A Pareto Corner Search Evolutionary Algorithm and Principal Component Analysis for Objective Dimensionality Reduction

Xuan Hung Nguyen
Faculty of Information Technology
Le Quy Don Technical University;
and Military Science Academy
Hanoi, Vietnam
Email: nguyenxuanhung@outlook.com

Lam Thu Bui
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: lambt@lqdtu.edu.vn

Cao Truong Tran
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: truongtc@lqdtu.edu.vn

***Abstract*—Many-objective optimization problems (MaOPs) cause serious difficulties for existing multi-objective evolutionary algorithms (MOEAs). One common way to alleviate these difficulties is to use objective dimensionality reduction. Most existing objective reduction methods are time-consuming because they require to run MOEAs to numerous generation. Pareto corner search evolutionary algorithm (PCSEA) was proposed in [21] to speed up objective reduction methods by only seeking corner solutions instead of whole solutions. However, the PCSEA-based objective reduction method in [21] needs to predefine a threshold to select objectives which strongly depends on problems and is not straightforward to obtain. This paper proposes a new objective dimensionality reduction method by integrating PCSEA and principal component analysis (PCA). Thanks to combining advantages of PCSEA and PCA, the proposed method not only can be efficient to eliminate redundant objectives, but also not require to define any parameter in advanced. The experimental results also show that the proposed method can perform objective reduction more successfully than the PCSEA-based objective reduction method.**

***Index Terms*—many-objective optimization, objective dimensionality reduction, feature selection, evolutionary computation**

I. Introduction

In the real world, there often exist problems with more than one objective conflicting to each other which is often referred as multi-objective problems (MOPs) [18]. In an MOP, different solutions are likely to have an advantage over other objectives, so the Pareto dominance concept is commonly-used to compare different solutions. Most common multi-objective optimization algorithms try to approximate the objective Pareto Front space so that no further enhancement on any objective can achieve without lossing the quality of other objectives [15].

One of the most common approach to dealing with these problems is to use evolutionary computation (EC) techniques such as genetic algorithms and particle swarm optimization [9]. EC has many advantages including the simplicity of the approach, broad applicability, outperforming traditional methods on real problems, and the capability for self-optimization [10]. Multi-objective evolutionary algorithms (MOEAs) refer to algorithms using EC technique to evolve solutions for MOPs. MOEAs can evolve multiple solutions in a single run, and they can achieve better solutions than traditional methods. For example, NSGA-II [6] is one of the most well-know MOEAs.

MOPs with more than three objectives are usually regarded as MaOPs. When dealing with these MaOPs, MOEAs encounter a number of obstacles. Firstly, with Pareto-based MOEAs, a large portion of population becomes the non-dominated, so it is difficult to select candidates for next generation. Moreover, the size of population increases exponentially when approximating the entire Pareto Front. Finally, it is difficult to envision solutions for decision makers to select a final solution [14].

Approach to solving MaOPs can be categorised 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 NSGA-III [5], KnEA [26], try to directly eliminate the difficulties encountered. In contrast, the second group supposes that there remain redundant objectives in the given problem, and the methods in second group, such as PCA-NSGA-II [7], L-PCA [20], try to remove the redundant objective before using MOEAs to search for Pareto Front. By removing redundant objectives, the objective reduction approach has three main advantages. Firstly, it can reduce the computational load of an MaOEA, i.e. it makes less time to operate and less space to store. Furthermore, the problem with less objectives can be solved by other MOEAs. Finally, it can help decision makers understand the MaOP better by pointing out the redundant objectives [17], [20].

Objective reduction methods can be divided into structure-based methods and correlation-based methods. The structure-based methods try to retain the dominance relations as much as possible when removing redundant objectives. The Exact and Greedy algorithms in [2] and the PCSEA-based objective dimensionality reduction in [21] are examples of the first approach. The correlation-based methods use metrics such as correlation or mutual information to evaluate the relation between objectives of non-dominated solutions, then objectives that are low conflict, or non-conflict to other are removed while others are retained [12], [19].

Both the two objective reduction approaches need an approximate nondominated solution set which is generated by MOEAs/MaOEAs. However, these evolutionary algorithms often target the set covering the whole true Pareto front. Because of requiring MOEAs/MaOEAs for evolving whole Pareto Front, most existing objective dimensionality 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 [21] only seeks for a few key solutions which locate in the corners of Pareto front. As a result, the complexity of the PCSEA-based objective reduction method is much faster than others methods. However, the PCSEA-based objective dimensionality reduction method requires to define a threshold to remove redundant objectives and keep relevant objectives. Moreover, the experimental results in this paper show that the threshold strongly depends on each problem, but the threshold was fixed in [21]. Therefore, this paper proposes a new method to alleviate the limitation of the PCSEA-based objective dimensionality reduction method by using principal component analysis for performing objective reduction. Results show that the proposed method can perform objective reduction effectively and efficiently.

The rest of this paper is organized as follows. Section II shows an overview of related work. Section III presents the proposed algorithm. Section IV shows experimental design. Section V presents the result and discussion. Section VI makes conclusion and state future work.

II. Related Work

A. Multi and Many-objective Optimization

MOP as defined as follows [18]:

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

where there are k (≥ 2) objective function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. The decision vectors $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ belongs to feasible region Ω, which is a subset of decision variable space $\mathbb{R}^n$. All of k objective functions need to be minimized simultaneously. 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 **f(x)** or $\mathbf{z} = (z_l, z_2, ..., z_k)^T$, where $z_i = f_i(\mathbf{x})$ for all $i = 1, ..., k$ 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 converting the problem into a single objective optimization problem. After conversion, the new problem has a single objective function, then it can be solved by using developed theory and methods for single objective optimization [18].

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) [6], Pareto archived evolution strategy (PAES) [16], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [24].

When MOPs with more than three objectives which are considered as MaOPs. When tackling these MaOPs, MOEAs encounter a number of difficulties. First, when applying Pareto-dominance based MOEAs such as NSGAII [6] to solve MaOPs, a large portion of population becomes non-dominated, so we cannot determine which solutions are better for next generation. When using aggregation-based or indicator-based approaches such as IBEA [27], 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 [14]. Third, visualization the solution set is difficult to help decision makers to choose the final solution [14].

MaOEAs, which are proposed to solve MaOPs, can be categorized into 2 classes. The first class supposes that there is no redundant objectives in a given problem and try to directly alleviate or eliminate the difficulties encountered. It includes sub-classes: preference ordering relation-based, preference incorporation-based, indicator-based, decomposition-based. MaOEAs such as reference-point based non-dominated sorting (NSGA-III) [5], grid-based evolutionary algorithm (GrEA) [22], and knee point driven evolutionary algorithm (KnEA) [26] belong to the first class. In contrast to the first class, the second one 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. This class is presented more detail in sub-section II-B below.

B. Objective Dimensionality Reduction

Dimensionality reduction is often used to avoid the curse of dimensionality from experimental life sciences. In general, dimensionality reduction methods is used to reduce large feature spaces to smaller feature spaces. 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. 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 [3], [4] 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 Pareto Front as the set of original objectives does. This approach can be classified into 2 sub-classes: Pareto 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 [1], [2]. The latter 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 low conflict, or non-conflict each other. This sub-class measures the conflict between objectives by using the correlation [20] or mutual information [11] of objective value of non-dominated solutions.

C. PCA for Objective Dimensionality Reduction

Principal component analysis (PCA) is one of the well-known dimensionality reduction technique. Based on this technique, the authors in [20] proposed an algorithm for linear objective reduction (L-PCA). The idea is to find the smallest set of conflicting objectives preserving the correlation structure in the given nondominated population, which is achieved by removing objectives that are non-conflict or low conflict along the significant eigenvectors of the correlation matrix. Algorithm 1 shows framework of L-PCA algorithm.

Algorithm 1: L-PCA objective dimensionality reduction

Input: $t \leftarrow 0$ and $F_t \leftarrow \{f_1, f_2, ..., f_M\}$
begin
  Obtain a set of non-dominated solutions corresponding to F_t by run an MOEA/MaOEA
  Compute a positive semi-definite matrix R($M \times M$) From R, compute the eigenvalues, eigenvectors
  Perform the Eigenvalue Analysis to identify the set of important objectives $F_e \subseteq F_t$
  Perform the Reduced Correlation Matrix Analysis to identify the identically correlated subset (S) in F_e. If there no such subset, $F_s \leftarrow F_e$
  Identify the most significant objective in each S, to arrive at F_s, such that $F_s \subseteq F_e \subseteq F_t$
  Computation of error
  if $F_s = F_t$ **then**
    Stop and declare F_t as the essential objectives;
    $T \leftarrow t$ and compute the total error
  else
    $t \leftarrow t + 1$, $F_t \leftarrow F_s$, and go to Step 2
  end
end

D. Pareto Corner Search Evolutionary Algorithm

Objective dimensionality reduction schemes essentially have two major components. Those are generation of an approximation of the Pareto front using a MOEA (NSGAII in [7], [20], $\epsilon-$MOEA in [20], MOEA/D in [13], SPEA2-SDE in [23]), and dimensionality analysis on the obtained approximation of the Pareto front. Those components may be applied once, or repetitively to come up with the reduced set of objectives. However, for generating approximate Pareto Front, an MOEA has to be run for a large number of generation. Even so, population may still far away from Pareto front, which might render extraction of any information regarding dimensionality reduction meaningless. Moreover, MOEAs often search for approximation to the entire Pareto front. This becomes impossible even with a large number of solutions set for high number of objectives. To overcome this difficulty, Pareto Corner Search Evolutionary Algorithm (PCSEA) [21] is shown in Algorithm 2, the search is focused on a few *key* solutions on the boundaries of Pareto front. These solutions are the "corner" solutions of the Pareto front where the boundaries intersect. The corners may be one of cases which are in circles in Fig. 1.

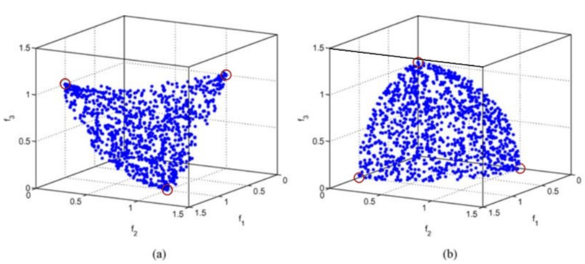

Fig. 1: Different types of Pareto corner solutions (extracted from [21])

Algorithm 2: Framework for PCSEA

Input: Population size N, number of generations N_G
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 [6]. While NSGA-II uses non-dominated sorting and crowding distance-based ranking, PCSEA uses corner-sort ranking procedure. In corner-sort, based on individual objective values and L_2 norm all-but-one objectives, the solutions are ranked and selected.

Once the corner solutions of the Pareto front are obtained using PCSEA (in Algorithm 2), the objective dimensionality

Algorithm 3: PCSEA objective reduction algorithm

Input: original objectives $F_R \leftarrow \{f_1, f_2, \ldots, f_M\}$, threshold C

Output: reduced objective set F_R

1 corner solutions $\leftarrow$ PCSEA (*choose the unique non-dominated solutions*)
2 **foreach** $m \in \{1, \ldots, M\}$ **do**
3     $R \leftarrow \dfrac{N_{F_R \setminus \{f_m\}}}{N_{F_R}}$
4   **if** $R > C$ **then**
5         $F_R \leftarrow F_R \setminus \{f_m\}$
6   **end**
7 **end**

Algorithm 4: PCS-LPCA objective dimensionality reduction algorithm

Input: $t \leftarrow 0$;
original objective set $F_t \leftarrow \{f_1, f_2, \ldots, f_M\}$

Output: reduced objective set F_s

1 **repeat**
2     $P \leftarrow$ PCSEA(F_t) // *get corner solutions*
3     $P_u \leftarrow$ Unique-Nondominated(P) // *Retain the unique nondominated solutions*
4     $F_s \leftarrow$ L-PCA(P_u)
5   **if** $F_t = F_s$ **then**
6         $stop \leftarrow true$
7   **else**
8         $t \leftarrow t + 1$;
9         $F_t \leftarrow F_s$;
10         $stop \leftarrow false$;
11   **end**
12 **until** *stop*;

reduction is performed. The idea is Pareto dominance among the solutions will be largely depend on the relevant objectives. That means if one redundant or irrelevant objective is removed then there is no (or negligible) change in the number of non-dominated solutions. On the contrary, if one of critical objectives is discarded, that number is changed significantly. The PCSEA objective dimensionality reduction algorithm is shown in Algorithm 3.

The parameter R in line 3 in Algorithm 3 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.

III. The Proposed Method

The Pareto corner sort evolutionary algorithm (PCSEA)-based objective reduction in [21] can efficiently remove redundant objectives. However, this algorithm has a number of limitations. The first limitation is that a threshold C (cutoff value of R) must be provided before performing objective reduction. Secondly, the objective dimensionality reduction algorithm did not consider the importance of the order of removing redundant objectives. Finally, the algorithm was tested on DTLZ5 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 Pareto corner sort evolutionary algorithm (PCSEA) to generate non-dominated solutions which are then used by principal component analysis (PCA) to eliminate redundant objectives. Algorithm 4 show the main steps of the proposed method. We call the algorithm PCS-LPCA.

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 Pareto front with lower complexity than other MOEAs/MaOEAs. The second idea, unlike Algorithm 3, the proposed algorithm avoids using the sensitive parameter *threshold C*.

The process of the proposed algorithm includes performing PCSEA (line 2), which finds only solutions lie on "corner" or intersection of boundaries of Pareto front corresponding remain objective set. Then, the unique non-dominated solutions are retained, while others are discarded (line 3). From solution set retained and current objective set F_t, L-PCA objective dimensionality reduction is executed. The reduction (line 4) works as the same pseudo-code as between line 3 and line 5 in Algorithm 1 do. F_s is objective set after reduction. If objective set after reduction F_s is same as before reduction F_t then algorithm exits, otherwise the algorithm loops with new current objective set ($F_t \leftarrow F_s$ as in line 9).

IV. Experimental Design

We do experiment with DTLZ5 problem, PCSEA for generating non-dominated solutions. Then we compare the proposed algorithm (PCS-LPCA) with the best corresponding of PCSEA objective dimensionality reduction.

A. Test Problem

To study, we use DTLZ5(*I,M*) problem [8], it is defined as:

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

$$
\begin{aligned}
where\quad &\theta_i = \frac{\pi}{2}x_i && for\ \ i = 1, 2, ..., (I-1)\\
&\theta_i = \frac{\pi}{4(1 + g(x_M))}(1 + 2g(x_M)x_i) && for\ \ i = I, \ldots, (M-1)\\
&g = \textstyle\sum_{x_i \in x_M}(x_i - 0.5)^2\\
&0 \le x_i \le 1 && for\ i = 1, 2, ..., n
\end{aligned}
$$

The total number of variables is $n = M + k - 1$, where $k = |x_M| = 10$. The first property of the problem is that the dimensionality (I) of the Pareto-optimal front can be changed by setting I to an integer 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.

B. Parameter Setting

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

In objective dimensionality reduction algorithms, threshold C is set equal from 0.55 to 0.95 at each step 0.05, and 0.975, 0.99 for PCSEA objective dimensionality reduction algorithm, threshold θ is set equal to 0.997 for L-PCA objective dimensionality reduction algorithm. We execute 30 independent runs for each instance.

TABLE I: 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

V. Results and Discussion

A. The dependence of PCSEA reduction algorithm's result on threshold

In order to evaluate the affection of threshold on result we do this experiment. In PCSEA reduction dimensionality algorithm in Algorithm 3, there is a threshold C parameter. The algorithm examines each objective by calculating the value of rate R (the ratio between number of non-dominated solutions after and before dropping that objective). If R is greater than threshold C then that objective is remove from objective set. We test the threshold with values: 0.55, 0.60, 0.65, 0.75, 0.80, 0.85, 0.90, 0.95, 0.975 and 0.99 for 3 instances of DTLZ5(*I, M*) problem. The problem with values of I are 5, 10 and 15; values of M are 20, 40, 60. We execute 30 runs independent for each cases. Fig. 2 shows the percentage of success in finding relevant objective set.

As we can see from Fig. 2, if I is equal to 5, the range of threshold C can be from 0.8 to 0.9 the success rate of finding correct redundant is up to 100 percent. When I values are set to 10, the percentage of success in finding relevant objectives is best at threshold C equal to 0.95 at the highest slight smaller than 90 percent. In the list of threshold C, the highest success is only 20 percent. That can infer that when the number of relevant objectives increases, the threshold C need to be set higher and seems to sensitive. Moreover, the percentage of success in finding correct relevant objective set becomes lower. So, we can conclude that threshold 0.8 as in [21] is not an optimal threshold for any problems.

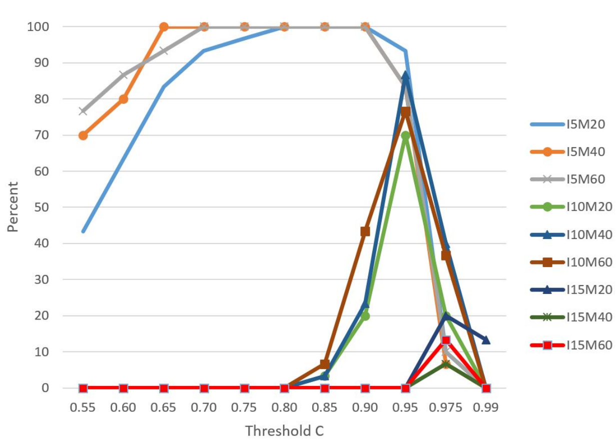

Fig. 2: The percentage of success in finding relevant objective set

B. Performance of PCS-LPCA objective dimensionality reduction

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

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, we get p-value is 3.8528E-05. It says that the null hypothesis is rejected or we accept the alternative hypothesis which means that the two algorithms

TABLE II: Comparison the number of successes in finding correct relevant objective set in total 30 runs

Problem			**Number of successes**	
	I	***M***	***PCSEA Reduction***	***PCS-LPCA***
DTLZ5	5	10	30	30
DTLZ5	5	20	30	30
DTLZ5	5	30	30	30
DTLZ5	5	40	30	30
DTLZ5	5	50	30	30
DTLZ5	5	60	30	30
DTLZ5	5	70	27	28
DTLZ5	5	80	28	29
DTLZ5	5	90	23	27
DTLZ5	5	100	22	28
DTLZ5	10	20	21	28
DTLZ5	10	30	25	27
DTLZ5	10	40	26	28
DTLZ5	10	50	27	28
DTLZ5	10	60	23	27
DTLZ5	10	70	22	26
DTLZ5	10	80	21	25
DTLZ5	10	90	23	27
DTLZ5	10	100	25	27
DTLZ5	15	20	6	20
DTLZ5	15	30	3	19
DTLZ5	15	40	2	17
DTLZ5	15	50	1	15
DTLZ5	15	60	4	19
DTLZ5	15	70	4	15
DTLZ5	15	80	5	17
DTLZ5	15	90	3	14
DTLZ5	15	100	4	19
Total			*525*	*690*

are different with significant level at 0.05. So we have enough basis to conclude that the PCS-LPCA is different to or better than PCSEA objective dimensionality reduction algorithm at level 0.05.

VI. Conclusion and Future Work

This paper has proposed a method of objective dimensionality reduction PCS-LPCA by combination between Pareto corner search evolutionary algorithm and linear principal component analysis objective reduction algorithm for identifying the relevant and redundant objectives. It is proved that the proposed method is better than the original one. The paper also examined the affection of threshold value for PCSEA objective dimensionality reduction. It indicated that PCSEA objective dimensionality reduction is very efficient for solving redundant problem with a small number of relevant objective. It also said that PCSEA objective dimensionality reduction become inefficient for a larger that number.

As the result of PCSEA objective dimensionality 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.

References

[1] D. Brockhoff and E. Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature-PPSN IX*, pages 533–542. Springer, 2006.

[2] D. Brockhoff and E. Zitzler. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary computation*, 17(2):135–166, 2009.

[3] Y.-m. Cheung and F. Gu. Online objective reduction for many-objective optimization problems. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1165–1171. IEEE, 2014.

[4] 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*, 20(5):755–772, 2016.

[5] 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*, 18(4):577–601, 2014.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[7] K. Deb and D. Saxena. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360, 2006.

[8] K. Deb and D. K. Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Kangal report*, 2005011, 2005.

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

[10] D. B. Fogel. The advantages of evolutionary computation. In *BCEC*, pages 1–11, 1997.

[11] 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*, pages 11–16. IEEE, 2012.

[12] X. Guo, Y. Wang, and X. Wang. Using objective clustering for solving many-objective optimization problems. *Mathematical Problems in Engineering*, 2013, 2013.

[13] X. Guo, Y. Wang, and X. Wang. An objective reduction algorithm using representative pareto solution search for many-objective optimization problems. *Soft Computing*, 20(12):4881–4895, 2016.

[14] 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*, pages 2419–2426, 2008.

[15] D. Kalyanmoy et al. *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons, 2001.

[16] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[17] B. Li, J. Li, K. Tang, and X. Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):13, 2015.

[18] K. Miettinen. Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science, 1999.

[19] 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*, pages 772–787. Springer, 2007.

[20] 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*, 17(1):77–99, 2013.

[21] 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*, 15:539–556, 2011.

[22] S. Yang, M. Li, X. Liu, and J. Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

[23] 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*, 22(2):189–210, 2018.

[24] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

[25] X. Zhang, Y. Tian, R. Cheng, and Y. Jin. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 19(2):201–213, 2014.

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

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