Clustering Method using Pareto Corner Search Evolutionary Algorithm for Objective Reduction in Many-Objective Optimization Problems

Xuan Hung Nguyen
(1) Le Quy Don Technical University
(2) Military Science Academy
Hanoi, Vietnam
nguyenxuanhung@outlook.com

Lam Thu Bui
Le Quy Don Technical University
Hanoi, Vietnam
lamtb@lqdtu.edu.vn

Cao Truong Tran
Le Quy Don Technical University
Hanoi, Vietnam
truongtc@lqdtu.edu.vn

ABSTRACT

Many-objective optimization problems (MaOPs) have been gained considerable attention for researcher, recently. MaOPs make a number of difficulties for multi-objective optimization evolutionary algorithms (MOEAs) when solving them. Although, there exist a number of many-objective optimization evolutionary algorithms (MaOEAs) for solving MaOPs, they still face difficulties when the number of objectives of MaOPs increases. One common method to reduce or alleviate these difficulties is to use objective dimensionality reduction (or objective reduction for briefly). Moreover, instead of searching the whole of objective space like existing MOEAs or MaOEAs, Pareto Corner Search Evolutionary (PCSEA) concentrates only on some places of objective space, so it decreases time consuming and then speeds up objective reduction. However, PCSEA-based objective reduction needs to specify a threshold to select or remove objectives, which is not straightforward to do. Based on the idea that more conflict two objectives are, more distant two objectives are; in this paper, we introduce a new objective reduction by integrating PCSEA and k-means, DBSCAN clustering algorithms for solving MaOPs which are assumed containing redundant objectives. The experimental results show that the introduced method can reducing redundant objectives better than PCSEA-based objective reduction. The results further strengthen the links between evolutionary computation and machine learning to address optimization problems.

CCS CONCEPTS

• **Mathematics of computing** → *Dimensionality reduction*; *Mathematical optimization*; *Evolutionary algorithms.*

KEYWORDS

many-objective optimization, objective reduction, clustering

ACM Reference Format:
Xuan Hung Nguyen, Lam Thu Bui, and Cao Truong Tran. 2019. Clustering Method using Pareto Corner Search Evolutionary Algorithm for Objective Reduction in Many-Objective Optimization Problems. In *The Tenth International Symposium on Information and CommunicationTechnology (SoICT 2019), December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3368926.3369720

1 INTRODUCTION

In social life, there often exist problems with two or more objectives which are conflict to each other. The problems are often referred as multi-objective problems (MOPs) [22]. In an MOP, different solutions are likely to have an edge of objective over other objectives, so the term Pareto dominance concept is used widely to differentiate them. Most common MOEAs try to approximate the objective Pareto Front (PF) space so that no we cannot enhance any objectives without sacrificing the quality of others [17].

There exist a number of evolutionary computation (EC) techniques for dealing with these MOPs [9] such as particle swarm optimization, differential evolution, or genetic algorithms. Many benefits can gained when using EC. They include the simplicity of the approach, broad applicability, outperforming traditional methods on real problems, and the capability for self-optimization [11]. MOEAs refer to algorithms using EC technique to evolve solutions for solving MOPs. MOEAs can evolve multiple solutions in a single run, and they can achieve better solutions than traditional methods, for instance, SPEA2 [34], NSGA-II [7].

MOPs with more than three objectives are usually regarded as MaOPs (Many-Objective Optimization Problems) [3, 14]. When dealing with these MaOPs, MOEAs encounter a number of difficulties. Firstly, with Pareto-based MOEAs, a large portion of population becomes incomparable, so it is difficult to identify which candidates for next generation. Moreover, the size of population increases exponentially when approximating the entire PF. Finally, it is not easy to visualize the results for decision makers to select a final solution [15].

Approach to solving MaOPs can be categorized to two groups. The first group assumes that there is not any redundant objective in a given problem, and the methods in first group, such as HypE [1], Two_Arch2 [27], try to directly eliminate the difficulties encountered. The algorithms in the first group are called Many-Objective Evolutionary Algorithms (MaOEAs). In contrast, the second group supposes that there remain redundant objectives in the given problem, and the methods in second group, such as NCIE [28], Exact or Greedy algorithms [2], try to remove the redundant objectives before using MOEAs or MaOEAs to search for PF. By removing redundant objectives, the objective reduction approach has three main

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam

ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369720

benefits. Firstly, it can reduce the computation of an MaOEA, i.e. it makes less time to operate and less space to store. Furthermore, the problem with less objectives can be even solved by other MOEAs. Finally, it can help decision makers better understand the MaOP by indicating the irrelevant objectives or redundant ones [20, 25].

There exist two methods in objective reduction. The first one is the structure-based method. It tries to retain the dominance relations as much as possible when removing redundant objectives. The δ-OR, η-OR in [29] and the PCSEA-based objective dimensionality reduction in [26] are examples of the first method. The second one is the correlation-based method. It uses metrics such as correlation to evaluate the relation between objectives of nondominated solutions, then the objectives that are low conflict, or non-conflict to other are removed while others are retained [13, 16].

Both the two objective reduction methods need an approximate nondominated solution set which is generated by MOEAs or MaOEAs. However, these evolutionary algorithms often target the set covering the whole true PF. Because of requiring MOEAs/MaOEAs for evolving whole PF, most existing objective reduction methods require a large of calculation, especially when solving problems having numerous objectives.

In contrast to most existing methods, Pareto corner search evolutionary algorithm (PCSEA) in [26] only locates in intersections of PF's boundaries then search for solutions. The result is that the PCSEA-based objective reduction method run much faster than other methods do. However, the PCSEA-based objective reduction method requires to define a threshold to remove or retain objectives. Moreover, the authors in [23] shows that the threshold strongly depends on each problem, but the threshold was fixed in [26]. Therefore, this paper proposes a new method to alleviate the limitation of the PCSEA-based objective reduction method by using clustering algorithms for objective reduction. Results display that the proposed method can perform objective reduction both effectively and efficiently.

The rest of this paper is organized as follows. Section 2 gives some related work. Section 3 presents the proposed method. Section 4 shows experimental design. Section 5 shows the results and discussion. Section 6 concludes and states the future work.

2 RELATED WORK

This section presents related work including multi and many-objective optimization, objective dimensionality reduction, and Pareto Corner Search Evolutionary Algorithm. Clustering algorithms including k-means and Density Based Spatial Clustering of Applications with Noise (DBSCAN) are also presented in this section.

2.1 Multi and Many-objective optimization

An MOP as defined as follows [22]:

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

where there are k objective functions[1], which are mapping of $\mathbb{R}^n$ to $\mathbb{R}$, and are fully or partially conflict each other and need to be minimised simultaneously. $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is the decision vector, Ω is a subset of $\mathbb{R}^n$.

[1] k is greater than or equal to 2

In order to solve MOPs, there exist two main techniques that are weighted sum technique and evolutionary computation based technique. The first technique solves MOP by transforming it to problem with a single objective (SOP). Then, the SOP can be solved by using existing methods for single objective problems [22].

The evolutionary computation-based technique solves the MOP by using evolutionary algorithms to approximate optimal solutions. 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 any problem that can be formulated as a function optimization task. Plenty of MOEAs have been proposed. Some well-known MOEAs are nondominated sorting genetic algorithm II (NSGA-II) [7], Pareto archived evolution strategy (PAES) [19], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [31].

When MOPs with more than three objectives which are considered as MaOPs [3],[14]. When tackling these MaOPs, MOEAs encounter a number of difficulties. First, when applying Pareto-dominance based MOEAs such as NSGAII [7] to solve MaOPs, a large portion of population becomes nondominated, so we cannot determine which solutions are better for next generation. When using aggregation-based or indicator-based approaches such as IBEA [33], they still have to search simultaneously in an exponentially increasing number of directions. Second, the size of population has to increase exponentially to describe the front result [15]. Third, visualization the solution set is difficult to help decision makers to choose the final solution [15].

MaOEAs, which are proposed to solve MaOPs, can be categorized into 2 classes. The first class supposes that the problems do not contain redundant objectives then directly eliminates the difficulties encountered. It includes sub-classes: decomposition-based, preference ordering relation-based, preference incorporation-based. MaOEAs such as a new dominance relation-based (θ-DEA) [30], reference-point based nondominated sorting (NSGA-III) [6], and knee point driven evolutionary algorithm (KnEA) [32] belong to the first class. In contrast to the first class, the second one assumes 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 PF as whole original objectives do. This class is showed more detail in sub-section 2.2 below.

2.2 Objective Dimensionality Reduction

In order to avoid the curse of dimensionality, dimensionality reduction is often used. In general, we use dimensionality reduction methods for transforming a large feature space to a smaller 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 find the smallest subset of the given features in order to represent the given data best. Dimensionality reduction brings several benefits such as reducing the storage space or time performance required.

In evolutionary multiobjective optimization, objectives are considered as features, and dimensionality reduction is used and is called objective dimensionality reduction or objective reduction for briefly. 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, authors in [4, 5] formulated the essential/reduced objective as a linear combination of the original objectives based on the correlations of each pair of the essential objectives.

Objective feature selection aims at finding the smallest subset of the given objectives in order to generate the same PF as the set of original objectives does. This approach can be classified into 2 sub-classes: Pareto dominance structure based and correlation based. Pareto dominance structure based sub-class is based on preserving the dominance relations in the given nondominated solutions. That means they are retained as many as possible after removing redundant objectives [12]. The correlation based sub-class bases on the correlation between each of pairs of objectives. Then it aims to keep the most conflict objectives and remove the objectives that are non-conflict each other. This sub-class measures the conflict between objectives by using the correlation [25] or mutual information [13] of objective values of solutions set.

2.3 Pareto Corner Search Evolutionary Algorithm (PCSEA)

Objective reduction procedures essentially have two major modules. In the first module, an MOEA ($\epsilon-$MOEA, NSGAII in [25], SPEA2-SDE in [29]) is used to generate an approximation of the PF. In the second module, dimensionality reduction is executed on the obtained approximation of the PF. Those modules may be applied in some loops to get the reduced set of objectives. However, for generating approximate PF, we often call for MOEA for a plenty of generations. Even so, from PF population may still distant, which can make it pointless to extract any information for reduction. Moreover, existing MOEAs often search for approximation to the entire PF. This becomes impossible even with the case of a large number of generations or/and solutions set for high number of objectives. To overcome these difficulties, PCSEA [26] focuses on only "corner" solutions of the PF where the boundaries intersect (Algorithm 1).

Algorithm 1: PCSEA algorithm

Input: N (Population size),
N_G (number of generations)
begin
  Initialize(pop_1)
  Evaluate(pop_1)
  for $i \leftarrow 2$ **to** N_G **do**
    $childpop_i \leftarrow$ Evolve(pop_{i-1})
    Evaluate($childpop_i$)
    $S \leftarrow$ CornerSort ($pop_{i-1} + childpop_i$)
    $pop_i \leftarrow S(1:N)$
  end
end

Working of PCSEA is similar to other evolutionary algorithms such as NSGA-II [7]. While NSGA-II uses nondominated sorting for comparing and ranks the solution by using crowding distance-based, PCSEA uses corner-sort ranking mechanism which ranks solutions basing on individual objective and L_2 norm all-but-one objectives.

Algorithm 2: The PCSEA-based objective reduction algorithm

Input: $F_R \leftarrow \{f_1, f_2, \ldots, f_M\}$ // *original objectives set*
C // *threshold*
Output: F_R // *reduced objective set*
1 run PCSEA to get corner solutions
2 **foreach** $m \in \{1, \ldots, M\}$ **do**
3   $R \leftarrow \frac{N_{F_R \setminus \{f_m\}}}{N_{F_R}}$
4   **if** $R > C$ **then**
5     $F_R \leftarrow F_R \setminus \{f_m\}$
6   **end**
7 **end**

Once the corner solutions of the PF are obtained using PCSEA (in Algorithm 1), the objective reduction module is performed. The idea is Pareto dominance among the solutions will be largely depend on the relevant objectives. That is if one redundant or irrelevant objective is removed then there is no (or negligible) change in the number of nondominated solutions. On the contrary, if one of critical objectives is discarded, that number is changed significantly. The PCSEA-based objective reduction algorithm is shown in Algorithm 2.

The parameter R in step 3 in Algorithm 2 is ratio between N_F and $N_{F_R \setminus \{f_m\}}$, where N_F and $N_{F_R \setminus \{f_m\}}$ are the number of non-dominated solutions in F and $F_R \setminus \{f_m\}$, respectively.

2.4 Clustering algorithms

Clustering involves the dividing a dataset into groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to all available variables [24]. Partition-based and density-based methods are two typical types of clustering method.

2.4.1 K-means clustering algorithm.

K-means [21], which is one of the simplest unsupervised learning algorithms, belongs to partition clustering method. K-means solves the clustering problems where each cluster is represented by center of gravity of the cluster. The k-means algorithm works as Algorithm 3

Algorithm 3: K-means algorithm

Input: k clusters
Initialize centroid of the k clusters
repeat
  Generate a new partition by assigning each data to its closest cluster centroid.
  Compute new cluster centroid for each cluster.
until *the cluster membership stabilized*;

2.4.2 *DBSCAN clustering algorithm.*

DBSCAN [10] is one of density-based clustering algorithm. Clusters are identified by looking at the density of points. Regions with a high density of points depict the existence of clusters whereas regions with a low density of points indicate clusters of noise or clusters of outliers. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. DBSCAN has two parameters: epsilon (*Eps*) and (*minPts*). They mean that the minimum number of points that must be existed in *Eps*. The algorithm starts with an arbitrary starting point that has not been visited. This point's *Eps*-neighborhood is retrieved recursively, and if it contains enough number of points, i.e greater than or equal to *minPts*, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized *Eps*-environment of a different point and hence be made part of a cluster.

If a cluster contains a small number of of points in usual clustering problems then the points are regarded as noise or outlier. However, bases on the features of objective in MaOPs, a point (an objective in MaOPs) which is contained in a cluster, is considered a good point (a *relevant* objective of problem).

3 THE PROPOSED METHOD

The PCSEA-based objective reduction in [26] can efficiently remove redundant objectives. However, this algorithm has a number of drawbacks. First, the cutoff value of R (*C* threshold) must be provided before running objective reduction. Secondly, the objective reduction algorithm did not consider the importance of the order of removing redundant objectives. Finally, the algorithm was tested on DTLZ5(*I,M*) problem with only a small number of relevant objectives (specifically 5).

The main purpose of this paper is to take the advantages and alleviate the limitations of the PCSEA-based objective reduction method. The proposed method uses the PCSEA to generate nondominated solutions. The objectives in the solution set are considered as object (or point) then for clustering to eliminate redundant objectives. Algorithm 4 shows the main steps of the proposed method. We call the algorithm PCS-Cluster.

The algorithm has two key ideas. The first idea, the proposed algorithm can take advantages of the PCSEA, which is able to find some key solutions in the PF with lower complexity than other MOEAs/MaOEAs. The second idea, unlike Algorithm 2, the proposed algorithm avoids using the sensitive parameter *threshold C*.

Algorithm 4: PCS-Cluster objective reduction algorithm

Input: $t \leftarrow 0$ // *step*
$F_t \leftarrow \{f_1, f_2, \dots, f_M\}$ // *original objective set*
Output: F_s // *reduced objective set*

1 **repeat**
2 $P \leftarrow$ PCSEA(F_t) ; `// Get corner solutions corresponding to remaining (current) objective set`
3 $\{C_1, C_2, \dots, C_k\} \leftarrow Clustering\,\{F_t(P)\}$
4 $F_s \leftarrow \varnothing$
`/* for each cluster: retain one, discard the others */`
5 **for** $i = 1$ **to** k **do**
6 $F_s \leftarrow F_s \cup$ (an objective in cluster C_i)
7 **end**
`/* Compare two sets before and after reduction */`
8 **if** $F_t = F_s$ `// If they are same`
9 **then**
10 $stop \leftarrow true$
11 **else**
12 $t \leftarrow t + 1;$
13 $F_t \leftarrow F_s;$
14 $stop \leftarrow false;$
15 **end**
16 **until** *stop*;

Pareto Corner Search Evolutionary Algorithm (PCSEA).
The PCSEA [26] is excuted to get nondominated solutions as line 2 in Algorithm 4. It is similar to NSGAII [7]. While NSGAII uses nondominated sort and crowding distance-based ranking, PCSEA uses corner-sort ranking in which solutions are ranked based on the individual objective values and L_2 norm of all-but-one objectives.

Clustering procedure.
Objectives of solution set are considered as objects in objective space. They are grouped in a number of set which are known as clusters. To measure the distance between objective x and objective y for clustering, we use distance d as formula (2):

$$d = 1 - \rho(x, y) \tag{2}$$

where $\rho(x, y)$ is the Pearson 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.

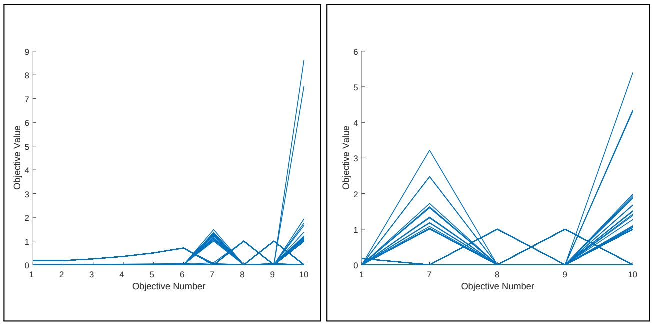
This procedure uses two kinds of clustering algorithms namely k-means [21] and DBSCAN [10]. The k-means divides the set of objectives into k clusters. The value of k is determined using ELBOW method [18]. The ELBOW method computes the distortions under different cluster number counting from 1 to n, and k is the cluster number corresponding 99.0% percentage of variance explained, which is the ratio of the between-group variance to the total variance. DBSCAN automatically divides the set of objectives into a number of clusters using density-based clustering algorithm instead of predetermination the number of clusters.

Working of PCS-Cluster

This sub-section shows the process of working of the algorithm when solving DTLZ5(5,10) problem[2]. Parameters for PCSEA are set as in Table 3 (step 2). Initially, 10 objectives are assigned for F_t set. Fig. 1a draws a parallel coordinates of F_t objectives of solutions set obtained by PCSEA (the solutions concentrate only on "corners" of objective space).

After having the solutions set, a matrix distance (followed formula 2) between objectives is calculated. The matrix distance is showed in Table 1. A cell of row i and column j contains the distance between objective i and objective j.

[2]It is defined in subsection 4.1



(a) ***The first loop (10 objectives)*** **(b)** ***The second loop (5 objectives)***

Figure 1: Parallel coordinate plots for objectives of solution set obtained by PCSEA for solving DTLZ5(5,10) problem

Table 1: The matrix distance between 10 objectives

0.0E+00	0.0E+00	1.1E-15	1.9E-15	2.0E-15	2.3E-15	1.1E+00	1.2E+00	1.2E+00	1.3E+00
0.0E+00	0.0E+00	0.0E+00	1.9E-15	0.0E+00	1.7E-15	1.1E+00	1.2E+00	1.2E+00	1.3E+00
1.1E-15	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	1.1E+00	1.2E+00	1.2E+00	1.3E+00
1.9E-15	1.9E-15	0.0E+00	0.0E+00	0.0E+00	0.0E+00	1.1E+00	1.2E+00	1.2E+00	1.3E+00
2.0E-15	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	1.1E+00	1.2E+00	1.2E+00	1.3E+00
2.3E-15	1.7E-15	0.0E+00	0.0E+00	0.0E+00	0.0E+00	1.1E+00	1.2E+00	1.2E+00	1.3E+00
1.1E+00	1.1E+00	1.1E+00	1.1E+00	1.1E+00	1.1E+00	0.0E+00	1.1E+00	1.1E+00	1.2E+00
1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.1E+00	0.0E+00	1.1E+00	1.2E+00
1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.2E+00	1.1E+00	1.1E+00	0.0E+00	1.3E+00
1.3E+00	1.3E+00	1.3E+00	1.3E+00	1.3E+00	1.3E+00	1.2E+00	1.2E+00	1.3E+00	0.0E+00

Based on matrix distance in Table 1, a clustering algorithm (k-means or DBSCAN) is executed. In the case of DBSCAN, the DBSCAN groups 10 objectives into clusters: each objectives is assigned cluster as $\{1, 1, 1, 1, 1, 1, 2, 3, 4, 5\}$, so there are 5 clusters in total. We retain each cluster one object (one objective in MOP/MaOP). Then five objectives $1, 7, 8, 9, 10$ are retained and assigned for F_s set, while others are removed. As a result, F_t is not equal to F_s, F_t is assigned to F_s and we continue the loop.

At the next loop, PCSEA generates a solutions set with F_t objectives set. This solutions set is drawn as in Figure 1b. The distance matrix is calculated as showed in Table 2. Clustering divides the F_t objectives set as $1, 2, 3, 4, 5$. Each cluster contains only one object (objective) and needs to keep one object. So we retain all objective. Other words, F_s is the same as F_t or the algorithm exists. Finally, the algorithm retains right five objectives $\{1, 7, 8, 9, 10\}$.

Table 2: The matrix distance between 5 objectives (1,7,8,9,10)

0.0E+00	1.1E+00	1.1E+00	1.2E+00	1.3E+00
1.1E+00	0.0E+00	1.1E+00	1.2E+00	1.2E+00
1.1E+00	1.1E+00	0.0E+00	1.2E+00	1.3E+00
1.2E+00	1.2E+00	1.2E+00	0.0E+00	1.4E+00
1.3E+00	1.2E+00	1.3E+00	1.4E+00	0.0E+00

4 EXPERIMENTAL DESIGN

We do experiment with DTLZ5(*I*,*M*) problem [8], PCSEA for generating nondominated solutions. Then we compare the proposed algorithm (PCS-Cluster) with the best corresponding of the PCSEA-based objective reduction.

4.1 Test Problem

The DTLZ5(*I*,*M*) problem is defined as follows:

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

$$
\begin{aligned}
&where\\
&r(x_M) = (1 + 100g(\mathbf{x}_M))\\
&\theta_i = \frac{\pi x_i}{2} && i = \overline{1, (I-1)}\\
&\theta_i = \frac{\pi(1 + 2g(\mathbf{x}_M)x_i)}{4(1 + g(\mathbf{x}_M))} && i = \overline{I, (M-1)}\\
&g = \textstyle\sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2\\
&0 \le x_i \le 1 && i = \overline{1, n}
\end{aligned}
$$

The total number of variables n in decision space is equal to $M + k - 1$, where $k = |\mathbf{x}_M| = 10$. There remain three properties of problem. The first property of the problem is that the dimensionality (I) of the PF can be altered by setting I to an integer between 2 and M. The second property is that PF is non-convex and follows the formula: $\sum_{i=1}^{M} (f_i^*)^2 = 1$. The third property is that there are $M - I$ first objectives correlated, while the others and one of $M - I$ first objectives are conflict each other.

4.2 Parameter Setting

The experiments are performed totally on 28 instances of DTLZ5(*I*, *M*) problem. While values of I is set 5, 10 and 15, values of M is set from 10 to 100 at each step 10. The parameters for PCSEA are set as Table 3.

In PCSEA-based objective reduction algorithms, threshold C is set equal from 0.55 to 0.95 at each step 0.05, and 0.975, 0.99. Parameters for PCSEA-Cluster objective reduction (namely for k-means and DBSCAN clustering algorithm) are set as Table 4. In which, distance type for both clustering algorithm is Pearson correlation, percentage is percentage of variance explained, minPts is minimum number of points required to form a cluster. Total 30 independent runs are executed for each instance.

Table 3: The parameters for PCSEA

Parameter	Value	Parameter	Value
population size	200	SBX crossover index	10
number of generations	500	mutation probability	0.1
crossover probability	0.9	polynomial mutation	20

5 RESULTS AND DISCUSSION

This section firstly examines and chooses values for one threshold for k-means and one threshold for DBSCAN. Next, performance of the proposed method is compared with the original method namely PCSEA-based objective reduction.

Table 4: The parameters for k-means, and DBSCAN

(a) For k-means

Parameter	*Value*
Distance type	correlation
Percentage*	99.0

* *Percentage of variance explained for ELBOW method in determining k clusters*

(b) For DBSCAN

Parameter	*Value*
Distance type	correlation
Distance threshold	0.25
minPts	1

5.1 Examination of thresholds for k-means, and DBSCAN in solving DTLZ5(10,*) problem

This section examines the effect of choosing the percentage threshold and distance threshold for clustering algorithm (namely k-means and DBSCAN respectively) on result of solving DTLZ5(10,*) problem. For distance threshold, the values are set from 0.2 to 0.9 in step 0.05. For percentage threshold, the values are set 95, 96, 97, 98, 99. For DTLZ5(*I,M*) problem, *I* is set fixed with value of 10, the values of *M* is set from 20 to 100 with step 10. We run 30 independently each instance, so we have 270 cases in total.

For percentage threshold, we found that results gain the best result at value of 99. For distance threshold, Fig 2 shows plot for number of successes in finding out the relevant objectives and removing redundant ones for solving DTLZ5(*10,**) problems on a number of different values of distance threshold for DBSCAN. The plot says that the worst result at the threshold of 0.9 and gradually better when threshold decreases. The result gains the best when threshold is small at values of 0.2, 0.25, and 0.3. Bases on this examination, we choose distance threshold as in Table 4b for sub-section 5.2.

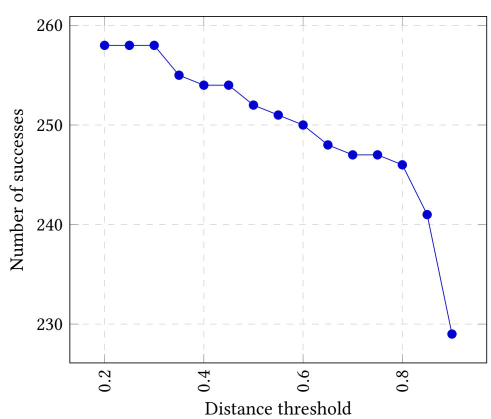


Figure 2: The number of successes in determining the relevant objectives in solving DTLZ5(10,*) problem by using DBSCAN

Table 5: The number of successes in finding correct relevant objective set in total 30 runs

Problems	I	M	Number of successes		
			PCSEA-based	k-means based	DBSCAN based
DTLZ5	5	10	30	30	30
DTLZ5	5	20	30	30	30
DTLZ5	5	30	30	30	30
DTLZ5	5	40	30	30	30
DTLZ5	5	50	30	30	30
DTLZ5	5	60	30	30	30
DTLZ5	5	70	27	30	30
DTLZ5	5	80	28	30	30
DTLZ5	5	90	23	30	30
DTLZ5	5	100	22	30	28
DTLZ5	10	20	21	29	29
DTLZ5	10	30	25	29	28
DTLZ5	10	40	26	30	29
DTLZ5	10	50	27	29	29
DTLZ5	10	60	23	29	29
DTLZ5	10	70	22	30	29
DTLZ5	10	80	21	27	24
DTLZ5	10	90	23	30	29
DTLZ5	10	100	25	29	28
DTLZ5	15	20	6	22	21
DTLZ5	15	30	3	23	22
DTLZ5	15	40	2	20	19
DTLZ5	15	50	1	19	18
DTLZ5	15	60	4	22	21
DTLZ5	15	70	4	19	17
DTLZ5	15	80	5	22	19
DTLZ5	15	90	3	19	16
DTLZ5	15	100	4	23	21
Total			*525*	*751*	*726*

5.2 Performance of PCS-Cluster objective reduction

This sub-section compares the performance of the proposed method (PCS-Cluster) with existing method, namely PCSEA-based objective reduction. Table 5 shows numbers of successes in finding correct relevant objective set for PCSEA and PCS-Cluster objective reduction algorithms. The values in the PCSEA-based objective reduction is chosen the best in all cases of threshold *C*.

Table 5 shows the number of successes in finding correct relevant objective set in total 30 runs independently for original and two algorithms of proposed method. It can be seen that, all the algorithms can remove redundant objective set exactly when value of *I* is 5 and values of *M* in not greater than 60. The table also gives decreasing trend in succeed when value of *I* increases, while the proposed method decreases moderately, the original one decreases significantly. It is clear that the number of successes in finding correct relevant objective set in the proposed method (751 and 726 for k-means based and DBSCAN based objective reductions) is greater than its original one (525 for PCSEA-based objective reduction).

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 level at 0.05, and the alternative hypothesis is that the performance of the two methods is significant different. From the table, PCSEA and PCS-Cluster k-means we get p-value of 3.94E-05. It says that the null hypothesis is rejected or we accept the alternative hypothesis which means that the two algorithms are different with significant level at 0.05. PCSEA and PCS-Cluster DBSCAN we get p-value of 3.83E-05. It says that the null hypothesis is rejected or we accept the alternative hypothesis which means that the two algorithms are different with significant level at 0.05. So we have enough basis to conclude that the both variants of PCS-Cluster are different to or better than PCSEA-based objective reduction algorithm at level 0.05.

6 CONCLUSION AND FUTURE WORK

This paper has proposed a method of objective reduction PCS-Cluster by combination between PCSEA evolutionary algorithm and clustering algorithms for identifying the relevant and redundant objectives. It takes advantages of PCSEA evolutionary algorithm and the simple of clustering algorithms namely k-means (a partitioning clustering algorithm) and DBSCAN (a densisty based clustering algorithm). We designed experimental with wider range instances of DTLZ5(I, M) problem and run 30 times independently. The results proved that the proposed method is better than the original one.

As the result of PCSEA-based objective reduction algorithm is not entirely independent of the order in which the objective is removed, therefore future works could investigate the order of objective is examined for retaining or discarding.

ACKNOWLEDGMENTS

This research is funded by project: A smart surveillance system based on facial recognition for border and immigration controls (code HNQT/SPĐP/14.19)

REFERENCES

[1] Johannes Bader and Eckart Zitzler. 2011. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation* 19, 1 (2011), 45–76.

[2] Dimo Brockhoff and Eckart Zitzler. 2006. Dimensionality reduction in multiobjective optimization with (partial) dominance structure preservation: Generalized minimum objective subset problems. *TIK Report* 247 (2006).

[3] Jixiang Cheng, Gary G Yen, and Gexiang Zhang. 2015. A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 592–605.

[4] Yiu-ming Cheung and Fangqing Gu. 2014. Online objective reduction for many-objective optimization problems. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 1165–1171.

[5] Yiu-ming Cheung, Fangqing Gu, and Hai-Lin Liu. 2016. Objective extraction for many-objective optimization problems: Algorithm and test problems. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 755–772.

[6] Kalyanmoy Deb and Himanshu Jain. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 577–601.

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[8] Kalyanmoy Deb and Dhish Kumar Saxena. 2005. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Kangal report* 2005011 (2005).

[9] Andries P Engelbrecht. 2007. *Computational intelligence: an introduction*. John Wiley & Sons.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[11] David B Fogel. 1997. The Advantages of Evolutionary Computation.. In *BCEC*. 1–11.

[12] Fangqing Gu, Hai-Lin Liu, and Yiu-ming Cheung. 2017. A Fast Objective Reduction Algorithm Based on Dominance Structure for Many Objective Optimization. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 260–271.

[13] Xiaofang Guo, Yuping Wang, and Xiaoli Wang. 2013. Using objective clustering for solving many-objective optimization problems. *Mathematical Problems in Engineering* 2013 (2013).

[14] Zhenan He and Gary G Yen. 2016. Many-objective evolutionary algorithms based on coordinated selection strategy. *IEEE Transactions on Evolutionary Computation* 21, 2 (2016), 220–233.

[15] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. 2008. Evolutionary many-objective optimization: A short review. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC2008, June 1-6, 2008, Hong Kong*. 2419–2426.

[16] Antonio López Jaimes, Carlos A Coello Coello, and Jesús E Urías Barrientos. 2009. Online objective reduction to deal with many-objective problems. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 423–437.

[17] Deb Kalyanmoy et al. 2001. *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons.

[18] David J Ketchen and Christopher L Shook. 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal* 17, 6 (1996), 441–458.

[19] Joshua D Knowles and David W Corne. 2000. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation* 8, 2 (2000), 149–172.

[20] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 13.

[21] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[22] Kaisa Miettinen. 1999. Nonlinear Multiobjective Optimization, volume 12 of International Series in Operations Research and Management Science.

[23] Hung Xuan Nguyen, Lam Thu Bui, and Cao Truong Tran. 2019. 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, in press.

[24] Kaushik H Raviya and Kunjan Dhinoja. 2013. An Empirical Comparison of K-Means and DBSCAN Clustering Algorithm. *PARIPEX Indian Journal of Research* 2, 4 (2013), 153–155.

[25] Dhish Kumar Saxena, João A Duro, Ashutosh Tiwari, Kalyanmoy Deb, and Qingfu Zhang. 2013. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation* 17, 1 (2013), 77–99.

[26] Hemant Kumar Singh, Amitay Isaacs, and Tapabrata Ray. 2011. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation* 15, 4 (2011), 539–556.

[27] Handing Wang, Licheng Jiao, and Xin Yao. 2015. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 524–541.

[28] Handing Wang and Xin Yao. 2016. Objective reduction based on nonlinear correlation information entropy. *Soft Computing* 20, 6 (2016), 2393–2407.

[29] Yuan Yuan, Yew-Soon Ong, Abhishek Gupta, and Hua Xu. 2018. Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive/critical analysis. *IEEE Transactions on Evolutionary Computation* 22, 2 (2018), 189–210.

[30] Yuan Yuan, Hua Xu, Bo Wang, and Xin Yao. 2016. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 16–37.

[31] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[32] Xingyi Zhang, Ye Tian, and Yaochu Jin. 2015. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 19, 6 (2015), 761–776.

[33] Eckart Zitzler and Simon Künzli. 2004. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*. Springer, 832–842.

[34] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103 (2001).