Information Sciences 423 (2018) 200–218

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# DECOR: Differential Evolution using Clustering based Objective Reduction for many-objective optimization

Monalisa Pal [a,*], Sriparna Saha [b], Sanghamitra Bandyopadhyay [a]

[a] Machine Intelligence Unit, Indian Statistical Institute, 203, Barrackpore Trunk Road, Kolkata - 700108, India
[b] Department of Computer Science and Technology, Indian Institute of Technology Patna, India

ARTICLE INFO

*Article history:*
Received 9 June 2017
Revised 14 August 2017
Accepted 18 September 2017
Available online 19 September 2017

*Keywords:*
Many-objective optimization
Differential evolution
Pareto-optimality
Objective reduction
DTLZ test problems

ABSTRACT

Challenges like scalability and visualization which make multi-objective optimization algorithms unsuitable for solving many-objective optimization problems, are often handled using objective reduction approaches. This work proposes a novel many-objective optimization algorithm, viz. Differential Evolution using Clustering based Objective Reduction (DECOR). Correlation distance based clustering of objectives from the approximated Pareto-front, followed by elimination of all but the centroid constituent of the most compact cluster (with special care to singleton cluster), yields the reduced objective set. During optimization, the objective set periodically toggles between full and reduced size to ensure both global and local exploration. For finer clustering, number of clusters is eventually increased until it is equal to the remaining number of objectives. DECOR is integrated with an Improved Differential Evolution for Multi-objective Optimization (IDEMO) algorithm which uses a novel elitist selection and ranking strategy to solve many-objective optimization problems. DECOR is applied on some DTLZ problems for 10 and 20 objectives which demonstrates its superior performance in terms of convergence and equivalence in terms of diversity as compared to other state-of-the-art optimization algorithms. The results have also been statistically validated. Source code of DECOR is available at http://decor.droppages.com/index.html.



## 1. Introduction

Optimization algorithms help to find the best alternative from a set of available choices for any problem at hand. Meta-heuristic optimization algorithms have gained popularity by providing a decent approximation of the optimal solution [15,16,30], even for hard problems. To solve the problems with multiple conflicting objectives, Multi-Objective Optimization (MOO) algorithms are used [14]. In this class of optimization algorithms, when number of objectives is four or more, several challenges come into play. Hence, this sub-class of problems forms an essential research topic and is called Many-Objective Optimization (MaOO) problems [20]. Some practical applications of MaOO algorithms from varied domains are nurse scheduling problem [27], problem of designing factory-shed truss [2], space trajectory design problem [22], feature selection problem for motor imagery brain signal classification [28], software refactoring problem [26] and cyclone geometry design problem [12].

* Corresponding author.
*E-mail addresses:* monalisapal_r@isical.ac.in (M. Pal), sriparna@iitp.ac.in (S. Saha), sanghami@isical.ac.in (S. Bandyopadhyay).

https://doi.org/10.1016/j.ins.2017.09.051
0020-0255/

Literature has in abundance several MOO algorithms. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [8], Decomposition based Multi-Objective Evolutionary Algorithms (MOEA/D) [32], Hypervolume Estimation (HypE) algorithm [1], Differential Evolution for Multi-objective Optimization (DEMO) [33], Pareto Envelope-based Selection Algorithm (PESA) [7], and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [35] are some of the popular ones. The most prominent issues of applying MOO algorithms to MaOO problems are as follows:

1. With increase in number of objectives, the number of solutions needed to approximate the Pareto-optimal set grows as polynomial in length of encoded input, if not exponentially [13,31]. Handling such a large population in every iteration of evolutionary algorithms, increases the computational burden.
2. The popular Pareto-dominance based selection, in several MOO algorithms [8], fails to provide sufficient selection pressure for MaOO problems. Thus, when MOO algorithms are scaled beyond 8 to 10 objectives, the population is saturated by non-dominated solutions at an early generation [2,17]. Relaxations to Pareto-dominance like $\epsilon$-dominance [20], favour relation [11,20], modified favour relation [20,34], fuzzy Pareto-dominance [17] and many more, are available in the literature to handle this situation.
3. Beyond three objectives, decision-making is difficult as the objective space cannot be visualized [3,18].

To assist in decision-making, one of the options is to rely on performance measures to assess the quality of the result. Different performance metrics [1,2,8,20] consider different features of the MaOO algorithms like the convergence of solution, the diversity of Pareto-front, the speed of convergence, or a combination of two or more of these features. For example, a study for optimizing four scalable functions, each having 2–8 objectives, was performed to compare NSGA-II (excelled in diversity and speed), PESA (excelled in convergence) and SPEA2 (excelled in diversity) [24]. Another study [29] emphasizes the need to visualize the Pareto-front by showing that MaOO can result in conflicting decisions on the basis performance measures. Even being associated with several problems, the most popular performance metrics are convergence metric and Hypervolume Indicator [29]. Thus, the other option for decision making is to develop visualization methods like Buddle Chart [18], Parallel Coordinates [5,18], Heatmaps [5,18], Radial Visualization (or RadViz) [5,18] and Self-organizing Maps [5,18]. Parallel coordinates is found as the most popular visualizing tool due to its simplicity.

The mentioned issues for MaOO problems are often tackled by objective reduction where the most conflicting $m$ objectives out of $M$ objectives ($m \le M$) are chosen. If this achieves $m \le 3$, the MaOO problem reduces to a MOO problem. Even when $4 \le m < M$, the computational burden reduces as lesser number of points could approximate the Pareto-front and thus the algorithm could converge faster [2,21]. The size of the full and reduced objective sets are denoted as $M$ and $m$, respectively, throughout the present work. There are two groups of objective reduction algorithms in literature [21,30]: (i) $m$ is specified by the users, and (ii) $m$ is automatically determined. Some of the recent objective reduction algorithms which have gained attention are $\delta$-Minimum Objective Sub-Set ($\delta$-MOSS) and Objective Sub-Set of size $k$ with Minimum Error ($k$-EMOSS) [4], Principal Component Analysis NSGA-II (PCA-NSGA-II) [9], $k$-sized Objective Sub-Set Algorithm (kOSSA) and mixed search scheme of kOSSA [21], $\alpha$-DEMO and $\alpha$-DEMO-revised [2]. All these schemes, except [4], quantify the conflict between objectives using correlation among the objectives. Motivated by these characteristics of objective reduction techniques, the authors propose an optimization algorithm in this paper which uses objective reduction in the background of a multi-objective optimization algorithm such that many-objective optimization problems are efficiently addressed.

Outline of the rest of the paper is as follows. A brief description of the previous works that are related to the proposed algorithm, is mentioned in Section 2 while highlighting the primary contributions of the proposed work. The modifications of the base optimization algorithm (DEMO) to yield Improved DEMO (or IDEMO) are presented in Section 3 and the proposed objective reduction based optimization approach (viz. Differential Evolution using Clustering based Objective Reduction or DECOR) is described in Section 4. Performance analysis and the results of statistical significance test are presented in Section 5, and the major observations are further discussed in Section 6. Finally, the conclusion is drawn and open research scopes are discussed in Section 7.

## 2. Motivation for the work

This section briefly describes the shortcomings of the existing works which help to understand the motivation behind studying the proposed work. Subsequently, the characteristic features of the proposed work are mentioned which distinguish it from the existing works by highlighting its novelties.

### *2.1. Related works and their drawbacks*

The existing approaches suffer from some severe drawbacks. In $\delta$-MOSS and $k$-EMOSS [4], a greedy approach is followed where the minimal alteration in induced Pareto-dominance relation is searched by removing one objective in every turn. The high time complexity of this approach makes it unsuitable for practical applications [4]. To quicken the process, online objective reduction [21] is adopted by performing reduction during search. However, when online reduction is of one objective at a time like in mixed search scheme of kOSSA [21], the procedure is still slow. Hence, the provision of removal of multiple objectives at a time is adopted, like in $\alpha$-DEMO and $\alpha$-DEMO-revised [2], and the approach proposed in [30]. The presented algorithms in [2] require the users to specify $m$. However, the apriori determination of $m$ is not possible, and for desirable performance, repeated evaluations by varying $m$ is time-consuming and not user-friendly. Considering all these disadvantages, an objective reduction approach has been proposed in [30] which is fast (online and has provision for elimination of multiple objectives at a time) and automatically finds $m$.

### 2.2. Proposed work and its novel characteristics

The work presented in this paper is an extension of the work reported in [30]. It is compared with several existing MaOO algorithms by applying these to the DTLZ test suite on 10 and 20 objectives. For comparison, not only the convergence metric and the hypervolume indicator are evaluated, but also the estimated Pareto-front have been visualized using parallel coordinates. The results conclude superior performance of the proposed approach, based on convergence, whereas equivalent performance, based on diversity, as compared to other MaOO algorithms. Moreover, the results improve on the convergence and diversity achieved by the work reported in [30].

On one hand, the proposed work has the following two key features which are similar to some existing algorithms:

- It uses Differential Evolution as the underlying optimization algorithm similar to [2,30].
- It uses a similar principle of correlation based clustering for online objective reduction as done in [21,30]. Hence, the proposed algorithm has been named as Differential Evolution using Clustering based Objective Reduction (DECOR).

On the other hand, the proposed work differs from which it has been extended [30] in the following aspects:

- DECOR presents both the versions of objective reduction (automatic determination and user-specified $m$).
- DECOR avoids premature termination due to appearance of singleton cluster, by determining whether it is significantly close to the nearest cluster.
- The proposed approach uses a novel elitist selection strategy to avoid early saturation of the population by non-dominated solutions.
- The proposed approach uses a novel ranking scheme which combines the distance of a solution from ideal point with the crowding distance to account for both convergence and diversity during online objective reduction.

Thus the contributions of the current work are two-fold. Firstly, a new MaOO technique, which is named Improved Differential Evolution for Multi-objective Optimization (IDEMO), is developed which introduces an enhanced ranking strategy utilizing a new selection operator based on crowding distance and distance with respect to ideal point, and also introduces an extended elitist strategy. Secondly, a new objective reduction technique is proposed which is further integrated with IDEMO to yield MaOO algorithm viz. DECOR.

To the best of our knowledge, a MaOO algorithm combining all these features has not been developed in recent years and this highlights the novelty of the proposed work.

## 3. Theory of proposed underlying optimization algorithm - IDEMO

For easy comprehension of the work, the basics of many-objective optimization problem formulation, the notion of Pareto-optimality, the formal concept of conflicting objectives, the steps of the base optimization algorithm i.e. DEMO, the details of the benchmark problems (DTLZ test suite) and the performance metrics used in this work, are outlined in the supplementary material (http://decor.droppages.com/index.html).

This section mainly covers a brief description of different steps of the newly developed underlying optimization technique, namely Improved Differential Evolution for Multi-objective Optimization (IDEMO). IDEMO differs from ordinary differential evolution based multi-objective optimization approach in selection operation. Here a new ranking strategy and a new elitist operation are proposed to make the selection process more pertinent to solve the complex many objective optimization problem.

### 3.1. Improved elitist strategy at the selection stage

An $M$-objective optimization problem maps an $N$-dimensional vector $X = [x_1, x_2, \cdots, x_N]^T$ in the decision space i.e., $X \in S$ to an $M$-dimensional vector $F(X) = [f_1(X), f_2(X), \cdots, f_M(X)]^T$ in the objective space such that searching the objective space yields a set of decision vectors (called as the Pareto-optimal Set) which represent the optimal trade-off in terms of all the $M$-objectives.

Elitist strategy is a kind of selection operations where the good solutions of parent population (represented by a matrix $X$ having $NP$ rows denoting $NP$ decision vectors) of the present generation ($G$) are passed onto the population for next generation ($G+1$) in order to preserve the good solutions.

#### 3.1.1. Conventional elitist strategy

In the elitist framework, at the selection stage, the trial solutions ($U_{i,G+1}$) which dominate the parent candidates ($X_{i,\,G}$) form the new candidates ($Q_{p,G+1}$) which are added to population pool for next generation ($Y_{G+1}$) along with the entire

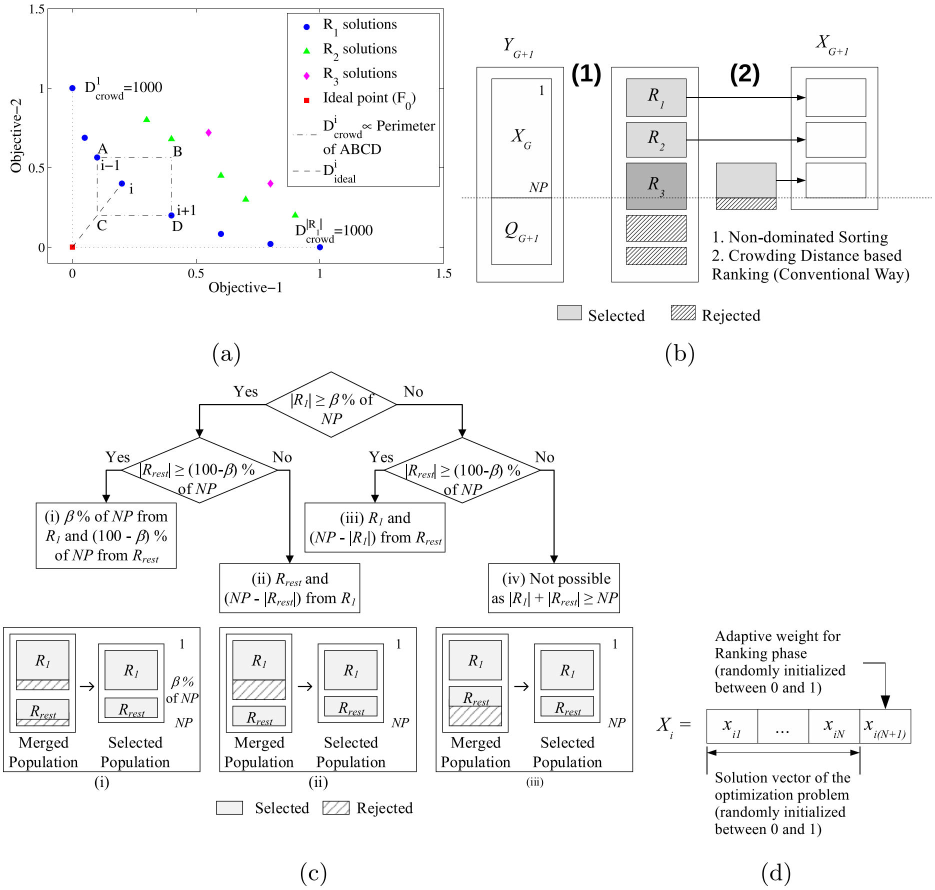


**Fig. 1.** (a) Illustration of ranks of solutions, crowding distance on $R_1$ solutions (along Objective-1) and distance from ideal point, (b) conventional elitist framework, (c) selection step of proposed elitist framework to create a population of size *NP*, (d) candidate representation for implementing the improved ranking strategy.

parent population of this generation ($X_G$). This is shown in Eq. (1).

$$\begin{aligned} Q_{p,G+1} &= U_{i,G+1}, \text{ if } X_{i,G} \nprec U_{i,G+1} \\ \text{and } Y_{G+1} &= [X_G, Q_{G+1}]^T \\ \text{where } i &= 1, \cdots, NP \text{ and } 1 \le p \le NP \end{aligned} \tag{1}$$

However, the size of this population pool ($Y_{G+1}$) might exceed *NP* depending on the number of $Q_{p,G+1}$ generated. To generate the population ($X_{G+1}$) of size *NP*, the population pool ($Y_{G+1}$) has to be trimmed by selecting the candidates which will go to $X_{G+1}$. For this elitist selection, non-dominated sorting [2,8] is the first step where the population is partitioned into several ranks such that the following properties are satisfied:

- Solutions within each of the $l^{\text{th}}$ rank (i.e., $Y_{i,G+1} \in R_l$) are non-dominated with respect to each other
- Each solution in $R_l$ is dominated by at least one of the solutions in $R_{l'}$ where $l' < l$
- Each solution in $R_l$ dominates at least one of the remaining solutions from $Y_{i,G+1} - \{R_1 \cup R_2 \cup \cdots \cup R_l\}$

The concept of non-dominated sorting is demonstrated in Fig. 1a where the population $Y_{G+1}$ has three ranks of solutions viz. $R_1$, $R_2$ and $R_3$.

The conventional next step is to reorder the solutions based on crowding distance [2,8] and allow rank-wise solutions, starting from the first rank, to pass to the next generation until the population size reaches *NP* [8]. This is demonstrated in Fig. 1b. This performs satisfactorily for multi-objective optimization algorithms.

### *3.1.2. Problem with the existing approach*

However, in several studies [2,17], it has been shown that the entire population is saturated with non-dominated solutions (rank-one solutions) towards early generations of evolutionary many-objective optimization algorithms with 10 or higher objectives. Thus, using conventional trimming of population pool leads to higher amount of non-dominated solutions and thereby, leading to premature termination of the algorithm.

*3.1.3. Alteration to conventional elitist strategy*

To prevent getting stuck at such sub-optimal scenario, the second step is modified. A novel ranking strategy (described in Section 3.2) is used to rearrange the solutions within each rank. Following this, a mixture of mostly rank-one solutions ($R_1$) and a few solutions from remaining ranks ($R_{rest}$) are used to create the population for next generation whose size is *NP*. This step of the novel elitist strategy which is performed after the non-dominated sorting on a population pool of size greater than *NP*, is outlined in Fig. 1c. The proportion of $R_1$ and $R_{rest}$ solutions are regulated by $\beta$ (in the range [0, 100]) which should usually be high to prefer the non-dominated solutions. In this work, $\beta$ is chosen as 75. This simple elitist selection overcomes the saturation problem and hence, is applicable for many-objective optimization with higher objectives.

*3.2. Improved ranking strategy at the selection stage*

For single-objective optimization, the candidates can be rearranged in ascending/descending order of fitness values for minimization/maximization problem. Ranking is not that simple for many-objective optimization as Pareto-dominance is not a total order relation.

*3.2.1. Conventional ranking strategy*

Conventionally, the approach adopted for selection and ranking is to perform non-dominated sorting, followed by reordering of the solutions within each rank using crowding distance ($D_{crowd}$) [2,8], and finally, starting from rank-one ($R_1$), allow the lower rank solutions to fill the final population until the population size reaches *NP*. This has been demonstrated in Fig. 1b.

For every $i^{\text{th}}$ candidate within the $l^{\text{th}}$ rank of solutions i.e. $\forall Y_i \in R_l$, crowding distance ($D^i_{crowd}$) assignment is accomplished as follows. Along the $j^{\text{th}}$ objective ($f_j$), the candidates having maximum ($f_j^{\max}$) and minimum ($f_j^{\min}$) objective values are assigned a crowding distance of 1000. For the remaining candidates, the crowding distance, along each objectives, is proportional to the perimeter of the hyper-rectangle formed by the normalized objectives of the candidates that precede ($Y_{i-1}$) and succeed ($Y_{i+1}$) the corresponding candidate ($Y_i$) in terms of objective value. Finally, the distances are summed across all the objectives (of full or reduced set) to give the crowding distance corresponding to a candidate ($D^i_{crowd}$). This is mathematically formulated by Eq. (2). The idea is that given a frontier (solutions of a particular rank), higher the perimeter means the neighbors of a candidate are far away and hence, lesser is the crowding of the candidate and its surrounding areas, which in turn implies that the diversity (spread of the solutions along the frontier) is better. This concept of crowding distance is demonstrated in Fig. 1a.

$$D^i_{crowd} = D_{crowd}(Y_i) = \sum_{j=1}^{a} \left( D_{crowd}(Y_i | f_j) \right)$$
$$\text{where } a = M \text{ or } a = m, \text{ and } D_{crowd}(Y_i|f_j) = \begin{cases} 1000 & \text{if } f_j(Y_i) = f_j^{max} \\ & \text{ or } f_j(Y_i) = f_j^{min} \\ \left| \dfrac{f_j(Y_{i-1}) - f_j(Y_{i+1})}{f_j^{max} - f_j^{min}} \right| & \text{where } \{Y_{i-1}, Y_i, Y_{i+1}\} \subseteq R_l \\ & \text{ and } R_l \text{ is sorted by } f_j \end{cases} \tag{2}$$

*3.2.2. Problem with existing approach*

In case of many-objective optimization, the problem with this approach is that it heavily weighs the boundary solutions of each objective and hence, as the number of objectives increases, the population ends up having high number of candidates representing the bordering points of the estimated Pareto-front. On one hand, these candidates are important for estimating the Pareto-front whereas, on the other hand, if the majority of the population is composed of these bordering solutions, the diversity is lost as there is non-uniform spread of solutions over the approximated Pareto-front.

*3.2.3. Alteration to conventional ranking strategy*

In order to consider the solutions towards the center of the Pareto-front, after non-dominated sorting, the candidates can be ranked in ascending order with respect to the Euclidean distance ($D^i_{ideal}$) between the objective vector ($F(Y_i)$) and the ideal point ($F_0 = [f_{01}, f_{02}, \cdots, f_{0a}]^T$). The concept of $D_{ideal}$ is illustrated in Fig. 1a and is mathematically given by Eq. (3) where $a = m$ (for reduced objective set) or $a = M$ (for full objective set).

$$D^i_{ideal} = D_{ideal}(Y_i) = \sqrt{\sum_{j=1}^{a} \left( f_j(Y_i) - f_{0j} \right)^2} \tag{3}$$

It should be highlighted that the ideal point ($F_0$) is a vector of minimum attainable value along each of the objectives, for minimization problems. However, $F_0$, in itself, might not be a part of the feasible objective space. For the test-suite (DTLZ) under consideration, $F_0$ is the origin of the objective space. The idea to rank the population $Y_{G+1}$ according to $D_{ideal}$ is that given a frontier, shorter distance to ideal point implies better convergence of the candidate in objective space. This ranking strategy considers equal preference among all the objectives. If the preference varies, weighted Euclidean distance or some other distance metric could be used which is suitable to the application under consideration. However, the problem with this method is that it ignores the information about the diversity of the solutions which approximate the Pareto-front.

As a trade-off between the two ranking strategies (ranking based on $D_{crowd}$ and ranking based on $D_{ideal}$), a metric ($D^i_{comb}$) for every $i^{th}$ candidate, is used which uses a weighted combination of the crowding distance ($D^i_{crowd}$) and distance from ideal point ($D^i_{ideal}$) as shown in Eq. (4).

The weight ($w^i$) is adaptively selected between 0 and 1 by optimizing it as the last element of the candidate vector i.e. $x_{i(N+1)}$. Hence, the dimension of the candidate vector increases from $N$ to $(N+1)$ as explained in Fig. 1d. It should be mentioned that a $(N+1)$-dimensional candidate (including the last element) goes through all the stages of DEMO, except during the selection stage, the $N$-dimensional sub-vector is considered for computing the objectives of the test problem.

$$D^i_{comb} = w^i \times \frac{1}{D^i_{crowd}} + (1 - w^i) \times D^i_{ideal}$$
$$\text{where } w^i = x_{i(N+1)} \tag{4}$$

Having described DEMO and its modifications, to yield IDEMO, to render it suitable as a MaOO algorithm, the proposed objective reduction approach and its integration with IDEMO is described in the next section.

## 4. Theory of proposed objective reduction based optimization approach - DECOR

This section presents the proposed MaOO approach viz. DECOR, which uses correlation based online objective reduction and IDEMO (described in Section 3) as the optimizer with improved elitist selection and ranking strategies.

### *4.1. Correlation distance*

Previous studies in literature [2,21,30] have presented correlation based objective reduction algorithm to perform many-objective optimization. This work uses the correlation distance which is defined between two $n$-dimensional vectors (say, $A$ and $B$) as shown in Eq. (5) and ranges between [0, 2].

$$Dist(A, B) = 1 - \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{5}$$

### *4.2. Objective reduction principle*

In order to implement objective reduction, the conflicting objectives are to be identified. The idea for this step is as follows. Based on the current population, the rank-one ($R_1$) solutions represent an estimate of the Pareto-front. The correlation distances between every objective pairs are noted where the estimate of the $i^{th}$ objective ($f_i(.)$) is given by the $i^{th}$ row vector of the transpose of the population matrix (only candidates from $R_1$ solutions) mapped in to objective space as shown in Eq. (6) and Fig. 2a. Hence, as more number of solutions contribute to form the $R_1$ solutions, better is the estimate of the objectives ($f_1$ to $f_a$, with $a = M$ for full objective set and $a = m$ for reduced objective set). Closer the objectives in terms of correlation distance, more is the correlation between these objectives and hence, these are less conflicting than other objective pairs. Based on this observation, the central idea for objective reduction is to cluster the objective representatives (obtained from estimated Pareto-front) and eliminate all the neighbours of the cluster center of the most compact cluster while retaining its center. This presents a provision for eliminating multiple objectives at a time. Hence, less conflicting objectives are eliminated. This basic idea is outlined in Fig. 2b.

$$f_i(.) = \left[f_i(X_1), \cdots, f_i(X_{|R_1|})\right], \text{ where } i = 1, \cdots, a \tag{6}$$

### *4.3. Specifications of clustering*

The clustering algorithm should be simple and fast. Hence, $k$-medoids clustering is used which selects a constituent (real) objective vector of a cluster as the cluster center. It is implemented using Partitioning Around Medoid (PAM) [23] with correlation distance representing the similarity among data points for clustering. When performed on the full objective set, clustering has the following specifications:

- Clustering partitions $M$ objectives $[f_1, f_2, \cdots, f_M]$ in to $k$ clusters $[C_1, C_2, \cdots, C_k]$
- $\exists j$ such that $f_i \in C_j$, and $\nexists (j_1, j_2)$ such that $f_i \in C_{j_1}$ and $f_i \in C_{j_2}$, where $i = 1, 2, \cdots, M$ (hard clustering)
- $\sum_{j=1}^{k} |C_j| = M$ and $1 \le |C_j| \le (M - k - 1)$, where $j = 1, 2, \cdots, k$
- $\forall f_i \in C_j$, either it is the medoid ($f_i = C_j^{med}$) or it belongs to the non-medoid set ($f_i \in C_j^{nmed}$)

For reduced objective set, the above specifications hold, where $M$ is replaced by $m$.

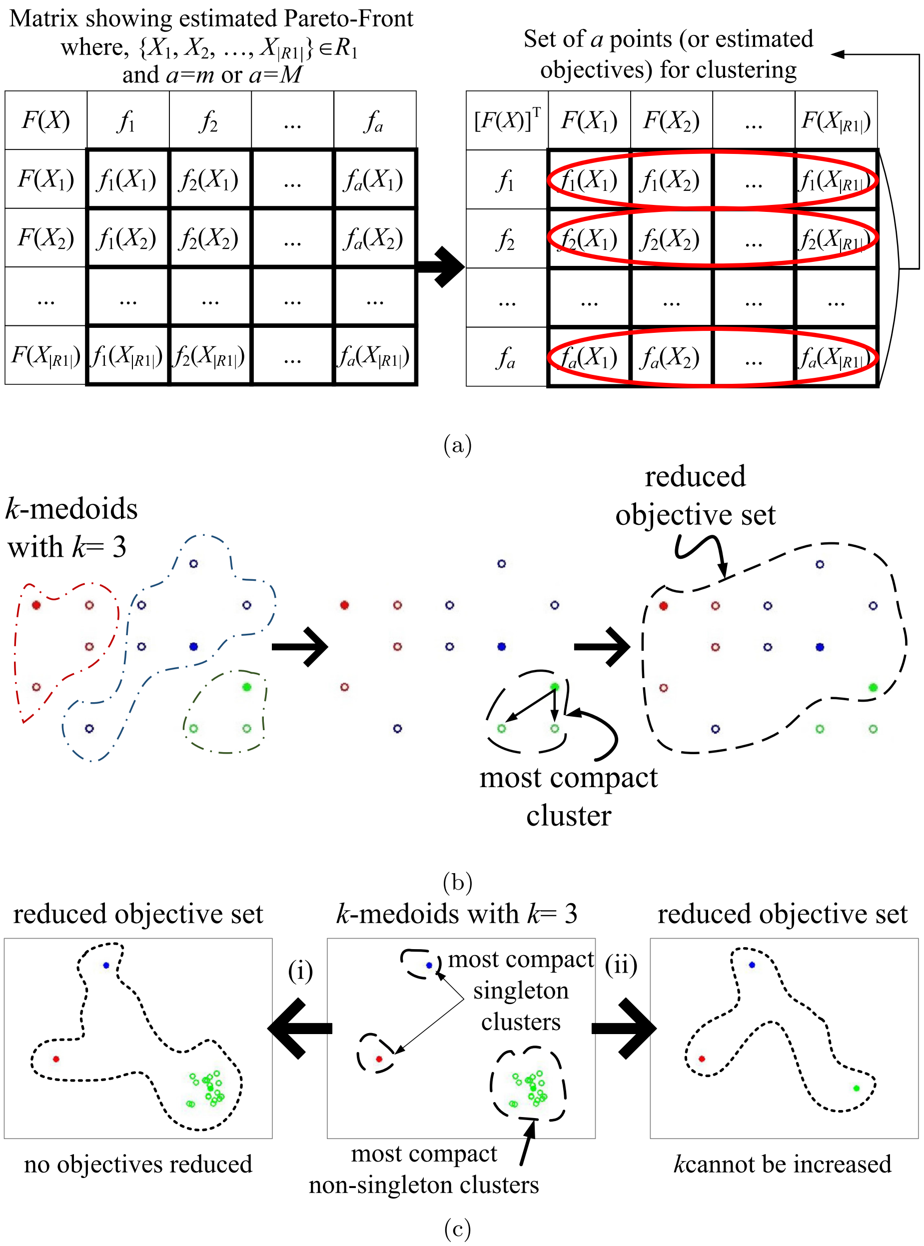


**Fig. 2.** (a) Construction of the datapoints (representatives of each of the objectives encircled) for clustering, (b) objective reduction principle, (c) issues in objective reduction due to the presence of singleton clusters: (i) scenario when a singleton cluster is considered as the most compact cluster, (ii) scenario when a non-singleton cluster is considered as the most compact cluster and there are $(k-1)$ singleton clusters and one non-singleton cluster.

### *4.4. Concept of most compact cluster*

The next step after clustering is to choose the most compact cluster. For each cluster $C_j$, the sum of medoid to non-medoid correlation distance is calculated and the cluster having minimum value of this sum is declared as the most compact cluster with the best-known estimate of Pareto-front at a given generation. By following Eq. (5), the most compact cluster ($C_{com}$) is given by Eq. (7).

$$C_{com} = \arg\min_{j=1}^{k} \sum_{|C_j^{nmed}|} Dist(f_{i_1}, f_{i_2})$$

$$\text{where } f_{i_1} = C_j^{med} \text{ and } f_{i_2} \in C_j^{nmed} \tag{7}$$

### 4.5. Problem with singleton clusters

A special case needs to be considered while determination of the most compact cluster. This special case arises when clustering results in one or more singleton clusters (i.e., $|C_j| = 1$) which trivially are the most compact clusters. As there is no neighbour in a singleton cluster, no objective reduction occurs. If such a singleton cluster keeps on appearing in successive stages, the objective reduction gets stuck.

The next possible way to avoid this problem of singleton cluster could be to consider the most compact non-singleton cluster for objective reduction. However, a scenario may arise when $M$ objectives are clustered in to $k$ clusters such that there are $(k-1)$ singleton clusters and one non-singleton cluster with $(M-k-1)$ objectives. Using the proposed principle of objective reduction from most compact non-singleton cluster, this will yield $k$ objectives in the reduced set and as incrementing $k$ will not be possible any further, the objective reduction procedure will terminate prematurely.

An illustration to demonstrate both these extreme scenarios arising in presence of singleton clusters is given in Fig. 2c. Hence, a trade-off approach has to be adopted to compromise between these two extreme cases in the objective reduction stage.

### 4.6. Proposed solution for handling singleton clusters

There can be two possible scenarios while singleton clusters are encountered. These scenarios and the strategies to tackle these scenarios are as follows:

1. *Case-1:* When the singleton cluster is significantly far away from the nearest cluster, it forms a crucial objective which is in conflict with other clusters of objectives. Hence, it is directly added to the reduced set of objectives, and the next most compact cluster is analysed, immediately. Thus, the objective reduction continues without getting stuck.
2. *Case-2:* On the other hand, when the singleton cluster is comparatively closer to the nearest cluster, there can be a possibility that the singleton cluster is resultant because of the present value of $k$ (number of clusters) and/or the present estimate of the Pareto-front. In such a scenario, the singleton cluster itself is considered as the most compact cluster so that no objective reduction occurs and $k$ is increased in successive turns for finer clustering.

In order to differentiate whether the singleton cluster ($C_j$) is significantly far away or comparatively closer to the nearest cluster, a ratio ($D^j_{ratio}$), defined in Eq. (10), is used. The correlation distance between the singleton cluster ($C_j$) to the nearest cluster's medoid is defined as $D^j_{near}$ (Eq. (8)) and the maximum of intra-cluster medoid to non-medoid correlation distance over all the clusters at the present state is defined as $D_{neigh}$ (Eq. (9)).

$$D^j_{near} = \min_{j'=1, j'\neq j}^{k} Dist(f_{i_1}, f_{i_2}) \quad \text{where } f_{i_1} = C^{med}_j \text{ and } f_{i_2} = C^{med}_{j'} \tag{8}$$

$$D_{neigh} = \max_{j=1}^{k} Dist(f_{i_1}, f_{i_2}) \quad \text{where } f_{i_1} = C^{med}_j \text{ and } f_{i_2} \in C^{nmed}_j \tag{9}$$

$$D^j_{ratio} = \frac{D^j_{near}}{D_{neigh}} \tag{10}$$

If $D^j_{ratio}$ is at least greater than some threshold ($th$), implying that the singleton cluster ($C_j$) is significantly far away from the nearest cluster, it is considered as a conflicting objective and is directly added to the reduced objective set (by following Case-1). On the other hand, if $D^j_{ratio}$ is less than the threshold ($th$) implying the singleton cluster ($C_j$) is comparatively closer to the nearest cluster, no objective reduction occurs in the present state (by following Case-2). In this work, the value of the threshold ($th$) is chosen by trial and error.

### 4.7. Selecting the number of clusters

Next issue with the clustering approach of objective reduction is choosing the number of clusters (or equivalently, the value for $k$). In the proposed work, the objective reduction procedure starts with a small value of $k$, and after certain generations of the optimization algorithm, the value of $k$ is increased by 1 at a time. This increment happens periodically until $k$ equals number of objectives because clustering is not possible with $k$ higher than this value. The number of generations after which $k$ is incremented is controlled by the $K_{op}$ parameter. Through this step, most of the clusters are explored eventually, even if these were not declared as the most compact cluster in earlier stages.

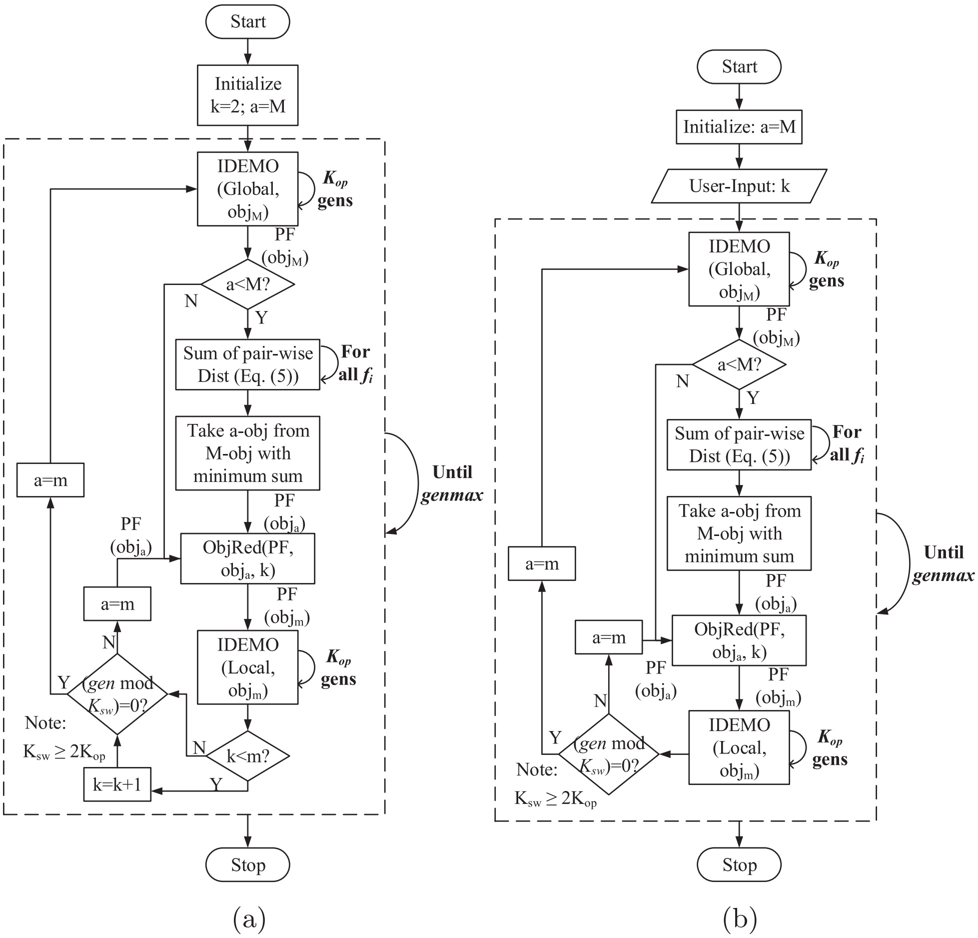


**Fig. 3.** Flowchart describing the proposed framework for: (a) automatic-DECOR (or aDECOR), (b) fixed-DECOR (or fDECOR).

*4.7.1. Reasons for online objective reduction*

As the rank-one ($R_1$) solution is just an estimate of the Pareto-front, the true correlation between the objectives or the exact pairs/groups of conflicting objectives cannot be determined at an early stage. Due to this evolving nature of the correlation structure, the online version of objective reduction is adopted. In this approach, the optimization starts with global exploration of the search space with the full objective set, then continues by local exploitation of the search space which is defined by the reduced objective space, after which global exploration resumes with the full objective set. The switching parameter, $K_{sw}$, regulates the number of generations after which global search (with full objective set) and the local search (with reduced objective set) toggles.

*4.7.2. Two different versions of DECOR*

The flowchart describing the proposed framework integrating the objective reduction and the optimization algorithm viz. automatic-DECOR or aDECOR is shown in Fig. 3a and Algorithm 1 which in turn calls the objective reduction procedure as described in Algorithm 2. It is called automatic because the number of objectives of the reduced set is automatically determined by the algorithm and does not involve any input from the user.

A problem arises while using the automatic version of the proposed MaOO algorithm viz. aDECOR. For sufficiently large $M$ (number of objectives) and very small $k$ (number of clusters), say $k = 2$, a situation may arise where there are very high number of objectives in each of the clusters, say $M/k$ in each cluster. In such situation, objective reduction from the most compact cluster leads to elimination of very high number of objectives (almost half of the objectives in the example scenario). This is undesirable at an early stage because the estimation of Pareto-front and thereby, the estimation of the resulting objective vectors is poor.

To avoid this situation, a fixed version of DECOR viz. fixed-DECOR or fDECOR, is proposed. It is called fixed because the size of the reduced objective set is lower bounded by a fixed user-specified value of $k$. Clustering yields $k$ clusters and in extreme case like in (ii) of Fig. 2b, the reduced objective set has at least $k$ objectives. So, for MaOO problems with very high $M$, specifying a high $k$ will not lead to drastic reduction in the number of objectives. It should be noted that although

**Algorithm 1** Complete framework of the proposed approach - aDECOR.

**Input:** $\{F(X)\}$ Objective functions for MaOO problem
**Output:** $PS$ Pareto-optimal Set; $PF$ Pareto-front

Initialize: $k = 2$; $a = M$; $gen = 1$; $flag = 1$;
**while** $gen <= genmax$ (GLOBAL SEARCH) **do**
  Execute IDEMO on full objective set ($obj_M$) for $K_{op}$ generations, set $flag = 0$ when population has no new candidate, and save $PS$ and $PF$
  $gen = gen + K_{op}$
  **if** flag=0 (if no new candidates are found) **then**
    Break;
  **end if**
  **if** $a < M$ **then**
    $\forall f_i$, $S_i = \sum_{j=1, j \neq i}^{M} Dist(f_i, f_j)$ (Eq. (5))
    Sort all $f_i$ based on $S_i$ in descending order
    Form $obj_a$ with top $a$ conflicting objectives
  **end if**
  **while** $(gen) mod (K_{sw}) \neq 0$ (LOCAL SEARCH) **do**
    $obj_m = ObjRed(PF, obj_a, k)$;
    Execute IDEMO on reduced objective set ($obj_m$) for $K_{op}$ generations, set $flag = 0$ when population has no new candidate, and save $PS$ and $PF$
    $gen = gen + K_{op}$
    **if** flag=0 (if no new candidates are found) **then**
      Break;
    **end if**
    **if** $k < m$ **then**
      $k = k + 1$;
    **end if**
    $a = m$;
  **end while**(END OF LOCAL SEARCH)
  **if** flag=0 (if no new candidates are found) **then**
    Break;
  **end if**
**end while**(END OF GLOBAL SEARCH)

**Algorithm 2** Procedure for objective reduction.

**Input:** $PF$ Pareto-front; $obj_a$ Objective set of size $a$; $k$ Number of clusters
**Output:** $obj_m$ Objective set of size $m$ where $m \leq a$

1: **procedure** ObjRed($PF$, $obj_a$, $k$)
2:   Execute $k$-medoids on $PF$ using $Dist$ (Eq. (5))
3:   Find $D_{neigh}$ (Eq. (9))
4:   **for** $i = 1$ to $k$ (for all clusters) **do**
5:     Initialize $Flag_i$ = 0
6:     Obtain $|C_i|$
7:     **if** $|C_i| = 1$ (if $C_i$ is a singleton cluster) **then**
8:       Find $D_{near}^i$ (Eq. (8)) and $D_{ratio}^i$ (Eq. (10))
9:       **if** $D_{ratio}^i \geq th$ **then**
10:         Set $Flag_i = 1$ (to ignore $C_i$ in Step 14)
11:       **end if**
12:     **end if**
13:   **end for**
14:   Find $C_{com}$ using Eq. (7) (ignore $C_i$ if $Flag_i = 1$)
15:   Construct $obj_m = obj_a - \{f_j | f_j \in C_{com}^{nmed}\}$
16:   Return $obj_m$
17: **end procedure**

$k$ is not incremented periodically in fDECOR, the objective reduction module is executed periodically to ensure that the correlation structure is evolved along with the evolution of the objective vectors.

The flowchart describing fDECOR is shown in Fig. 3b and Algorithm 3 which in turn also calls the objective reduction procedure as described in Algorithm 2.

**Algorithm 3** Complete framework of the proposed approach - fDECOR.

**Input:** $\{F(X)\}$ Objective functions for MaOO problem; $k$ Number of clusters (lower bound of $m$)
**Output:** $PS$ Pareto-optimal Set; $PF$ Pareto-front

Initialize: $a = M$; $gen = 1$; $flag = 1$;
User-Input: $k$;
**while** $gen <= genmax$ (GLOBAL SEARCH) **do**
    Execute IDEMO on full objective set ($obj_M$) for $K_{op}$ generations, set $flag = 0$ when population has no new candidate, and save $PS$ and $PF$
    $gen = gen + K_{op}$
    **if** flag=0 (if no new candidates are found) **then**
        Break;
    **end if**
    **if** $a < M$ **then**
        $\forall f_i$, $S_i = \sum_{j=1, j\neq i}^{M} Dist(f_i, f_j)$ (Eq. (5))
        Sort all $f_i$ based on $S_i$ in descending order
        Form $obj_a$ with top $a$ conflicting objectives
    **end if**
    **while** $(gen) mod (K_{sw}) \neq 0$ (LOCAL SEARCH) **do**
        $obj_m = ObjRed(PF, obj_a, k)$;
        Execute IDEMO on reduced objective set ($obj_m$) for $K_{op}$ generations, set $flag = 0$ when population has no new candidate, and save $PS$ and $PF$
        $gen = gen + K_{op}$
        **if** flag=0 (if no new candidates are found) **then**
            Break;
        **end if**
        $a = m$;
    **end while**(END OF LOCAL SEARCH)
    **if** flag=0 (if no new candidates are found) **then**
        Break;
    **end if**
**end while**(END OF GLOBAL SEARCH)

The time complexity of Algorithm 2 (objective reduction) considering line 2 ($\mathcal{O}(k.(a-k)^2.|R_1|^2.I)$) [23], line 3 ($\mathcal{O}((a-k).|R_1|)$), line 4 to line 13 ($\mathcal{O}(a+k^2.|R_1|)$), line 14 ($\mathcal{O}((a-k).|R_1|)$) and line 15 ($\mathcal{O}(a)$) is $\mathcal{O}(k.(a-k)^2.|R_1|^2.I)$ where $a$ is the size of $obj_a$, $k$ is the number of cluster, $|R_1|$ represents the dimension of each objective vector (Eq. (6)) for clustering and $I$ is the number of iterations needed by $k$-medoid.

After designing the algorithmic framework, it is applied on several benchmark problems in order to assess the efficacy of the proposed approach. The results are presented and compared with other state-of-the-art MaOO algorithms in the subsequent sections.

## 5. Results

The performances of aDECOR and fDECOR are presented and analysed in this section. The algorithm is run on a computer having 4GB RAM and Intel Core i3 processor @2.30 GHz, using the 32-bit version of MATLAB R2012b. Each of the algorithms is executed 50 times and the average and the standard deviation of the results are noted for comparison.

The performance of DECOR is compared with the performance of six other popular algorithms viz. NSGA-II [8], MOEA/D [32], HypE [1], DEMO [33], $\alpha$-DEMO-revised [2] and the optimization algorithm presented in [30]. The reasons for selecting these algorithms for comparison are as follows:

- Some of the popular and well-cited algorithms that utilize different strategies to address MOO problems are NSGA-II [8] (pioneering work for crowding distance based ranking), HypE [1] (indicator based ranking) and MOEA/D [32] (decomposition into sub-problems). Hence, these have been selected for comparison of performance of DECOR.
- As DECOR uses IDEMO which is an improved version of DEMO, hence DEMO [33] forms an important reference against which the performance of DECOR should be compared.
- As DECOR is an extension of the approach proposed in [30], hence the results of [30] are important for analysing the performance of DECOR.

**Table 1**
Values of different parameters used to run and analyse DECOR.

| Parameters | Explanation | Values |
|---|---|---|
| $NP$ | Population size | 100 |
| $genmax$ | Maximum generations | 2000 |
| $CR$ | Crossover Rate | 0.8 |
| $K_{op}$ | Number of generations for which IDEMO runs at a time | 20 |
| $K_{sw}$ | Number of generations after which DECOR switches from $obj_m$ to $obj_M$ | 100 |
| $\beta$ | Percentage of $R_1$ solutions | 75 |
| $th$ | Threshold for considering singleton cluster for objective reduction | Chosen from {1.2, 1.5, 2.0} |
| $\|H\|$ | Cardinality of reference set | |
| | (1) Convergence Metric | 500 |
| | (2) Hypervolume Indicator | 10,000 |
| $r$ | Reference point (objective vector) for Hypervolume Indicator | $\{3, \overset{M}{\cdots}, 3\}$ |

- Finally, $\alpha$-DEMO-revised [2] is one of the contemporary algorithms that has demonstrated significant performance improvements in comparison to existing many-objective optimization algorithms. Further $\alpha$-DEMO-revised utilizes objective reduction technique to handle the many objective optimization problems. Hence, it forms an important basis for comparison.

Besides all these reasons, the works in Refs. [2,30] have compared their proposed approaches against the above-mentioned algorithms. Therefore, DECOR is also compared with these algorithms where Bandyopadhyay and Mukherjee [2] and Pal et al. [30] have been consulted for the performance measures of NSGA-II, MOEA/D, HypE and DEMO.

As a case study, DECOR is tested on four test problems viz. DTLZ1, DTLZ2, DTLZ3 and DTLZ4 for 10 and 20 objectives. The performance of the competing algorithms are measured using convergence metric [2,29,30] and hypervolume indicator [2,29,30]. The resulting Pareto-front from DECOR is visualized for comparison with the true Pareto-front of the DTLZ problems and the resulting Pareto-front from $\alpha$-DEMO-revised [2] using parallel coordinates [5,18]. Finally, statistical tests are performed to validate the results.

### 5.1. Parameter specifications

To execute DECOR and to use the performance metrics, several parameters have to be set. The algorithm is executed with various parameter settings and best results are obtained for the parameters whose specifications are mentioned in Table 1.

Increasing $NP$ while keeping $genmax$ fixed, does not improve performance as comparatively more random initialization of candidates occur rather than mutation and recombination, which imply proportionately lesser number of good solutions being propagated. Again, keeping $NP$ fixed and increasing $genmax$, does not improve the results any further. The parameter $CR$ is kept high in order to generate a solution far from the parent candidate. However, with $CR > 0.8$, the trial vector is mostly independent of the parent candidate which leads to poor performance. The parameter $K_{op}$ is the minimum number of generations over which some significant change in performance (change in total $D_{ideal} > 10^{-2}$) is observed. Incrementing $K_{sw}$ leads to more local search and thus poor performance at global level whereas decreasing $K_{sw}$ slows down objective reduction. When $\beta > 75$, more sub-optimal non-dominated solutions are passed on to next generations, whereas when $\beta < 75$, more number of potential $R_1$ solutions are not propagated, resulting in poor performance. For setting the reference point ($r$) and $|H|$ for performance metrics, the work in [2] is consulted.

### 5.2. Performance of aDECOR

This algorithm involves a thresholding which is associated with $D_{ratio}$ for deciding whether to consider a singleton cluster in objective reduction procedure. It should be mentioned that $D_{ratio}$ should be greater than 1. As $D_{near}$ is always greater than $D_{neigh}$. But a $D_{ratio}$ higher than 2 implies the singleton cluster is sufficiently far away from the nearest cluster to be considered as an essential cluster. Hence, the threshold ($th$) on $D_{ratio}$ should be chosen in the range of (1, 2]. As a case study, this work tries for $th = \{1.2, 1.5, 2.0\}$ and reports that value in Tables 2 and 3 which gives best performance corresponding to aDECOR and fDECOR, respectively. The execution of aDECOR is mentioned in Table 2, in terms of convergence metric and hypervolume indicator. For evaluating convergence metric, a sampled version representing the true Pareto-front is needed for which [6] has been referred. NSGA-II, MOEA/D, HypE and DEMO are not objective reduction based optimization algorithms. Hence, the results obtained using the full objective set ($obj_M$) are noted for these four algorithms. As aDECOR automatically determines the size of the reduced objective set ($m$), the final value of $m$, as yielded by aDECOR, is used as an input for $\alpha$-DEMO-revised which requires the user to specify $m$ in terms of $\alpha$. Hence, for NSGA-II, MOEA/D, HypE, and DEMO, $M = m = 10$ (or 20) and for $\alpha$-DEMO-revised, and aDECOR, $m$ values for corresponding $M$ values are mentioned in Table 2 in the format DTLZ$\eta$_$M, m$ where $\eta$ indicates the type of the DTLZ problem.

**Table 2**
Comparing aDECOR with other optimization algorithms.

(a) aDECOR vs. others (in terms of Convergence Metric)

| Problem type | Threshold ($th$) on $D_{ratio}$ | NSGA-II | MOEA/D | HypE | DEMO | $\alpha$-DEMO-revised | aDECOR |
|---|---|---|---|---|---|---|---|
| DTLZ1_10,6 | 1.2 | 225.4502 ± 5.9816 | 2.4800 ± 1.0351 | 146.3039 ± 2.2147 | 142.2519 ± 3.1073 | 1.0291 ± 0.0061 | **0.3993 ± 0.0042** |
| DTLZ2_10,6 | 1.5 | 1.4716 ± 0.0317 | 0.7419 ± 0.0101 | 1.3979 ± 0.0156 | 1.3891 ± 0.0161 | 1.3858 ± 0.0907 | **0.4088 ± 0.0111** |
| DTLZ3_10,6 | 1.2 | 1048.0740 ± 39.3631 | 24.8627 ± 4.5587 | 409.5137 ± 3.9870 | 939.7426 ± 9.8824 | 1.0011 ± 0.0245 | **0.5256 ± 0.0153** |
| DTLZ4_10,6 | 1.5 | 1.1784 ± 0.0264 | 0.7461 ± 0.0102 | 0.8914 ± 0.0106 | 1.2663 ± 0.0347 | 1.5048 ± 0.0306 | **0.4768 ± 0.0092** |
| DTLZ1_20,10 | 1.5 | 176.2357 ± 3.6600 | 3.2397 ± 1.1651 | 305.1945 ± 9.7488 | 143.5408 ± 2.7434 | 1.2356 ± 0.0210 | **0.3307 ± 0.0310** |
| DTLZ2_20,10 | 1.5 | 1.9273 ± 0.0224 | 1.3116 ± 0.0050 | 1.9240 ± 0.0144 | 1.9009 ± 0.0092 | 1.1112 ± 0.0172 | **0.4696 ± 0.0177** |
| DTLZ3_20,10 | 1.5 | 978.3490 ± 44.9975 | 37.8409 ± 7.2125 | 911.8077 ± 5.5582 | 1024.4046 ± 12.5577 | 94.6363 ± 1.5260 | **0.4925 ± 0.0293** |
| DTLZ4_20,11 | 1.5 | 1.4337 ± 0.0309 | 1.0818 ± 0.0070 | 0.9572 ± 0.0077 | 1.6816 ± 0.0370 | 1.5048 ± 0.0098 | **0.4768 ± 0.0307** |

(b) aDECOR vs. others (in terms of Hypervolume Indicator)

| Problem type | Threshold ($th$) on $D_{ratio}$ | NSGA-II | MOEA/D | HypE | DEMO | $\alpha$-DEMO-revised | aDECOR |
|---|---|---|---|---|---|---|---|
| DTLZ1_10,6 | 1.2 | 0.0044 ± 0.0061 | 0.8132 ± 0.0984 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.1385 ± 0.0092 | **0.9915 ± 0.0098** |
| DTLZ2_10,6 | 1.5 | 0.8399 ± 0.0079 | **1.0000 ± 0.0000** | 0.9514 ± 0.0034 | 0.8863 ± 0.0059 | 0.9103 ± 0.0675 | 0.8765 ± 0.0018 |
| DTLZ3_10,6 | 1.2 | 0.0000 ± 0.0000 | 0.0235 ± 0.0388 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.2967 ± 0.0013 | **0.9879 ± 0.0105** |
| DTLZ4_10,6 | 1.5 | 0.9765 ± 0.0056 | **1.0000 ± 0.0000** | 0.8741 ± 0.0169 | 0.9956 ± 0.0012 | 0.1786 ± 0.0313 | 0.9488 ± 0.0072 |
| DTLZ1_20,10 | 1.5 | 0.0000 ± 0.0000 | 0.7233 ± 0.1172 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.9779 ± 0.0176 | **0.9994 ± 0.0206** |
| DTLZ2_20,10 | 1.5 | 0.8280 ± 0.0070 | **1.0000 ± 0.0000** | 0.9372 ± 0.0019 | 0.8487 ± 0.0059 | 0.9213 ± 0.0150 | 0.8016 ± 0.0050 |
| DTLZ3_20,10 | 1.5 | 0.0000 ± 0.0000 | 0.0301 ± 0.0391 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.3213 ± 0.0055 | **0.9964 ± 0.0098** |
| DTLZ4_20,11 | 1.5 | 0.9914 ± 0.0030 | **1.0000 ± 0.0000** | 0.8963 ± 0.0103 | 0.9829 ± 0.0111 | 0.9018 ± 0.0761 | 0.9420 ± 0.0155 |

**Table 3**
Comparing fDECOR with other optimization algorithms.

(a) fDECOR vs. others (in terms of Convergence Metric)

| Problem type | Threshold ($th$) on $D_{ratio}$ | NSGA-II | MOEA/D | HypE | DEMO | $\alpha$-DEMO-revised | Approach in [30] | fDECOR |
|---|---|---|---|---|---|---|---|---|
| DTLZ1_10,7 | 1.2 | 225.4502 ± 5.9816 | 2.4800 ± 1.0351 | 146.3039 ± 2.2147 | 142.2519 ± 3.1073 | 1.1718 ± 0.0094 | 0.3991 ± 0.0017 | **0.3421 ± 0.0029** |
| DTLZ2_10,6 | 1.2 | 1.4716 ± 0.0317 | 0.7419 ± 0.0101 | 1.3979 ± 0.0156 | 1.3891 ± 0.0161 | 1.3858 ± 0.0907 | 0.5214 ± 0.0069 | **0.3843 ± 0.0070** |
| DTLZ3_10,6 | 1.2 | 1048.0740 ± 39.3631 | 24.8627 ± 4.5587 | 409.5137 ± 3.9870 | 939.7426 ± 9.8824 | 1.0011 ± 0.0245 | 0.5877 ± 0.0014 | **0.5256 ± 0.0153** |
| DTLZ4_10,7 | 1.5 | 1.1784 ± 0.0264 | 0.7461 ± 0.0102 | 0.8914 ± 0.0106 | 1.2663 ± 0.0347 | 0.5218 ± 0.0059 | 0.4780 ± 0.0014 | **0.3815 ± 0.0067** |
| DTLZ1_20,14 | 1.2 | 176.2357 ± 3.6600 | 3.2397 ± 1.1651 | 305.1945 ± 9.7488 | 143.5408 ± 2.7434 | 1.7954 ± 0.0565 | 1.0095 ± 0.0041 | **0.4391 ± 0.0046** |
| DTLZ2_20,12 | 1.5 | 1.9273 ± 0.0224 | 1.3116 ± 0.0050 | 1.9240 ± 0.0144 | 1.9009 ± 0.0092 | 1.3915 ± 0.0175 | 1.1610 ± 0.0076 | **0.8412 ± 0.0183** |
| DTLZ3_20,13 | 1.2 | 978.3490 ± 44.9975 | 37.8409 ± 7.2125 | 911.8077 ± 5.5582 | 1024.4046 ± 12.5577 | 1.4153 ± 0.0111 | 0.8591 ± 0.0068 | **0.5211 ± 0.0034** |
| DTLZ4_20,11 | 1.2 | 1.4337 ± 0.0309 | 1.0818 ± 0.0070 | 0.9572 ± 0.0077 | 1.6816 ± 0.0370 | 1.4034 ± 0.0197 | 0.4716 ± 0.0021 | **0.3111 ± 0.0101** |

(b) fDECOR vs. others (in terms of Hypervolume Indicator)

| Problem type | Threshold ($th$) on $D_{ratio}$ | NSGA-II | MOEA/D | HypE | DEMO | $\alpha$-DEMO-revised | Approach in [30] | fDECOR |
|---|---|---|---|---|---|---|---|---|
| DTLZ1_10,7 | 1.2 | 0.0044 ± 0.0061 | 0.8132 ± 0.0984 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.9281 ± 0.0016 | **0.9544 ± 0.0545** | 0.9182 ± 0.0087 |
| DTLZ2_10,6 | 1.2 | 0.8399 ± 0.0079 | **1.0000 ± 0.0000** | 0.9514 ± 0.0034 | 0.8863 ± 0.0059 | 0.9103 ± 0.0675 | 0.6054 ± 0.0037 | 0.6314 ± 0.0114 |
| DTLZ3_10,6 | 1.2 | 0.0000 ± 0.0000 | 0.0235 ± 0.0388 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.2967 ± 0.0013 | **0.9848 ± 0.0759** | 0.9525 ± 0.0130 |
| DTLZ4_10,7 | 1.5 | 0.9765 ± 0.0056 | **1.0000 ± 0.0000** | 0.8741 ± 0.0169 | 0.9956 ± 0.0012 | 0.8632 ± 0.054 | 0.8850 ± 0.0273 | 0.9095 ± 0.0301 |
| DTLZ1_20,14 | 1.2 | 0.0000 ± 0.0000 | 0.7233 ± 0.1172 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.8743 ± 0.0081 | 0.9704 ± 0.0356 | **0.9791 ± 0.0234** |
| DTLZ2_20,12 | 1.5 | 0.8280 ± 0.0070 | **1.0000 ± 0.0000** | 0.9372 ± 0.0019 | 0.8487 ± 0.0059 | 0.9117 ± 0.0477 | 0.6444 ± 0.0028 | 0.9127 ± 0.0167 |
| DTLZ3_20,13 | 1.2 | 0.0000 ± 0.0000 | 0.0301 ± 0.0391 | 0.0000 ± 0.0000 | 0.0000 ± 0.0000 | 0.4830 ± 0.0035 | 0.9825 ± 0.0468 | **0.9870 ± 0.0099** |
| DTLZ4_20,11 | 1.2 | 0.9914 ± 0.0030 | **1.0000 ± 0.0000** | 0.8963 ± 0.0103 | 0.9829 ± 0.0111 | 0.9018 ± 0.0761 | 0.9608 ± 0.0383 | 0.9053 ± 0.0083 |

### *5.3. Performance of fDECOR*

The execution of fDECOR is also compared with NSGA-II, MOEA/D, HypE, and DEMO. These are executed on the full objective set ($obj_M$) as these cannot perform objective reduction. Besides these, the performance of the algorithm in [30], which has overcome several drawbacks of previous objective reduction algorithms is noted for comparison. As the approach in [30] automatically determines the size of the reduced objective set ($m$), the final $m$ values provided in [30] are used as input for fDECOR and $\alpha$-DEMO-revised. Thus, the performance of fDECOR is compared with six other algorithms (four optimization algorithms without objective reduction and two algorithms with objective reduction) where for NSGA-II, MOEA/D, HypE, and DEMO, $M = m = 10$ (or 20) and for fDECOR, $\alpha$-DEMO-revised, and the approach in [30], $m$ values for corresponding $M$ values are mentioned in Table 3 in the same DTLZ$\eta$_$M, m$ format ($\eta$ represents the type of the DTLZ problem). The optimization performance is estimated in terms of convergence metrics and hypervolume indicator as noted from Table 3. It should be noted that the approach in [30] and aDECOR cannot be compared directly as both of these automatically determine the value of $m$, and thus, the final $m$ values are often different.

### 5.4. Comparison by visualization

Due to the similarities of DECOR with $\alpha$-DEMO as (i) both are objective reduction based MaOO algorithms using the search capability of differential evolution, and (ii) both use Pearson's correlation coefficient to quantify conflict among objectives, the Pareto-fronts resulting from both the algorithms are compared. Also, the results are compared with the true Pareto-fronts which are used in forming the reference set while estimating the convergence metric.

Using different *th* and *m* values from Tables 2 and 3, the Pareto-fronts (true, results of $\alpha$-DEMO-revised, aDECOR and fDECOR) are compared using the parallel coordinates for all the four test problems (with 10 and 20 objectives) which are shown in Fig. 4.

### 5.5. Statistical analysis

For statistical validation of the results, Friedman Test [10], McNemar's Test [25] and Holm-Bonferroni Test [19] have been performed.

#### 5.5.1. Friedman test

Assumptions made for the Friedman test are as follows: the null hypothesis ($H_0$) states that all the algorithms are ranked equally while the alternate hypothesis ($H_a$) states that the algorithms are ranked as proposed in Table 4. Friedman statistic (Eq. (11)) follows a $\chi^2$-distribution with $(k_a - 1)$ degrees of freedom where $N_d$ represents number of datasets which is four in the present case, $k_a$ represents number of algorithms being compared which is six while aDECOR is considered and seven while fDECOR is considered and $R_{F,i}$ represents the average rank of the $i^{\text{th}}$ algorithm being compared as specified in Table 4.

$$\chi_F^2 = \frac{12N_d}{k_a(k_a+1)}\left\{\sum_{i=1}^{k_a} R_{F,i}^2 - \frac{k_a(k_a+1)^2}{4}\right\} \tag{11}$$

For aDECOR, convergence metric attains $\chi_F^2 > \chi_{5,0.05}^2 = 11.07$ (critical value of $\chi^2$ for five degrees of freedom and 95% confidence interval) and thus, the test rejects $H_0$, favouring $H_a$. This is not the case for hypervolume indicator where $\chi_F^2 < \chi_{5,0.05}^2 = 11.07$ and hence, it fails to reject $H_0$. For fDECOR, similar Friedman test results are observed i.e., based on convergence metric, $H_0$ could be rejected as $\chi_F^2 > \chi_{6,0.05}^2 = 12.59$ (critical value for six degrees of freedom and 95% confidence interval) while based on hypervolume indicator, the test fails to reject $H_0$ as $\chi_F^2 < \chi_{6,0.05}^2 = 12.59$.

#### 5.5.2. McNemar's test

McNemar's test is another non-parametric test following $\chi^2$-distribution which compares algorithms on a one-against-one basis. McNemar's test statistics are given by Eq. (12) where $n_{AB}$ denotes the number of times algorithm A performs better than algorithm B and $n_{BA}$ denotes the number of times algorithm B performs better than algorithm A. Each algorithm is executed 50 times on each problem type and there are four problem types (DTLZ1 to DTLZ4) within each category (10/20 number of objectives under one performance measure as shown in Table 4). Hence, the number of discordant pairs are found from $200(= 50 \times 4)$ comparisons of the results of the algorithms. The null hypothesis ($H_{0,i}$) is assumed as the two algorithms being compared (i.e., aDECOR/fDECOR with the $i^{\text{th}}$ algorithm listed in Table 4) to have equal tendencies to approximate Pareto-front. The hypothesis $H_{0,i}$ is rejected when the test statistics is greater than $\chi_{1,0.005}^2 = 3.84$ (critical value for one degree of freedom and 95% confidence interval) validating that the tendencies to approximate the Pareto-front are significantly different. This test gives a more detailed insight about DECOR outperforming specific competitor algorithms.

$$\chi_M^2 = \frac{(|n_{AB} - n_{BA}| - 1)^2}{(n_{AB} + n_{BA})} \tag{12}$$

#### 5.5.3. Holm-Bonferroni test

This is a post-hoc test which is performed to counteract the problem of multiple comparisons. It controls the family-wise error rate by applying Bonferroni corrections to the significance level of each individual hypotheses. The $p$-values of the algorithms with which aDECOR/fDECOR have been compared, are listed under the McNemar's test in Table 4. The hypotheses, within each family, are ranked/sorted from lowest to highest $p$-values, under this test, as given by $R_{H,i}$ in Table 4. The adjusted significance level ($\alpha'$) for 95% confidence interval is given by Eq. (13) and noted in Table 4. All the hypotheses are rejected which have $p$-values smaller than the corresponding $\alpha'$, the rest of the hypotheses are not rejected.

$$\alpha' = \frac{0.05}{(k_a + 1 - R_{H,i})} \tag{13}$$

## 6. Discussion

In the previous Sections, DECOR has been proposed and is applied to DTLZ problems. The performance of DECOR is evaluated in terms of convergence metric and hypervolume indicator, and the results are also visualized using the parallel

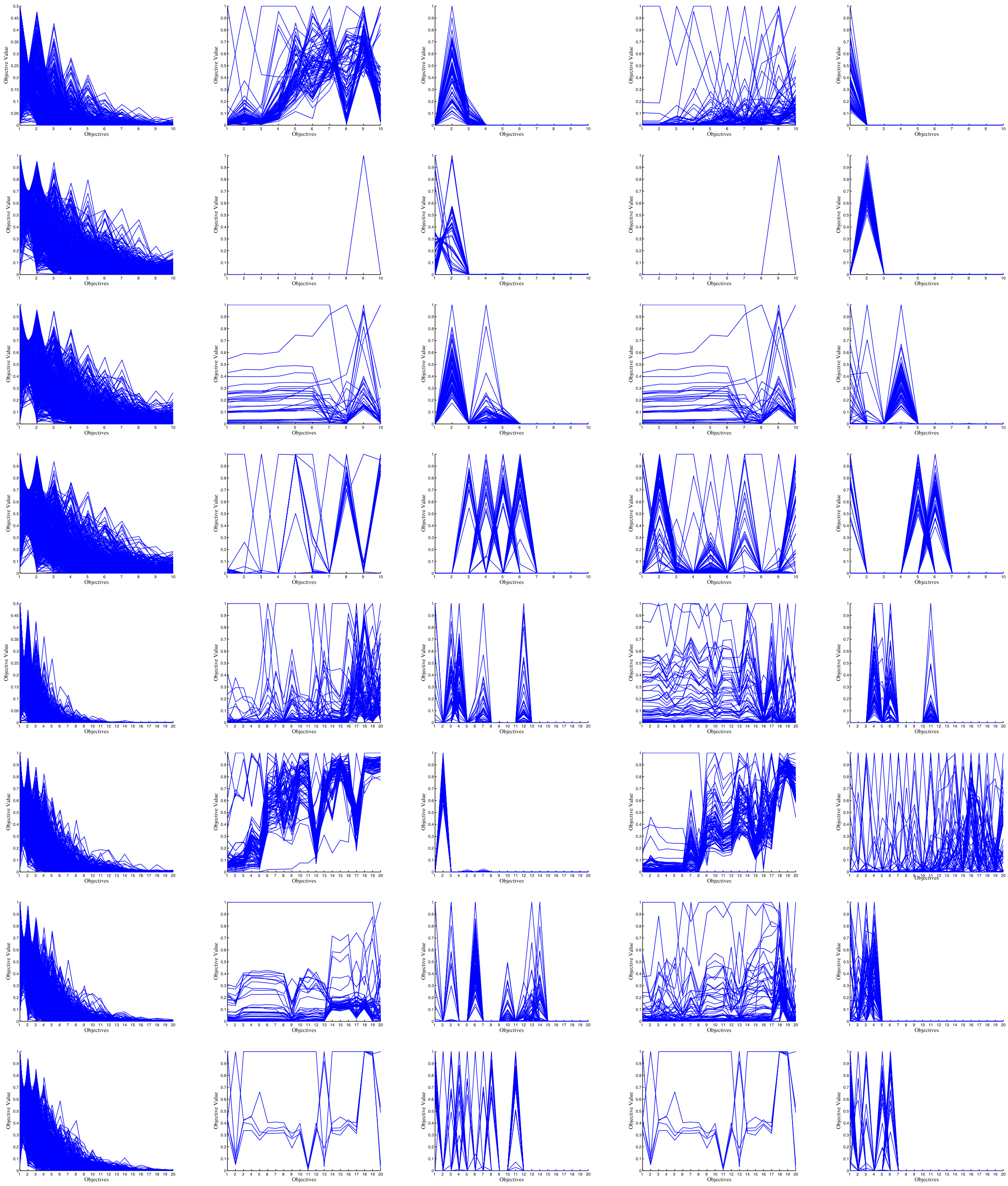

**Fig. 4.** Parallel coordinates. First four rows indicate DTLZ1 to DTLZ4 problems with 10 objectives, and the other four rows indicate the same for 20 objectives. First column shows true Pareto-optimal fronts, the second and the third column show estimated Pareto-optimal fronts by $\alpha$-DEMO-revised and aDECOR approaches using parameters from Table 2, and the fourth and the last column show estimated Pareto-optimal fronts by $\alpha$-DEMO-revised and fDECOR using parameters from Table 3.

**Table 4**
Parameters and results of Friedman Test (FT), McNemar's Test (MNT) and Holm-Bonferroni Test (HBT).

| Optimization Algorithms | (a) aDECOR vs. others | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Convergence Metric | | | | | | Hypervolume Indicator | | | | | |
| | 10 objectives | | | 20 objectives | | | 10 objectives | | | 20 objectives | | |
| | FT[a] | MNT[b] | HBT[c] | FT | MNT | HBT | FT | MNT | HBT | FT | MNT | HBT |
| **NSGA-II** | 5.50 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 5.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 4.50 | 150, 50, 49.005 $< 10^{-5}$, R | 1, 0.01, R | 3.75 | 100, 100, 0.005, 0.94363, A | 2, 0.0125, A |
| **MOEA/D** | 2.50 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 2.75 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 1.75 | 100, 100, 0.005, 0.94363, A | 2, 0.0125, A | 2.00 | 100, 100, 0.005, 0.94363, A | 2, 0.0125, A |
| **HypE** | 4.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 4.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 3.75 | 150, 50, 49.005, $< 10^{-5}$, R | 1, 0.01, R | 4.00 | 150, 50, 49.005 $< 10^{-5}$, R | 1, 0.01, R |
| **DEMO** | 4.50 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 5.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 3.50 | 100, 100, 0.005, 0.94363, A | 2, 0.0125, A | 3.75 | 100, 100, 0.005, 0.94363, A | 2, 0.0125, A |
| **$\alpha$-DEMO-revised** | 3.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 3.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.01, R | 3.50 | 160, 40, 70.805, $< 10^{-50}$, R | 1, 0.01, R | 3.00 | 140, 60, 31.205, $< 10^{-50}$, R | 1, 0.01, R |
| **aDECOR** | 1.00 | – | – | 1.00 | – | – | 2.75 | – | – | 3.00 | – | – |
| $\chi^2_{5,0.05}$, $H_0$ (FT) | 14.71, R | – | – | 13.86, R | – | – | −4.64, A | – | – | −8.43, A | – | – |
| | (b) fDECOR vs. others | | | | | | | | | | | |
| **NSGA-II** | 6.75 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 6.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 4.50 | 100, 100, 0.005, 0.94363, A | 4, 0.0167, A | 4.50 | 150, 50, 49.005, $< 10^{-5}$, R | 1, 0.0083, R |
| **MOEA/D** | 4.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 3.75 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 2.50 | 100, 100, 0.005, 0.94363, A | 4, 0.0167, A | 2.50 | 100, 100, 0.005, 0.94363, A | 3, 0.0125, A |
| **HypE** | 5.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 5.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 4.75 | 150, 50, 49.005, $< 10^{-5}$, R | 1, 0.0083, R | 4.75 | 145, 55, 39.605, $< 10^{-5}$, R | 1, 0.0083, R |
| **DEMO** | 5.75 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 6.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 4.25 | 100, 100, 0.005 0.94363, A | 4, 0.0167, A | 4.50 | 150, 50, 49.005, $< 10^{-5}$, R | 1, 0.0083, R |
| **$\alpha$-DEMO-revised** | 3.25 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 3.75 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 3.75 | 95, 105, 0.405 0.52452, A | 3, 0.0125, A | 4.00 | 140, 60, 31.205, $< 10^{-5}$, R | 1, 0.0083, R |
| **Approach in [30]** | 2.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 2.00 | 200, 0, 198.005, $< 10^{-5}$, R | 1, 0.0083, R | 3.50 | 120, 80, 7.605, 0.00582, R | 2, 0.01, R | 3.75 | 125, 75, 12.005, 0.00053, R | 2, 0.01, R |
| **fDECOR** | 1.00 | – | – | 1.00 | – | – | 3.75 | – | – | 2.50 | – | – |
| $\chi^2_{6,0.05}$, $H_0$ (FT) | 22.07, R | – | – | 20.36, R | – | – | −3.86, A | – | – | −5.46, A | – | – |

[a] For Friedman Test (FT): Average Ranks ($R_{F,i}$).
[b] For McNemar's Test (MNT): $n_{AB}$, $n_{BA}$, $\chi^2_M$, $p$-value, Acceptance(A) or Rejection(R) of $H_{0,i}$ at 95% confidence interval.
[c] For Holm-Bonferroni Test (HBT): Rank ($R_{H,i}$), Bonferroni corrected significance level ($\alpha'$) for 95% confidence interval, Acceptance(A) or Rejection(R) of $H_{0,i}$.

coordinates while comparing it with other popular MaOO approaches. A discussion on the performance analysis of DECOR and the key features of this work are presented in this Section. The major observations from the results reported in Section 5 are as follows:

- *Advantage of Thresholding while Clustering:* The threshold ($th$) values (in Tables 2 and 3) determine the course of action in presence of singleton cluster. Such a thresholding has been introduced, for the first time, in DECOR. As a result, unlike the work in [30], the objective reduction is not stuck due to appearance of singleton cluster, rather more number of objectives are reduced leading to reduced number of objective computation. The reduced number of objectives ($m$) in Table 2, automatically determined for aDECOR, are less than those in Table 3, automatically determined for [30] (reused as input for fDECOR).
- *Disadvantage of Thresholding while Clustering:* As seen from the results, for small changes in $m$ and $M$ values, the corresponding $th$ value changes. Hence, for determining the appropriate $th$ for practical problems, repeated trials have to be done such that performance measures which are independent of the true Pareto-front (like hypervolume indicator) are acceptably high and the Pareto-front is satisfactory, when visualized through tools like parallel coordinates.
- *Resulting Convergence by DECOR:* The convergence metric of aDECOR and fDECOR are presented in Tables 2 and 3, respectively, where the best values are written in bold. It is observed that both aDECOR and fDECOR clearly outperform all the other competitor algorithms considered. The superior performance of DECOR in terms of convergence metric is also statistically validated in Table 4. From the parallel coordinates (Fig. 4), it can be observed that along several objectives DECOR has converged better than $\alpha$-DEMO-revised. However, in several cases, the solutions are accumulated in a region on the Pareto-front surface. Also, Pareto-fronts obtained by DECOR are far worse from being identical to the true Pareto-front and thereby, indicating that further scope of improvement in convergence exists.
- *Resulting Diversity by DECOR:* In Tables 2 and 3, DECOR is not declared as an outright winner among the competitors. However, it can be stated that as DECOR has superior convergence, smaller hypervolume indicates poorer diversity. In this aspect, a few observations on hypervolume are as follows:
  1. The dominated hypervolume in several cases in Tables 2 and 3 are noted to be zero. This can happen when the entire estimated Pareto-front lies outside the hyper-rectangle defined by the origin and the reference point in the objective space. DECOR, on the other hand, has attained non-zero hypervolume in all cases indicating better performance.
  2. DECOR beats the contemporary objective reduction based MaOO algorithms [2,30] in 6 out of 8 cases for aDECOR (Table 2) and in 4 out of 8 cases for fDECOR (Table 3).
  3. When DECOR has failed to outperform its competitors, the hypervolume is not much lesser than the highest value attained by other algorithms (Tables 2 and 3).

  Hence, it can be concluded that the performance of DECOR, in terms of diversity, is better than some of the competitor algorithms and equivalent to its other competitors. This equivalence is statistically validated, in Table 4, where $H_0$ has been accepted.

  Moreover, the estimated Pareto-front by DECOR is visually compared to that by $\alpha$-DEMO-revised through the parallel coordinates (Fig. 4). In all the cases, the result of DECOR does not exactly match with the ideal scenario (indicating poor estimation of true Pareto-front), but in most of the cases, DECOR has better diversity than $\alpha$-DEMO-revised and in few of the cases, the solutions constituting the Pareto-front have accumulated within a small region which resulted in poor diversity.

The key features of DECOR which help it to perform better than the other existing approaches are attributed to the integrated combination of the following:

- *Reason for objective reduction:* The proposed work uses objective reduction to reduce the complexity of the MaOO problem.
- *Reason for online objective reduction:* The online objective reduction strategy is considered as it is fast which searches for optima while reducing the number of objectives, and toggles between full and reduced objective set, at some intervals, in order to maintain balance exploration and exploitation of the search space. Also the proposed objective reduction approach has provision for elimination of multiple objectives at a time.
- *Reason for choosing IDEMO as the optimizer:* The proposed approach, DECOR, is designed such that any MOO/MaOO algorithm could be used in the optimizer module. However, owing to the following two reasons, IDEMO (customized version of DEMO) is chosen as the optimization method in the proposed algorithmic framework. Firstly, the recent years have seen objective reduction based MaOO algorithm which uses DEMO [2,30]. Secondly, DEMO has shown its superior performance as compared to other optimization algorithms [33].
- *Reason for introducing the novel elitism to DEMO:* Although DEMO is a multi-objective optimization algorithm, it has been used in the improved elitist framework, in order to make the algorithm suitable as a MaOO algorithm. For overcoming the saturation of population by sub-optimal non-dominated solutions, the strategy of choosing a proportion of $R_1$ solution for the next population has previously been introduced in $\alpha$-DEMO-revised [2] and has been reused in [30]. Yet the existing method does not account for the three separate cases of proportions of $R_1$ and $R_{rest}$ solutions which is achieved by DECOR, as shown in Fig. 1c.
- *Reason for introducing the novel ranking to DEMO:* While the most popular approach uses ranking based on crowding distance i.e. diversity-based ranking, DECOR uses ranking based on combined information ($D_{comb}$) from crowding distance

and distance from ideal point i.e. convergence-based as well as diversity-based ranking. The superior performance of DECOR, in terms of convergence, can be attributed to this novel ranking strategy.

- *Reason for good diversity:* On one hand, crowding distance ($D_{crowd}$) prefers bordering solutions of the Pareto-front. On the other hand, distance from ideal point ($D_{ideal}$) prefers solutions towards the center of the Pareto-front which has been used to improve convergence. An interesting observation follows when ranking by $D_{ideal}$ is used along with objective reduction. Due to objective reduction, the local optimizer (Fig. 3a and b) operates on different subsets of the objectives. Thus, ranking of solution by $D_{ideal}$, not only helps in convergence along the center of the entire Pareto-front (global), but also along the center of those Pareto-fronts which are induced by the subset of objectives (local).

Overall performance of DECOR can be summarised as follows. Due to its efficient managing of singleton clusters, DECOR achieves a lower value of $m$ (number of objectives in reduced set) and thereby must require less number of objective computation than the recent approach proposed in [30]. Due to the new ranking scheme which combines crowding distance with distance from the ideal point, the convergence has improved as evident from all the test cases. As DECOR not only prefers the solutions that bound the Pareto-front but also other solutions that lie on the surface of the Pareto-front, it exhibits better diversity in most of the cases than other recent objective reduction based MaOO algorithms. However, the determination of appropriate $th$ (threshold for declaring singleton cluster as essential objective) for satisfactory results by repeated trials is not an user-friendly strategy. Moreover, as the ranking scheme improves diversity by improving the convergence of induced Pareto-front (not by improving any true diversity measure), it has failed to enhance diversity in a few of the cases.

## 7. Conclusions

Given the popularity and the need for many-objective optimization algorithms across multiple domains, robust MaOO algorithms are often being designed with minor improvement over existing algorithms. In this work, IDEMO, with revised elitist selection and ranking scheme, is used in order to improve the selection pressure, convergence and diversity of the solutions. The proposed approach (DECOR) integrates IDEMO in a fast and online objective reduction framework with provision for elimination of multiple objectives in a turn. DECOR is applied on DTLZ problems for 10 and 20 objectives and results are noted in terms of convergence metric and hypervolume indicator. DECOR shows superior convergence to Pareto-front as compared to several other algorithms. The diversity of the Pareto-front resulting from DECOR, is better than some of the MaOO approaches and is equivalent to a few other popular MaOO approaches. DECOR not only outperforms the recent objective reduction based MaOO approaches but also overcomes their drawbacks.

In future, the performance of DECOR has to be tested on other benchmark problems. Also, as different performance metrics capture different information, performance in terms of other metrics are yet to be analysed. Moreover, applying DECOR to real-life application problems will validate the practical applicability of the algorithm. The biggest drawback of DECOR is that there is no user-friendly way of determining a suitable value of $th$ (threshold for declaring singleton cluster as essential objective), except by trial and error method which is very tedious. Moreover, $th$ is not only problem-specific but also a small change in the size of full ($M$) or reduced ($m$) objective set results in a different $th$. Hence, in future, it is essential to develop a strategy in order to make the parameter $th$ adaptive such that it is automatically determined depending on the application. Another important observation is that there is huge scope of improvement in terms both convergence and diversity as seen from the parallel coordinates plots. Thus, further studies and extensions of DECOR along the mentioned directions are open for research.

## References


[1] J. Bader, E. Zitzler, Hype: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[2] S. Bandyopadhyay, A. Mukherjee, An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution, Evol. Comput. IEEE Trans. 19 (3) (2015) 400–413.
[3] X. Blasco, G. Reynoso-Meza, E.A.S. Prez, J.V.S. Prez, Asymmetric distances to improve n-dimensional Pareto fronts graphical analysis, Inf. Sci. 340 (2016) 228–249, doi:10.1016/j.ins.2015.12.039.
[4] D. Brockhoff, E. Zitzler, Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization, in: Parallel Problem Solving from Nature-PPSN IX, Springer, 2006, pp. 533–542.
[5] S. Chand, M. Wagner, Evolutionary many-objective optimization: a quick-start guide, Surv. Oper. Res. Manage. Sci. 20 (2) (2015) 35–42.
[6] C.A.C. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, 242, Springer, 2002.
[7] D. Corne, J. Knowles, M. Oates, The Pareto envelope-based selection algorithm for multiobjective optimization, in: Parallel Problem Solving from Nature PPSN VI, Springer, 2000, pp. 839–848.
[8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Evol. Comput. IEEE Trans. 6 (2) (2002) 182–197, doi:10.1109/4235.996017.
[9] K. Deb, 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), 2006, pp. 3352–3360.
[10] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[11] N. Drechsler, R. Drechsler, B. Becker, Multi-objective optimisation based on relation favour, in: Evolutionary Multi-criterion Optimization, Springer, 2001, pp. 154–166.
[12] K. Elsayed, C. Lacor, Cfd modeling and multi-objective optimization of cyclone geometry using desirability function, artificial neural networks and genetic algorithms, Appl. Math. Model. 37 (8) (2013) 5680–5704.
[13] T. Erlebach, H. Kellerer, U. Pferschy, Approximating multiobjective knapsack problems, Manage. Sci. 48 (12) (2002) 1603–1612, doi:10.1287/mnsc.48.12.1603.445.
[14] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1) (1995) 1–16.
[15] W.F. Gao, S.Y. Liu, L.L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, Inf. Sci. 270 (2014) 112–133, doi:10.1016/j.ins.2014.02.104.

[16] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184, doi:10.1016/j.ins.2012.08.023.
[17] Z. He, G. Yen, J. Zhang, Fuzzy-based Pareto optimality for many-objective evolutionary algorithms, Evol. Comput. IEEE Trans. 18 (2) (2014) 269–285, doi:10.1109/TEVC.2013.2258025.
[18] Z. He, G.G. Yen, An improved visualization approach in many-objective optimization, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 1618–1625.
[19] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. (1979) 65–70.
[20] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008, pp. 2419–2426, doi:10.1109/CEC.2008.4631121.
[21] A.L. Jaimes, C.A.C. Coello, J.E.U. Barrientos, Online objective reduction to deal with many-objective problems, in: Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 423–437.
[22] A.L. Jaimes, A. Oyama, K. Fujii, Space trajectory design: analysis of a real-world many-objective optimization problem, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE, 2013, pp. 2809–2816.
[23] L. Kaufman, P.J. Rousseeuw, Partitioning around medoids (program pam), in: Finding Groups in Data: An Introduction to Cluster Analysis, 1990, pp. 68–125.
[24] V. Khare, X. Yao, K. Deb, Performance scaling of multi-objective evolutionary algorithms, in: Evolutionary Multi-Criterion Optimization, Springer, 2003, pp. 376–390.
[25] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika 12 (2) (1947) 153–157.
[26] M.W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, M. Ó Cinnéide, High dimensional search-based software engineering: finding tradeoffs among 15 objectives for automating software refactoring using nsga-iii, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, 2014, pp. 1263–1270.
[27] S. Otake, T. Yoshikawa, T. Furuhashi, Basic study on aggregation of objective functions in many-objective optimization problems, in: World Automation Congress (WAC), 2010, IEEE, 2010, pp. 1–6.
[28] M. Pal, S. Bandyopadhyay, Many-objective feature selection for motor imagery eeg signals using differential evolution and support vector machine, in: IEEE International Conference on Microelectronics, Computing and Communication, MicroCom 2016, 2016, pp. 1–6.
[29] M. Pal, S. Bandyopadhyay, Reliability of convergence metric and hypervolume indicator for many-objective optimization, in: Control, Instrumentation, Energy & Communication (CIEC), 2016 2nd International Conference on, IEEE, 2016, pp. 511–515, doi:10.1109/CIEC.2016.7513806.
[30] M. Pal, S. Saha, S. Bandyopadhyay, Clustering based online automatic objective reduction to aid many-objective optimization, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 1131–1138, doi:10.1109/CEC.2016.7743915.
[31] C.H. Papadimitriou, M. Yannakakis, On the approximability of trade-offs and optimal access of web sources, in: Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on, IEEE, 2000, pp. 86–92.
[32] Z. Qingfu, L. Hui, Moea/d: a multiobjective evolutionary algorithm based on decomposition, Evol. Comput. IEEE Trans. 11 (6) (2007) 712–731, doi:10.1109/TEVC.2007.892759.
[33] T. Robi, B. Filipi, Demo: differential evolution for multiobjective optimization, in: C. Coello Coello, A. Hernndez Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 3410, Springer Berlin Heidelberg, 2005, pp. 520–533, doi:10.1007/978-3-540-31880-4_36.
[34] A. Sülflow, N. Drechsler, R. Drechsler, Robust multi-objective optimization in high dimensional spaces, in: Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 715–726.
[35] E. Zitzler, M. Laumanns, L. Thiele, Spea2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Evolutionary Methods for Design, Optimization, and Control, 2002, pp. 95–100.