Applied Soft Computing Journal 74 (2019) 274–290

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

MCEDA: A novel many-objective optimization approach based on model and clustering

Xiaoxu Duan

Central South University, Changsha, Hunan 410075, China

HIGHLIGHTS

- Propose a new framework of estimation of distribution algorithms (EDAs), MCEDA.
- Propose a new promising estimation model, the bits model.
- Extend the application of EDAs to solve many-objective optimization problems.
- Combine clustering and decomposition strategy in evolutionary algorithms.

ARTICLE INFO

Article history:
Received 10 September 2017
Received in revised form 26 September 2018
Accepted 19 October 2018
Available online 24 October 2018

Keywords:
Many-objective optimization
Estimation of distribution algorithm
Clustering
Evolutionary algorithm

ABSTRACT

To solve many-objective optimization problems (MaOPs) by evolutionary algorithms (EAs), the maintenance of convergence and diversity is essential and difficult. Improved multi-objective optimization evolutionary algorithms (MOEAs), usually based on the genetic algorithm (GA), have been applied to MaOPs, which use the crossover and mutation operators of GAs to generate new solutions. In this paper, a new approach, based on decomposition and the MOEA/D framework, is proposed: model and clustering based estimation of distribution algorithm (MCEDA). MOEA/D means the multi-objective evolutionary algorithm based on decomposition. The proposed MCEDA is a new estimation of distribution algorithm (EDA) framework, which is intended to extend the application of estimation of distribution algorithm to MaOPs. MCEDA was implemented by two similar algorithm, MCEDA/B (based on bits model) and MCEDA/RM (based on regular model) to deal with MaOPs. In MCEDA, the problem is decomposed into several subproblems. For each subproblem, clustering algorithm is applied to divide the population into several subgroups. On each subgroup, an estimation model is created to generate the new population. In this work, two kinds of models are adopted, the new proposed bits model and the regular model used in RM-MEDA (a regularity model based multi-objective estimation of distribution algorithm). The non-dominated selection operator is applied to improve convergence. The proposed algorithms have been tested on the benchmark test suite for evolutionary algorithms (DTLZ). The comparison with several state-of-the-art algorithms indicates that the proposed MCEDA is a competitive and promising approach.

1. Introduction

Evolutionary multi-objective optimization (EMO) algorithms have performed very well in different two and three objective optimization problems, while the theory proof of its convergence and diversity is insufficient [1,2]. Many engineering optimization problems have been solved with the help of multi-objective optimization evolutionary algorithms (MOEAs) [3–6]. For example, during designing a certain construction, the civil engineer may seek after the minimal total cost, the minimal danger coefficient, the maximal available space and the maximal stiffness.

However, when it comes to many-objective optimization problems (MaOPs), it is difficult to maintain the convergence and diversity. Most MOEAs are based on the selection directed by domination relation. As Kalyanmoy Deb et al. say, the proportion of non-dominated solutions in the population tends to be very large which will slow down the search process considerably as the number of objectives increases [7]. This will make the non-domination based algorithms extremely confused where to go since most algorithms rely on the domination relation among solutions to guide the search process. Also many evolution algorithms cannot maintain

E-mail address: xiaoxuduan@outlook.com.

https://doi.org/10.1016/j.asoc.2018.10.039
1568-4946/© 2018 Elsevier B.V. All rights reserved.

enough diversity, faced with MaOPs, which may result in local optimal solutions or even wrong answer.

While most EAs simply adopt the traditional genetic operators (e.g., crossover and mutation), estimation of distribution algorithms (EDAs), trying to generate the new solutions more reasonably, direct the search process of the Pareto fronts by building explicit probabilistic models of the promising candidate solutions and generating new solutions from those estimation models [8]. Although EDAs have successfully solved many problems, they are mainly used to handle problems with discrete variables and low number of objectives [9]. As Luis Marti et al. point out, the traditional EDAs have several shortcomings: handling the data outliers in a wrong way, losing diversity and too complex to compute [10].

In this paper, a new approach (MCEDA), utilizing estimation models and clustering strategy, is proposed to deal with many-objective optimization problems. MCEDA is based on decomposition and the MOEA/D framework [11,12]. It mainly focuses on unconstrained MaOPs at present. In MCEDA, the problem is decomposed into several subproblems. To deal with each subproblem, the population is divided into several clusters and the estimation model is built for each cluster. Then new solutions are generated from the estimation models. Non-dominated selection, which has been used in non-dominated sorting genetic algorithm III (NSGA-III) [7,13], is adopted to select the candidate solutions from all new solutions. The candidate solutions are compared with the solutions of the current population to determine whether the current solutions should be replaced or not so as to update the whole population. Since the search progress is directed by both decomposition and estimation models, the data outliers can be handled and the population diversity is maintained well.

The main new contributions of this paper are listed as follows.

(1) Propose a new EDA framework, MCEDA. The traditional EDA framework may be too ideal when it comes to complex problems, e.g. MaOPs.
(2) Propose a new estimation model, the bits model. It is a promising model according to the experiments.
(3) Extend the application of EDAs to MaOPs. A few attempts have been made to applied the EDAs to MaOPs. In this work, the new proposed EDAs prove to be qualified to solve the MaOPs.
(4) Combine clustering and decomposition strategy. In this paper, the new proposed EDA framework utilizes both the clustering and decomposition procedure, which has not been tried.

The remaining of this paper is organized as follows. In Section 2, the related work and some background concepts are described. Section 3 presents the MCEDA in detail. Experimental studies with parameter studies and comparison to other algorithms are reported in Section 4. And finally, Section 5 presents the conclusions.

2. Related work and preliminaries

In this section, current research about MaOPs, EDAs and decomposition are introduced. And for the sake of completeness, some background concepts are presented.

2.1. Many-objective optimization

An unconstrained multi-objective optimization problem (MOOP) can be defined as follows [14]. And many-objective optimization problems (MaOPs) are defined as problems with four or more objectives [7].

$$\begin{aligned} &\text{Minimize } f_j(\boldsymbol{x}), && j = 1, 2, \ldots, m; \\ &\text{subject to } x_i^{(L)} \le x_i \le x_i^{(U)}, && i = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

A solution $\boldsymbol{x}$ is a vector of n decision variables: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$. The corresponding objective value is $\boldsymbol{f} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$. The last set of constraints are called variable bounds. The decision space is formed by all the variables constituted by the variable bounds.

A solution $\boldsymbol{x}^{(1)}$ is said to *dominate* the other solution $\boldsymbol{x}^{(2)}$, if both following conditions are true:

(1) $f_j(\boldsymbol{x}^{(1)}) \le f_j(\boldsymbol{x}^{(2)})$ for all $j = 1, 2, \ldots, m$.
(2) $f_j(\boldsymbol{x}^{(1)}) < f_j(\boldsymbol{x}^{(2)})$ for at least one $j \in 1, 2, \ldots, m$.

Among a set of solutions $\boldsymbol{P}$, the non-dominated set of solution $\boldsymbol{P}$ are those that are not dominated by any member of the set $\boldsymbol{P}$.

MOEAs have been widely used in real-world applications [3–6]. However, most domination based MOEAs find difficulties in solving many-objective optimization problems, since the selection ability is reduced considerably with rising number of objectives [15,7,16].

To deal with MaOPs, a number of many-objective evolutionary algorithms (MaOEAs) have been proposed. Bingdong Li et al., according to the key operator, categorized MaOEAs into seven classes [2]: relaxed dominance based, diversity-based, aggregation-based, indicator-based, reference set based, preference-based and dimensionality reduction approaches. Hernán Aguirre and Kiyoshi Tanaka proposed a method to search effectively on MaOPs by partitioning the objective space into subspaces and they used an adaptive ϵ-ranking procedure to re-rank solution [17]. Convergence and diversity of the solutions can be improved remarkably on 4 to 10 objectives. Hisao Ishibuchi et al. examined two different algorithms: NSGA-II [1] and MOEA/D [11]. According to their computational experiments, when it comes to large populations, NSGA-II showed inefficient while MOEA/D performed well [18]. Xiaofang Guo et al. employed PAM clustering algorithm to identify redundant objectives [19]. Non-dominated sorting genetic algorithm III (NSGA-III), proposed by Kalyanmoy Deb et al., is a reference-point-based MaOEA based on NSGA-II framework, which uses reference points and clustering operator to maintain the population diversity and convergence [7]. And constrained non-dominated relation is applied in constrained MaOPs in their subsequent work [20]. Clustering-ranking evolutionary algorithm (crEA) implements clustering and ranking sequentially to improve convergence and diversity [21]. Xinye Cai et al. proposed a new variant of MOEA/D with sorting-and-selection (MOEA/D-SAS) in which decomposition-based-sorting (DBS) and angle-based-selection (ABS) was used to maintain the balance between convergence and diversity [22]. In their experiments, MOEA/D-SAS performed very well. Huanlai Xing et al. proposed a modified multi-objective evolutionary algorithm based on decomposition (MOEA/D-PBIL) [23], which is based on the multi-objective evolutionary algorithm based on decomposition (MOEA/D) and the population based incremental learning (PBIL) [24]. The MOEA/D-PBIL is intended to solve a specific multi-objective optimization problem in the context of multicasting with network coding and the problem have three objectives. Jing Bai et al. proposed a multi-objective artificial bee algorithm (MOABC) which is based on decomposition by PBI method [25]. They apply artificial bee colony algorithm to enhance the performance and the experimental results show that MOABC outperforms others. Yuyan Han et al. proposed an evolutionary multi-objective robust scheduling algorithm (REMO) to deal with the flow shop scheduling problems in the presence of machine breakdowns [26]. Their experimental results show that REMO can tackle the blocking lot-streaming flow shop scheduling problems very well. Xiangwei Zheng et al. proposed a novel multi-objective group search optimizer based on multiple producers and crossover operator of genetic algorithm (MCGSO) [27]. Experimental results show that MCGSO is effective and efficient. Junqing Li et al. proposed an energy-aware multi-objective optimization algorithm (EA-MOA) to solve hybrid flow shop (HFS) scheduling problem when the setup energy consumptions are considered [28]. The experimental results show that EA-MOA is highly effective, compared with other efficient algorithms.

2.2. Estimation of distribution algorithms

As mentioned before, most MaOEAs are based on genetic algorithms (GAs). GAs are optimization methods based on the mechanics of artificial selection and genetic recombination operators [9], while estimation of distribution algorithms (EDAs) based optimization algorithms direct the search of the optimum by building and sampling explicit probabilistic models of promising candidate solutions [8]. The estimation model is used to capture the probability distribution of the promising, which is the most important feature of EDAs [29]. However, it is not intended to exactly represent the population, which could be very time consuming and almost impossible, but instead to build an estimation model that can generate better solutions. The basic procedure of traditional EDA is described in Algorithm 1 [8].

Algorithm 1. Basic procedure of traditional EDA

```
$t \leftarrow 0$;
Generate population $P(0)$ of random solutions;
while termination criteria not satisfied do
    Evaluate all candidate solutions in $P(t)$;
    Select promising solutions $S(t)$ from $P(t)$;
    Build a probabilistic model $M(t)$ for $S(t)$;
    Generate new solutions $O(t)$ by sampling $M(t)$;
    Create $P(t+1)$ by combining $O(t)$ and $P(t)$;
    $t \leftarrow t+1$;
return
```

Many traditional EDAs are focus on solving problems with discrete variables (e.g., Bayesian network based EDAs [30–34,9,35,36]). However, solutions for many problems are represented by continuous variables. According to Mark Hauschild and Martin Pelikan's survey, there are two primary approaches to applying EDAs to problems with continuous variables: map the continuous variables into discrete domain and use a discrete EDA to solve it or use EDAs designed for continuous variables [29]. The selection method also has an important impact on the performance of EDAs. J.R. Caldwell et al. analyzed two different selection method used in EDAs: filter selection and constructive selection [37]. And the results show that constructive selection can significantly improve the EDAs' performance. Marcella S. R. Martins et al. proposed the heuristic selection based on estimation of distribution algorithm (HSEDA), which considers EDA as a heuristic selection technique [38]. The HSEDA performs well at the traveling thief problem in their experiments. Tudor Dan Mihoc proposed a new estimation of distribution algorithm, which is intended to detect a sample of Nash equilibria for games [39]. Murilo Zangari et al. proposed a multi-objective decomposition-based EDA using Kernels of Mallows models (MEDA/D-MK) to deal with multi-objective permutation-based optimization problems [40]. MEDA/D-MK outperforms others in their experiments. Fergal Lane et al. introduced a bivariate model for discrete EDAs, based on dichotomized Gaussian models (DICE) [41]. And DICE turns out to be competitive compared with other EDAs in their experiments.

Many EDAs have emerged in the past several decades [42,34,9,43,35,36]. RM-MEDA, a regularity model based multi-objective estimation of distribution algorithm, employs the local principal component analysis algorithm to build the estimation model at each generation [44]. A non-dominated sorting based selection is used for choosing solutions for the next generation [44]. Li Mo et al. employed elitist strategy theory to improve RM-MEDA, using part of the parent population with better performance instead of the entire parent population to establish a more accurate manifold [45]. Yangyang Li et al. introduced local learning to improve the local search ability of RM-MEDA, which obtained much better convergence capability according to their experiments [46]. The Bayesian optimization algorithm (BOA), proposed by Martin Pelikan et al., estimates the distribution of promising solutions by Bayesian network to generate new candidate solutions, which mainly used to solve problems with discrete variables [47,34]. Other models have also been used. The naive MIDEA, proposed by Peter A.N. Bosman and Dirk Thierens, performed well in their experiments. C-Multi, a competent multi-swarm approach for many-objective problems, is based on a particle swarm optimization (PSO) algorithm and an EDA [48]. The PSO algorithm is applied to discover the Pareto front and divide different regions. The EDA used in C-Multi is the real-coded Bayesian optimization algorithm (rBOA) [43], focusing on convergence to its allocated region. It performed very well in their experiments. Yanan Sun et al. proposed a reference line-based estimation of distribution algorithm for many-objective optimization problems (MaOEDA/RL) /citeRN3270. MaOEDA/RL samples new solutions with the help of generated uniformly distributed reference points and performs well in their experiments. Luis Marti et al. proposed a scalable multi-objective optimization with a neural network-based estimation of distribution algorithm (MONEDA), which utilizes a modified growing neural gas network for model-building (MB-GNG) [49,50]. MONEDA performs better than others in their experiments.

2.3. Decomposition

MOEA/D, proposed by Qingfu Zhang et al., uses decomposition strategy to decompose MOOPs [11]. In their paper, three approaches to decompose MOOPs are introduced: weighted sum approach, Tchebycheff approach and boundary intersection (BI) approach. In this paper, the proposed MCEDA is based on MOEA/D, in which the Tchebycheff approach is adopted. In the Tchebycheff approach, the original problem is decomposed into N subproblems as follows [11].

$$\begin{aligned} &\text{minimize } g(\boldsymbol{x}|\boldsymbol{\lambda}^j, \boldsymbol{z}^*);\\ &\qquad g(\boldsymbol{x}|\boldsymbol{\lambda}^j, \boldsymbol{z}^*) = \max_{1\le i\le m}\{\lambda_i^j|f_i(x) - z_i^*|\}; \qquad (2)\\ &\text{subject to } \boldsymbol{x} \in \Omega. \end{aligned}$$

where m is the number of objectives of the problem and $\boldsymbol{z}^* = (z_1^*, \ldots, z_m^*)^T$ is the ideal point, i.e., $z_i^* = min\{f_i(x)|x \in \Omega\}$ for each $i = 1, \ldots, m$. $\boldsymbol{\lambda}^j = (\lambda_1^j, \lambda_2^j, \ldots, \lambda_m^j)^T, j = 1, 2, \ldots, N$, is the weight vector. For each Pareto optimal solution $\boldsymbol{x}^*$ of Eq. (1), there is a corresponding weight vector λ^j and vice versa. Therefore, different Pareto optimal solutions can be obtained by solving the single objective optimization problems generated by varying the weight vectors [12]. This is how the Tchebycheff approach works.

3. Proposed algorithm

3.1. Framework of the proposed algorithm

The framework of the proposed MCEDA is summarized in Algorithm 2, which is different from most other EDAs. To begin with, the problem is decomposed into several subproblems [11,12]. The initial population P_0, whose size is assigned with the number of the size of weights vector, are randomly generated and the ideal point $\boldsymbol{z}^*$ are initialized. Then the normalization is applied to the population. For a solution $\boldsymbol{x}$, the normalized objective value is defined as [21]

$$\tilde{f}_j(\boldsymbol{x}) = \frac{f_j(\boldsymbol{x}) - z_j^*}{z_j^{max} - z_j^*}, j = 1, 2, \ldots, m \qquad (3)$$

Table 1
Mean and standard deviation values of IGD, obtained by MCEDA/B on DTLZ instances.

Crossover parameter	Obj.	0.5	0.1	0.8
DTLZ1	3	$2.76E-04_{8.00E-06}$	$2.79E-03_{5.80E-03}(-)$	$2.67E-04_{1.50E-05}(+)$
	5	$2.63E-03_{1.90E-06}$	$1.11E-02_{5.50E-03}(-)$	$2.63E-03_{2.60E-06}(-)$
	8	$1.62E-03_{5.00E-05}$	$9.47E-03_{4.80E-03}(-)$	$1.64E-03_{6.40E-05}(-)$
	10	$3.05E-03_{2.50E-04}$	$2.97E-03_{3.30E-04}(+)$	$3.58E-03_{8.20E-04}(-)$
	15	$6.45E-03_{3.50E-03}$	$1.19E-02_{5.40E-03}(-)$	$4.65E-03_{1.50E-03}(+)$
DTLZ2	3	$4.12E-04_{3.90E-05}$	$4.28E-04_{3.50E-05}(-)$	$4.22E-04_{3.40E-05}(-)$
	5	$1.49E-03_{8.30E-05}$	$1.97E-03_{4.20E-04}(-)$	$1.58E-03_{1.50E-04}(-)$
	8	$4.63E-03_{9.70E-04}$	$4.65E-03_{3.40E-04}(-)$	$4.09E-03_{6.40E-04}(+)$
	10	$5.45E-03_{1.10E-03}$	$4.37E-03_{9.90E-04}(+)$	$5.45E-03_{1.30E-03}(-)$
	15	$2.58E-03_{6.10E-04}$	$2.25E-03_{4.50E-04}(+)$	$3.16E-03_{7.80E-04}(-)$
DTLZ3	3	$8.65E-04_{5.80E-05}$	$2.14E-02_{7.90E-03}(-)$	$8.99E-04_{7.00E-05}(-)$
	5	$8.42E-03_{4.30E-03}$	$1.73E-02_{6.20E-03}(-)$	$4.93E-03_{3.70E-05}(+)$
	8	$9.10E-03_{1.50E-03}$	$1.05E-02_{3.20E-03}(-)$	$9.45E-03_{1.60E-03}(-)$
	10	$5.51E-03_{2.80E-03}$	$6.84E-03_{4.30E-03}(-)$	$4.97E-03_{3.00E-03}(+)$
	15	$1.90E-02_{6.80E-03}$	$2.46E-02_{8.60E-03}(-)$	$2.76E-02_{8.50E-03}(-)$
DTLZ4	3	$1.07E-02_{5.00E-03}$	$1.45E-02_{5.60E-03}(-)$	$8.17E-03_{7.90E-03}(+)$
	5	$1.08E-02_{3.80E-03}$	$1.11E-02_{4.00E-03}(-)$	$8.26E-03_{4.20E-03}(+)$
	8	$6.51E-03_{1.70E-03}$	$5.72E-03_{1.40E-03}(+)$	$7.34E-03_{9.50E-04}(-)$
	10	$1.03E-02_{9.60E-04}$	$8.06E-03_{5.60E-04}(+)$	$1.23E-02_{1.70E-03}(-)$
	15	$7.07E-03_{4.60E-04}$	$6.36E-03_{2.40E-04}(+)$	$8.10E-03_{4.70E-04}(-)$
DTLZ5	3	$1.77E-05_{1.40E-05}$	$2.51E-05_{1.50E-05}(-)$	$3.07E-05_{1.80E-05}(-)$
	5	$1.86E-03_{2.70E-04}$	$2.33E-03_{1.90E-04}(-)$	$1.88E-03_{3.00E-04}(-)$
	8	$2.22E-03_{2.60E-04}$	$2.20E-03_{1.30E-04}(+)$	$2.33E-03_{2.00E-04}(-)$
	10	$2.79E-03_{3.20E-04}$	$2.97E-03_{2.20E-04}(-)$	$3.06E-03_{4.00E-04}(-)$
	15	$1.07E-03_{1.10E-04}$	$1.02E-03_{6.40E-05}(+)$	$1.20E-03_{1.60E-04}(-)$
DTLZ6	3	$3.98E-05_{3.90E-05}$	$4.55E-05_{3.50E-05}(-)$	$3.91E-05_{2.40E-05}(+)$
	5	$2.14E-03_{1.60E-04}$	$2.66E-03_{2.50E-04}(-)$	$2.23E-03_{1.30E-04}(-)$
	8	$3.43E-03_{6.20E-04}$	$3.27E-03_{6.00E-04}(+)$	$3.25E-03_{9.80E-04}(+)$
	10	$5.11E-03_{1.20E-03}$	$4.70E-03_{7.70E-04}(+)$	$5.80E-03_{1.70E-03}(-)$
	15	$3.44E-03_{6.30E-05}$	$3.43E-03_{6.00E-05}(+)$	$3.37E-03_{2.70E-04}(+)$
DTLZ7	3	$1.65E-03_{3.10E-03}$	$4.64E-03_{5.20E-03}(-)$	$1.65E-03_{3.20E-03}(-)$
	5	$7.95E-03_{1.90E-03}$	$7.35E-03_{4.10E-03}(+)$	$6.25E-03_{3.60E-03}(+)$
	8	$7.74E-03_{1.30E-03}$	$5.22E-03_{8.00E-04}(+)$	$1.03E-02_{2.00E-03}(-)$
	10	$1.30E-02_{1.80E-03}$	$9.79E-03_{7.50E-04}(+)$	$1.40E-02_{1.20E-03}(-)$
	15	$1.08E-02_{3.70E-04}$	$7.99E-03_{1.60E-04}(+)$	$1.20E-02_{1.50E-03}(-)$

The symbol "+" and "−" denote whether the IGD results of the algorithm with the compared parameter are better than or worse than that of the algorithms with the first parameter value according to the mean value and standard deviation.

where z_j^{max} is the maximum value for every objective f_j in the population. The weight vectors are created recursively under the conditions as follows.

$$\begin{aligned} &\boldsymbol{\lambda}^j = (\lambda_1^j, \lambda_2^j, \ldots, \lambda_m^j)^T, j = 1, 2, \ldots, N; \\ &\sum_{i=1}^{m} \lambda_i^j = 1, \lambda_i^j \in [0, 1]. \end{aligned} \tag{4}$$

Each weight vector λ^j is related to a subproblem and a solution S which has the best fitness for the subproblem in terms of the solutions have been evaluated and compared.

The fitness is represented as follows [12].

$$\text{fitness}(S) = \max_{1 \le i \le m} \{\lambda_i |f_i(x) - z_i^*|\} \tag{5}$$

The solution with the smaller fitness value is better.

For each subproblem, the neighborhood type Nt is randomly chose from *Neighbor* and *Public*, which will be used in the updating population procedure. The neighborhood relationship among all the subproblems is defined based on the distances of their weight vectors which is developed by Zhang et al. [11]. The current solution set P_t is normalized to generate Pn_t. Then the non-dominated solution set Ps_t is selected from Pn_t. Then Ps_t is divided into m clusters C_t according to the Euclidean distances between the objective values of the solutions by k means. For each cluster, a estimation model is built to generate new solutions, used to form Q_t. Here, a new proposed bits model and the regular model from RM-MEDA [44] are used to estimate the new solutions Section 3.3. During the model procedure, the generated solutions, whose variables exceed the variable bounds, will be dropped. And a new solution will be generated until the specified amount of new solutions is filled. To focus on the current subproblem, the new solutions Q_t are crossed with the current solution which relates to the current subproblem to obtain Qc_t. The uniform crossover operator from differential evolution is adopted [51]. In order to explore more decision space, Qc_t is mutated to get Qcm_t. The polynomial mutation is used here. And the non-dominated solutions $Qcms_t$ are selected from the mutated solutions Qcm_t. For each solution in $Qcms_t$, z^* is updated according to following principle, which have been described in Section 2.3. $\boldsymbol{z}^* = (z_1^*, \ldots, z_m^*)^T$ is the ideal point, i.e., $z_i^* = min\{f_i(x)|x \in \Omega\}$ for each $i = 1, \ldots, m$. Then, compared with the fitness of the non-dominated solutions, the current population P_t is updated to generate the next population P_{t+1}. Finally, the algorithm will terminate when the number of the evaluation reaches the max.

The reason why the normalization is used here is to make the algorithm more robust, i.e. 1) to avoid the too large or to small value of the objective function misleads the direction of the selection 2) and to avoid the too large value exceeds the scope the computer can hold. It would be better to add the normalization operator before line 21 in Algorithm 2 as well. However, thinking about the small amount of the new generated solutions and to make the algorithm more concise, the normalization operator is not added before line 21 in Algorithm 2.

The main difference between MCEDA and MOEA/D is the way to generate new solutions. While MOEA/D uses operators of the differential evaluation, MCEDA generate new solutions by clustering and modeling procedure.

Algorithm 2. The framework of MCEDA

```
Input: $P_r$: the problem; $N$: the size of population and weight
       vectors; $m$: the number of the objectives; $n$: the
       number of the variables of the problem; $MaxRSS$: the
       max RSS in k means; $MaxIterations$: the max iterations
       in k means;
$P_0 \leftarrow initializePopulation(N)$;
$P_0 \leftarrow evaluatePopulation(P_0)$;
Update evaluation number;
$z^* \leftarrow initializeIdealPoint()$;
$P_0 \leftarrow normalize(P_0)$;
$\lambda \leftarrow generateWeight(N)$;
$Nb \leftarrow initialNeighborhood(\lambda)$;
while termination criteria not satisfied do
    randomly permutate subproblem;
    foreach subproblem do
        $Id \leftarrow subproblemId$;
        $Nt \leftarrow RandomlyChooseNeighborType()$;
        $Pn_t \leftarrow normalize(P_t, z_t^*)$;
        $Ps_t \leftarrow nonDominatedSelect(Pn_t)$;
        $C_t \leftarrow generateClusters(Ps_t, m, MaxIterations, MaxRSS)$;
        $Q_t \leftarrow modeling(C_t, m|C_t|, n)$;
        $Qc_t \leftarrow crossover(Q_t, Id)$;
        $Qcm_t \leftarrow mutate(Qc_t)$;
        $Qcm_t \leftarrow evaluatePopulation(Qcm_t)$;
        Update evaluation number;
        $Qcms_t \leftarrow nonDominatedSelect(Qcm_t)$;
        foreach solution in $Qcms_t$ do
            $z_{t+1}^* \leftarrow updateIdealPoint(solution)$;
            $P_{t+1} \leftarrow$
            $updatePopulation(P_t, solution, Id, Nt, \lambda, z_{t+1}^*, Nb)$ ;
            // see Algorithm 3;
return
```

Algorithm 3. Update population [11]

```
Input: $P_t$: the previous population; $S$: the new solution; $Id$:
       the subproblem number; $Nt$: the neighborhood type;
       $\lambda$: the weight vectors; $z_{t+1}^*$: the updated ideal point;
       $Nb$: the neighborhood;
Output: $P_{t+1}$: the new population;
if $Nt$ == "Neighbor" then
    Update only the solutions which are the neighbor of the
    solution corresponding to the subproblem Id;
    Compare the new solution with the neighbor solution,
    the one has a smaller fitness is retained;
    And the max number of solutions to be updated is limited
    to two;
else
    Update all the solutions in $P_t$;
    Compare the new solution with each solution in $P_t$, the
    one has a smaller fitness is retained;
    And the max number of solutions to be updated is limited
    to two;
return the updated population as $P_{t+1}$;
```

3.2. Clustering algorithm

Several clustering strategies have been used [21,52,19,53,36,54–56] and comparison was conducted as well [57]. Castro et al. compared different clustering strategy for MaOPs: clustering in decision space, objective space or both [57]. After experimental study, they recommended clustering the solutions in the objective space although they concluded that the best choice of clustering space can be impacted by specific properties of the problem. Castro et al. also compared the results obtained when use different clustering metrics, after which they recommended to use the Euclidean distance since it was the most robust in their comparison.

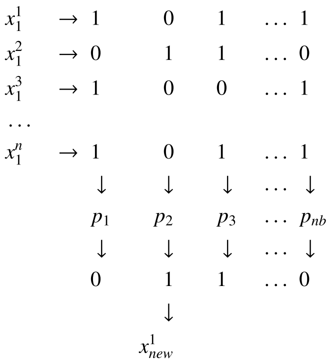


Fig. 1. Generating a new variable in bits model.

Therefore, in this paper, m (i.e., the objective number) clusters of solutions C_t are created from Ps_t by the k means clustering algorithm as a result of comparing the Euclidean distance between solutions. The Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be defined by [58]

$$distance = (\sum |x_i - y_i|^p)^{1/r} \tag{6}$$

where $p = r = 2$.

The max iterations of k means is set to m and the residual sum of squares (RSS) is set to $MaxRSS$. Only if the iteration number is less than m, the residual sum of squares is greater than $MaxRSS$ and the new generated mean vector is different from the previous one, will the k means iteration continue. Otherwise, it will terminate.

3.3. Estimation model

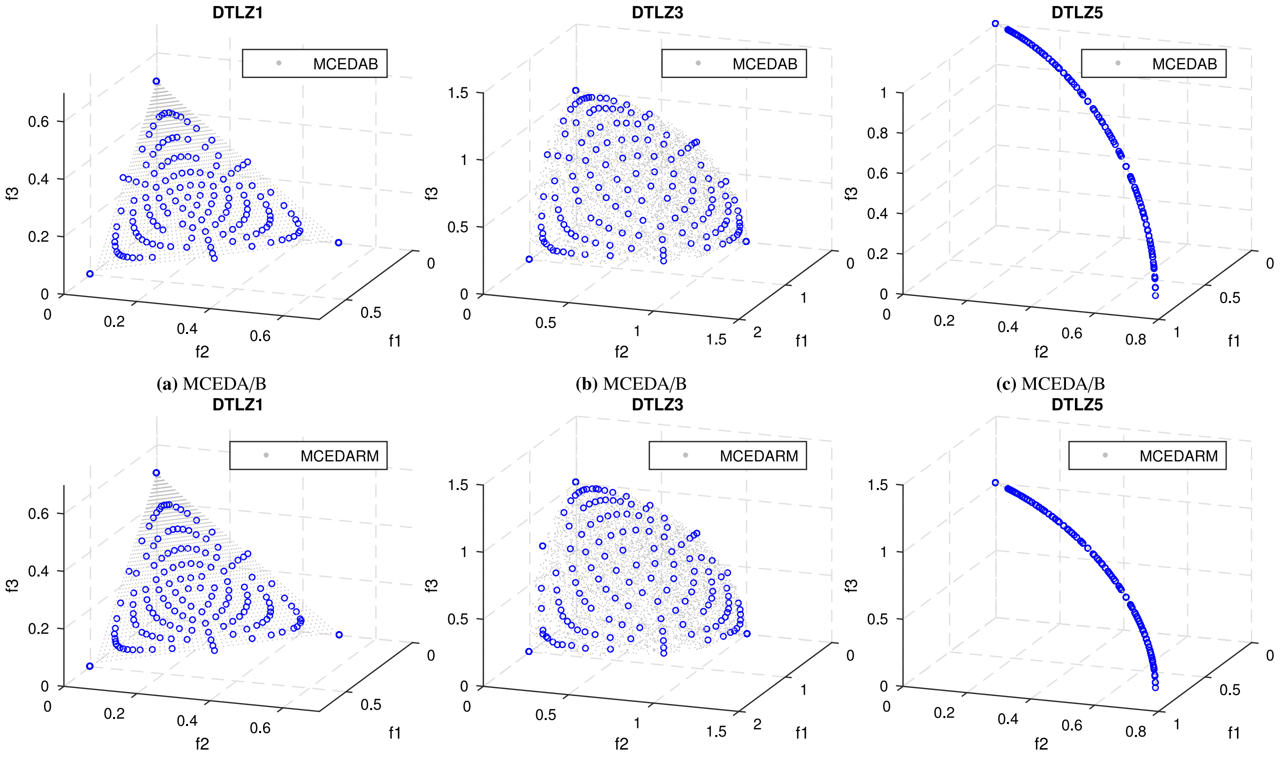
A too complicated model is not affordable and is not necessary, because modeling procedure needs to be conducted at each generation of the EAs [44]. And it is almost impossible to build a complete exact estimation model. In this paper, to deal with continuous MaOPs, two models are used to build estimation models for each cluster: a new estimation model based on bits probability (MCEDA/B) and the model adopted in RM-MEDA (MCEDA/RM).

3.3.1. Bits model

The principle of bits model is inspired by the Bayesian optimization algorithm (BOA) [34] which uses conditional probability and Bayesian network to estimate and that is mainly applied to problems with discrete variables. In bits model, the decision variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ are assumed to be independent each other. Notes that the bits model is different from BOA since the BOA builds model based on the connection among the variables while the bits model simply assumes they are independent. That is to say

$$P(\boldsymbol{X} = \boldsymbol{x}) = P(X_1 = x_1)P(X_2 = x_2)\ldots P(X_n = x_n) \tag{7}$$

The variables of each solution are transferred into a set of bits. Each bit is represented by '0' or '1'. All the bits are assumed to be independent each other. Then the probability distribution of each bit is used to generate the new variable. The reason why make this transform is in the hope of exploring more information from the solution sets as BOA has done. Experiments have shown that the new solutions generated by the bits model is explosive which can



(a) MCEDA/B **(b)** MCEDA/B **(c)** MCEDA/B

(d) MCEDA/RM **(e)** MCEDA/RM **(f)** MCEDA/RM

Fig. 2. Plots for the solution set in the best run obtained by MCEDA/B (first row) and MCEDA/RM (second row) on 3 objectives DTLZ problems.

cover enough decision space. In this section, a simple version of bits model is presented. The number of the bits of each variable is represented by *nb*.

In a solution set P, the ith solution S_i can be represented by its decision variables, $(x_i^1, x_i^2, \dots, x_i^n)$. The formula transferring a double variable x to the corresponding binary xb is presented as follows.

$$xb = toBinary(round(\frac{(x - lowLimit)(2^{nb} - 1)}{upperLimit - lowLimit})) \tag{8}$$

where $lowLimit$ and $upperLimit$ are the variable bounds of the variable x, which depend on the specific problem, $round(d)$ obtains the closest integer of d and $toBinary(t)$ transfers the integer t to the corresponding binary.

And the formula transferring a binary xb to the corresponding double variable x is presented below.

$$x = \frac{toInteger(xb)(upperLimit - lowLimit)}{(2^{nb} - 1)} + lowLimit \tag{9}$$

where $toInteger(b)$ transfers the binary b to the corresponding integer.

For example, the procedure of generating the first variable x_{new}^1 of the new solution S_{new} from a solution set P is illustrated in Fig. 1. In the figure, x_i^1 represents the first decision variable of the ith solution of P. First, x_i^1 is mapped to binary according to Eq. (8). Then the probability of the occurrence of '1' of the jth column p_j is calculated. To generate new variable, the new bits are generated at the probability of p and the new bits is mapped to the new variable x_{new}^1 by Eq. (9).

All the new generated variables $(x_{new}^1, x_{new}^2, \dots, x_{new}^n)$ are combined together to construct a new solution S_{new}. The model procedure is introduced in Algorithm 4.

3.3.2. Regular model

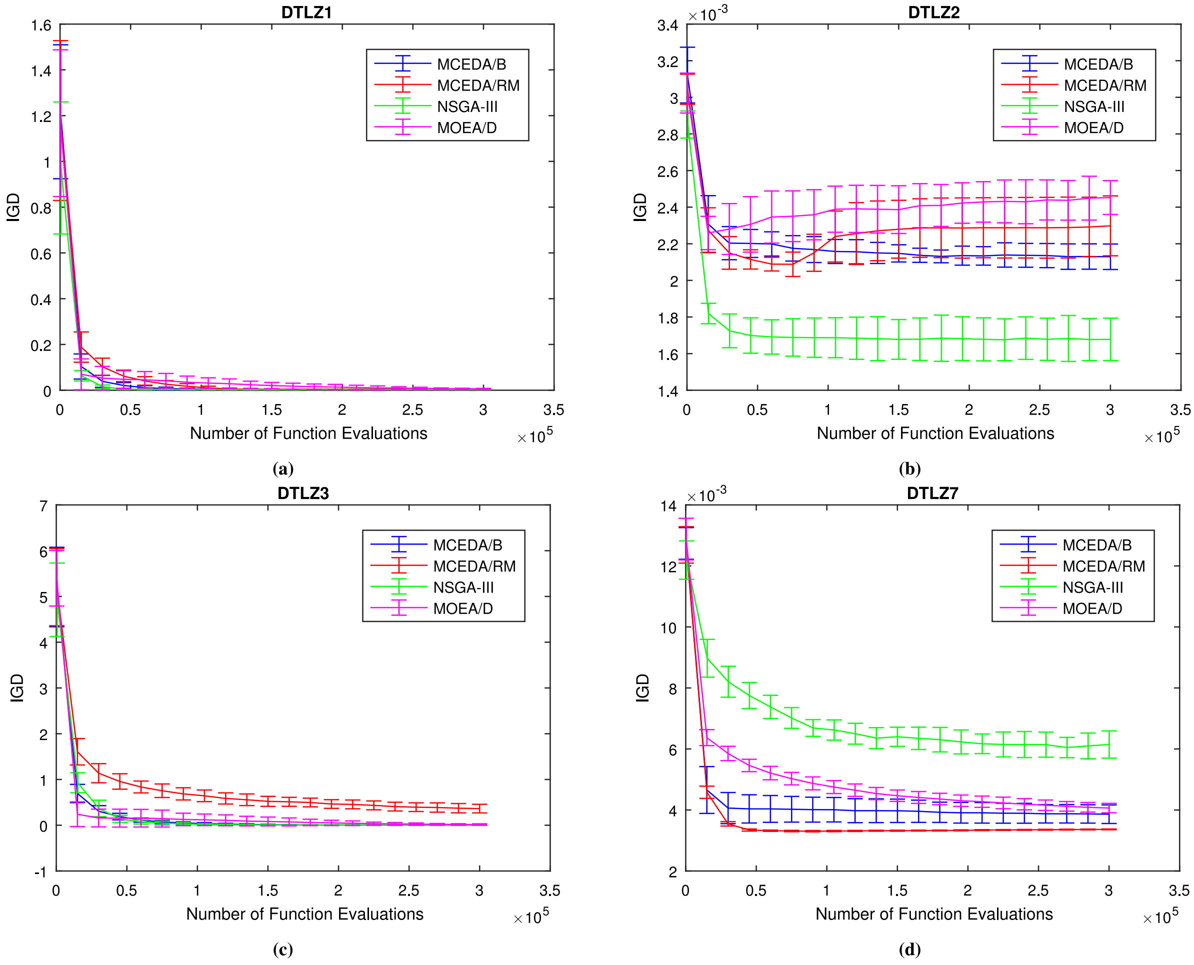
The RM-MEDA showed prominent performance in their experiments and it has been improved by latter researches [46,45]. In RM-MEDA, the model is built with the help of local principal component analysis (PCA) algorithm [59] under the assumption of Karush–Kuhn–Tucker condition. They assume that, with the search going on, the population in the decision space in an EA will approximate the Pareto Set (PS) (i.e., the decision variables of the corresponding Pareto Front) and be uniformly distribute around the PS, from which they make the assumption that the points in the population, whose centroid is the PS, are independent observations of a random vector $\xi \in R^n$.

$$\xi = \zeta + \varepsilon \tag{10}$$

where ζ is uniformly distributed over a piecewise continuous (m-1)-dimensional manifold. $\varepsilon \sim N(0, \sigma_j I)$ is an n-dimensional zero-mean noise vector, where I is the $n \times n$ identity matrix and $\sigma_j > 0$.

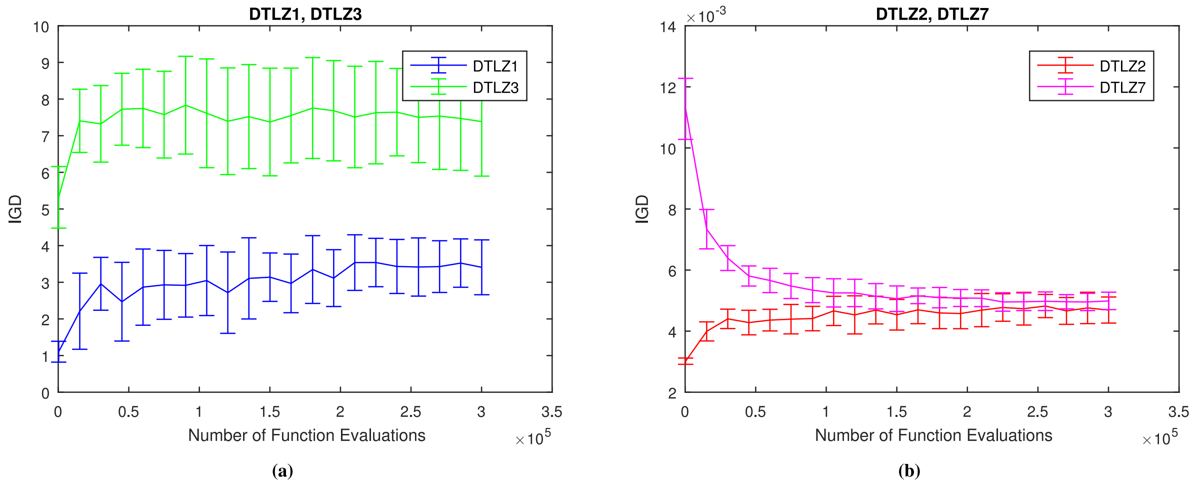
The modeling procedure is briefly described and the pseudocode is presented in Algorithm 5. Here is a detailed description of Algorithm 5. In this work, the regular model is applied to generate models for each cluster. For a certain solution cluster C_t^i, to begin with, it is partitioned into several sub clusters $S_1, S_2, \dots, S_h$ by the local PCA algorithm [59], where the partition number h is adaptively defined according to C_t^i. In this work, the partition number is simply set to one since the local PCA is time consuming. Then, each sub cluster is modeled by Eq. (10). Before generating new solutions, the mean of S_j ($\bar{\boldsymbol{x}}(i,j)$), the covariance matrix of the points in S_j ($Cov(i,j)$) should be calculated. Then, the sorted eigenvalue of $Cov(i,j)$ ($\lambda(i,j)$), i.e., $\lambda(i,j)(1,1) \geq \lambda(i,j)(2,2) \geq \cdots \geq \lambda(i,j)(n,n)$, is calculated. The eigenvector of $Cov(i,j)$ (U(i,j)), corresponding to $\lambda(i,j)$, is calculated as well. Now it is ready to generate the new solutions. The total number of new solutions to be generated is $m|C_t|$, and for each cluster in C_t $(C_t^i).n_i \leftarrow (m|C_t|)\frac{|C_t^i|}{size}$ solutions is generated. For each cluster in S (S_j), $n_i \frac{|S_j|}{|C_t^i|}$ new solutions are generated. And here is how to generate one new solution based on the previous calculation.

$$\boldsymbol{x}_{new} = \Psi(i,j) + \varepsilon'_j \tag{11}$$



(a) (b) (c) (d)

Fig. 3. Variation of IGD metric value with MCEDA/B, MCEDA/RM, NSGA-III and MOEA/D on four DTLZ instances with 8 objectives.



(a) (b)

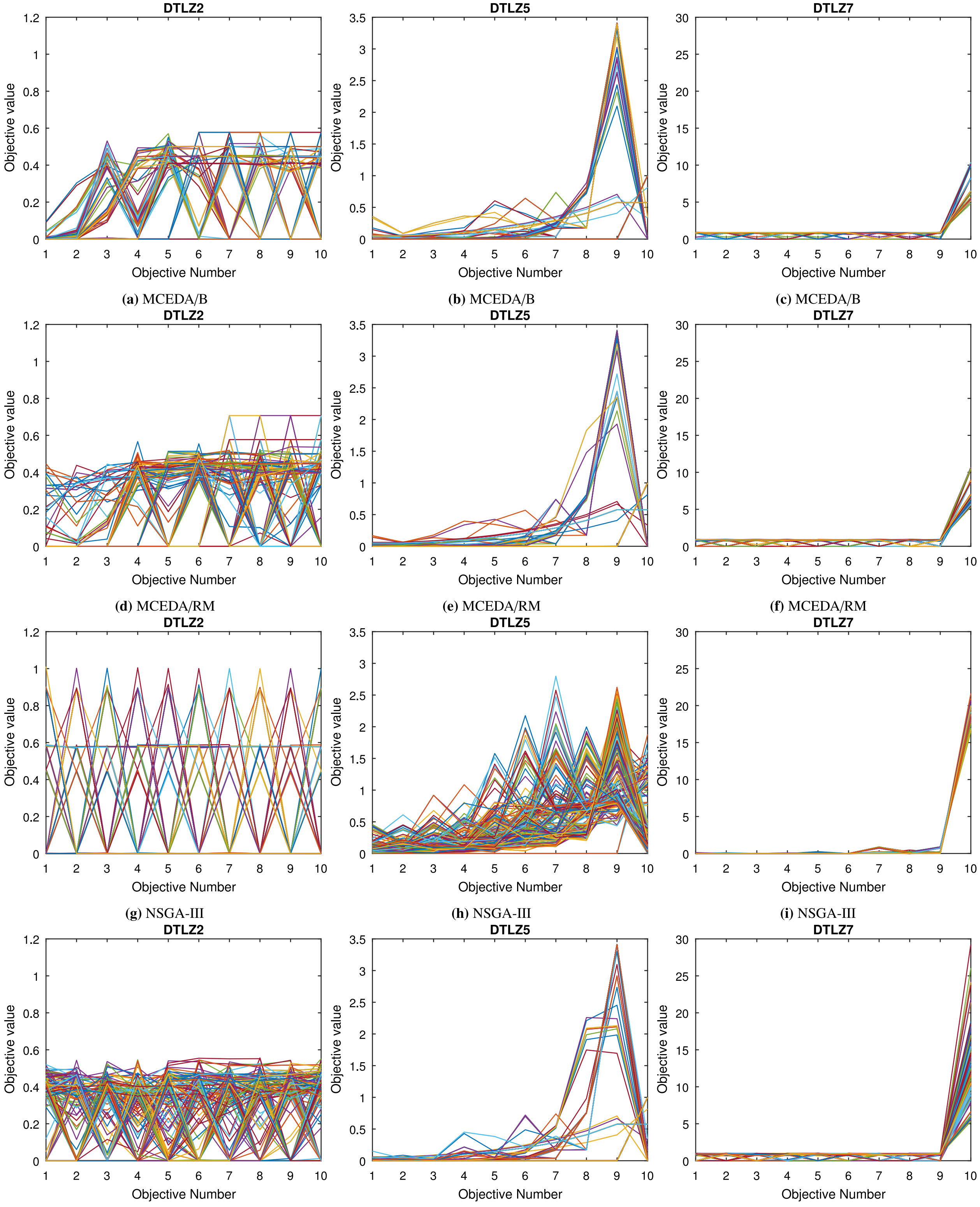
Fig. 4. Variation of IGD metric value with RM-MEDA on four DTLZ instances with 8 objectives.

where $\Psi(i,j) = \bar{\mathbf{x}}(i,j)+\sum_{w=1}^{m-1}\alpha_w U(i,j)(:,w)$, in which $\bar{\mathbf{x}}(i,j)$ is the mean of S_j and α_w is calculated in line 16 to line 18. ε'_j is a random value generated by normal distribution number where mean equals 0 and σ_j is the standard deviation. σ_j has been calculated in line 20.

The new population Q_t is formed by all the new solutions. For more details of the regular model, readers are referred to the original papers [59,44].

3.4. Computational complexity

Before analyzing the computational complexity of the proposed approach, some symbols are defined first. N is the population size. m is the number of the objectives of the problem. n is the number of the variables of the problem. And, in order to simplify the analysis, the solutions in each cluster is assumed to be N/m. N is usually greater than m and n. In Algorithm 2, the initialization process from



(a) MCEDA/B (b) MCEDA/B (c) MCEDA/B

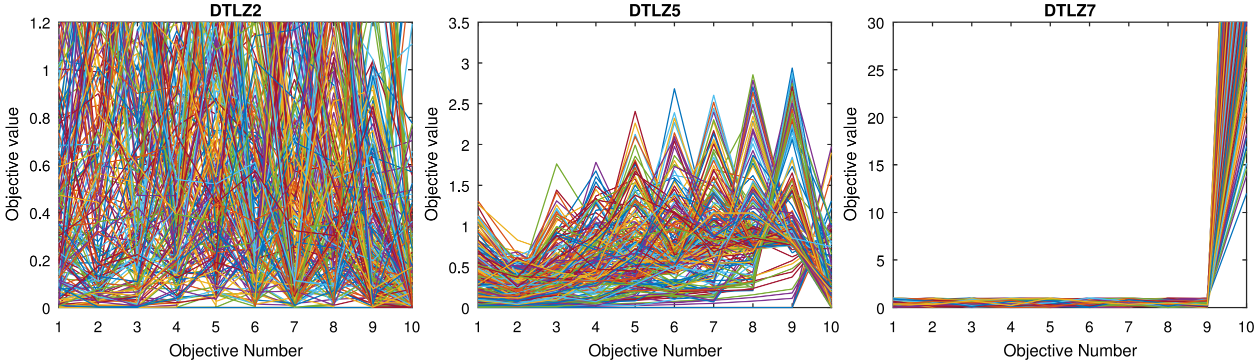
(d) MCEDA/RM (e) MCEDA/RM (f) MCEDA/RM

(g) NSGA-III (h) NSGA-III (i) NSGA-III

(j) MOEA/D (k) MOEA/D (l) MOEA/D

Fig. 5. Parallel coordinate plots for the solution set in the best run obtained by MCEDA/B (first row), MCEDA/RM (second row), NSGA-III (third row) and MOEA/D (fourth row) on 10 objectives DTLZ problems.

line 1 to line 7, requires $O(N + N + 1 + Nm + Nm + N + N^2)$ computations, i.e., $O(N^2)$. The while loop from line 8 to line 24 will stop when the maximum evaluations is reached. And line 9 requires $O(N)$ computations. The for loop from line 10 to line 24 will be executed N times. Line 11 to line 15 requires $O(1 + 1 + mN + N^2 + m^2N)$ computations, i.e., $O(N^2)$. Line 16 requires $O(N^2/m)$ computations for Algorithm 4 of MCEDA/B and $O(mN)$ computations for Algorithm 5 of MCEDA/RM. Line 17 to line 21 requires $O(m^2n + m^2n + m^2 + 1 + m^4)$ computations, i.e., $O(2m^2n + m^4)$. The for loop from line 22 to line 24 will be executed less than or



(a) RM-MEDA

(b) RM-MEDA

(c) RM-MEDA

Fig. 6. Parallel coordinate plots for the solution set in the best run obtained by RM-MEDA on 10 objectives DTLZ problems.

Table 2
Mean and standard deviation values of IGD, obtained by MCEDA/RM on DTLZ instances.

Crossover parameter	Obj.	0.5	0.1	0.8
	3	$5.97E-03_{1.50E-03}$	$7.50E-03_{2.80E-03}(-)$	$5.43E-03_{1.40E-03}(+)$
	5	$2.93E-03_{2.30E-03}$	$1.75E-03_{1.00E-04}(+)$	$1.84E-03_{3.00E-04}(+)$
DTLZ1	8	$4.81E-03_{5.70E-04}$	$4.38E-03_{7.90E-04}(+)$	$4.65E-03_{6.20E-04}(+)$
	10	$4.38E-03_{4.10E-04}$	$4.49E-03_{4.90E-04}(-)$	$4.63E-03_{4.20E-04}(-)$
	15	$8.01E-03_{7.20E-04}$	$7.56E-03_{7.20E-04}(+)$	$8.07E-03_{6.00E-04}(-)$
	3	$3.36E-04_{2.60E-05}$	$3.22E-04_{1.10E-05}(+)$	$3.37E-04_{2.30E-05}(-)$
	5	$3.47E-03_{2.90E-04}$	$3.22E-03_{4.80E-04}(+)$	$3.53E-03_{4.20E-04}(-)$
DTLZ2	8	$5.41E-03_{7.20E-04}$	$4.82E-03_{5.90E-04}(+)$	$5.08E-03_{7.90E-04}(+)$
	10	$4.53E-03_{3.80E-04}$	$4.35E-03_{4.40E-04}(+)$	$4.31E-03_{8.00E-04}(+)$
	15	$4.19E-03_{3.90E-04}$	$4.40E-03_{5.50E-04}(-)$	$4.48E-03_{5.00E-04}(-)$
	3	$1.25E-02_{4.10E-03}$	$1.23E-02_{2.40E-03}(+)$	$1.39E-02_{2.10E-03}(-)$
	5	$3.72E-03_{1.60E-04}$	$3.88E-03_{3.90E-04}(-)$	$3.79E-03_{2.10E-04}(-)$
DTLZ3	8	$9.64E-03_{1.20E-03}$	$9.30E-03_{1.50E-03}(+)$	$9.28E-03_{2.30E-03}(+)$
	10	$9.55E-03_{1.40E-03}$	$9.19E-03_{4.00E-04}(+)$	$9.33E-03_{1.50E-03}(+)$
	15	$1.19E-02_{8.10E-04}$	$1.16E-02_{7.00E-04}(+)$	$1.17E-02_{6.50E-04}(+)$
	3	$4.73E-03_{6.50E-03}$	$4.54E-03_{6.30E-03}(+)$	$4.04E-03_{5.00E-03}(+)$
	5	$2.80E-03_{1.00E-03}$	$3.78E-03_{2.90E-03}(-)$	$3.15E-03_{1.60E-03}(-)$
DTLZ4	8	$7.06E-03_{6.20E-04}$	$6.86E-03_{5.40E-04}(+)$	$6.97E-03_{4.60E-04}(+)$
	10	$8.96E-03_{2.60E-04}$	$9.22E-03_{4.30E-04}(-)$	$9.09E-03_{4.40E-04}(-)$
	15	$1.42E-02_{3.40E-04}$	$1.43E-02_{3.60E-04}(-)$	$1.41E-02_{4.00E-04}(+)$
	3	$2.01E-05_{1.80E-06}$	$2.17E-05_{7.60E-06}(-)$	$2.22E-05_{8.30E-06}(-)$
	5	$2.27E-03_{1.90E-04}$	$2.21E-03_{1.90E-04}(+)$	$2.16E-03_{1.10E-04}(+)$
DTLZ5	8	$4.10E-03_{4.20E-04}$	$3.99E-03_{3.20E-04}(+)$	$4.10E-03_{3.60E-04}(-)$
	10	$5.35E-03_{4.20E-04}$	$5.41E-03_{5.50E-04}(-)$	$5.74E-03_{7.80E-04}(-)$
	15	$7.41E-03_{4.00E-04}$	$7.63E-03_{3.60E-04}(-)$	$7.32E-03_{3.30E-04}(+)$
	3	$9.14E-03_{1.60E-03}$	$8.12E-03_{1.50E-03}(+)$	$9.04E-03_{1.50E-03}(+)$
	5	$1.84E-03_{3.50E-04}$	$1.66E-03_{2.20E-04}(+)$	$1.77E-03_{4.10E-04}(+)$
DTLZ6	8	$3.51E-03_{8.50E-04}$	$3.43E-03_{4.60E-04}(+)$	$3.60E-03_{5.40E-04}(-)$
	10	$2.87E-03_{3.50E-04}$	$3.38E-03_{6.30E-04}(-)$	$3.08E-03_{4.70E-04}(-)$
	15	$3.62E-03_{3.10E-04}$	$3.52E-03_{2.90E-04}(+)$	$3.58E-03_{6.60E-04}(+)$
	3	$6.63E-03_{4.10E-03}$	$3.79E-03_{4.20E-03}(+)$	$6.15E-03_{4.40E-03}(+)$
	5	$3.89E-03_{1.50E-03}$	$3.66E-03_{5.40E-04}(+)$	$3.61E-03_{6.00E-04}(+)$
DTLZ7	8	$6.51E-03_{1.90E-04}$	$6.64E-03_{2.00E-04}(-)$	$6.54E-03_{2.60E-04}(-)$
	10	$7.98E-03_{6.80E-05}$	$7.95E-03_{1.10E-04}(+)$	$7.98E-03_{1.60E-04}(-)$
	15	$1.16E-02_{2.20E-04}$	$1.14E-02_{1.00E-04}(+)$	$1.16E-02_{1.60E-04}(-)$

The symbol "+" and "−" denote whether the IGD results of the algorithm with the compared parameter are better than or worse than that of the algorithms with the first parameter value according to the mean value and standard deviation.

equal to m^2 times. Line 23 requires $O(m)$ computations. And line 24 requires $O(N+Nm)$ computations. The for loop from line 22 to line 24 requires $O(m^3N)$ computations. To execute the while loop from line 8 to line 24 once, the MCEDA/B requires $O(N(N^2+N^2/m+2m^2n+m^4+m^3N))$ computations and the MCEDA/RM require $O(N(N^2+mN+2m^2n+m^4+m^3N))$ computations, so that both the MCEDA/B and the MCEDA/RM requires $O(N^3+m^3N)$ computations when calculating the while loop once. The number of times that the while loop will be executed is depended on the max evaluations of parameter setting.

4. Experimental study

In this section, some parameter studies are conducted to show the performance of the proposed approach at a wide range of parameters. And the proposed MCEDA/B and MCEDA/RM are applied to the benchmark test suite for evolutionary algorithms (DTLZ) [60]. All algorithms are coded in Java and the experiments is conducted in the jMetal[1] platform [61–63]. The test results are compared with NSGA-III, MOEA/D and RM-MEDA.

[1] jMetal: https://jmetal.github.io/jMetal/.

Table 3
Mean and standard deviation values of IGD, obtained by MCEDA/B on DTLZ instances.

Number of bits	Obj.	16	8	30
DTLZ1	3	$4.37E-04_{1.10E-05}$	$4.78E-03_{1.90E-03}(-)$	$4.38E-04_{4.50E-06}(-)$
	5	$3.79E-03_{1.90E-03}$	$1.15E-02_{1.60E-03}(-)$	$2.85E-03_{1.00E-04}(+)$
	8	$2.40E-03_{1.90E-04}$	$6.87E-03_{1.20E-03}(-)$	$2.26E-03_{7.50E-05}(+)$
	10	$4.93E-03_{1.80E-04}$	$8.21E-03_{1.50E-03}(-)$	$4.97E-03_{2.10E-04}(-)$
	15	$5.99E-03_{1.90E-03}$	$8.93E-03_{1.40E-03}(-)$	$7.41E-03_{4.00E-03}(-)$
DTLZ2	3	$3.41E-04_{3.30E-05}$	$3.75E-04_{3.10E-05}(-)$	$3.32E-04_{2.20E-05}(+)$
	5	$1.76E-03_{2.10E-04}$	$1.77E-03_{2.60E-04}(-)$	$1.84E-03_{2.90E-04}(-)$
	8	$4.80E-03_{1.20E-03}$	$4.32E-03_{8.10E-04}(+)$	$4.88E-03_{9.80E-04}(-)$
	10	$7.08E-03_{1.10E-03}$	$6.29E-03_{9.50E-04}(+)$	$7.72E-03_{9.70E-04}(-)$
	15	$3.02E-03_{5.30E-04}$	$2.68E-03_{3.70E-04}(+)$	$3.16E-03_{6.10E-04}(-)$
DTLZ3	3	$1.56E-03_{1.20E-06}$	$2.29E-02_{2.20E-03}(-)$	$1.55E-03_{1.40E-06}(+)$
	5	$2.16E-02_{3.10E-03}$	$3.61E-02_{2.00E-03}(-)$	$2.54E-02_{7.00E-03}(-)$
	8	$1.35E-02_{2.00E-03}$	$1.37E-02_{1.10E-03}(-)$	$1.66E-02_{6.60E-03}(-)$
	10	$9.80E-03_{5.40E-04}$	$1.30E-02_{1.30E-03}(-)$	$9.48E-03_{6.10E-04}(+)$
	15	$6.63E-03_{5.40E-04}$	$8.10E-03_{5.00E-03}(-)$	$6.37E-03_{2.50E-04}(+)$
DTLZ4	3	$8.53E-03_{5.10E-03}$	$8.98E-03_{5.70E-03}(-)$	$1.36E-02_{2.70E-03}(-)$
	5	$1.13E-02_{3.10E-03}$	$1.08E-02_{3.60E-03}(+)$	$9.92E-03_{5.80E-03}(+)$
	8	$6.06E-03_{1.60E-03}$	$6.49E-03_{1.00E-03}(-)$	$6.29E-03_{9.60E-04}(-)$
	10	$1.21E-02_{1.10E-03}$	$1.18E-02_{8.40E-04}(+)$	$1.26E-02_{2.20E-03}(-)$
	15	$8.39E-03_{5.70E-04}$	$8.71E-03_{4.00E-04}(-)$	$8.10E-03_{5.40E-04}(+)$
DTLZ5	3	$1.99E-05_{1.00E-05}$	$2.55E-05_{7.40E-06}(-)$	$2.74E-05_{1.10E-05}(-)$
	5	$2.35E-03_{4.50E-04}$	$2.11E-03_{5.40E-04}(+)$	$1.75E-03_{3.30E-04}(+)$
	8	$1.93E-03_{1.80E-04}$	$2.01E-03_{2.70E-04}(-)$	$1.96E-03_{1.90E-04}(-)$
	10	$2.49E-03_{2.40E-04}$	$2.64E-03_{3.20E-04}(-)$	$2.48E-03_{2.00E-04}(+)$
	15	$1.15E-03_{6.40E-05}$	$1.15E-03_{7.10E-05}(-)$	$1.14E-03_{1.10E-04}(+)$
DTLZ6	3	$2.66E-05_{1.00E-05}$	$2.96E-05_{1.30E-05}(-)$	$2.49E-05_{2.20E-05}(+)$
	5	$2.06E-03_{1.90E-04}$	$2.04E-03_{2.50E-04}(+)$	$2.12E-03_{1.80E-04}(-)$
	8	$2.83E-03_{4.00E-04}$	$3.08E-03_{6.70E-04}(-)$	$2.86E-03_{3.20E-04}(-)$
	10	$5.33E-03_{1.80E-03}$	$5.16E-03_{9.20E-04}(+)$	$4.90E-03_{1.10E-03}(+)$
	15	$2.19E-03_{2.50E-04}$	$2.17E-03_{3.10E-04}(+)$	$2.25E-03_{3.40E-04}(-)$
DTLZ7	3	$2.77E-03_{4.50E-03}$	$1.70E-03_{3.30E-03}(+)$	$2.76E-03_{4.50E-03}(+)$
	5	$7.30E-03_{2.80E-03}$	$8.41E-03_{2.20E-03}(-)$	$7.27E-03_{3.50E-03}(+)$
	8	$8.44E-03_{1.40E-03}$	$1.02E-02_{1.50E-03}(-)$	$1.03E-02_{1.20E-03}(-)$
	10	$1.42E-02_{1.30E-03}$	$1.40E-02_{1.60E-03}(+)$	$1.51E-02_{7.40E-04}(-)$
	15	$1.02E-02_{5.30E-04}$	$1.05E-02_{5.70E-04}(-)$	$1.05E-02_{6.10E-04}(-)$

The symbol "+" and "−" denote whether the IGD results of the algorithm with the compared parameter are better than or worse than that of the algorithms with the first parameter value according to the mean value and standard deviation.

4.1. Test sets

The scalable test problems DTLZ, proposed by Kalyanmoy Deb et al. [60], have been widely used to test MOEAs. In their paper, three test problem construction approaches are suggested and DTLZ1 ∼ DTLZ9 are proposed. Test problem DTLZ1 has a linear Pareto front (PF). And DTLZ3 has many local Pareto fronts and one global Pareto fronts in order to investigate an MOEA's ability to converge to the global PF. DTLZ4 is designed to investigate an MOEA's ability to maintain a good distribution of solutions. Both DTLZ5 and DTLZ6 have a curve PF. And DTLZ7 has a disconnected set of Pareto-optimal regions, which tests an algorithm's ability to maintain subpopulation in different Pareto-optimal regions. In this work, the 3 to 15 objectives DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 are tested. The number of variables are set to $(m + k - 1)$, where m is the number of objectives and $k = 5$ for DTLZ1, while $k = 10$ for DTLZ2, DTLZ3, DTLZ4, DTLZ5 and DTLZ6, and $k = 20$ for DTLZ7 [7]. Table 6 illustrates the test setting.

4.2. Performance metrics

The inverse generational distance (IGD) metric [7] is chosen as the performance metric, since it can provide a combined information about the convergence and diversity of the obtained solutions. It measures the distances between the optimal solution set, obtained by EAs, and the computed approximated Pareto front [61]. It can be defined as follows.

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{12}$$

where n is the number of solutions in the optimal solution set, obtained by EAs, and d_i is the Euclidean distance (measured in objective space) between each point of the optimal solution set and the nearest member of approximated PF. A solution set with a smaller IGD value is better.

4.3. Parameter studies

Parameter studies of the proposed approach are conducted in this section. Parts of the results are presented in Tables 1, 2, 3, 4, 5. Note that When one parameter is changing, all other parameters are fixed. The results show that MCEDA/B and MCEDA/RM can work well at a wide range of parameters. It is a good choice to set crossover parameter as 0.5 for MCEDA/B and MCEDA/RM. And the number of bits used in MCEDA/B can be set to 16, which works better. Usually, the EDAs need more populations to produce a good results because the more the number of the populations, the more accurate of the model. The results of iterations number of k means are shown in Tables 4 and 5. Notes that the data from different tables should not be compared since other parameters may be different. However, the parameter setting would be related to the problem to be solved. It is recommended to choose the parameter for a specific problem.

Table 4
Mean and standard deviation values of IGD, obtained by MCEDA/RM on DTLZ instances.

Number of k means iterations	Obj.	200	50	500
DTLZ1	3	$5.84E-03_{4.20E-03}$	$5.44E-03_{3.20E-03}(+)$	$4.95E-03_{1.20E-03}(+)$
	5	$2.37E-03_{2.10E-04}$	$2.27E-03_{1.80E-04}(+)$	$2.31E-03_{1.80E-04}(+)$
	8	$4.73E-03_{6.80E-04}$	$4.33E-03_{3.90E-04}(+)$	$4.34E-03_{7.90E-04}(+)$
	10	$5.01E-03_{4.00E-04}$	$5.04E-03_{5.20E-04}(-)$	$4.76E-03_{4.70E-04}(+)$
	15	$5.23E-03_{2.00E-04}$	$5.53E-03_{4.90E-04}(-)$	$5.90E-03_{4.40E-04}(-)$
DTLZ2	3	$3.21E-04_{1.50E-05}$	$3.32E-04_{1.70E-05}(-)$	$3.31E-04_{2.40E-05}(-)$
	5	$2.75E-03_{3.80E-04}$	$3.10E-03_{5.20E-04}(-)$	$3.19E-03_{5.40E-04}(-)$
	8	$4.86E-03_{4.20E-04}$	$4.97E-03_{8.10E-04}(-)$	$5.12E-03_{6.80E-04}(-)$
	10	$4.36E-03_{4.80E-04}$	$4.42E-03_{4.50E-04}(-)$	$4.70E-03_{8.00E-04}(-)$
	15	$4.41E-03_{7.90E-04}$	$4.57E-03_{5.50E-04}(-)$	$4.66E-03_{7.60E-04}(-)$
DTLZ3	3	$1.34E-02_{1.70E-03}$	$1.20E-02_{3.10E-03}(+)$	$1.35E-02_{2.10E-03}(-)$
	5	$5.40E-03_{2.70E-03}$	$4.52E-03_{3.60E-04}(+)$	$4.35E-03_{5.10E-04}(+)$
	8	$7.01E-03_{8.60E-04}$	$6.71E-03_{6.70E-04}(+)$	$6.90E-03_{6.70E-04}(+)$
	10	$1.27E-02_{2.80E-03}$	$1.23E-02_{1.30E-03}(+)$	$1.22E-02_{2.10E-03}(+)$
	15	$9.71E-03_{8.40E-04}$	$9.70E-03_{8.20E-04}(+)$	$1.01E-02_{1.20E-03}(-)$
DTLZ4	3	$3.57E-03_{5.50E-03}$	$7.35E-03_{6.80E-03}(-)$	$8.77E-03_{8.80E-03}(-)$
	5	$2.81E-03_{1.10E-03}$	$3.50E-03_{2.80E-03}(-)$	$2.73E-03_{8.90E-04}(+)$
	8	$6.70E-03_{3.00E-04}$	$6.67E-03_{2.70E-04}(+)$	$6.50E-03_{3.00E-04}(+)$
	10	$8.78E-03_{1.20E-04}$	$9.10E-03_{3.50E-04}(-)$	$8.85E-03_{2.90E-04}(-)$
	15	$1.33E-02_{4.30E-04}$	$1.34E-02_{2.90E-04}(-)$	$1.34E-02_{4.30E-04}(-)$
DTLZ5	3	$1.61E-05_{2.60E-06}$	$1.53E-05_{3.00E-07}(+)$	$1.79E-05_{6.40E-06}(-)$
	5	$2.11E-03_{1.70E-04}$	$2.09E-03_{2.00E-04}(+)$	$2.10E-03_{9.30E-05}(+)$
	8	$3.78E-03_{2.40E-04}$	$3.89E-03_{3.60E-04}(-)$	$4.15E-03_{7.00E-04}(-)$
	10	$4.56E-03_{4.60E-04}$	$4.72E-03_{6.20E-04}(-)$	$4.80E-03_{6.30E-04}(-)$
	15	$6.85E-03_{2.80E-04}$	$7.02E-03_{4.90E-04}(-)$	$6.97E-03_{3.60E-04}(-)$
DTLZ6	3	$9.25E-03_{1.50E-03}$	$9.13E-03_{2.30E-03}(+)$	$9.52E-03_{1.20E-03}(-)$
	5	$1.72E-03_{2.90E-04}$	$1.61E-03_{1.70E-04}(+)$	$1.84E-03_{4.20E-04}(-)$
	8	$3.36E-03_{1.20E-03}$	$2.66E-03_{5.00E-04}(+)$	$2.81E-03_{5.30E-04}(+)$
	10	$3.23E-03_{5.00E-04}$	$2.90E-03_{4.80E-04}(+)$	$3.27E-03_{8.70E-04}(-)$
	15	$3.28E-03_{2.90E-04}$	$3.44E-03_{5.60E-04}(-)$	$3.44E-03_{4.90E-04}(-)$
DTLZ7	3	$4.68E-03_{3.70E-03}$	$5.60E-03_{3.70E-03}(-)$	$6.63E-03_{3.60E-03}(-)$
	5	$3.44E-03_{6.70E-04}$	$3.57E-03_{6.40E-04}(-)$	$3.71E-03_{9.20E-04}(-)$
	8	$6.41E-03_{1.80E-04}$	$6.45E-03_{2.50E-04}(-)$	$6.45E-03_{2.20E-04}(-)$
	10	$7.80E-03_{1.60E-04}$	$7.89E-03_{9.80E-05}(-)$	$7.82E-03_{1.80E-04}(-)$
	15	$1.15E-02_{2.10E-04}$	$1.15E-02_{1.90E-04}(-)$	$1.14E-02_{1.50E-04}(+)$

The symbol "+" and "−" denote whether the IGD results of the algorithm with the compared parameter are better than or worse than that of the algorithms with the first parameter value according to the mean value and standard deviation.

4.4. Compared algorithms

The proposed algorithms, MCEDA/B and MCEDA/RM, are compared with three other MOEAs: NSGA-III, MOEA/D and RM-MEDA.

The NSGA-III [7] is a reference-point-based many-objective evolutionary algorithm following NSGA-II framework based on genetic algorithms. It emphasizes population members that are non-dominated and which are close to a set of supplied reference points. In their work, Das's method is adopted to create reference points [64], which has been widely used [21,7]. The NSGA-III has been applied to test problems with 3 to 15 objectives, which produced satisfactory results. And it has been extended to deal with constrained MaOPs [20]. The MOEA/D [11] is a multi-objective evolutionary algorithm based on decomposition and differential evolution (DE), which decomposes a MOOP into a number of scalar optimization subproblems and optimizes them simultaneously. It performs very well in multi-objective optimization problems as well as MaOPs [65,21]. In this experiment, the Tchebycheff approach is adopted to decompose problems. The RM-MEDA [44] is a multi-objective estimation of distribution algorithm, which uses a regularity model to generate new solutions from the previous populations. And it has only a few parameters. In the paper, describing RM-MEDA, the authors have conducted systematically experiments to compare RM-MEDA with other competitive algorithms, e.g. generalized differential evolution (GDE3), Parent-Centric Recombination (PCX-NSGA-II) and a competitive EDA for MOPs, MIDEDA. The experimental results show that RM-MEDA performs better than others. Therefore, it should be proper to use RM-MEDA as a representation of EDAs in this paper.

4.5. Parameter setting

For all the competing algorithms, the same parameter setting of problems is used to ensure fair comparison, which is presented in Tables 6 and 7.

More parameter setting is presented in Table 8. The parameters of the algorithms compared is set according to the original papers. In NSGA-III, the simulated binary crossover (SBX) [66] probability p_c is set to 1 and the crossover parameter in SBX η_c is set to 30 according to the original paper [7]. In MOEDA/D, the differential crossover parameter cr is assigned to 1.0 and f is set to 0.5 [11,12]. The crossover parameter in MCEDA c_r is set to 0.5. The polynomial mutation probability p_m of all algorithms is assigned as $1/n$ where n is the number of variables and the η_m is set to 20. For MOEA/D and MCEDA, the neighborhood size is assigned with 20 [11]. The partition number used in RM-MEDA is set to 5 [44]. All algorithms will terminate when their evaluations reach 300,000 and all of them will be run 30 times independently.

4.6. Results

4.6.1. Three objectives problems

The Pareto fronts of problems with 3 objectives can be visualized in a Cartesian coordinate. There have been many graphs for NSGA-III and MOEA/D in terms of 3 objectives on DTLZ problems [7,20,11,12]. Therefore, in this section, only the proposed MCEDA/B and MCEDA/RM on 3 objectives DTLZ problems are plotted in Fig. 2. It is not surprising that the distribution of the solution

Table 5
Mean and standard deviation values of IGD, obtained by MCEDA/B on DTLZ instances.

Number of k means iterations	Obj.	200	50	500
DTLZ1	3	$2.44E-04_{2.60E-05}$	$2.36E-04_{2.20E-05}(+)$	$2.32E-04_{3.50E-05}(+)$
	5	$1.32E-03_{3.10E-04}$	$1.23E-03_{5.70E-05}(+)$	$1.28E-03_{1.10E-04}(+)$
	8	$1.99E-03_{6.30E-04}$	$3.28E-03_{3.20E-03}(-)$	$1.79E-03_{2.40E-04}(+)$
	10	$5.43E-03_{5.60E-04}$	$5.20E-03_{7.20E-04}(+)$	$5.00E-03_{7.40E-04}(+)$
	15	$1.07E-02_{5.90E-03}$	$7.29E-03_{3.50E-03}(+)$	$1.02E-02_{5.90E-03}(+)$
DTLZ2	3	$3.53E-04_{3.40E-05}$	$3.44E-04_{2.70E-05}(+)$	$3.41E-04_{2.60E-05}(+)$
	5	$2.24E-03_{5.00E-04}$	$2.15E-03_{3.70E-04}(+)$	$2.09E-03_{2.10E-04}(+)$
	8	$4.69E-03_{5.80E-04}$	$4.85E-03_{4.40E-04}(-)$	$4.87E-03_{7.00E-04}(-)$
	10	$5.64E-03_{8.70E-04}$	$5.25E-03_{7.00E-04}(+)$	$5.95E-03_{1.20E-03}(-)$
	15	$2.63E-03_{6.10E-04}$	$2.66E-03_{2.20E-04}(-)$	$2.54E-03_{4.20E-04}(+)$
DTLZ3	3	$8.41E-04_{8.00E-05}$	$8.35E-04_{1.10E-04}(+)$	$1.16E-03_{1.10E-03}(-)$
	5	$2.16E-02_{2.00E-02}$	$1.56E-02_{9.70E-03}(+)$	$1.45E-02_{1.10E-02}(+)$
	8	$1.07E-02_{4.20E-03}$	$1.08E-02_{3.80E-03}(-)$	$1.28E-02_{7.30E-03}(-)$
	10	$8.09E-03_{6.00E-03}$	$6.64E-03_{2.80E-03}(+)$	$6.60E-03_{1.90E-03}(+)$
	15	$1.11E-02_{2.90E-03}$	$8.97E-03_{1.50E-03}(+)$	$1.24E-02_{5.00E-03}(-)$
DTLZ4	3	$1.04E-02_{7.00E-03}$	$1.25E-02_{6.70E-03}(-)$	$6.99E-03_{8.10E-03}(+)$
	5	$1.37E-02_{2.80E-03}$	$1.35E-02_{3.40E-03}(+)$	$1.29E-02_{3.60E-03}(+)$
	8	$7.00E-03_{1.50E-03}$	$6.57E-03_{8.80E-04}(+)$	$6.62E-03_{8.20E-04}(+)$
	10	$1.21E-02_{1.10E-03}$	$1.25E-02_{1.50E-03}(-)$	$1.30E-02_{1.30E-03}(-)$
	15	$8.15E-03_{6.50E-04}$	$8.15E-03_{5.00E-04}(-)$	$8.11E-03_{4.40E-04}(+)$
DTLZ5	3	$1.89E-05_{9.30E-06}$	$1.67E-05_{1.00E-05}(+)$	$1.88E-05_{1.80E-05}(+)$
	5	$2.04E-03_{3.00E-04}$	$1.86E-03_{3.10E-04}(+)$	$2.07E-03_{3.30E-04}(-)$
	8	$2.18E-03_{2.40E-04}$	$2.31E-03_{2.10E-04}(-)$	$2.29E-03_{3.30E-04}(-)$
	10	$2.45E-03_{1.30E-04}$	$2.58E-03_{4.20E-04}(-)$	$2.47E-03_{1.80E-04}(-)$
	15	$1.68E-03_{8.80E-05}$	$1.76E-03_{1.30E-04}(-)$	$1.75E-03_{1.40E-04}(-)$
DTLZ6	3	$4.74E-05_{4.50E-05}$	$5.75E-05_{6.00E-05}(-)$	$2.79E-05_{2.00E-05}(+)$
	5	$2.08E-03_{2.50E-04}$	$2.15E-03_{2.60E-04}(-)$	$2.05E-03_{1.20E-04}(+)$
	8	$2.98E-03_{6.50E-04}$	$2.84E-03_{5.10E-04}(+)$	$2.85E-03_{4.40E-04}(+)$
	10	$5.30E-03_{9.60E-04}$	$4.89E-03_{6.20E-04}(+)$	$5.92E-03_{1.40E-03}(-)$
	15	$2.23E-03_{2.80E-04}$	$2.13E-03_{9.60E-05}(+)$	$2.43E-03_{4.90E-04}(-)$
DTLZ7	3	$6.41E-04_{2.70E-05}$	$2.69E-03_{4.30E-03}(-)$	$4.68E-03_{5.20E-03}(-)$
	5	$7.30E-03_{3.40E-03}$	$6.12E-03_{3.00E-03}(+)$	$8.14E-03_{2.90E-03}(-)$
	8	$8.37E-03_{2.40E-03}$	$1.02E-02_{2.40E-03}(-)$	$7.67E-03_{1.60E-03}(+)$
	10	$1.37E-02_{9.10E-04}$	$1.45E-02_{8.90E-04}(-)$	$1.45E-02_{1.60E-03}(-)$
	15	$1.08E-02_{5.00E-04}$	$1.04E-02_{5.80E-04}(+)$	$1.02E-02_{7.30E-04}(+)$

The symbol "+" and "−" denote whether the IGD results of the algorithm with the compared parameter are better than or worse than that of the algorithms with the first parameter value according to the mean value and standard deviation.

sets are similar to that of MOEA/D because MCEDA is based on MOEA/D framework. As the graphs show, the obtained solutions well distribute and converge to the Pareto fronts of DTLZ1, DTLZ3 and DTLZ5.

4.6.2. IGD values

The mean and standard deviation values of IGD, obtained by MCEDA/B, MCEDA/RM, NSGA-III and MOEA/D on DTLZ instances after 30 independent runs, is presented in Tables 9 and 10.

Here is how Tables 9 and 10 are produced. (1) Apply the Mann–Whitney rank test to justify the results of different algorithms to find whether they are statistically identical at the 5% significance level. (2) If they are statistically identical, their performance will be regarded as equal. (3) Otherwise, mean value and standard deviation are used to decide whether one result is better/worse than the other.

The data from Tables 9 and 10 can be compared together. According to the IGD values table, MCEDA/RM performs better in DTLZ6 and DTLZ7 while NSGA-III performs well in DTLZ1, DTLZ2 and DTLZ3. And it can be seen that MCEDA/RM has better results than MOEA/D and RM-MEDA on many problems, which should give the credit to the estimation model and the clustering strategy used in MCEDA/RM. MCEDA/B has better performance than MOEA/B and RM-MEDA in many problems although its results are worse than NSGA-III. It is noted that MCEDA/RM performs better than MCEDA/B , which may be due to the too ideal assumption of the bits model.

4.6.3. IGD variation with increasing number of function evaluations

Since the IGD value can represent the convergence and diversity of the obtained solutions at the same time, it is valuable to explore the change of IGD value with the increase of function evaluations. Variation of IGD metric value with MCEDA/B, MCEDA/RM, NSGA-III and MOEA/D on four DTLZ instances with 8 objectives is represented in Fig. 3. Variation of IGD metric value with RM-MEDA on four DTLZ instances with 8 objectives is represented in Fig. 4. These results are obtained after 300,000 function evaluations for each algorithm. And each experiment has been run 20 times to get the mean and standard deviation of the IGD value. And the Pareto fronts adopted here is generated by the MOEAFramework.[2] As we can see, MCEDA/B and MCEDA/RM have better performance on DTLZ1 and DTLZ7 with 8 objectives, while NSGA-III performs better on DTLZ2. It seems that the regular model based MCEDA/RM is not so suitable for the DTLZ3 with 8 objectives, while the MCEDA/B has normal performance on it. MOEA/D shows unstable fluctuation on DTLZ2 while MCEDA/B and MCEDA/RM have intended to lower the IGD value. According to the graphs, MCEDA/B and MCEDA/RM have outperformed others on DTLZ7 with 8 objectives. As for RM-MEDA, it seems that it cannot handle problems like DTLZ1 and DTLZ3 and has intended to lower the IGD on DTLZ2 and DTLZ4. According to the graphs, RM-MEDA, the pure EDA, is not good at dealing with problems with many objectives.

[2] MOEAFramework: http://moeaframework.org/index.html.

Algorithm 4. Bits model

```
Input: $C_t$: the solution cluster set of current population,
       where $C_t = \{C_t^1, C_t^2, \ldots, C_t^k\}$. $C_t^i$ contains the ith
       solution cluster; $m|C_t|$: the number of new solutions;
       $n$: the number of the variables of the problem;
Output: $Q_t$: new solutions;
$size \leftarrow$ the number of all solutions in $C_t$;
// Model each cluster;
for $i \leftarrow 1$ to $|C_t|$ do
    $Cs_t^i \leftarrow nonDominatedSelect(C_t^i)$;
    $Probability_i \leftarrow \emptyset$;
    for $j \leftarrow 1$ to $n$ do
        $singleVariableVector \leftarrow \emptyset$;
        for $v \leftarrow 1$ to $|Cs_t^i|$ do
            $singleVariableVector(v) \leftarrow$
            the jth variable of $Cs_t^i(v)$;
        $variableBitsMatrix \leftarrow getBitsMatrix($
        $singleVariableVector, lowLimit, upperLimit)$;
        $bitsProbability \leftarrow calculateBitsProbability($
        $variableBitsMatrix)$;
        $Probability_i \leftarrow Probability_i \cup bitsProbability$;
    // $n_i$ is the number of the new solutions to
       be generated for the $i$th cluster;
    $n_i \leftarrow (m|C_t|)\frac{|C_t^i|}{size}$;
    $counter \leftarrow 0$;
    while $counter < n_i$ do
        $newSolution \leftarrow createSolution()$;
        for $v \leftarrow 1$ to $n$ do
            $newVariable \leftarrow getNewVariable(\ Probability_i(v),$
            $lowLimit, upperLimit)$;
            if newVariable exceeds the variable bounds then
                Regenerate the newVariable
            Set the $v$th variable of the newSolution to
            newVariable;
        $Q_t \leftarrow Q_t \cup newSolution$;
        $counter++$;
return $Q_t$;
```

Algorithm 5. Regular model

```
Input: $C_t$: the solution cluster set of current population,
       where $C_t = \{C_t^1, C_t^2, \ldots, C_t^k\}$. $C_t^i$ contains the ith
       solution cluster; $m|C_t|$: the number of new solutions;
Output: $Q_t$: new solutions;
$C_t' \leftarrow \emptyset$;
$size \leftarrow$ the number of all solutions in $C_t$;
// Model each cluster;
for $i \leftarrow 1$ to $|C_t|$ do
    // The partition number is set to 1;
    $h \leftarrow 1$;
    // $S = \{S_1, S_2, \ldots, S_h\}$;
    $S \leftarrow PCApartition(C_t(i), h)$;
    $S' \leftarrow \emptyset$;
    // Model each sub cluster;
    for $j \leftarrow 1$ to $h$ do
        $\bar{\mathbf{x}}(i,j) \leftarrow mean(S_j)$;
        $Cov(i,j) \leftarrow \frac{1}{|S_j|-1}\sum_{\mathbf{x}\in S_j}(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}}(i,j))^T$;
        // $\lambda(i,j)$ is the eigenvalue matrix, which
           is sorted (i.e.,
           $\lambda(i,j)(1,1) \geq \lambda(i,j)(2,2) \geq \ldots \geq \lambda(i,j)(n,n)$) ;
        $\lambda(i,j) \leftarrow Eigenvalue(Cov(i,j)).sort()$;
        // $U(i,j)$ is the eigenvector matrix
           corresponding to $\lambda(i,j)$;
        $U(i,j) \leftarrow Eigenvector(Cov(i,j)).sort()$;
    // Generate new solutions;
    // $n_i$ is the number of the new solutions to
       be generated for the $i$th cluster;
    $n_i \leftarrow (m|C_t|)\frac{|C_t^i|}{size}$;
    for $j \leftarrow 1$ to $h$ do
        for $jj \leftarrow 1$ to $n_i\frac{|S_j|}{|C_t^i|}$ do
            for $w \leftarrow 1$ to $m-1$ do
                $a(i,j,w) \leftarrow \min_{\mathbf{x}\in S_j}(\mathbf{x}-\bar{\mathbf{x}})^T U(i,j)(:,w)$;
                $b(i,j,w) \leftarrow \max_{\mathbf{x}\in S_j}(\mathbf{x}-\bar{\mathbf{x}})^T U(i,j)(:,w)$;
                $\alpha_w \leftarrow uniformlyRandomlyGenerate($
                $a(i,j,w) - 0.25(b(i,j,w) - a(i,j,w)),$
                $b(i,j,w) + 0.25(b(i,j,w) - a(i,j,w)))$;
            $\Psi(i,j) \leftarrow \bar{\mathbf{x}}(i,j) + \sum_{w=1}^{m-1}\alpha_w U(i,j)(:,w)$;
            $\sigma_j \leftarrow \frac{1}{n-m+1}\sum_{w=m}^{n}\lambda(i,j)(w,w)$;
            for $ii \leftarrow 1$ to $n_i$ do
                // Generate normal distribution
                   number where mean equals 0 and
                   standard deviation equals $\sigma_j$;
                $\varepsilon_j(ii,ii) \leftarrow$
                $generateNormalDistributionNumber(0, \sigma_j)$;
                $\varepsilon_j'(ii) \leftarrow \varepsilon_j(ii,ii)$;
            $S_j'(jj) \leftarrow \Psi(i,j) + \varepsilon_j'$;
        $C_t'(i) \leftarrow C_t'(i) \cup S_j'$;
$Q_t \leftarrow$ all solution in $C_t'$;
return $Q_t$;
```

Note that the time cost by the proposed algorithms and other EDAs are a little larger than non-EDAs. However the difference is not too large and the time cost by the proposed algorithms is acceptable. And many methods can be used to speed up the run of algorithms, e.g. parallelization, vectorization, using cache.

4.6.4. Parallel coordinates

While the results of problems withe 2 or 3 objectives can be plotted by scatter in Cartesian coordinates, the visualization of results of MaOPs remains a challenge. When it comes to MaOPs, there have been some researches in visualizing their results [67, 68]. Parallel coordinates are such a tool, which illustrate an m-dimensional space on a 2D figure by plotting points in the parallel vertical parallel axes [69]. The parallel coordinates with different order of parallel vertical parallel axes will present different information.

Fig. 5 shows the parallel coordinate plots for the solution set in the best run obtained by MCEDA/B (first row), MCEDA/RM (second row), NSGA-III (third row) and MOEA/D (fourth row) on 10 objectives DTLZ problems. Fig. 6 shows the parallel coordinate plots for the solution set in the best run obtained by RM-MEDA on 10 objectives DTLZ problems. As we can see from the graphs, as for DTLZ2, MCEDA/B, MCEDA/RM and MOEA/D do not converge to the Pareto fronts as fast as NSGA-III, which indicates that they may need more evaluations. According to the parallel coordinates, these algorithms has similar performance on DTLZ5 with 10 objectives. RM-MEDA shows large oscillation for DTLZ2 and DTLZ7. However, it is important to note that parallel coordinates may not be able to accurately represent those properties since the problem could be very complex [69].

Table 6
DTLZ setting.

Problems	Objective number	Variable number
DTLZ1	3	7
	5	9
	8	12
	10	14
	15	19
DTLZ2	3	12
	5	14
	8	17
	10	19
	15	24
DTLZ3	3	12
	5	14
	8	17
	10	19
	15	24
DTLZ4	3	12
	5	14
	8	17
	10	19
	15	24
DTLZ5	3	12
	5	14
	8	17
	10	19
	15	24
DTLZ6	3	12
	5	14
	8	17
	10	19
	15	24
DTLZ7	3	22
	5	24
	8	27
	10	29
	15	34

Table 7
Population size.

No. of objectives	NSGA-III divisions	Population size
3	18	192
5	5	210
8	4	330
10	3	220
15	3	680

4.7. Discussion

The proposed approach has been applied to 3-objective to 15-objective test problems of DTLZ. On those problems, the proposed algorithms, MCEDA/B and MCEDA/RM, have been able to successfully find a well-converged and well-diversified set of points over multiple runs. MCEDA/RM have showed better results than MOEA/D and RM-MEDA in many problems and perform best in some problems. It is noted that MCEDA/B does not perform as good as MCEDA/RM, which may be because the assumption of the bits model adopted in the work is too ideal and it will be improved further. It can be expected that a better model incorporated with MCEDA will generate better results.

As a matter of fact, the best approach may be related to the specific problem to be solved and it may be useful to clearly define the class of problems an algorithm is designed for. However, how to choose the best approach will be the subsequent problem. It may be more difficult to choose a better algorithm than to apply a universe approach directly, since the choose of approaches could be very time consuming and a universe algorithm can also produce well distributed and converged results.

5. Conclusion

In this paper, a new approach to deal with many-objective optimization problems, utilizing estimation models and clustering strategy, is proposed, which is based on decomposition and the MOEA/D framework. And two algorithms, MCEDA/B and MCEDA/RM, have been implemented. Most other EDAs try to build the estimation model for the all variables as a whole and are based on the traditional EDA framework Algorithm 1. It may work very well at low number of objectives and simple problems. However, when it comes to MaOPs and problems with complex Pareto fronts, it would be unimaginable to build such a complex model. Therefore, unlike the traditional EDAs, the proposed approach builds the model after the clustering strategy, which will enhance the precision and the reliability of the latter estimation.

The performance of MCEDA/B and MCEDA/RM have been compared with NSGA-III, MOEA/D and RM-MEDA. The NSGA-III is based on GA and the MOEA/D is based on differential evolution (DE), which do not use any prior knowledge. While the proposed MCEDA/B and MCEDA/RM try to build estimation model by the bits model and the regular model in order to find more explicit and promising solutions so as to speed up the convergence procedure. The results have shown that the proposed MCEDA is a competitive and promising approach to solve MaOPs.

In the future, more explicit and more efficient estimation model will be tested in the hope of enhancing the convergence and diversity of the algorithm and increasing its speed. More reliable performance metrics will be used to justify the algorithms as well, since the accurate Pareto fronts are difficult to find in MaOPs. Also MCEDA will be extended to handle constrained problems and it will be applied to real-word engineering problems.

Table 8
Parameter setting.

Parameters	NSGA-III	MOEA/D	MCEDA/B	MCEDA/RM	RM-MEDA
SBX probability, p_c	1				
Crossover parameter in SBX, η_c	30				
Differential crossover, *cr*		1			
Differential crossover, *f*		0.5			
Crossover parameter in MCEDA			0.5	0.5	
Max RSS in k means, *MaxRSS*			1e−5	1e−5	
Max iterations in k means, *MaxIterations*			200	200	
Polynomial mutation probability, p_m	$1/n$	$1/n$	$1/n$	$1/n$	
Parameter in polynomial mutation, η_m	20	20	20	20	
Neighborhood size		20	20	20	
Partition number					5
Max evaluations	300,000	300,000	300,000	300,000	300,000
Independent runs	30	30	30	30	30

Table 9
Mean and standard deviation values of IGD, obtained by MCEDA/RM, NSGA-III, MOEA/D and RM-MEDA on DTLZ instances.

Problem	Obj.	MCEDA/RM	NSGA-III	MOEA/D	RM-MEDA
DTLZ1	3	$2.32E-04_{2.50E-05}$	$1.40E-04_{1.70E-04}(+)$	$2.21E-04_{2.80E-05}(+)$	$5.37E-03_{2.80E-03}(-)$
	5	$3.19E-04_{7.00E-05}$	$2.50E-04_{7.20E-05}(+)$	$1.72E-03_{2.50E-03}(-)$	$3.23E-03_{2.10E-04}(-)$
	8	$1.40E-03_{1.50E-04}$	$2.44E-03_{1.10E-03}(-)$	$4.41E-03_{3.50E-03}(-)$	$1.70E-03_{4.70E-05}(-)$
	10	$4.51E-03_{5.20E-04}$	$2.80E-03_{6.10E-04}(+)$	$1.92E-03_{4.00E-04}(+)$	$2.48E-03_{1.10E-04}(+)$
	15	$1.34E-03_{4.00E-04}$	$8.71E-04_{1.60E-04}(+)$	$4.09E-03_{6.80E-04}(-)$	$1.62E-03_{1.30E-04}(-)$
DTLZ2	3	$3.04E-04_{4.20E-05}$	$4.40E-05_{1.50E-05}(+)$	$3.99E-04_{3.90E-05}(-)$	$3.16E-03_{6.40E-04}(-)$
	5	$7.01E-04_{1.70E-04}$	$1.82E-04_{1.20E-05}(+)$	$2.39E-03_{3.60E-04}(-)$	$3.31E-03_{3.31E-03}(-)$
	8	$3.89E-03_{6.50E-04}$	$2.20E-03_{2.80E-03}(+)$	$5.04E-03_{4.50E-04}(-)$	$3.02E-03_{7.40E-05}(+)$
	10	$7.64E-03_{8.60E-04}$	$5.20E-03_{1.50E-03}(+)$	$9.62E-03_{5.20E-04}(-)$	$4.36E-03_{1.50E-04}(+)$
	15	$3.10E-03_{4.00E-04}$	$2.22E-03_{8.10E-04}(+)$	$6.54E-03_{1.10E-04}(-)$	$2.79E-03_{5.80E-05}(+)$
DTLZ3	3	$6.60E-04_{6.70E-05}$	$1.18E-04_{3.20E-05}(+)$	$4.15E-04_{3.00E-05}(+)$	$5.46E-03_{1.10E-03}(-)$
	5	$3.58E-02_{1.30E-02}$	$7.72E-04_{5.60E-04}(+)$	$8.43E-03_{3.20E-03}(+)$	$3.42E-03_{2.30E-04}(+)$
	8	$5.31E-03_{5.70E-04}$	$2.75E-03_{2.60E-03}(+)$	$4.02E-03_{1.90E-03}(+)$	$2.94E-03_{7.70E-05}(+)$
	10	$4.93E-03_{3.20E-03}$	$3.44E-03_{1.40E-03}(+)$	$2.16E-03_{2.70E-04}(+)$	$4.95E-03_{1.40E-04}(-)$
	15	$9.58E-03_{7.00E-04}$	$8.90E-04_{8.50E-05}(+)$	$5.61E-03_{9.50E-04}(+)$	$2.86E-03_{6.60E-05}(+)$
DTLZ4	3	$2.76E-03_{5.70E-03}$	$3.58E-04_{1.60E-03}(+)$	$1.64E-03_{2.00E-03}(+)$	$1.56E-02_{9.20E-03}(-)$
	5	$6.88E-04_{1.30E-04}$	$1.53E-03_{2.40E-03}(-)$	$4.23E-03_{1.70E-03}(-)$	$2.46E-03_{2.00E-04}(-)$
	8	$3.44E-03_{6.10E-04}$	$1.29E-04_{6.00E-06}(+)$	$8.76E-03_{6.30E-04}(-)$	$1.68E-03_{4.90E-05}(+)$
	10	$7.22E-03_{8.70E-04}$	$7.49E-04_{1.10E-03}(+)$	$1.65E-02_{8.30E-04}(-)$	$2.61E-03_{5.50E-05}(+)$
	15	$7.05E-03_{5.50E-04}$	$1.24E-04_{1.80E-05}(+)$	$7.45E-03_{7.90E-04}(-)$	$1.54E-03_{8.80E-06}(+)$
DTLZ5	3	$1.85E-05_{1.30E-05}$	$1.58E-04_{2.10E-05}(-)$	$2.85E-05_{1.90E-05}(-)$	$1.84E-03_{1.10E-03}(-)$
	5	$1.68E-03_{1.60E-04}$	$1.90E-03_{4.80E-04}(-)$	$2.36E-03_{4.70E-04}(-)$	$2.61E-03_{2.80E-04}(-)$
	8	$1.81E-03_{1.60E-04}$	$1.20E-03_{3.70E-05}(+)$	$4.07E-03_{7.30E-04}(-)$	$2.46E-03_{2.50E-04}(-)$
	10	$3.38E-03_{3.00E-04}$	$2.60E-03_{7.10E-04}(+)$	$6.98E-03_{5.70E-04}(-)$	$4.00E-03_{4.10E-04}(-)$
	15	$1.29E-03_{8.90E-05}$	$1.43E-03_{3.60E-04}(-)$	$4.44E-03_{2.30E-04}(-)$	$1.77E-03_{9.50E-05}(-)$
DTLZ6	3	$3.28E-04_{2.00E-04}$	$1.62E-03_{3.10E-04}(-)$	$2.09E-05_{1.80E-05}(+)$	$8.86E-03_{2.80E-03}(-)$
	5	$1.99E-03_{2.80E-04}$	$3.02E-03_{1.50E-04}(-)$	$1.52E-03_{1.70E-04}(+)$	$2.80E-03_{1.30E-04}(-)$
	8	$2.03E-03_{3.50E-04}$	$2.23E-03_{1.50E-03}(-)$	$3.47E-03_{5.90E-04}(-)$	$2.62E-03_{6.00E-05}(-)$
	10	$2.91E-03_{3.50E-04}$	$6.25E-03_{1.20E-03}(-)$	$5.09E-03_{7.30E-04}(-)$	$3.93E-03_{7.50E-05}(-)$
	15	$1.29E-03_{1.10E-04}$	$1.39E-03_{3.80E-04}(-)$	$2.71E-03_{2.20E-04}(-)$	$2.46E-03_{4.10E-05}(-)$
DTLZ7	3	$1.31E-03_{3.10E-03}$	$3.17E-04_{9.20E-06}(+)$	$8.10E-04_{2.20E-03}(+)$	$2.60E-02_{4.90E-03}(-)$
	5	$6.69E-04_{3.30E-04}$	$6.69E-03_{1.90E-03}(-)$	$2.85E-03_{7.10E-04}(-)$	$8.56E-03_{7.20E-04}(-)$
	8	$3.17E-03_{3.60E-04}$	$5.60E-03_{7.00E-04}(-)$	$4.48E-03_{5.60E-04}(-)$	$5.78E-03_{2.20E-04}(-)$
	10	$1.46E-03_{2.40E-05}$	$6.41E-03_{4.20E-04}(-)$	$7.43E-03_{1.10E-03}(-)$	$8.06E-03_{1.30E-04}(-)$
	15	$4.92E-03_{5.30E-05}$	$2.70E-03_{1.80E-04}(+)$	$6.92E-03_{3.70E-04}(-)$	$5.74E-03_{2.50E-05}(-)$

The symbol "=" denote that the IGD results of the proposed algorithm and algorithms compared are statistically identical according to the Mann–Whitney rank test at the 5% significance level. The symbol "+" and "−" denote whether the IGD results of the compared algorithm are better or worse than that of the proposed algorithms according to the mean value and standard deviation.

Table 10
Mean and standard deviation values of IGD, obtained by MCEDA/B, NSGA-III, MOEA/D and RM-MEDA on DTLZ instances.

Problem	Obj.	MCEDA/B	NSGA-III	MOEA/D	RM-MEDA
DTLZ1	3	$1.85E-04_{1.50E-05}$	$1.40E-04_{1.70E-04}(+)$	$2.21E-04_{2.80E-05}(-)$	$5.37E-03_{2.80E-03}(-)$
	5	$3.65E-04_{7.90E-05}$	$2.50E-04_{7.20E-05}(+)$	$1.72E-03_{2.50E-03}(-)$	$3.23E-03_{2.10E-04}(-)$
	8	$2.67E-03_{4.20E-04}$	$2.44E-03_{1.10E-03}(+)$	$4.41E-03_{3.50E-03}(-)$	$1.70E-03_{4.70E-05}(+)$
	10	$6.59E-03_{1.20E-03}$	$2.80E-03_{6.10E-04}(+)$	$1.92E-03_{4.00E-04}(+)$	$2.48E-03_{1.10E-04}(+)$
	15	$2.21E-03_{3.00E-04}$	$8.71E-04_{1.60E-04}(+)$	$4.09E-03_{6.80E-04}(-)$	$1.62E-03_{1.30E-04}(+)$
DTLZ2	3	$2.53E-04_{4.40E-05}$	$4.40E-05_{1.50E-05}(+)$	$3.99E-04_{3.90E-05}(-)$	$3.16E-03_{6.40E-04}(-)$
	5	$6.69E-04_{1.20E-04}$	$1.82E-04_{1.20E-05}(+)$	$2.39E-03_{3.60E-04}(-)$	$3.31E-03_{3.31E-03}(-)$
	8	$4.17E-03_{7.00E-04}$	$2.20E-03_{2.80E-03}(+)$	$5.04E-03_{4.50E-04}(-)$	$3.02E-03_{7.40E-05}(+)$
	10	$7.02E-03_{1.30E-03}$	$5.20E-03_{1.50E-03}(+)$	$9.62E-03_{5.20E-04}(-)$	$4.36E-03_{1.50E-04}(+)$
	15	$3.23E-03_{5.80E-04}$	$2.22E-03_{8.10E-04}(+)$	$6.54E-03_{1.10E-04}(-)$	$2.79E-03_{5.80E-05}(+)$
DTLZ3	3	$4.71E-04_{5.20E-05}$	$1.18E-04_{3.20E-05}(+)$	$4.15E-04_{3.00E-05}(+)$	$5.46E-03_{1.10E-03}(-)$
	5	$2.77E-03_{3.90E-04}$	$7.72E-04_{5.60E-04}(+)$	$8.43E-03_{3.20E-03}(-)$	$3.42E-03_{2.30E-04}(-)$
	8	$4.17E-03_{7.00E-04}$	$2.75E-03_{2.60E-03}(+)$	$4.02E-03_{1.90E-03}(+)$	$2.94E-03_{7.70E-05}(+)$
	10	$7.02E-03_{1.30E-03}$	$3.44E-03_{1.40E-03}(+)$	$2.16E-03_{2.70E-04}(+)$	$4.95E-03_{1.40E-04}(+)$
	15	$1.13E-02_{2.20E-03}$	$8.90E-04_{8.50E-05}(+)$	$5.61E-03_{9.50E-04}(+)$	$2.86E-03_{6.60E-05}(+)$
DTLZ4	3	$1.05E-02_{6.30E-03}$	$3.58E-04_{1.60E-03}(+)$	$1.64E-03_{2.00E-03}(+)$	$1.56E-02_{9.20E-03}(-)$
	5	$1.08E-02_{4.40E-03}$	$1.53E-03_{2.40E-03}(+)$	$4.23E-03_{1.70E-03}(+)$	$2.46E-03_{2.00E-04}(+)$
	8	$6.21E-03_{1.40E-03}$	$1.29E-04_{6.00E-06}(+)$	$8.76E-03_{6.30E-04}(-)$	$1.68E-03_{4.90E-05}(+)$
	10	$1.13E-02_{2.20E-03}$	$7.49E-04_{1.10E-03}(+)$	$1.65E-02_{8.30E-04}(-)$	$2.61E-03_{5.50E-05}(+)$
	15	$9.37E-03_{8.50E-04}$	$1.24E-04_{1.80E-05}(+)$	$7.45E-03_{7.90E-04}(+)$	$1.54E-03_{8.80E-06}(+)$

(*continued on next page*)

Table 10 (*continued*).

Problem	Obj.	MCEDA/B	NSGA-III	MOEA/D	RM-MEDA
DTLZ5	3	$1.84E-05_{1.80E-05}$	$1.58E-04_{2.10E-05}(-)$	$2.85E-05_{1.90E-05}(-)$	$1.84E-03_{1.10E-03}(-)$
	5	$1.68E-03_{4.70E-04}$	$1.90E-03_{4.80E-04}(-)$	$2.36E-03_{4.70E-04}(-)$	$2.61E-03_{2.80E-04}(-)$
	8	$2.06E-03_{1.70E-04}$	$1.20E-03_{3.70E-05}(+)$	$4.07E-03_{7.30E-04}(-)$	$2.46E-03_{2.50E-04}(-)$
	10	$2.88E-03_{2.60E-04}$	$2.60E-03_{7.10E-04}(+)$	$6.98E-03_{5.70E-04}(-)$	$4.00E-03_{4.10E-04}(-)$
	15	$1.05E-03_{1.60E-04}$	$1.43E-03_{3.60E-04}(-)$	$4.44E-03_{2.30E-04}(-)$	$1.77E-03_{9.50E-05}(-)$
DTLZ6	3	$5.31E-05_{5.30E-05}$	$1.62E-03_{3.10E-04}(-)$	$2.09E-05_{1.80E-05}(+)$	$8.86E-03_{2.80E-03}(-)$
	5	$2.07E-03_{3.10E-04}$	$3.02E-03_{1.50E-04}(-)$	$1.52E-03_{1.70E-04}(+)$	$2.80E-03_{1.30E-04}(-)$
	8	$3.22E-03_{6.60E-04}$	$2.23E-03_{1.50E-03}(+)$	$3.47E-03_{5.90E-04}(-)$	$2.62E-03_{6.00E-05}(+)$
	10	$5.13E-03_{1.20E-03}$	$6.25E-03_{1.20E-03}(-)$	$5.09E-03_{7.30E-04}(+)$	$3.93E-03_{7.50E-05}(+)$
	15	$2.88E-03_{2.50E-04}$	$1.39E-03_{3.80E-04}(+)$	$2.71E-03_{2.20E-04}(+)$	$2.46E-03_{4.10E-05}(+)$
DTLZ7	3	$1.39E-03_{2.90E-03}$	$3.17E-04_{9.20E-06}(+)$	$8.10E-04_{2.20E-03}(+)$	$2.60E-02_{4.90E-03}(-)$
	5	$5.28E-03_{2.80E-03}$	$6.69E-03_{1.90E-03}(-)$	$2.85E-03_{7.10E-04}(+)$	$8.56E-03_{7.20E-04}(-)$
	8	$8.06E-03_{1.60E-03}$	$5.60E-03_{7.00E-04}(+)$	$4.48E-03_{5.60E-04}(+)$	$5.78E-03_{2.20E-04}(+)$
	10	$1.33E-02_{1.60E-03}$	$6.41E-03_{4.20E-04}(+)$	$7.43E-03_{1.10E-03}(+)$	$8.06E-03_{1.30E-04}(+)$
	15	$1.03E-02_{7.20E-04}$	$2.70E-03_{1.80E-04}(+)$	$6.92E-03_{3.70E-04}(+)$	$5.74E-03_{2.50E-05}(+)$

The symbol "=" denote that the IGD results of the proposed algorithm and algorithms compared are statistically identical according to the Mann–Whitney rank test at the 5% significance level. The symbol "+" and "−" denote whether the IGD results of the compared algorithm are better or worse than that of the proposed algorithms according to the mean value and standard deviation.

References

[1] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[2] B. Li, J. Li, K. Tang, X. Yao, Many-objective evolutionary algorithms: A survey, ACM Comput. Surv. 48 (1) (2015) 13.

[3] J.H. Zhu, W.H. Zhang, L. Xia, Topology optimization in aircraft and aerospace structures design, Arch. Comput. Methods Eng. 23 (4) (2016) 595–622, http://dx.doi.org/10.1007/s11831-015-9151-2.

[4] R. Jafarkhani, S.F. Masri, Finite element model updating using evolutionary strategy for damage detection, Comput.-Aided Civil Infrastruct. Eng. 26 (3) (2011) 207–224, http://dx.doi.org/10.1111/j.1467-8667.2010.00687.x.

[5] A.M. Raich, T.R. Liszkai, Multi-objective optimization of sensor and excitation layouts for frequency response function-based structural damage identification, Comput.-Aided Civil Infrastruct. Eng. 27 (2) (2012) 95–117, http://dx.doi.org/10.1111/j.1467-8667.2011.00726.x.

[6] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim. 26 (6) (2004) 369–395, http://dx.doi.org/10.1007/s00158-003-0368-6.

[7] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601, http://dx.doi.org/10.1109/tevc.2013.2281535.

[8] M. Pelikan, M.W. Hauschild, F.G. Lobo, Introduction to estimation of distribution algorithms, MEDAL Rep. (2012003) (2012).

[9] M. Pelikan, D.E. Goldberg, E.E. Cant, Linkage problem, distribution estimation, and bayesian networks, Evol. Comput. 8 (3) (2000) 311–340, http://dx.doi.org/10.1162/106365600750078808.

[10] L. Mart, J. Garcia, A. Berlanga, C. Coello, J.M. Molina, On current model-building methods for multi-objective estimation of distribution algorithms: shortcommings and directions for improvement, Department of Informatics of the Universidad Carlos III de Madrid, Madrid, Spain, Tech. Rep. GIAA2010E001, 2010.

[11] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/tevc.2007.892759.

[12] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: 2009 IEEE Congress on Evolutionary Computation, pp. 203–208, http://dx.doi.org/10.1109/CEC.2009.4982949.

[13] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C.A.C. Coello, A survey of multiobjective evolutionary algorithms for data mining: part i, IEEE Trans. Evol. Comput. 18 (1) (2014) 4–19, http://dx.doi.org/10.1109/tevc.2013.2290086.

[14] K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Vol. 16, Wiley, 2001.

[15] K. Deb, D.K. 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), pp. 3352–3360.

[16] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Ieee, Evolutionary many-objective optimization: A short review, in: 2008 Ieee Congress on Evolutionary Computation, Vols 1-8, 2008, pp. 2419–2426.

[17] H. Aguirre, K. Tanaka, Many-objective optimization by space partitioning and adaptive epsilon-ranking on MNK-landscapes, in: M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.K. Hao, M. Sevaux (Eds.), Evolutionary Multi-Criterion Optimization: 5th International Conference, Emo 2009, in: Lecture Notes in Computer Science, vol. 5467, 2009, pp. 407–+.

[18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Ieee, Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, in: 2009 Ieee International Conference on Systems, Man and Cybernetics, in: IEEE International Conference on Systems Man and Cybernetics Conference Proceedings, 2009, pp. 1758–1763, http://dx.doi.org/10.1109/icsmc.2009.5346628.

[19] X. Guo, Y. Wang, X. Wang, Using objective clustering for solving many-objective optimization problems, Math. Probl. Eng. 2013 (2013).

[20] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2014) 602–622, http://dx.doi.org/10.1109/tevc.2013.2281534.

[21] L. Cai, S.R. Qu, Y. Yuan, X. Yao, A clustering-ranking method for many-objective optimization, Appl. Soft Comput. 35 (2015) 681–694, http://dx.doi.org/10.1016/j.asoc.2015.06.020.

[22] X. Cai, Z. Yang, Z. Fan, Q. Zhang, Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization, IEEE Trans. Cybern. PP (99) (2017) 1–14, http://dx.doi.org/10.1109/TCYB.2016.2586191.

[23] H. Xing, Z. Wang, T. Li, H. Li, R. Qu, An improved MOEA/D algorithm for multi-objective multicast routing with network coding, Appl. Soft Comput. 59 (2017) 88–103, http://dx.doi.org/10.1016/j.asoc.2017.05.033.

[24] X. Meng, J. Li, M. Zhou, X. Dai, J. Dou, Population-based incremental learning algorithm for a serial colored traveling salesman problem, IEEE Trans. Syst. Man Cybern.: Syst. 48 (2) (2018) 277–288, http://dx.doi.org/10.1109/TSMC.2016.2591267.

[25] J. Bai, H. Liu, Multi-objective artificial bee algorithm based on decomposition by PBI method, Appl. Intell. 45 (4) (2016) 976–991.

[26] Y. Han, D. Gong, Y. Jin, Q. Pan, Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns, IEEE Trans. Cybern. 42 (C) (2017) 229–245.

[27] X.-W. Zheng, X.-M. Yu, Y. Li, H. Liu, An enhanced multi-objective group search optimizer based on multi-producer and crossover operator, J. Inf. Sci. Eng. 33 (1) (2017) 37–50.

[28] J.-q. Li, H.-y. Sang, Y.-y. Han, C.-g. Wang, K.z. Gao, Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions, J. Cleaner Prod. 181 (2018) 584–598.

[29] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm Evol. Comput. 1 (3) (2011) 111–128, http://dx.doi.org/10.1016/j.swevo.2011.08.003.

[30] M. Laumanns, J. Ocenasek, Bayesian Optimization Algorithms for Multi-objective Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 298–307, http://dx.doi.org/10.1007/3-540-45712-7_29.

[31] J. Ocenasek, S. Kern, N. Hansen, P. Koumoutsakos, A mixed bayesian optimization algorithm with variance adaptation, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 352–361.

[32] M. Pelikan, Bayesian optimization algorithm: from single level to hierarchy, Thesis, 2002.

[33] M. Pelikan, D.E. Goldberg, Escaping hierarchical traps with competent genetic algorithms, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 2001, pp. 511–518.

[34] M. Pelikan, D.E. Goldberg, E. Cant-Paz, BOA: the bayesian optimization algorithm, in: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1, Morgan Kaufmann Publishers Inc., 1999, pp. 525–532.

[35] M. Pelikan, D.E. Goldberg, S. Tsutsui, Hierarchical bayesian optimization algorithm: toward a new generation of evolutionary algorithms, DBLP, 2003, pp. 2738–2743 Vol.3.

[36] M. Pelikan, K. Sastry, D.E. Goldberg, Multiobjective hBOA, clustering, and scalability, in: Proceedings of the 7th annual conference on Genetic and evolutionary computation, ACM, 2005, pp. 663–670.

[37] J.R. Caldwell, R.A. Watson, How to get more from your model: the role of constructive selection in estimation of distribution algorithms, 2017, http://dx.doi.org/10.1145/3067695.3076049.

[38] M.S. Martins, M. El Yafrani, M.R. Delgado, M. Wagner, B. Ahiod, R. Lders, HSEDA: a heuristic selection approach based on estimation of distribution algorithm for the travelling thief problem, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 2017, pp. 361–368.

[39] T.D. Mihoc, A New Estimation of Distribution Algorithm for Nash Equilibria Detection, Springer, 2017, pp. 99–107.

[40] M. Zangari, A. Mendiburu, R. Santana, A. Pozo, Multiobjective decomposition-based mallows models estimation of distribution algorithm. a case of study for permutation flowshop scheduling problem, Inform. Sci. 397 (2017) 137–154.

[41] F. Lane, R.M.A. Azad, C. Ryan, DICE: a new family of bivariate estimation of distribution algorithms based on dichotomised multivariate Gaussian distributions, in: European Conference on the Applications of Evolutionary Computation, Springer, 2017, pp. 670–685.

[42] S. Baluja, Population-based Incremental Learning. A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Report, Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, 1994.

[43] C.W. Ahn, R.S. Ramakrishna, D.E. Goldberg, Real-coded Bayesian Optimization Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 51–73, http://dx.doi.org/10.1007/3-540-32494-1_3.

[44] Q.F. Zhang, A.M. Zhou, Y.C. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63, http://dx.doi.org/10.1109/tevc.2007.894202.

[45] L. Mo, G. Dai, J. Zhu, The RM-MEDA based on elitist strategy, Adv. Comput. Intell. (2010) 229–239.

[46] Y. Li, X. Xu, P. Li, L. Jiao, Improved RM-MEDA with local learning, Soft Comput. 18 (7) (2014) 1383–1397.

[47] D. Heckerman, D. Geiger, D.M. Chickering, Learning bayesian networks: the combination of knowledge and statistical data, Mach. Learn. 20 (3) (1995) 197–243, http://dx.doi.org/10.1023/a:1022623210503.

[48] O.R. Castro, R. Santana, A. Pozo, C-Multi: A competent multi-swarm approach for many-objective problems, Neurocomputing 180 (2016) 68–78, http://dx.doi.org/10.1016/j.neucom.2015.06.097.

[49] L. Mart, J. Garca, A. Berlanga, J.M. Molina, MONEDA: scalable multi-objective optimization with a neural network-based estimation of distribution algorithm, J. Global Optim. 66 (4) (2016) 729–768, http://dx.doi.org/10.1007/s10898-016-0415-7.

[50] L. Mart, J. Garca, A. Berlanga, C.A. Coello Coello, J.M. Molina, MB-GNG: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms, Oper. Res. Lett. 39 (2) (2011) 150–154, http://dx.doi.org/10.1016/j.orl.2011.01.002.

[51] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31, http://dx.doi.org/10.1109/tevc.2010.2059031.

[52] R. Denysiuk, L. Costa, I.E. Santo, Clustering-Based Selection for Evolutionary Many-Objective Optimization, in: Lecture Notes in Computer Science, vol. 8672, 2014, pp. 538–547.

[53] F. Lavinia, C. Corina, Multi-objective genetic algorithm with clustering-based ranking and direct control of diversity, in: 2013 17th International Conference on System Theory, Control and Computing, 2013, pp. 213–218.

[54] D. Pradhan, S. Wang, S. Ali, T. Yue, M. Liaaen, Ieee, CBGA-ES: A cluster-based genetic algorithm with elitist selection for supporting multi-objective test optimization, in: 2017 10th Ieee International Conference on Software Testing, Verification and Validation, in: IEEE International Conference on Software Testing Verification and Validation, 2017, pp. 367–378, http://dx.doi.org/10.1109/icst.2017.40.

[55] G.T. Pulido, C.A.C. Coello, Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer, in: Genetic and Evolutionary Computation - Gecco 2004, Pt 1, Proceedings, in: Lecture Notes in Computer Science, vol. 3102, Springer-Verlag Berlin, Berlin, 2004, pp. 225–237.

[56] C. von Lucken, C. Brizuela, B. Baran, Clustering Based Parallel Many-Objective Evolutionary Algorithms Using the Shape of the Objective Vectors, in: Lecture Notes in Computer Science, vol. 9019, 2015, pp. 50–64, http://dx.doi.org/10.1007/978-3-319-15892-1_4.

[57] O.R. Castro, A. Pozo, J.A. Lozano, R. Santana, An investigation of clustering strategies in many-objective optimization: the I-Multi algorithm as a case study, Swarm Intell. 11 (2) (2017) 101–130, http://dx.doi.org/10.1007/s11721-017-0134-9.

[58] M.M. Deza, E. Deza, Encyclopedia of distances, in: Encyclopedia of Distances, Springer, 2009, pp. 1–583.

[59] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, Neural Comput. 9 (7) (1997) 1493–1516, http://dx.doi.org/10.1162/neco.1997.9.7.1493.

[60] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, Springer London, London, 2005, pp. 105–145, http://dx.doi.org/10.1007/1-84628-137-7_6.

[61] J.J. Durillo, A.J. Nebro, jMetal: A Java framework for multi-objective optimization, Adv. Eng. Softw. 42 (10) (2011) 760–771, http://dx.doi.org/10.1016/j.advengsoft.2011.05.014.

[62] J.J. Durillo, A.J. Nebro, E. Alba, The jMetal framework for multi-objective optimization: Design and architecture, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, 2010, pp. 1–8.

[63] A.J. Nebro, J.J. Durillo, M. Vergne, Redesigning the jMetal multi-objective optimization framework, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, 2015, pp. 1093–1100.

[64] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657, http://dx.doi.org/10.1137/s1052623496307510.

[65] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, in: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, IEEE, pp. 1758–1763.

[66] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (3) (1994) 1–15.

[67] Z.A. He, G.G. Yen, Visualization and performance metric in many-objective optimization, IEEE Trans. Evol. Comput. 20 (3) (2016) 386–402, http://dx.doi.org/10.1109/tevc.2015.2472283.

[68] D.J. Walker, R.M. Everson, J.E. Fieldsend, Visualizing mutually nondominating solution sets in many-objective optimization, IEEE Trans. Evol. Comput. 17 (2) (2013) 165–184, http://dx.doi.org/10.1109/tevc.2012.2225064.

[69] M. Li, L. Zhen, X. Yao, How to read many-objective solution sets in parallel coordinates [educational forum], IEEE Comput. Intell. Mag. 12 (4) (2017) 88–100.