$ has maximum value. However, according to PAM, the result sensitively depends on the initialization of “centrally located” objects. To overcome this circumstance, we execute PAM with predetermined number of runs and choose the best result. The predetermined number greater is, the better result is. However, it is a limitation of the algorithm.

4. Experimental Design

4.1. Comparison Method

To evaluate the quality of the proposed method, we compare it with other existing clustering-based methods. Both the proposed method and existing clustering-based ones are used to remove redundant objectives as shown in Figure 1

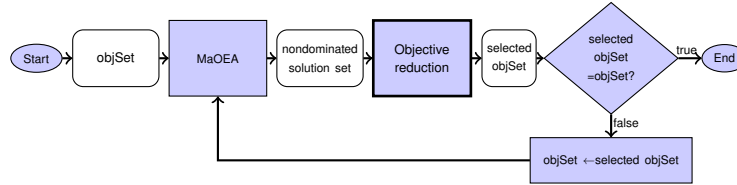


Fig. 1. Working of objective reduction algorithm and many-objective evolutionary algorithm

Figure 1 shows the working of an objective reduction algorithm which integrates with a MaOEA. Stemming from requirements in life, many-objective optimization problem is formulated. We do not know whether all objectives are essential or not. These objectives are regarded as original objectives (or *objSet* in the figure). Firstly, from the original objectives, MaOEA generates a set of nondominated solutions. In the set, columns are objectives. Then, the set is input for objective reduction algorithm (*Objective reduction* block in the figure). The objective reduction algorithms may be OC-ORA [20] (as mentioned in subsection 2-C), or L-PCA [8] (in subsection 4-C). The reduction algorithm removes redundant objectives and retains essential objectives (remainder objectives). The remainder objectives then is input objective set for the MaOEA. The process is continued until the objective reduction algorithm cannot find out any redundant objectives.

We test both the proposed and existing methods on instances of redundant problem DTLZ5IM(I, M) with I relevant objectives and $M - I$ redundant objectives.

4.2. Test Problem

To study, we use DTLZ5IM(I, M) problem [37], it is defined as:

$$\begin{aligned}
 \min f_1(\mathbf{x}) &= (1 + 100g(\mathbf{x}_M))\cos(\theta_1)\cos(\theta_2) \dots \cos(\theta_{M-2})\cos(\theta_{M-1}) \\
 \min f_2(\mathbf{x}) &= (1 + 100g(\mathbf{x}_M))\cos(\theta_1)\cos(\theta_2) \dots \cos(\theta_{M-2})\sin(\theta_{M-1}) \\
 \min f_3(\mathbf{x}) &= (1 + 100g(\mathbf{x}_M))\cos(\theta_1)\cos(\theta_2) \dots \sin(\theta_{M-2}) \\
 &\dots \\
 \min f_{M-1}(\mathbf{x}) &= (1 + 100g(\mathbf{x}_M))\cos(\theta_1)\sin(\theta_2) \\
 \min f_M(\mathbf{x}) &= (1 + 100g(\mathbf{x}_M))\sin(\theta_1)
 \end{aligned}$$

$$\begin{aligned}
 \text{where } \theta_i &= \frac{\pi}{2}x_i & \text{for } i = 1, 2, \dots, (I - 1) \\
 \theta_i &= \frac{\pi}{4(1 + g(\mathbf{x}_M))}(1 + 2g(\mathbf{x}_M)x_i) & \text{for } i = I, \dots, (M - 1) \\
 g &= \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 \\
 0 \leq x_i &\leq 1 & \text{for } i = 1, 2, \dots, n
 \end{aligned}$$

The first property of the problem is that the value² of I can be set in a range between 2 and M . The second one is that Pareto-optimal front is non-convex and follows the relationship: $\sum_{i=1}^M (f_i^*) = 1$. The another property is that there are $M - I$ first objectives correlated, while the others and one of $M - I$ first objective are conflict each other. The experiments are performed on 5 instances of DTLZ5IM(I, M) problem: DTLZ5IM(2,5), DTLZ5IM(3,5), DTLZ5IM(5,10), DTLZ5IM(7,10), and DTLZ5IM(5,20).

4.3. Benchmark Method

We compare the performance of the proposed algorithm with other existing clustering based methods which are OC-ORA [20], and L-PCA [8]. We also use six MOEAs to get non-dominated solutions for both the proposed and existing methods. They are grid-based evolutionary algorithm (GrEA) [26], knee point driven evolutionary algorithm (KnEA) [27], reference-point based non-dominated sorting (NSGA-III) [7], reference vector guided evolutionary algorithm (RVEA*) [38], new dominance relation-based evolutionary algorithm (θ -DEA) [39], and LMEA [40].

4.4. Parameters Setting

All MaOEAs used the experiments are implemented by PlatEMO [41]. The population size is set to 200, and the number of generation is set to 2000. The probability of crossover and mutation is set to 0.9 and 0.1, respectively. The distribution index for crossover is set to 5, and the distribution index for mutation is set to 20. We suggest the predetermined number (in 3-B) is $20 * M$ (where M is current number of objectives of problem).

5. Result and Discussion

This section presents 2 parts. The first part is comparison table between the proposed algorithm with two other objective reduction algorithms integrating with six MaOEAs. The second part is the average number of times COR appeals MaOEAs to finish their reduction process.

To compare the performance of objective reduction algorithms when integrating with MaOEAs, we do the test and show the results in Table 1. The table shows the number of successes in finding the true non-redundant objective set when objective reduction algorithms combining with MaOEAs. The first row of the table lists name of 2 objective reduction algorithms and the proposed algorithm (COR) with two variants CORi and CORc. MaOEAs are written in the first column. The objective reduction algorithms include LPCA (Linear-PCA) [8], OC-ORA [20], and the two variants of COR³.

To investigate whether results of the objective reduction algorithms are significant different to each other in a statistical sense, Wilcoxon signed-rank test is performed. The null hypothesis is that the performance of the two methods are similar with significant

² I is the dimensionality of the Pareto-optimal front

³the proposed algorithm with two methods of calculation distance coefficient matrix

Table 1. The number of times out of 20 runs, objective reduction algorithms has successfully found a set of conflicting objectives when combining with MaOEAs

MaOEAs	DTLZ5IM		Objective Reductions			
	I	M	LPCA	OC-ORA	CORi	CORc
GrEA	2	5	20	15	20	20
	3	5	20	20	20	20
	5	10	20	20	20	20
	7	10	20	19	20	20
	5	20	1	11	1	1
KnEA	2	5	20	15	20	20
	3	5	20	19	18	20
	5	10	20	19	18	20
	7	10	20	16	10	20
	5	20	20	19	18	11
NSGAIII	2	5	20	20	20	20
	3	5	20	17	20	20
	5	10	19	20	7	20
	7	10	20	20	15	20
	5	20	18	0	2	14
RVEA	2	5	20	15	20	20
	3	5	20	19	20	20
	5	10	20	20	20	20
	7	10	20	20	16	20
	5	20	19	3	4	13
θ -DEA	2	5	20	20	20	20
	3	5	20	17	20	20
	5	10	20	19	3	20
	7	10	20	20	20	20
	5	20	19	12	4	17
LMEA	2	5	20	20	20	20
	3	5	20	20	20	20
	5	10	20	20	17	20
	7	10	20	20	20	20
	5	20	20	20	12	9

Table 2. *p* values and hypotheses when do comparison using Wilcoxon signed-rank test

(a) *p*-values

	LPCA	OC-ORA	CORi	CORc
LPCA	1.00	0.01	0.00	0.06
OC-ORA	0.01	1.00	0.17	0.12
CORi	0.00	0.17	1.00	0.01
CORc	0.06	0.12	0.01	1.00

(b) hypotheses

	LPCA	OC-ORA	CORi	CORc
LPCA	0	1	1	0
OC-ORA	1	0	0	0
CORi	1	0	0	1
CORc	0	0	1	0

level at 0.05, and the alternative hypothesis is that the performance of the two methods is significant different. The Table 2a shows *p* values when comparing two algorithms with each other. The Table 2b presents the hypotheses are accepted or rejected. The value 0 means that the null hypothesis is accepted, in other words, the two algorithms are similar with significant level at 0.05. The value 1 means that the null hypothesis is rejected or we accept the alternative hypothesis which mean that the two algorithms are different with significant level at 0.05. As can be seen from Table 2b, the LPCA is different to OC-ORA, or CORi but similar to CORc at significant level 0.05. CORi is similar to OC-ORA but different to LPCA, and CORc.

Table 3. The average number of times objective reduction algorithms calling the MOEAs when it finds the set of conflicting objectives

MaOEAs	DTLZ5IM		objective reductions			
	<i>I</i>	<i>M</i>	<i>LPCA</i>	<i>OC-ORA</i>	<i>CORi</i>	<i>CORc</i>
GrEA	2	5	2.0	3.8	2.0	2.0
	3	5	2.0	3.0	2.0	2.0
	5	10	2.0	6.0	2.0	2.0
	7	10	2.0	4.0	2.0	2.0
	5	20	1.2	15.7	2.1	2.1
KnEA	2	5	2.0	3.8	2.0	2.0
	3	5	2.0	3.1	2.3	2.0
	5	10	2.0	6.0	2.3	2.0
	7	10	2.0	4.1	2.4	2.0
	5	20	2.0	16.0	2.1	2.6
NSGAIII	2	5	2.0	4.0	2.0	2.0
	3	5	2.0	2.9	2.0	2.0
	5	10	2.0	6.0	2.0	2.0
	7	10	2.0	4.0	2.0	2.0
	5	20	2.2	15.9	2.5	3.1
RVEA	2	5	2.0	3.8	2.0	2.0
	3	5	2.0	3.0	2.0	2.0
	5	10	2.0	6.0	2.0	2.0
	7	10	2.0	4.0	2.1	2.0
	5	20	2.0	17.4	2.2	2.1
θ -DEA	2	5	2.0	4.0	2.0	2.0
	3	5	2.0	2.9	2.0	2.0
	5	10	2.0	6.0	2.0	2.0
	7	10	2.0	4.0	2.1	2.0
	5	20	2.0	16.5	2.1	2.6
LMEA	2	5	2.0	4.0	2.1	2.0
	3	5	2.0	3.0	2.1	2.0
	5	10	2.0	6.0	2.1	2.0
	7	10	2.0	4.0	2.1	2.0
	5	20	2.3	16.0	2.1	2.3

Beside the number of times successful in finding non-redundant objective set above, the number of repeating in reduction process is considered when do comparison the performance of the objective reduction algorithm. For OC-ORA algorithm, each loop can reduce at most one redundant objective. Table 3 shows that COR and LPCA can execute almost 2 loops to finish the process, while OC-ORA need $M - I + 1$ loops. As a result, COR and LPCA are equivalent, and they more efficient than the OC-ORA, specially in high redundant instances.

6. Conclusion and Future Works

The paper proposed a clustering-based objective reduction method which uses PAM algorithm for clustering objectives in MaOPs for determining redundant objectives. The proposed algorithm used Silhouette index to self-determine the number of clusters

instead requiring it as a parameter. We compared the proposed method with two other existing methods on different instances of DTLZ5IM(I, M) problem. The result showed that the proposed algorithm is not only equivalent to or more efficient than existing methods but also it can retain comparable results as existing methods can do.

There are other methods which are able to automatically determine the number of clusters, future works could investigate these methods for the proposed method.

References

- [1] K. Miettinen, "Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science," 1999.
- [2] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.
- [3] D. B. Fogel, "The advantages of evolutionary computation," in *BCEC*, 1997, pp. 1–11.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC2008, June 1-6, 2008, Hong Kong*, 2008, pp. 2419–2426.
- [6] S. Bechikh, M. Elarbi, and L. B. Said, "Many-objective optimization using evolutionary algorithms: a survey," in *Recent Advances in Evolutionary Multi-objective Optimization*. Springer, 2017, pp. 105–137.
- [7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [8] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.
- [9] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, "Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive/critical analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2018.
- [10] X. H. Nguyen, L. T. Bui, and C. T. Tran, "Improving many objective optimisation algorithms using objective dimensionality reduction," *Evolutionary Intelligence*, pp. 1–16, 2019.
- [11] H. K. Singh, A. Isaacs, and T. Ray, "A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.
- [12] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary computation*, vol. 17, no. 2, pp. 135–166, 2009.
- [13] F. Gu, H.-L. Liu, and Y.-m. Cheung, "A fast objective reduction algorithm based on dominance structure for many objective optimization," in *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 2017, pp. 260–271.
- [14] D. Brockhoff and E. Zitzler, "On objective conflicts and objective reduction in multiple criteria optimization," *TIK Report*, vol. 243, 2006.
- [15] D. K. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 772–787.
- [16] A. L. Jaimes, C. A. C. Coello, and J. E. U. Barrientos, "Online objective reduction to deal with many-objective problems," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2009, pp. 423–437.
- [17] X. H. Nguyen, L. T. Bui, and C. T. Tran, "Clustering method using pareto corner search evolutionary algorithm for objective reduction in many-objective optimization problems," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, ser. SoICT 2019. New York, NY, USA: ACM, 2019, pp. 78–84. [Online]. Available: <http://doi.acm.org/10.1145/3368926.3369720>
- [18] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proceedings of the 10th annual conference on Genetic and Evolutionary Computation*. ACM, 2008, pp. 673–680.
- [19] X. Guo, X. Wang, M. Wang, and Y. Wang, "A new objective reduction algorithm for many-objective problems:

- employing mutual information and clustering algorithm,” in *Computational Intelligence and Security (CIS), 2012 Eighth International Conference on*. IEEE, 2012, pp. 11–16.
- [20] X. Guo, Y. Wang, and X. Wang, “Using objective clustering for solving many-objective optimization problems,” *Mathematical Problems in Engineering*, vol. 2013, 2013.
 - [21] E. Zitzler, M. Laumanns, and L. Thiele, “Spea2: Improving the strength pareto evolutionary algorithm,” *TIK-report*, vol. 103, 2001.
 - [22] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the pareto archived evolution strategy,” *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
 - [23] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.
 - [24] E. J. Hughes, “Multiple single objective pareto sampling,” in *Congress on Evolutionary Computation 2003*, 2003, pp. 2678–2684.
 - [25] E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.
 - [26] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.
 - [27] X. Zhang, Y. Tian, and Y. Jin, “A knee point-driven evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.
 - [28] C. O. S. Sorzano, J. Vargas, and A. P. Montano, “A survey of dimensionality reduction techniques,” *arXiv preprint arXiv:1403.2877*, 2014.
 - [29] Y.-m. Cheung and F. Gu, “Online objective reduction for many-objective optimization problems,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1165–1171.
 - [30] Y.-m. Cheung, F. Gu, and H.-L. Liu, “Objective extraction for many-objective optimization problems: Algorithm and test problems,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 755–772, 2016.
 - [31] D. Brockhoff and E. Zitzler, “Dimensionality reduction in multiobjective optimization with (partial) dominance structure preservation: Generalized minimum objective subset problems,” *TIK Report*, vol. 247, 2006.
 - [32] —, “Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization,” in *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 533–542.
 - [33] H. X. Nguyen, L. T. Bui, and C. T. Tran, “A Pareto Corner Search Evolutionary Algorithm and Principal Component Analysis for Objective Dimensionality Reduction,” in *2019 11th International Conference on Knowledge and Systems Engineering (KSE) (KSE’19)*, Da Nang, Vietnam, Oct. 2019.
 - [34] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, “Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2018.
 - [35] L. Kaufman and P. Rousseeuw, *Clustering by means of medoids*. North-Holland, 1987.
 - [36] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
 - [37] K. Deb and D. K. Saxena, “On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems,” *Kangal report*, vol. 2005011, 2005.
 - [38] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
 - [39] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.
 - [40] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2018.
 - [41] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

Manuscript received 07-5-2019; Accepted 18-12-2019.

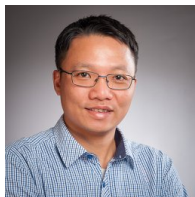




Nguyen Xuan Hung received his B.Sc (1998) and M.Sc (2008) degrees in computer science and technology from Le Quy Don Technical University. Currently he is a PhD Candidate at the Faculty of Information Technology, Le Quy Don Technical University. His current research interests include Evolutionary Multi-Objective/Many-Objective Optimization and Objective Dimensionality Reduction. His email is nguyensexuanhung@outlook.com



Bui Thu Lam received the Ph.D. degree in computer science from the University of New South Wales (UNSW), Australia, in 2007. He did postdoctoral training at UNSW from 2007 until 2009. He has been involved with academics including teaching and research since 1998. Currently, he is an Associate Professor at Le Quy Don Technical University, Hanoi, Vietnam. He is doing research in the field of evolutionary computation, specialized with evolutionary multi-objective optimization.



Tran Cao Truong received the B.Sc in applied mathematics and informatics from from Ha Noi University of Science, Vietnam, in 2005 and the M.S. degree from Le Qui Don Technical University, Viet Nam in 2009. He received the PhD degree in computer science from Victoria University of Wellington, New Zealand, in 2018. He is researching in the field of evolutionary computation, specialized with evolutionary machine learning for data mining with missing data. He servers as a reviewer of international journals, including IEEE Transactions on Evolutionary Computation, IEEE Transactions on Cybernetics, Pattern Recognition, Knowledge-Based Systems, Applied Soft Computing and Engineering Application of Artificial Intelligence. He is also a PC member of international conferences, including IEEE Congress on Evolutionary Computation, IEEE Symposium Series on Computational Intelligence, the Australasian Joint Conference on Artificial Intelligence and the AAAI Conference on Artificial Intelligence.

GIẢI THUẬT GIẢM CHIỀU MỤC TIÊU CẢI TIẾN SỬ DỤNG PHƯƠNG PHÁP PHÂN CỤM CHO BÀI TOÁN TỐI ƯU NHIỀU MỤC TIÊU

Tóm tắt

Bài toán tối ưu nhiều mục tiêu đã nhận được sự quan tâm đáng kể từ các nhà nghiên cứu gần đây. Khi số lượng mục tiêu tăng lên, chúng gây ra những khó khăn cho các thuật toán tiến hóa đa mục tiêu hiện tại về toán tử lựa chọn, yêu cầu kích thước quần thể lớn, việc trực quan hóa tập kết quả, đặc biệt là chi phí tính toán. Một số bài toán có mục tiêu dư thừa và có thể bị thoái hóa thành các bài toán có kích thước thấp hơn. Việc loại bỏ các mục tiêu dư thừa sẽ tiết kiệm thời gian và không gian khi giải quyết bài toán. Bài báo này cải tiến phương pháp giảm kích thước sử dụng thuật toán PAM thực hiện phân cụm các mục tiêu sau đó loại bỏ các mục tiêu dư thừa. Hai biến thể của thuật toán đề xuất tự xác định số mục tiêu dư thừa và hiệu quả hơn thuật toán gốc.