Knowledge-Based Systems 245 (2022) 108582

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection

Jeng-Shyang Pan [a,b], Nengxian Liu [b,*], Shu-Chuan Chu [a]

[a] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
[b] College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China

ARTICLE INFO

Article history:
Received 17 April 2021
Received in revised form 21 February 2022
Accepted 9 March 2022
Available online 17 March 2022

Keywords:
multi-objective algorithm
Differential evolution
Competitive mechanism
Feature selection

ABSTRACT

A large number of evolutionary algorithms have been introduced for multi-objective optimization problems in the past two decades. However, the compromise of convergence and diversity of the non-dominated solutions is still the main difficult problem faced by optimization algorithms. To handle this problem, an efficient competitive mechanism based multi-objective differential evolution algorithm (CMODE) is designed in this work. In CMODE, the rank based on the non-dominated sorting and crowding distance is first adopted to create the leader set, which is utilized to lead the evolution of the differential evolution (DE) algorithm. Then, a competitive mechanism using the shift-based density estimation (SDE) strategy is employed to design a new mutation operation for producing offspring, where the SDE strategy is beneficial to balance convergence and diversity. Meanwhile, two variants of the CMODE using the angle competitive mechanism and the Euclidean distance competitive mechanism are proposed. The experimental results on three test suites show that the proposed CMODE performs better than six state-of-the-art multi-objective optimization algorithms on most of the twenty benchmark functions in terms of hypervolume and inverted generation distance. Furthermore, the proposed CMODE is applied to the feature selection problem. The comparison results on feature selection also demonstrate the efficiency of our proposed CMODE.

1. Introduction

Multi-objective optimization problems (MOPs) exist widely in the practical applications, such as air traffic control system design [1], community detection problem [2], signal processing [3], and software engineering [4]. MOPs usually contains multiple conflicting objectives [5,6], such as the task scheduling problem which has two objectives, namely, reducing the makespan and enhancing resource utilization [7], and there does not have a single solution that is capable of optimizing all objectives at the same time. Therefore, MOPs are more difficult to be dealt with than the single-objective optimization problems.

In the past few decades, multi-objective evolutionary algorithms (MOEAs) have obtained increasing interests since MOEAs can handle MOPs effectively. Since Schaffer introduced the vector evaluated genetic algorithm (VEGA) [8], plenty of MOEAs have been presented on the basis of various stochastic meta-heuristic algorithms, such as genetic algorithm (GA) [9], firefly algorithm [10], particle swarm optimization (PSO) [11], whale optimization algorithm [12], sine cosine algorithm (SCA) [13] and differential evolution (DE) algorithm [14]. Generally, MOEA can be roughly grouped into three types based on their different selection mechanisms: dominance-based MOEAs, decomposition-based MOEAs and indicator-based MOEAs.

The dominance-based MOEAs evaluate solutions based on dominance relationship and select solutions according to Pareto-based and diversity-based selection criteria. The first MOEA using Pareto was introduced by Goldberg et al. in 1989 [15]. Its essential idea is to find all Pareto front solutions in the current population via the dominance-based relationship. Srinivas et al. presented multi-objective GA via non-dominated sorting (NSGA) [16]. To reduce the difficulty in setting the parameters of NSGA, Deb et al. presented a representative and famous MOEA, i.e., the fast elite multi-objective GA (NSGA-II) [17]. What is more, to design approaches for solving many objective optimization problems, researchers modified the dominance rule in various ways, such as ϵ-dominance [18], fuzzy dominance [19], and grid dominance [20]. Recently, Zhang et al. proposed multi-objective PSO with competitive mechanism, called CMOPSO, which uses non-dominated sorting along with crowding distance based ranking to select elite particles [21]. The experiment results have shown that CMOPSO outperformed the compared algorithms. Using the same elite particles as CMOPSO, Zhang et al. presented an MOEA, called EMOSO, based on a level swarm optimizer [22]. Got et al. [23]

* Corresponding author.
E-mail addresses: jspan@cc.kuas.edu.tw (J.-S. Pan), lylnx@fzu.edu.cn (N. Liu), scchu0803@gmail.com (S.-C. Chu).

https://doi.org/10.1016/j.knosys.2022.108582
0950-7051/

proposed a multi-objective manta ray foraging optimizer to address engineering design problems, in which an external-archive is utilized to save the non-dominated solutions achieved so far.

The decomposition-based MOEAs are represented by MOEA/D [24], where the problem is transformed into a series of single-objective sub-problems with the use of an aggregation function, and each sub-problem is solved with the neighborhood information. The same authors of MOEA/D further proposed a new version of MOEA/D based on DE, named MOEA/D-DE [25]. Furthermore, some variants of MOEA/D are presented to solve more complex MOPs, such as MOEA/D-AM2M [26] with adaptively allocating search effort and MOEA/D-PaS [27] with Pareto adaptive scalarizing methods. Based on decomposition strategy, Cai et al. [28] introduced a multi-objective PSO (MPSOD), which adopted a special set of direction vectors to enable each subspace to have a solution, and to preserve the diversity. Cheng et al. [29] introduced an MOEA guided by a set of reference vectors, called RVEA, and Zhao et al. [30] proposed surrogate-ensemble assisted MOEA based on RVEA to solve expensive problems. Recently, to solve constrained project scheduling problem, Zhu et al. [31] developed a multi-objective genetic programming hyper-heuristic approach based on decomposition strategy. In [32], a classification-based surrogate-assisted MOEA was developed for expensive optimization.

Regard to indicator-based MOEAs, the performance indicators are utilized to environmental selection and to guide the evolution process. Zitzler et al. [33] presented indicator-based evolutionary algorithm (IBEA). In IBEA, an indicator (a binary performance measure) is directly adopted in the process of selection. Beume et al. [34] proposed an indicator-based MOEA (SMS-EMOA) by combining hypervolume measure with non-dominated sorting concept. Bader et al. [35] presented hypervolume-based MOEA, called Hype, which utilizes the Monte Carlo simulation to estimate the hypervolume values, and to reduce the calculation cost. Tian et al. [36] introduced an indicator-based MOEA with reference point adaptation (AR-MOEA) to enhance the performance of irregular Pareto frontier problems. In AR-MOEA, a new indicator is proposed and the position of reference points are adjusted according to the contributions of solutions in computing indicator.

In addition, several scholars tried to take advantage of the merits provided by the above three categories. Wang et al. [37] introduced a Two-arch2 algorithm via using both the dominance and the performance. Based on decomposition and dominance mechanisms, Li et al. [38] proposed an MOEA/DD algorithm. Deb et al. extended the famous NSGA-II and presented the NSGA-III with the help of non-dominated sorting mechanism as well as decomposition strategy. Although a lot of MOEAs have made great efforts in the improvement of convergence performance and diversity preservation and obtained better performance, the trade-off of diversity and convergence remains a great challenge in multi-objective optimization. Because there is no common criterion for balancing conflicting objectives, and concerns will vary according to the features of different problems. On the other hand, according to the classical No Free Lunch theorem [39], there does not exist a single optimization method which can address all types of optimization problems [40]. These motivate us to propose new MOEAs to better deal with different types of MOPs.

To further improve the effectiveness of MOEA in handling MOPs, we introduce a multi-objective DE algorithm with the competitive mechanism in this paper. The idea of competition adopted in the competitive swarm optimizer [41] is an effectiveness mechanism which has also been adopted to improve other algorithms' performance [12,21,42,43]. Paper [41] has demonstrated that the competitive mechanism can obtain a better trade-off between convergence and diversity than classical PSO via theoretical analysis and empirical results. On the other hand, DE is one of the powerful population-based algorithms [44–46], which has not be integrated with this competitive mechanism to deal with MOPs. Inspired by the competitive mechanism and to take advantages of DE, a competitive mechanism based multi-objective DE, called CMODE, is proposed in this paper. In CMODE, a leader set is first created to lead the evolution of DE by using the rank based on the non-dominated sorting and crowding distance. Then, a new competitive mechanism based mutation operation is proposed to generate offspring. Finally, the environmental selection is utilized to select individuals from parent and offspring populations to the next population. The main contributions of this work are given as follows.

(1) A competitive mechanism based on the shift-based density estimation (SDE) strategy is adopted to design a novel DE mutation strategy for generating offspring. In this strategy, an individual with better SDE value is used to guide the evolution process. Different from the existing DE mutation strategies, this mutation strategy can take advantages of the SDE competitive mechanism to achieve a well trade-off between diversity and convergence.
(2) Two variants of the CMODE are proposed, which use the angle competitive mechanism and Euclidean distance competitive mechanism, respectively. Experiments are done to compare the performance of these two variants with CMODE and experimental results demonstrate the effectiveness of the competitive mechanism with SDE.
(3) The performance of our proposed CMODE is comprehensively assessed by making a comparison with six popular MOEAs on twenty benchmarks with different shape of Pareto front. Experimental results indicate that our CMODE can obtain better results in terms of convergence and diversity on most of the benchmarks. Moreover, we further apply our proposed CMODE to solve the feature selection problem. The comparison results of the feature selection problem also reveal the competitive performance of our proposed algorithm.

The paper is structured as follows. Section 2 briefly presents preliminaries and related work about multi-objective DE. The details of our proposed CMODE is described in Section 3. Section 4 gives the empirical results on three test suites with some discussions. The application of our proposed CMODE to feature selection is presented in Section 5. Finally, Section 6 draws the conclusions.

2. Preliminaries and related work

2.1. Multi-objective optimization problems (MOPs)

Multi-objective optimization problems (MOPs) contain two or more objectives to be optimized simultaneously and these objectives often conflict with others. The minimum MOP is described as follows:

$$\begin{aligned} \min \quad & F(X) = (f_1(X), f_2(X), \ldots, f_m(X)) \\ s.t. \quad & g_p(X) \le 0 \quad (p = 1, 2, \ldots, k) \\ & h_q(X) = 0 \quad (q = 1, 2, \ldots, k) \end{aligned} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_D)$ represents the D-dimensional vector in decision space and m is the number of objectives, $f_i(X)$, $i = 1, 2, \ldots, m$ are objective functions in m-dimensional objective space. g_p and h_q represents inequality and equality constraints, respectively. Given two feasible solutions X_a and X_b, X_a dominates X_b, if and only if for $\forall i$, $f_i(X_a) \le f_i(X_b)$ and $\exists j$, $f_j(X_a) < f_j(X_b)$, $i, j \in \{1, 2, \ldots, m\}$. If no other solution dominates X^*, then X^* is known as a Pareto-optimal solution. All the Pareto-optimal solutions constitute the Pareto-optimal set (PS) and their objective values constitute the Pareto front (PF).

2.2. Differential evolution (DE)

DE is a population-based search method, introduced by Price and Storn [47]. It adopts three main operators (i.e. mutation, crossover and selection) to optimize problems. Initially, a population with ps individuals (i.e. initial population) is generated randomly using Eq. (2), and then the mutation operation (one of the Eqs. (3)–(7)) and crossover operation (Eq. (8)) are utilized to create ps trial vectors, and finally Eq. (9) is adopted to select ps new individuals (known as offsprings) from the parents and trial vectors.

$$\begin{aligned} &X_{i,g} = (x_{i,1,g}, x_{i,2,g}, \ldots, x_{i,D,g}), \quad i = 1, 2, \ldots, ps \\ &x_{i,j,0} = x_{\min,j} + rand(0,1).(x_{\max,j} - x_{\min,j}), \quad j = 1, 2, \ldots, D \end{aligned} \tag{2}$$

where g, i and j are the generation number, the index of the individual and the variable index, respectively. $x_{\max,j}$ and $x_{\min,j}$ stand for the upper and lower bounds of the $x_{i,j}$.

After initiation, DE firstly adopts mutation operation to produce a mutant vector $V_{i,g}$ for each individual (target vector) in the population. Five common mutation schemes are summarized as follows [44].

$$\text{DE/rand/1:} \quad V_{i,g} = X_{r1,g} + F.(X_{r2,g} - X_{r3,g}) \tag{3}$$

$$\text{DE/rand/2:} \quad V_{i,g} = X_{r1,g} + F.(X_{r2,g} - X_{r3,g}) + F.(X_{r4,g} - X_{r5,g}) \tag{4}$$

$$\text{DE/best/1:} \quad V_{i,g} = X_{best,g} + F.(X_{r1,g} - X_{r2,g}) \tag{5}$$

$$\text{DE/best/2:} \quad V_{i,g} = X_{best,g} + F.(X_{r1,g} - X_{r2,g}) + F.(X_{r3,g} - X_{r4,g}) \tag{6}$$

$$\begin{aligned} \text{DE/current-to-best/1:} \quad V_{i,g} = X_{i,g} + F.(X_{best,g} - X_{i,g}) \\ + F.(X_{r1,g} - X_{r2,g}) \end{aligned} \tag{7}$$

where F and $X_{best,g}$ indicate the mutation factor in (0, 1) and the best vector of generation g, respectively. $r_j \in \{1, 2, \ldots, ps\}$, r_j is different to each other and $r_j \neq i, j = 1, 2, \ldots, 5$.

Then, the crossover operation is employed to blend the mutant vector with the target vector to create a trial vector $U_{i,g}$ via using the following equation.

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if}(rand(0,1) \leq Cr \quad or \quad j = j_{rand}) \\ x_{i,j,g} & \text{otherwise} \end{cases} \tag{8}$$

where Cr is crossover probability in [0, 1] and $j_{rand} \in 1, 2, \ldots, D$ is adopted to Eq. (9) to assure that $U_{i,g}$ is different to $X_{i,g}$.

Lastly, the selection operation is utilized to compare trial vector with target vector via their objective values and the better one will survive to the next generation. As for the minimization function, the selection equation is as follows.

$$X_{i,g+1} = \begin{cases} U_{i,g} & if\ f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} & otherwise \end{cases} \tag{9}$$

The above-mentioned three evolution operations continues until a stopping criteria is met.

2.3. Existing multi-objective DE algorithms

Many DE algorithms have been formulated by the scholars to deal with MOPs in the past few decades. In this section, some representative multi-objective DE (MODEs) algorithms are briefly reviewed.

Chang et al. first extended the DE to handle MOPs based on the idea of Pareto dominance [48]. The authors adopted an external archive to record the non-dominated solutions during the evolution and incorporated fitness sharing to reserve diversity. Abbass et al. [49] introduced a Pareto-frontier DE (PDE) algorithm for solving MOPs and obtained competitive results in comparing with some other MOEAs in literature. In [50], Abbass further improved the PDE with self-adaptive strategy (SPDE). Lampinen et al. [51] proposed generalized differential evolution (GDE) by extending DE/rand/1/bin. After this paper, two improved versions of GDE (i.e. GDE2 [52] and GDE3 [14]) have been introduced by Kukkonen et al.

Xue et al. [53] proposed a multi-objective DE (MODE) which incorporates the best individual to create the offspring, where the best individual is selected by the Pareto-based approach. Based on the framework of NSGA-II, Iorio et al. [54] designed the non-dominated sorting DE (NSDE). Based on decomposition, Li and Zhang [25] introduced a multi-objective DE algorithm (MOEA/D-DE) to solving continuous MOPs. In this approach, DE/rand/1/bin scheme is employed to produce new trial solutions. Zhao et al. [55] adopted an ensemble strategy to replace the tuning of neighborhood size of the MOEA/D-DE and proposed an improved approach which can obtain an overall better performance. Robic et al. [56] designed a DE for multi-objective optimization (named DEMO), which integrates the advantages of DE with the ranking mechanisms of NSGA-II. Rakshit et al. [57] extended this DEMO to address MOPs in a noisy environment. To solve MOPs with variable linkages, Jiang et al. [58] introduced a Multi-objective DE with dynamic covariance matrix learning (DCML), where DCML is used to establish a proper coordinate system for the crossover operation. In [59], Saini et al. proposed a self-organizing multi-objective DE to extract single document summarization. Wang et al. [60] proposed a MODE with a set of personal archives as well as biased self-adaptive mutation selection (called BiasMOSaDE). Computational results demonstrated that BiasMOSaDE has a competitive performance. Tian et al. developed a novel DE with the neighborhood information and a restart mechanism to effectively balance diversity and convergence [61]. Jamali [62] introduced a multi-objective DE algorithm with fuzzy inference-based dynamic adaptive mutation factor to solve the Pareto optimization of problems. Altay et al. [63] introduced a novel hybrid multi-objective approach with DE and SCA for numerical association rule mining. Yue et al. [64] designed a multi-objective DE to address multi-modal MOPs, where two separate crowding distance calculation approaches are utilized to objective space and decision space, respectively.

More reviews about MODEs can be found in [44]. Literature reviews show that no works use the competitive mechanism to design the DE for multi-objective optimization problems.

3. The proposed CMODE

In this part, our proposed CMODE is described in detail, which aims to achieve well balance between diversity and convergence of non-dominance solutions. First, The complete framework of the algorithm CMODE is provided. Then the novel mutation strategy is designed based on the SDE competition mechanism. Finally, we will give two variants of the proposed CMODE.

3.1. The framework of CMODE

The proposed CMODE has a concise framework as presented in Algorithm 1, where the main cycle contains two parts, i.e., DE evolution with the competition mechanism and environment selection. Note that CMODE employs the same environment selection strategy as that of SPEA2 [65] in the evolution process. To make it more clear, Fig. 1 gives the main framework of our proposed CMODE.

The procedure to implement the CMODE is given as follows. First, the population is randomly initialized. Then the individuals in the population $\boldsymbol{P}$ are updated via the proposed DE mutation strategy and crossover operation to generate offspring $\boldsymbol{P}'$. Finally, the environmental selection is performed on the parent $\boldsymbol{P}$ and offspring $\boldsymbol{P}'$. The above procedure (i.e., DE evolution and environmental selection) stops when the termination criteria is met.

Algorithm 1 The framework of the proposed CMODE

Input: population size ps, dimension of the problem D, $MaxNFE$, NFE.
Output: All non-dominant individuals in $\boldsymbol{P}$.
Randomly initialize the population $\boldsymbol{P}$
while ($NFE < MaxNFE$) **do**
  $\boldsymbol{P}' \leftarrow$ CompetitionBasedDEevolution($\boldsymbol{P}$) ;
  $\boldsymbol{P} \leftarrow$ EnvironmentalSelection($\boldsymbol{P}, \boldsymbol{P}'$) ;
end while
Return $\boldsymbol{P}$

3.2. The competition mechanism based DE evolution

There are four parts in our proposed competition mechanism based DE evolution, namely, creating the leader set L, fitness evaluation, pairwise competition and DE evolution. The pseudo code for our proposed competition mechanism based DE evolution is provided in Algorithm 2.

Algorithm 2 CompetitionBasedDEevolution($\boldsymbol{P}$)

Input: current population $\boldsymbol{P}$, size of leader set α, $F = 0.5$, $Cr = 0.5$.
Output: trial population (offspring) $\boldsymbol{P}'$.
1: $\boldsymbol{P}' \leftarrow \Phi$;
2: /*Selecting the elite individuals to form the leader set*/;
3: $\boldsymbol{L} \leftarrow$ Select α individuals from $\boldsymbol{P}$ based on the front index and crowding distance.
4: $Fitness \leftarrow$ Calculate the fitness for each individual in $\boldsymbol{L}$ by Eq. (10);
5: **for** each individual X_i in $\boldsymbol{P}$ **do**
6:   $\{k, l\} \leftarrow$ Randomly select two individuals from $\boldsymbol{L}$
7:   **if** $Fitness(k) < Fitness(l)$ **then**
8:     $X_l = k$; //loser of the competition
9:     $X_w = l$; //winner of the competition
10:   **else**
11:     $X_l = l$;
12:     $X_w = k$;
13:   **end if**
14:   Generate the mutation individual V_i by Eq.(11);
15:   Generate the trial individual U_i by Eq.(8);
16:   $\boldsymbol{P}' \leftarrow \boldsymbol{P}' \bigcup \{U_i\}$;
17: **end for**
18: **Return** $\boldsymbol{P}'$

Firstly, the leader set is created (line 3 in Algorithm 2). The leader set plays important role in our proposed DE mutation strategy, since it offers the candidate individuals to be adopted in the pairwise competition to guide the evolution of the population. The leader set is similar to the elite particle set in CMOPSO. Both of them use the method proposed in NSGA-II [17] to create the leader set, namely, individuals in the leader set are selected according to the non-dominated sorting as well as crowding distance based ranking. Note that the individuals in the leader set are directly selected from the current population so that CMODE does not need the external archive to store non-dominated solutions. It is also worth noting that the size α of the leader set has a great impact on the performance of the algorithm CMODE, and we will analyze it in the experimental part.

Then, we calculate the fitness value of each individual in the leader set $\boldsymbol{L}$ (line 4 in Algorithm 2), which employs the shift-based density estimation (SDE) approach [66]. Specifically, the minimum SDE-based distance (Eq. (10)) [43] is used to define the fitness of an individual p.

$$Fitness(p) = \min_{q \in P \setminus \{p\}} \sqrt{\sum_{i=1}^{m} (\max\{0, f_i(q) - f_i(p)\})^2} \tag{10}$$

where m is the number of objectives and $f_i(p)$ denotes the ith objective value of p. The SDE approach is able to assess the quality of a candidate solution in terms of both diversity and convergence, and has also been adopted in several MOEAs [22,43]. Hence, the SDE-based distance is adopted in our proposed CMODE to evaluate the convergence and diversity of each individual in the population.

Algorithm 3 CompetitionBasedDEevolution_Angle($\boldsymbol{P}$)

Input: current population $\boldsymbol{P}$, size of leader set α, $F = 0.5$, $Cr = 0.5$.
Output: trial population (offspring) $\boldsymbol{P}'$.
$\boldsymbol{P}' \leftarrow \Phi$;
/*Selecting the elite individuals to form the leader set*/;
$\boldsymbol{L} \leftarrow$ Select α individuals from $\boldsymbol{P}$ based on the front index and crowding distance.
for each individual X_i in $\boldsymbol{P}$ **do**
  $\{k, l\} \leftarrow$ Randomly select two individuals from $\boldsymbol{L}$
  calculate the angle φ_1 between k and X_i, φ_2 between l and X_i
  if $\varphi_2 < \varphi_1$ **then**
    $X_l = k$; //loser of the competition
    $X_w = l$; //winner of the competition
  else
    $X_l = l$;
    $X_w = k$;
  end if
  Generate the mutation individual V_i by Eq.(11);
  Generate the trial individual U_i by Eq.(8);
  $\boldsymbol{P}' \leftarrow \boldsymbol{P}' \bigcup \{U_i\}$;
end for
Return $\boldsymbol{P}'$

Afterward, two individuals are randomly chosen from the leader set $\boldsymbol{L}$ and a pair competition is performed (lines 6–13 in Algorithm 2). The one with larger fitness value is denoted as the winner X_w and the other one is denoted as the loser X_l. Then these two individuals are utilized to take part in the mutation operation (line 14 in Algorithm 2). The proposed mutation operation is defined as below.

$$\text{DE/current-to-competition/1:} \quad V_{i,g} = X_{i,g} + F.(X_{w,g} - X_{i,g}) + F.(X_{w,g} - X_{l,g}) \tag{11}$$

The proposed mutation operation can take advantages of individuals in the leader set. Since these individuals are selected by the method proposed in NSGA-II and they are more close to the Pareto front with less density. On the other hand, the SDE-based competition can select the better individual to guide the search of evolution so as to maintain the well balance between convergence and diversity.

After the mutation operation, the crossover operation is performed to further improved the diversity of the population (line

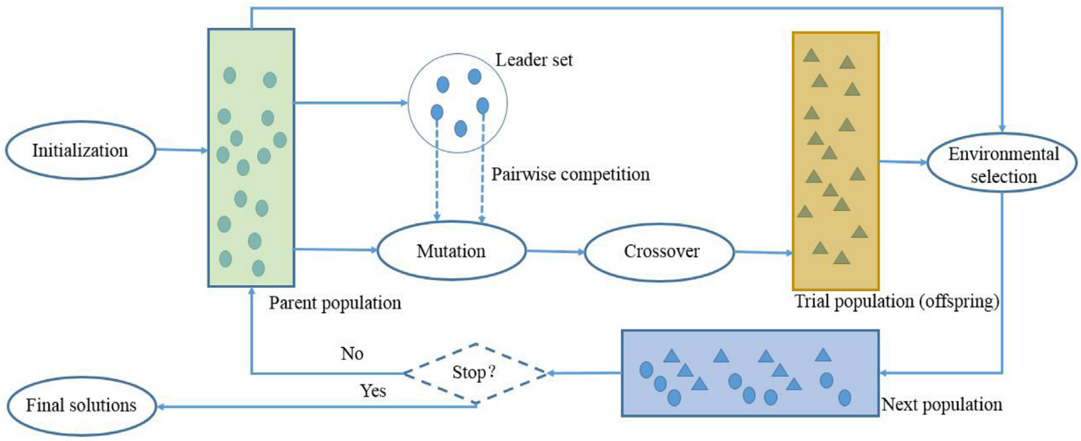


Fig. 1. The framework of the proposed CMODE.

15 in Algorithm 2). Finally, the generated trial vector U_i is put into the population P'. Note that the original DE selection operation is not performed in our proposed CMODE. Instead, after the evolution of DE, the procedure will return to Algorithm 1 to perform the multi-objective environment selection operation.

3.3. Two variants of the proposed CMODE

Algorithm 4 CompetitionBasedDEevolution_Distance($\boldsymbol{P}$)

Input: current population $\boldsymbol{P}$, size of leader set α, $F = 0.5$, $Cr = 0.5$.
Output: trial population (offspring) $\boldsymbol{P'}$.

$\boldsymbol{P'} \leftarrow \Phi$;
/*Selecting the elite individuals to form the leader set*/;
$\boldsymbol{L} \leftarrow$ Select α individuals from $\boldsymbol{P}$ based on the front index and crowding distance.
Distance $\leftarrow$ Calculate the Euclidean distance for each individual in $\boldsymbol{L}$;
for each individual X_i in $\boldsymbol{P}$ **do**
  $\{k, l\} \leftarrow$ Randomly select two individuals from $\boldsymbol{L}$
  if $Distance(l) < Distance(k)$ **then**
    $X_l = k$; //loser of the competition
    $X_w = l$; //winner of the competition
  else
    $X_l = l$;
    $X_w = k$;
  end if
  Generate the mutation individual V_i by Eq.(11);
  Generate the trial individual U_i by Eq.(8);
  $\boldsymbol{P'} \leftarrow \boldsymbol{P'} \bigcup \{U_i\}$;
end for
Return $\boldsymbol{P'}$

The angle between two individuals (vectors) and the Euclidean distance between the individual and origin are two other strategies which can also be used to compare the quality of a candidate solutions [21,29]. Base on these two strategies, we proposed two variants of the proposed CMODE, which are called CMODEA and CMODED, respectively. The evolution pseudo codes of these two variants are presented in Algorithms 3 and 4, respectively. In general, the angle strategy is used to measure the diversity of the solutions, and the distance strategy is used to measure convergence of the solutions. However, the SDE strategy is able to measure the quality of a solution in terms of both diversity and convergence. Therefore, the proposed CMODE has better performance than the other two variants and this conclusion will be confirmed in the experimental section.

4. Experimental results

4.1. Benchmark functions and performance measures

To evaluate the performance of our proposed CMODE, we compare it with six representative methods: NSGAII [17], MOEA/D [24], MOEA/D-DE [25], GDE3 [14], CMOPSO [21], EMOSO [22]. Among these methods, NSGAII and MOEA/D are well-known MOEAs using GA operators, MOEA/D-DE and GDE3 are two classical MOEAs using DE operators, and CMOPSO and EMOSO are recently proposed MOEAs using PSO operators. All these methods are coded with Matlab and embedded into the platform PlatEMO by Tian et al. [67]. Twenty benchmark functions from three multi-objective test suits, DTLZ [68], WFG [69] and ZDT [70] are employed to validate the performance of the methods. Among them, test suites DTLZ and WFG are variable objective functions, and test suite ZDT is two-objective functions. For two-objective functions, the dimension of decision variables D is set to 6 for DTLZ1, to 11 for WFG1-WFG9 and DTLZ2-DTLZ6, 21 for DTLZ7, 30 for ZDT1-ZDT3 and 10 for ZDT4. Regard to three-objective functions, the dimension of decision variables D is set to 7 for DTLZ1, to 12 for WFG1-WFG9 and DTLZ2-DTLZ6, and 22 for DTLZ7.

Two commonly used performance metrics, namely, inverted generational distance (IGD) [71] and hypervolume (HV) [72], are utilized to compare CMODE with other MOEAs. For computing the IGD values, about 10 000 reference points are selected from the real Pareto front of each test function. The lower the IGD value, the superior performance of the MOEA is. For calculating the HV values, the reference points are set to $(1, 1, \ldots, 1)$ and all objectives are normalized. The higher the HV value, the superior performance of the MOEA is.

4.2. Experimental settings

For the sake of fairness, all parameters of the competing methods are consistent with the their original papers. In our experiments, the parameter α, F and Cr of the proposed CMODE are set to 5, 0.5, 0.5, respectively. We will analysis the value of α in the following section and the values of F and Cr are set according to the paper [47]. The population size ps is set to 100 for all competing algorithms. The maximum number of function evaluations (NFEs) is utilized as the termination criteria for all competing algorithms. The maximum NFEs is set to 30 000 for

Table 1
Comparison of IGD values obtained by CMODE and other competing MOEAs.

Problem	M	NSGA-II	MOEA/D	MOEA/D-DE	GDE3	CMOPSO	EMOSO	CMODE
DTLZ1		2.4332e−3 (2.97e−4)+	**2.2777e−3 (3.71e−4)**+	8.9472e−3 (2.94e−2)+	3.4723e+0 (1.12e+0)−	5.2498e−1 (1.24e+0)+	1.7905e+1 (4.70e+0)−	1.3739e+0 (2.01e+0)
DTLZ2		5.1329e−3 (2.17e−4)−	**3.9669e−3 (2.27e−6)**+	3.9783e−3 (4.01e−6)+	5.0681e−3 (1.41e−4)−	4.3947e−3 (7.46e−5)−	4.5772e−3 (6.16e−5)−	4.0931e−3 (3.07e−5)
DTLZ3		**1.5984e−1 (4.31e−1)**+	2.0680e−1 (3.70e−1)+	7.4414e+0 (1.07e+1)+	3.4466e+1 (2.59e+1)+	3.4325e+1 (2.23e+1)+	1.8297e+2 (1.02e+1)−	6.0261e+1 (2.19e+1)
DTLZ4	2	2.0158e−1 (3.32e−1)−	2.2542e−1 (3.44e−1)−	**4.1041e−3 (9.96e−5)**+	5.1047e−3 (1.31e−4)−	3.4868e−1 (3.74e−1)−	1.2751e−1 (2.80e−1)−	4.2875e−3 (2.30e−4)
DTLZ5		5.1328e−3 (2.31e−4)−	**3.9666e−3 (5.87e−7)**+	3.9799e−3 (9.80e−6)+	5.0582e−3 (1.23e−4)−	4.4040e−3 (5.70e−5)−	4.5757e−3 (4.07e−5)−	4.0944e−3 (2.87e−5)
DTLZ6		5.7630e−3 (3.07e−4)+	**3.9661e−3 (4.93e−8)**+	3.9664e−3 (8.92e−8)+	4.3246e−3 (4.24e−5)+	4.1212e−3 (3.64e−5)+	4.5003e−3 (4.10e−5)+	3.1114e−2 (1.48e−1)
DTLZ7		5.3575e−3 (2.12e−4)−	2.1720e−2 (7.95e−2)−	2.1074e−1 (2.21e−1)−	5.1907e−1 (1.03e−1)−	6.2949e−2 (1.52e−1)−	1.9194e−2 (8.01e−2)−	**4.3741e−3 (3.25e−5)**
WFG1		**1.0920e−1 (3.51e−2)**+	1.4820e−1 (1.20e−1)+	7.2070e−1 (1.33e−1)−	1.4305e+0 (3.73e−2)−	9.0611e−1 (6.24e−2)−	1.4794e+0 (3.99e−2)−	2.3619e−1 (1.25e−1)
WFG2		1.3222e−2 (5.53e−4)−	3.6143e−2 (4.15e−2)−	2.2733e−2 (9.23e−4)−	2.0481e−2 (2.40e−3)−	**1.1780e−2 (3.43e−4)**+	1.3666e−2 (6.07e−4)−	1.2549e−2 (1.28e−3)
WFG3		1.5565e−2 (8.36e−4)−	1.5565e−2 (2.28e−3)−	1.5726e−2 (6.42e−4)−	1.9475e−2 (9.00e−4)−	1.3913e−2 (3.94e−4)−	1.4196e−2 (4.27e−4)−	**1.1670e−2 (5.07e−5)**
WFG4		1.5764e−2 (4.57e−4)=	**1.5639e−2 (5.42e−3)**+	5.2290e−2 (7.13e−3)−	7.1891e−2 (5.25e−3)−	5.1851e−2 (1.14e−2)−	6.9515e−2 (5.26e−3)−	1.5845e−2 (1.60e−3)
WFG5	2	**6.5451e−2 (1.18e−3)**+	6.9768e−2 (6.61e−4)−	6.8933e−2 (1.75e−3)−	6.7494e−2 (2.19e−3)=	6.8073e−2 (2.23e−3)=	6.5917e−2 (1.87e−3)=	6.7266e−2 (2.35e−3)
WFG6		7.6991e−2 (2.14e−2)−	8.8523e−2 (3.37e−2)−	7.0625e−2 (8.73e−2)=	5.4425e−2 (7.82e−2)=	1.8380e−2 (5.80e−3)+	**1.7788e−2 (5.33e−3)**+	3.4654e−2 (1.67e−2)
WFG7		1.7112e−2 (6.49e−4)−	1.5001e−2 (5.02e−3)−	1.4250e−2 (3.63e−4)−	1.9468e−2 (8.88e−4)−	1.4449e−2 (2.87e−4)−	1.4473e−2 (3.36e−4)−	**1.2665e−2 (1.41e−4)**
WFG8		1.1124e−1 (1.32e−3)−	1.1440e−1 (9.35e−3)−	1.0649e−1 (3.91e−3)+	**1.0354e−1 (1.09e−3)**+	1.1750e−1 (4.15e−3)−	1.2232e−1 (2.62e−3)−	1.0855e−1 (8.64e−4)
WFG9		2.8700e−2 (3.74e−2)+	4.0506e−2 (4.10e−2)=	2.7006e−2 (2.42e−3)=	1.2254e−1 (1.02e−1)−	**2.6775e−2 (1.69e−3)**=	3.5712e−2 (3.65e−2)=	1.0026e−1 (9.82e−2)
ZDT1		4.8083e−3 (1.90e−4)−	4.1704e−3 (4.26e−4)−	1.1806e−2 (3.25e−3)−	1.9663e−1 (4.26e−2)−	4.1572e−3 (9.56e−5)−	4.6524e−3 (2.12e−4)−	**3.8410e−3 (3.84e−5)**
ZDT2	2	4.8328e−3 (1.56e−4)−	3.9526e−3 (2.10e−4)−	7.7468e−3 (2.06e−3)−	5.6996e−1 (1.46e−1)−	4.1105e−3 (7.11e−5)−	4.2531e−3 (5.89e−5)−	**3.8487e−3 (3.24e−5)**
ZDT3		6.7100e−3 (5.46e−3)−	1.2662e−2 (6.36e−3)−	1.8190e−2 (7.81e−3)−	2.3651e−1 (6.67e−2)−	**4.6404e−3 (4.28e−5)**+	5.7274e−3 (3.16e−4)−	5.2390e−3 (6.43e−4)
ZDT4		**5.5198e−3 (1.16e−3)**+	6.4124e−3 (1.38e−3)+	2.2597e−1 (2.10e−1)+	2.1633e+1 (4.08e+0)−	3.1523e−1 (3.49e−1)+	1.7666e+1 (4.91e+0)−	3.6907e+0 (2.18e+0)
DTLZ1		**2.7584e−2 (1.39e−3)**+	3.0663e−2 (1.99e−4)+	1.0014e−1 (2.47e−1)=	2.3985e+0 (1.11e+0)−	7.9577e+0 (5.01e+0)−	1.5982e+1 (4.19e+0)−	4.3335e−1 (4.31e−1)
DTLZ2		6.8855e−2 (2.39e−3)−	7.5179e−2 (3.16e−4)−	7.5995e−2 (8.15e−4)−	8.0096e−2 (2.85e−3)−	5.7734e−2 (1.08e−3)−	5.8297e−2 (8.73e−4)−	**5.3833e−2 (6.27e−4)**
DTLZ3		3.2837e−1 (6.22e−1)+	**1.8096e−1 (2.89e−1)**+	6.4828e+0 (1.35e+1)+	1.9168e+1 (2.08e+1)+	6.8951e+1 (2.93e+1)−	1.7985e+2 (1.57e+1)−	4.0432e+1 (2.28e+1)
DTLZ4	3	9.7076e−2 (1.60e−1)−	3.7871e−1 (2.71e−1)−	1.6188e−1 (8.78e−2)−	8.0505e−2 (3.20e−3)−	**6.0861e−2 (1.79e−3)**+	9.5827e−2 (1.45e−2)−	6.0882e−2 (3.70e−2)
DTLZ5		5.7727e−3 (3.27e−4)−	1.4595e−2 (1.41e−5)−	1.4371e−2 (1.11e−4)−	6.2018e−3 (2.83e−4)−	6.3774e−3 (6.27e−4)−	7.5052e−3 (6.17e−4)−	**4.1533e−3 (3.14e−5)**
DTLZ6		5.7830e−3 (2.73e−4)−	1.4599e−2 (9.15e−6)−	1.4504e−2 (4.49e−5)−	4.5107e−3 (6.62e−5)−	4.1963e−3 (4.10e−5)−	4.5875e−3 (1.14e−4)−	**4.1597e−3 (4.20e−5)**
DTLZ7		7.6933e−2 (4.37e−3)−	2.6326e−1 (1.28e−1)−	2.4785e−1 (1.67e−1)−	7.7795e−1 (2.45e−1)−	2.0569e−1 (2.35e−1)−	6.4936e−2 (1.97e−3)−	**5.9853e−2 (1.20e−3)**
WFG1		3.6794e−1 (4.64e−2)+	**3.1371e−1 (2.21e−2)**+	1.4717e+0 (3.75e−2)−	1.6373e+0 (1.83e−2)−	1.5117e+0 (2.18e−2)−	1.8754e+0 (3.28e−2)−	4.4350e−1 (9.37e−2)
WFG2		2.2104e−1 (1.09e−2)−	3.0327e−1 (4.84e−3)−	3.4241e−1 (2.02e−2)−	2.4445e−1 (9.71e−3)−	1.8109e−1 (5.62e−3)=	**1.6883e−1 (3.51e−3)**+	1.7893e−1 (4.36e−3)
WFG3		1.0690e−1 (1.38e−2)=	1.1093e−1 (5.57e−3)−	1.7592e−1 (3.68e−2)−	2.2659e−1 (2.86e−2)−	1.5547e−1 (9.05e−3)−	2.3205e−1 (1.71e−2)−	**1.0224e−1 (1.17e−2)**
WFG4		2.6964e−1 (7.44e−3)−	3.5553e−1 (3.93e−3)−	3.8841e−1 (9.70e−3)−	3.4330e−1 (1.43e−2)−	2.6166e−1 (4.42e−3)−	2.6476e−1 (4.87e−3)−	**2.1593e−1 (3.27e−3)**
WFG5	3	2.8019e−1 (1.11e−2)−	3.5951e−1 (7.60e−3)−	3.3948e−1 (5.36e−3)−	2.8387e−1 (1.71e−2)−	2.4932e−1 (5.88e−3)−	2.4949e−1 (9.41e−3)−	**2.2292e−1 (2.00e−3)**
WFG6		3.1229e−1 (1.83e−2)−	3.7980e−1 (1.23e−2)−	3.9676e−1 (3.57e−2)−	3.6156e−1 (3.15e−2)−	2.4251e−1 (7.14e−3)−	2.2711e−1 (5.76e−3)−	**2.2150e−1 (2.37e−2)**
WFG7		2.7850e−1 (8.00e−3)−	3.5580e−1 (4.50e−3)−	3.6203e−1 (4.82e−3)−	3.4531e−1 (1.17e−2)−	2.3355e−1 (4.76e−3)−	2.2141e−1 (3.47e−3)−	**2.1400e−1 (1.98e−3)**
WFG8		3.6573e−1 (1.21e−2)−	3.9219e−1 (5.27e−3)−	4.2909e−1 (1.39e−2)−	4.4213e−1 (1.16e−2)−	3.3815e−1 (7.02e−3)−	3.3419e−1 (5.80e−3)−	**2.9979e−1 (4.77e−3)**
WFG9		2.8661e−1 (3.58e−2)=	3.4565e−1 (1.91e−2)=	3.3721e−1 (1.83e−2)=	3.8282e−1 (4.62e−2)−	**2.2107e−1 (3.58e−3)**+	2.4827e−1 (4.54e−2)=	3.0873e−1 (6.48e−2)
+/ − / =		10/23/3	11/23/2	9/23/4	4/30/2	9/24/3	3/30/3	–/–/–

all two and three-objective functions. All the methods are run 30 times independently on each test function, and the mean and the standard deviation of the IGD and HV values are recorded. In addition, Wilcoxon Signed Rank Test is performed at a significance level 0.05 to statistically analyze the differences between the results achieved by CMODE and the competing methods, where the symbols "+", "=" and "−" mean that the results of competing methods are significantly better, similar, and worse to those achieved by CMODE, respectively. All experiments are conducted on PC with an two Intel(R) Core(TM) i5 3.3 GHz and 8.0 GB memory and Windows 7 Operating System with MATLAB 2016b.

4.3. Comparisons between CMODE and other competing MOEAs

Table 1 provides the IGD values of all competing MOEAs on DTLZ, WFG and ZDT test functions. From Table 1, we can clearly find that the proposed CMODE obtains better performance than the six competing methods. The proposed CMODE achieves the best mean IGD values on 15 out of the 36 test functions, NSGA-II on 5 test functions, MOEA/D on 7 test functions, MOEA/D-DE on 1 test function, GDE3 on 1 test function, CMOPSO on 5 test functions and EMOSO on 1 test functions. Specifically, The proposed CMODE performs better (similar) on 23 (3) out of 36 functions in comparing with NSGA-II. The CMODE has better performance mainly because it uses the competition mechanism and the DE evolution operations. The proposed CMODE performs better (similar) on 23 (2) out of 36 functions in comparing with MOEA/D. MOEA/D performs better on the two-objective functions, while CMODE performs better on the three-objective functions. MOEA/D-DE performs slightly worse than MOEA/D and also performs better on the two-objective functions than the proposed CMODE. The proposed CMODE performs better (similar) on 30 (2) out of 36 functions in comparing with GDE3. The proposed CMODE performs better (similar) on 24 (3) out of 36 functions in comparing with CMOPSO. These two methods have a similar framework, but they have different competition mechanism and the evolution operations so they have different performance. The proposed CMODE performs better (similar) on 30 (3) out of 36 functions in comparing with EMOSO. Both CMODE and EMOSO use the leader set to guide the evolution, but they employ different evolution operations.

Table 2 reports the HV values of the all competing MOEAs on DTLZ, WFG and ZDT test functions. It can also be observed from Table 2 that our proposed CMODE attains better HV values than the other six competing methods on most of test functions. CMODE attains the best mean value of HV on 19 out of 36 functions. NSGA-II, MOEA/D, MOEA/D-DE, GDE3, CMOPSO and EMOSO attain the best mean value of HV on 10, 2, 0, 3, 1, 1 out of 36 functions, respectively. From Table 2, we can also find that some HV values are zero, which indicates that the corresponding method is unable to get any candidate solution to dominate the reference point on those test functions. For instance, The proposed CMODE gets zero HV values on two-objective and three-objective DTLZ3. This shows that CMODE cannot effectively solve the highly multimodel DTLZ3. The HV values of method GDE3 become zero on two-objective DTLZ1 and ZDT4. CMOPSO and EMOSO get zero HV values on two-objective DTLZ3, three-objective DTLZ1 and three-objective DTLZ3. These results also show the limitation of GDE3, CMOPSO and EMOSO.

Figs. 2–5 show the non-dominated solutions achieved by the CMODE and other competing MOEAs on ZDT3, two and three-objective DTLZ7 and WFG6. From Figs. 2 and 3, it can be observed that the non-dominated solutions achieved by CMODE on two-objective DTLZ7 and two-objective ZDT3 functions can well approximate the PF and have a good distribution. This reveals that our proposed CMODE can achieve a well trade-off between the convergence and diversity. GDE3 performs worst among the seven competing methods. From Figs. 4 and 5, we can also seen that CMODE achieves non-dominated solutions with well convergence and diversity on three-objective DTLZ7 and WFG6.

From the above experimental results, on most of the 36 test instances, CMODE significantly outperforms the other 6 multi-objective methods in solving MOP, which reveals that our proposed CMODE is a competitive algorithm. The main reason is that CMODE adopts a mutation strategy based on the SDE competition mechanism, which can generate better offspring in terms of convergence and diversity.

Table 2
Comparison of HV values obtained by CMODE and other competing MOEAs.

Problem	M	NSGA-II	MOEA/D	MOEA/D-DE	GDE3	CMOPSO	EMOSO	CMODE
DTLZ1		**5.8003e−1 (1.36e−3)**+	5.7998e−1 (1.40e−3)+	5.6409e−1 (6.95e−2)+	0.0000e+0 (0.00e+0)−	2.9592e−1 (2.78e−1)=	0.0000e+0 (0.00e+0)−	1.3779e−1 (2.41e−1)
DTLZ2		3.4656e−1 (1.70e−4)−	3.4721e−1 (4.10e−6)−	3.4702e−1 (3.06e−5)−	3.4538e−1 (1.77e−4)−	3.4637e−1 (1.73e−4)−	3.4720e−1 (7.82e−5)−	**3.4756e−1 (3.91e−5)**
DTLZ3		**2.8416e−1 (1.02e−1)**+	2.4431e−1 (1.16e−1)+	8.5673e−2 (1.33e−1)+	4.1061e−3 (2.25e−2)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)
DTLZ4	2	2.7840e−1 (1.15e−1)−	2.7031e−1 (1.19e−1)−	3.4682e−1 (8.54e−5)−	3.4537e−1 (2.05e−4)−	2.2709e−1 (1.30e−1)−	3.0425e−1 (9.70e−2)−	**3.4764e−1 (6.15e−5)**
DTLZ5		3.4654e−1 (1.99e−4)−	3.4721e−1 (3.42e−6)−	3.4703e−1 (2.29e−5)−	3.4541e−1 (2.10e−4)−	3.4634e−1 (1.79e−4)−	3.4722e−1 (7.12e−5)−	**3.4755e−1 (3.95e−5)**
DTLZ6		3.4632e−1 (2.42e−4)+	3.4721e−1 (9.91e−8)+	3.4721e−1 (1.02e−7)+	**3.4762e−1 (2.03e−5)**+	3.4756e−1 (4.07e−5)=	3.4737e−1 (7.37e−5)+	3.3598e−1 (6.35e−2)
DTLZ7		2.4270e−1 (4.90e−5)−	2.4000e−1 (1.21e−2)−	2.1098e−1 (3.35e−2)−	7.1239e−2 (2.75e−2)−	2.3393e−1 (2.31e−2)−	2.4066e−1 (1.22e−2)−	**2.4297e−1 (6.44e−6)**
WFG1		**6.5471e−1 (2.05e−2)**+	6.1529e−1 (6.58e−2)=	3.4715e−1 (5.77e−2)−	2.5255e−2 (1.65e−2)−	2.4471e−1 (2.24e−2)−	3.5309e−3 (7.59e−3)−	5.9845e−1 (5.76e−2)
WFG2		6.3214e−1 (6.85e−4)−	6.2805e−1 (5.33e−3)−	6.2811e−1 (9.76e−4)−	6.2584e−1 (1.21e−3)−	6.3210e−1 (3.28e−4)−	6.3090e−1 (3.88e−4)−	**6.3344e−1 (2.34e−4)**
WFG3		5.7912e−1 (7.74e−4)−	5.7871e−1 (1.81e−3)−	5.7830e−1 (5.84e−4)−	5.7559e−1 (5.71e−4)−	5.7968e−1 (2.87e−4)−	5.8000e−1 (3.08e−4)−	**5.8238e−1 (4.98e−5)**
WFG4		**3.4571e−1 (3.56e−4)**+	3.4455e−1 (2.68e−3)+	3.2441e−1 (3.15e−3)−	3.1529e−1 (1.06e−3)−	3.2554e−1 (5.33e−3)−	3.1766e−1 (1.29e−3)−	3.4347e−1 (1.65e−3)
WFG5	2	**3.1260e−1 (1.06e−3)**+	3.0709e−1 (2.87e−4)−	3.0769e−1 (1.45e−3)−	3.1050e−1 (2.02e−3)=	3.0993e−1 (2.10e−3)=	3.1134e−1 (1.77e−3)+	3.1000e−1 (2.12e−3)
WFG6		3.0640e−1 (1.20e−2)−	2.9938e−1 (1.82e−2)−	3.1165e−1 (4.83e−2)=	3.2288e−1 (4.38e−2)−	3.4188e−1 (4.26e−3)+	**3.4339e−1 (4.22e−3)**+	3.3124e−1 (1.05e−2)
WFG7		3.4500e−1 (2.96e−4)−	3.4517e−1 (3.06e−3)−	3.4469e−1 (3.36e−4)−	3.4236e−1 (3.30e−4)−	3.4522e−1 (2.70e−4)−	3.4625e−1 (1.82e−4)−	**3.4757e−1 (3.09e−5)**
WFG8		2.8765e−1 (7.46e−4)−	2.8791e−1 (3.94e−3)=	2.8967e−1 (2.07e−3)=	**2.9168e−1 (5.56e−4)**+	2.8444e−1 (2.17e−3)−	2.8294e−1 (9.88e−4)−	2.8951e−1 (3.99e−4)
WFG9		**3.3593e−1 (2.08e−2)**+	3.2446e−1 (2.18e−2)=	3.3254e−1 (2.48e−3)=	2.8243e−1 (5.44e−2)−	3.3435e−1 (1.60e−3)=	3.2995e−1 (1.98e−2)=	2.9535e−1 (5.35e−2)
ZDT1		7.1909e−1 (2.53e−4)−	7.1981e−1 (5.85e−4)−	7.0719e−1 (4.37e−3)−	4.7291e−1 (4.93e−2)−	7.1937e−1 (2.39e−4)−	7.1918e−1 (3.33e−4)−	**7.2033e−1 (1.38e−4)**
ZDT2	2	4.4390e−1 (1.72e−4)−	4.4456e−1 (6.27e−4)−	4.3478e−1 (4.23e−3)−	3.0711e−2 (3.40e−2)−	4.4401e−1 (1.50e−4)−	4.4468e−1 (6.72e−5)−	**4.4511e−1 (2.93e−5)**
ZDT3		**6.0226e−1 (1.62e−2)**=	6.0027e−1 (1.75e−2)+	5.9333e−1 (6.57e−3)−	4.8468e−1 (5.45e−2)−	5.9968e−1 (9.98e−5)+	5.9885e−1 (2.93e−4)−	5.9892e−1 (9.54e−4)
ZDT4		**7.1702e−1 (2.16e−3)**+	7.1496e−1 (2.22e−3)+	4.7155e−1 (2.13e−1)+	0.0000e+0 (0.00e+0)=	4.1463e−1 (2.15e−1)+	0.0000e+0 (0.00e+0)=	5.4465e−3 (2.82e−2)
DTLZ1		**8.2055e−1 (5.76e−3)**+	8.0296e−1 (2.08e−3)+	7.0856e−1 (2.29e−1)=	2.2096e−2 (8.41e−2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	3.4487e−1 (3.85e−1)
DTLZ2		5.3243e−1 (4.38e−3)−	5.2617e−1 (5.65e−4)−	5.2538e−1 (2.18e−3)−	4.9711e−1 (4.30e−3)−	5.4226e−1 (3.12e−3)−	5.4380e−1 (2.32e−3)−	**5.6068e−1 (8.85e−4)**
DTLZ3		3.9045e−1 (1.83e−1)+	**4.4815e−1 (1.54e−1)**+	2.4993e−1 (2.40e−1)+	9.0033e−4 (4.93e−3)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)
DTLZ4	3	5.1900e−1 (8.09e−2)−	3.9975e−1 (1.23e−1)−	5.0613e−1 (2.81e−2)−	4.9816e−1 (5.48e−3)−	5.3382e−1 (4.05e−3)−	5.3445e−1 (4.91e−3)−	**5.5910e−1 (1.29e−2)**
DTLZ5		1.9915e−1 (1.67e−4)−	1.9472e−1 (7.06e−6)−	1.9437e−1 (8.67e−5)−	1.9796e−1 (2.18e−4)−	1.9773e−1 (4.98e−4)−	1.9656e−1 (6.88e−4)−	**2.0017e−1 (3.22e−5)**
DTLZ6		1.9942e−1 (1.38e−4)−	1.9471e−1 (6.06e−6)−	1.9476e−1 (2.47e−5)−	**2.0024e−1 (2.32e−5)**+	2.0017e−1 (3.51e−5)=	2.0005e−1 (4.96e−5)−	2.0018e−1 (3.19e−5)
DTLZ7		2.6742e−1 (1.88e−3)−	2.3205e−1 (5.79e−3)−	2.0817e−1 (1.75e−2)−	7.5594e−2 (4.77e−2)−	2.5472e−1 (2.23e−2)−	2.7196e−1 (1.22e−3)−	**2.7896e−1 (4.46e−4)**
WFG1		8.0398e−1 (2.76e−2)+	**8.7036e−1 (3.11e−2)**+	3.0483e−1 (1.97e−2)−	2.2823e−1 (7.71e−3)−	2.9238e−1 (9.83e−3)−	9.6722e−2 (1.64e−2)−	7.7606e−1 (5.10e−2)
WFG2		9.1817e−1 (2.71e−3)−	8.7011e−1 (5.03e−3)−	8.7631e−1 (6.88e−3)−	8.8145e−1 (4.09e−3)−	9.2728e−1 (1.25e−3)−	9.1768e−1 (2.22e−3)−	**9.3571e−1 (9.15e−4)**
WFG3		**3.9481e−1 (3.46e−3)**+	3.7765e−1 (4.24e−3)−	3.3869e−1 (2.46e−2)−	3.3276e−1 (9.55e−3)−	3.5384e−1 (4.81e−3)−	3.1167e−1 (7.96e−3)−	3.8643e−1 (6.58e−3)
WFG4		5.1789e−1 (4.57e−3)−	5.0455e−1 (4.66e−3)−	4.6220e−1 (6.05e−3)−	4.5090e−1 (6.87e−3)−	4.8892e−1 (2.90e−3)−	4.9417e−1 (3.32e−3)−	**5.4695e−1 (5.34e−3)**
WFG5	3	4.8980e−1 (4.62e−3)−	4.5656e−1 (2.66e−3)−	4.5562e−1 (3.35e−3)−	4.7079e−1 (1.26e−2)−	4.7868e−1 (5.82e−3)−	4.7951e−1 (8.61e−3)−	**5.0978e−1 (3.39e−3)**
WFG6		4.6857e−1 (1.49e−2)−	4.3579e−1 (1.95e−2)−	4.3312e−1 (4.48e−2)−	4.3866e−1 (2.72e−2)−	5.1477e−1 (7.85e−3)−	5.3386e−1 (7.01e−3)−	**5.3588e−1 (2.80e−2)**
WFG7		5.2118e−1 (3.94e−3)−	5.0104e−1 (4.00e−3)−	4.9049e−1 (5.93e−3)−	4.5257e−1 (6.22e−3)−	5.2448e−1 (4.37e−3)−	5.4030e−1 (2.51e−3)−	**5.6164e−1 (6.72e−4)**
WFG8		4.4153e−1 (4.30e−3)−	4.3754e−1 (3.11e−3)−	3.9177e−1 (9.78e−3)−	3.8140e−1 (6.25e−3)−	4.2983e−1 (4.42e−3)−	4.3313e−1 (3.83e−3)−	**4.5847e−1 (3.32e−3)**
WFG9		4.8794e−1 (3.02e−2)=	4.7504e−1 (2.07e−2)=	4.7277e−1 (2.22e−2)=	3.9676e−1 (3.77e−2)−	**5.1277e−1 (4.10e−3)**+	4.8874e−1 (3.83e−2)=	4.3914e−1 (5.67e−2)
+/ − / =		12/22/2	9/23/4	5/26/5	3/29/4	4/25/7	3/28/5	–/–/–

4.4. Analysis of the parameter

In CMODE, the parameter α determine the size of leader set $\boldsymbol{L}$ and affect the performance of the proposed CMODE, the individuals in which are used to guide the evolution of population. We conduct the proposed CMODE with different α values, ranging from 2 to 30. The other parameter values of CMODE are set according to Section 4.2. The comparison results of 30 independent runs on these three test suites are reported in Table 3. From Table 3, it can be found that CMODE with the parameter $\alpha = 5$ has relative better results. Therefore, in this work we adopt the 5 as the recommended value.

4.5. Comparisons among different CMODE variants

To evaluate the effects of the SDE competitive mechanism on the performance of the CMODE. Two variants of CMODE are employed for comparison, namely, CMODEA using the angle competitive mechanism and CMODED using the Euclidean distance competitive mechanism. The parameter values of these three CMODE variants are set according to Section 4.2. The comparison results of 30 independent runs on these three test suites are reported in Table 4. From Table 4, we can find that the proposed CMODE has better performance than the other two variants. The reason is that SDE approach can assess the quality of solution in terms of diversity and convergence.

5. Application of our proposed CMODE to feature selection problem

In this subsection, we apply CMODE algorithm to solve the feature selection problem. With the continuous improvements in science and technology, tremendous data is being produced everyday. Handling such large amounts of data is a challenge for humans [73], which increase the need for advanced data processing technologies such as data mining. One of the most widely used data preprocessing technologies in data mining is feature selection, which aims to eliminate the irrelevant features without reducing the prediction accuracy and to pick out the most useful features which make contribution to build the model effectively and efficiently. Hence, feature selection involves two chief conflicting goals, namely, maximizing the accuracy and minimizing features number [74]. And feature selection can be considered as a multi-objective problem to achieve the trade-off between these two conflicting goals. In the literature, many EAs have been adopted to deal with the feature selection problem [6,74–76]. In this work, to further verify the performance of CMODE in dealing with the real-world problems, we apply CMODE to handle the feature selection problem and compare it with other MOEAs.

5.1. Objective function

In terms of MOEA, feature selection problem is often mathematically modeled as the following two-objective MOP [75].

$$\min \quad \begin{aligned} & F(x) = (f_1(x), f_2(x)) \\ & f_1(x) = |x| \\ & f_2(x) = Err(x) \end{aligned} \tag{12}$$

where x represents a solution containing a set of selected features and $Err(x)$ stands for the error of the model. The purpose of the first objective f_1 in Eq. (12) is to minimize the number of selected features. Its objective value can be easily obtained by calculating the number of selected features in solution x. The purpose of the second objective f_2 in Eq. (12) is to minimize the error rate of the model. In other words, it is to maximize the accuracy rate. In this study, the feature selection problem is consider in classification problem, thus the error rate of the classifier is utilized as f_2. And the classifier K Nearest Neighbored (KNN) with $k = 3$ is used in this study. In the experiment, each dataset is randomly partitioned into two sets, 80% for training set, and 20% for test set. Then, error rate (i.e., objective value of f_2) of the model is calculated with regard to the test set.

5.2. Individual encoding

In this study, the numerical encoding of individual is adopted to solve the feature selection problem. Each individual stands

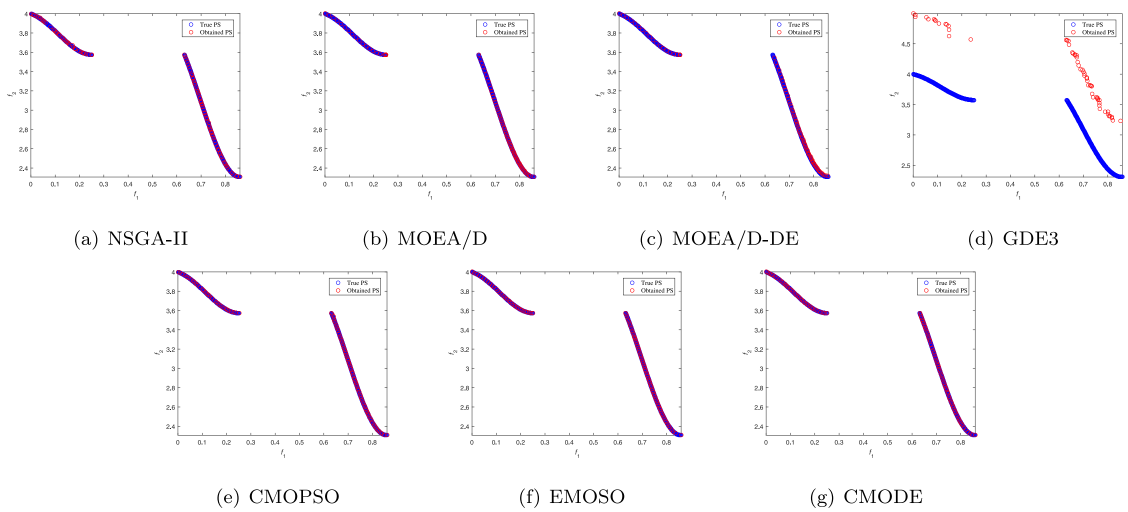


Fig. 2. The non-dominated solutions achieved by the competing MOEAs and CMODE on two-objective DTLZ7 function.

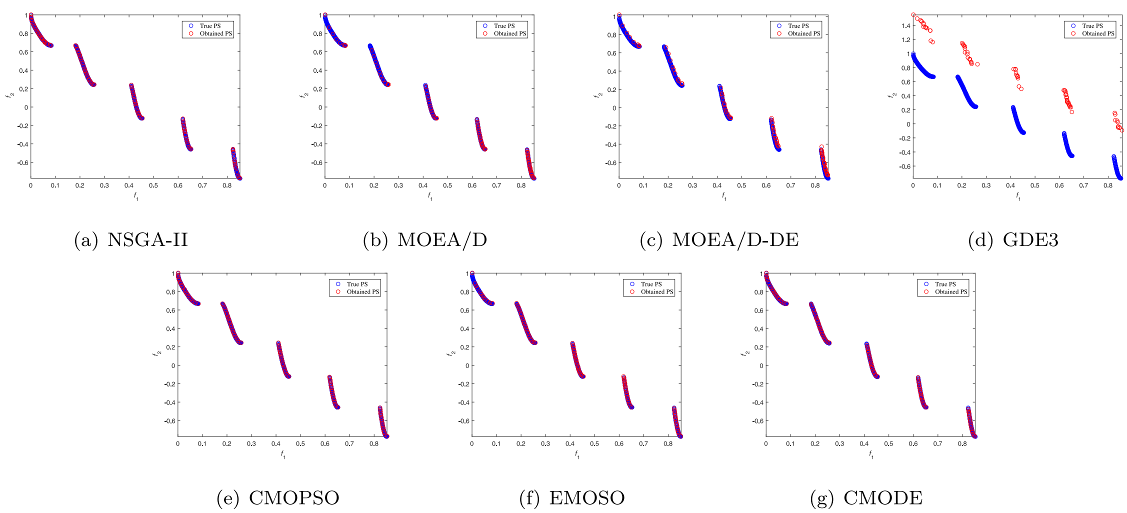


Fig. 3. The non-dominated solutions achieved by the competing MOEAs and CMODE on two-objective ZDT3 function.

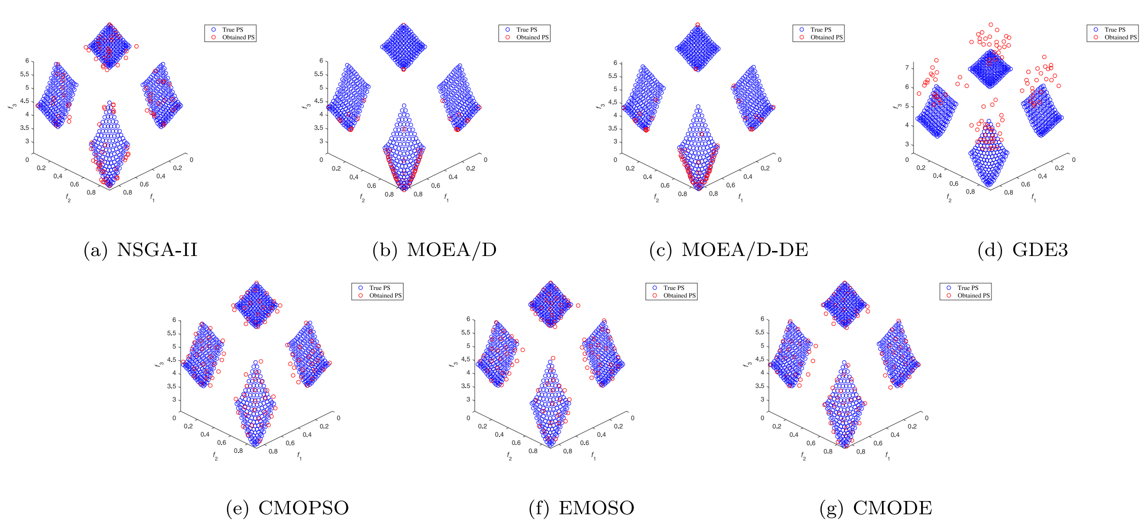
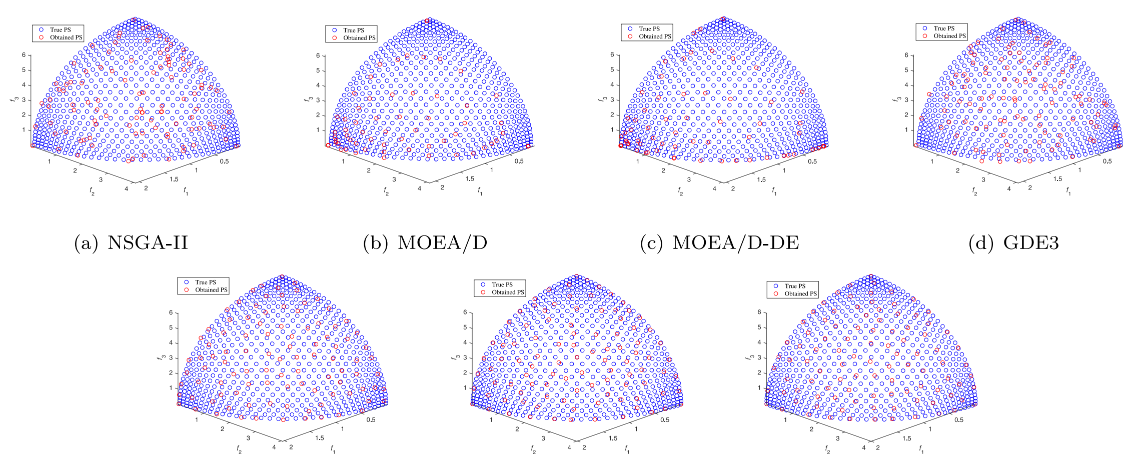


Fig. 4. The non-dominated solutions achieved by the competing MOEAs and CMODE on three-objective DTLZ7 function.

Table 3
Comparison of IGD values obtained by CMODE with different α values.

Problem	M	CMODE$\alpha=2$	CMODE$\alpha=5$	CMODE$\alpha=10$	CMODE$\alpha=15$	CMODE$\alpha=20$	CMODE$\alpha=25$	CMODE$\alpha=30$
DTLZ1	2	5.3191e+0 (3.36e+0)−	1.3739e+0 (2.01e+0)	3.4500e−1 (4.97e−1)+	1.6621e−1 (3.56e−1)+	8.3876e−2 (1.51e−1)+	9.3121e−2 (1.80e−1)+	**3.7005e−2 (1.07e−1)**+
DTLZ2	2	1.2826e−2 (5.88e−3)−	**4.0931e−3 (3.07e−5)**	4.1053e−3 (2.58e−5)=	4.1184e−3 (2.94e−5)−	4.1108e−3 (2.97e−5)−	4.1204e−3 (3.81e−5)−	4.1127e−3 (2.78e−5)−
DTLZ3	2	1.1321e+2 (3.21e+1)−	6.0261e+1 (2.19e+1)	1.8734e+1 (1.56e+1)+	1.5101e+1 (1.24e+1)+	9.1591e+0 (9.29e+0)+	**4.1195e+0 (3.48e+0)**+	4.3726e+0 (4.72e+0)+
DTLZ4	2	1.9101e−1 (1.15e−1)−	4.2875e−3 (2.30e−4)	**4.1646e−3 (6.88e−5)**=	4.1808e−3 (6.89e−5)=	4.1958e−3 (7.45e−5)=	4.1906e−3 (6.52e−5)=	4.1900e−3 (8.83e−5)=
DTLZ5	2	1.2445e−2 (3.86e−3)−	**4.0944e−3 (2.87e−5)**	4.1126e−3 (2.32e−5)−	4.1118e−3 (2.99e−5)−	4.1171e−3 (3.07e−5)−	4.1193e−3 (3.43e−5)−	4.1149e−3 (1.87e−5)−
DTLZ6	2	2.9842e+0 (1.27e+0)−	3.1114e−2 (1.48e−1)	**4.1121e−3 (3.93e−5)**=	4.1156e−3 (2.54e−5)=	4.1232e−3 (3.25e−5)+	4.1182e−3 (3.08e−5)+	4.1194e−3 (3.06e−5)+
DTLZ7	2	6.1980e−1 (2.75e−1)−	4.3741e−3 (3.25e−5)	**4.3659e−3 (3.35e−5)**=	4.3866e−3 (3.43e−5)=	4.3723e−3 (3.41e−5)=	4.3742e−3 (4.53e−5)=	4.3759e−3 (3.61e−5)=
WFG1	2	1.2876e+0 (1.44e−1)−	**2.3619e−1 (1.25e−1)**	4.3714e−1 (1.76e−1)−	6.0748e−1 (1.96e−1)−	6.9679e−1 (1.27e−1)−	6.9403e−1 (1.30e−1)−	6.8463e−1 (1.50e−1)−
WFG2	2	1.7335e−1 (3.68e−2)−	1.2549e−2 (1.28e−3)	1.0769e−2 (1.73e−4)+	1.0801e−2 (1.50e−4)+	1.0779e−2 (1.12e−4)+	**1.0761e−2 (1.20e−4)**+	1.0774e−2 (1.28e−4)+
WFG3	2	4.3533e−2 (2.11e−2)−	**1.1670e−2 (5.07e−5)**	1.1696e−2 (6.56e−5)=	1.1718e−2 (7.36e−5)−	1.1720e−2 (5.32e−5)−	1.1726e−2 (8.74e−5)−	1.1743e−2 (5.89e−5)−
WFG4	2	4.0322e−2 (8.93e−3)−	**1.5845e−2 (1.60e−3)**	1.6619e−2 (2.16e−3)=	2.0842e−2 (2.87e−3)−	2.4043e−2 (2.33e−3)−	2.6315e−2 (3.16e−3)−	2.7028e−2 (3.41e−3)−
WFG5	2	1.0480e−1 (2.77e−2)−	**6.7266e−2 (2.35e−3)**	6.8123e−2 (2.10e−3)=	6.7704e−2 (2.35e−3)=	6.8010e−2 (2.23e−3)=	6.8109e−2 (2.23e−3)=	6.7436e−2 (2.41e−3)=
WFG6	2	1.1056e−1 (7.38e−2)−	3.4654e−2 (1.67e−2)	3.1590e−2 (3.86e−2)+	1.8744e−2 (9.26e−3)+	2.4030e−2 (3.84e−2)+	**1.6307e−2 (4.58e−3)**+	1.8866e−2 (7.46e−3)+
WFG7	2	1.5031e−1 (2.86e−2)−	1.2665e−2 (1.41e−4)	**1.2658e−2 (1.45e−4)**=	1.2719e−2 (1.23e−4)=	1.2874e−2 (1.74e−4)−	1.2982e−2 (1.52e−4)−	1.3079e−2 (1.64e−4)−
WFG8	2	1.4086e−1 (2.45e−2)−	**1.0855e−1 (8.64e−4)**	1.0991e−1 (1.05e−3)−	1.1040e−1 (8.64e−4)−	1.1094e−1 (8.25e−4)−	1.1041e−1 (9.40e−4)−	1.1059e−1 (1.33e−3)−
WFG9	2	1.4467e−1 (8.30e−2)−	1.0026e−1 (9.82e−2)	6.1020e−2 (7.62e−2)=	7.7015e−2 (8.54e−2)=	8.2806e−2 (9.03e−2)=	**5.5269e−2 (6.94e−2)**=	6.9062e−2 (8.12e−2)=
ZDT1	2	4.4360e−1 (1.10e−1)−	**3.8410e−3 (3.84e−5)**	3.8435e−3 (3.22e−5)=	3.8471e−3 (3.65e−5)=	3.8453e−3 (3.70e−5)=	3.8422e−3 (3.49e−5)=	3.8480e−3 (3.47e−5)=
ZDT2	2	7.8487e−1 (3.04e−1)−	**3.8487e−3 (3.24e−5)**	3.8695e−3 (3.12e−5)−	3.8678e−3 (3.22e−5)−	3.8646e−3 (2.42e−5)−	3.8757e−3 (3.35e−5)−	3.8671e−3 (3.19e−5)−
ZDT3	2	4.0850e−1 (1.14e−1)−	5.2390e−3 (6.43e−4)	4.5599e−3 (5.03e−5)+	4.5117e−3 (5.57e−5)+	**4.4760e−3 (5.80e−5)**+	4.4985e−3 (5.03e−5)+	4.4839e−3 (6.02e−5)+
ZDT4	2	1.0954e+1 (3.99e+0)−	3.6907e+0 (2.18e+0)	2.2476e+0 (1.71e+0)+	1.5325e+0 (1.13e+0)+	6.9759e−1 (5.99e−1)+	3.9143e−1 (3.25e−1)+	**2.9187e−1 (2.59e−1)**+
DTLZ1	3	2.0943e+0 (2.15e+0)−	4.3335e−1 (4.31e−1)	8.0838e−2 (1.12e−1)+	6.5684e−2 (1.03e−1)+	8.9899e−2 (2.24e−1)+	8.3790e−2 (1.17e−1)+	**4.7598e−2 (8.28e−2)**+
DTLZ2	3	**5.3718e−2 (4.62e−4)**=	5.3833e−2 (6.27e−4)	5.3895e−2 (5.72e−4)=	5.3991e−2 (4.65e−4)=	5.4333e−2 (6.04e−4)−	5.4270e−2 (5.00e−4)−	5.4470e−2 (4.58e−4)−
DTLZ3	3	5.6599e+1 (2.60e+1)−	4.0432e+1 (2.28e+1)	1.1922e+1 (9.41e+0)+	9.0519e+0 (8.25e+0)+	8.3142e+0 (9.80e+0)+	3.2145e+0 (3.91e+0)+	**2.0293e+0 (2.76e+0)**+
DTLZ4	3	1.6869e−1 (1.80e−1)−	6.0882e−2 (3.70e−2)	**5.3907e−2 (4.60e−4)**=	5.4090e−2 (4.38e−4)=	5.4146e−2 (5.07e−4)=	5.4241e−2 (4.93e−4)=	5.4251e−2 (4.86e−4)=
DTLZ5	3	1.7038e−2 (9.52e−3)−	**4.1533e−3 (3.14e−5)**	4.1857e−3 (4.69e−5)−	4.1877e−3 (6.88e−5)=	4.1944e−3 (4.68e−5)−	4.2104e−3 (4.45e−5)−	4.2016e−3 (6.10e−5)−
DTLZ6	3	5.4506e−2 (5.90e−2)−	**4.1597e−3 (4.20e−5)**	4.1668e−3 (3.36e−5)=	4.1739e−3 (3.73e−5)=	4.1776e−3 (4.26e−5)=	4.1831e−3 (5.71e−5)=	4.1812e−3 (4.89e−5)=
DTLZ7	3	7.3016e−2 (3.02e−2)=	5.9853e−2 (1.20e−3)	6.0070e−2 (1.22e−3)=	5.9678e−2 (1.26e−3)=	5.9685e−2 (1.00e−3)=	7.8945e−2 (7.28e−2)=	**5.9640e−2 (8.24e−4)**=
WFG1	3	9.8568e−1 (1.34e−1)−	**4.4350e−1 (9.37e−2)**	7.0009e−1 (1.20e−1)−	8.4308e−1 (9.79e−2)−	9.1344e−1 (1.25e−1)−	1.0302e+0 (9.49e−2)−	1.0650e+0 (1.13e−1)−
WFG2	3	**1.7703e−1 (5.59e−3)**+	1.7893e−1 (4.36e−3)	1.7847e−1 (3.66e−3)=	1.7896e−1 (4.31e−3)=	1.7975e−1 (3.74e−3)=	1.7884e−1 (3.46e−3)=	1.7983e−1 (3.41e−3)=
WFG3	3	1.9409e−1 (3.57e−2)−	**1.0224e−1 (1.17e−2)**	1.0376e−1 (9.33e−3)=	1.1917e−1 (9.04e−3)−	1.3382e−1 (1.08e−2)−	1.4282e−1 (1.04e−2)−	1.4661e−1 (1.32e−2)−
WFG4	3	**2.1455e−1 (2.39e−3)**=	2.1593e−1 (3.27e−3)	2.3803e−1 (4.11e−3)−	2.4359e−1 (4.35e−3)−	2.4557e−1 (3.86e−3)−	2.4748e−1 (4.56e−3)−	2.4800e−1 (4.14e−3)−
WFG5	3	2.2444e−1 (3.88e−3)=	**2.2292e−1 (2.00e−3)**	2.2353e−1 (2.67e−3)=	2.2785e−1 (3.86e−3)−	2.3061e−1 (5.03e−3)−	2.3242e−1 (4.86e−3)−	2.3295e−1 (3.83e−3)−
WFG6	3	2.4562e−1 (4.45e−2)−	**2.2150e−1 (2.37e−2)**	2.2405e−1 (2.37e−2)=	2.2502e−1 (2.17e−2)−	2.2613e−1 (8.43e−3)−	2.3177e−1 (2.13e−2)−	2.2680e−1 (4.76e−3)−
WFG7	3	2.1433e−1 (2.13e−3)=	**2.1400e−1 (1.98e−3)**	2.1541e−1 (2.42e−3)−	2.1766e−1 (1.96e−3)−	2.2264e−1 (3.89e−3)−	2.2645e−1 (4.17e−3)−	2.2825e−1 (3.39e−3)−
WFG8	3	3.2800e−1 (2.50e−2)−	**2.9979e−1 (4.77e−3)**	3.0888e−1 (5.63e−3)−	3.1283e−1 (5.87e−3)−	3.1578e−1 (5.08e−3)−	3.1712e−1 (4.70e−3)−	3.1821e−1 (6.08e−3)−
WFG9	3	3.0317e−1 (6.25e−2)=	3.0873e−1 (6.48e−2)	2.7404e−1 (6.67e−2)=	2.6852e−1 (6.39e−2)=	2.9602e−1 (6.58e−2)=	2.9221e−1 (6.49e−2)=	**2.6800e−1 (5.98e−2)**=
+/ − / =		1/30/5	-/-/-	8/9/19	8/13/15	9/16/11	9/16/11	9/16/11



(a) NSGA-II (b) MOEA/D (c) MOEA/D-DE (d) GDE3

(e) CMOPSO (f) EMOSO (g) CMODE

Fig. 5. The non-dominated solutions achieved by the competing MOEAs and CMODE on three-objective WFG6 function.

for the number of features in a given dataset and each variable represents the probability to select the corresponding feature. For instance, given a dataset with D features, a individual with D variables is shown as follows.

$$X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D}) \quad x_{i,j} \in [0, 1] \\ i = 1, 2, \ldots, ps; \quad j = 1, 2, \ldots, D \tag{13}$$

where $x_{i,j}$ denotes the probability of the jth feature be selected. In the experiment, a threshold η is set to compare with each variable. The jth feature is selected if and only if its value is larger than η. In this study, the η is set to 0.6 as recommended in [75] to limit the number of selected features.

5.3. Datasets and parameter settings

In the experiment, seven representative real-world datasets cited from the UCI machine learning repository [77] are adopted for comparison. Table 5 gives a detailed description of these datasets, including the number of patterns (ranging from 178 to 2310), number of features (ranging from 13 to 256) and number of classes (ranging from 2 to 10). The experimental results of CMODE are compared with NSGA-II, MOEA/D, MOEA/D-DE, GDE3, CMOPSO and EMOSO methods. The population size of all competing MOEAs is set to 100. The maximum number of function evaluations is set to 10 000 and is employed as the termination criterion. The reference point is set to $(D, 1)$, i.e., the maximum values of the objectives. The other parameters are the same as Section 4.2.

5.4. Results and analysis

The statistical results of HV values for the seven competing MOEAs on seven feature selection datasets are reported in Table 6. From this table, we can see that NSGA-II, GDE3 and the proposed CMODE achieve the same number of highest HV values on 3 out of 7 datasets. MOEA/D, MOEA/D-DE, CMOPSO

Table 4
Comparison of IGD values obtained by three CMODE variants.

Problem	M	D	CMODEA	CMODED	CMODE
DTLZ1	2	6	**2.3633e−1 (3.24e−1)**+	1.6560e+0 (1.84e+0)=	1.3739e+0 (2.01e+0)
DTLZ2		11	4.1149e−3 (3.12e−5)−	4.1015e−3 (3.04e−5)=	**4.0931e−3 (3.07e−5)**
DTLZ3		11	**1.9726e+1 (1.70e+1)**+	5.3744e+1 (2.60e+1)=	6.0261e+1 (2.19e+1)
DTLZ4		11	4.1267e−3 (4.03e−5)+	**4.1112e−3 (3.20e−5)**+	4.2875e−3 (2.30e−4)
DTLZ5		11	4.5957e−3 (1.04e−3)−	4.0962e−3 (3.69e−5)=	**4.0944e−3 (2.87e−5)**
DTLZ6		11	3.7626e−2 (1.82e−1)−	7.0505e−2 (2.53e−1)=	**3.1114e−2 (1.48e−1)**
DTLZ7		21	7.0831e−3 (1.11e−3)−	5.7759e−3 (3.62e−3)−	**4.3741e−3 (3.25e−5)**
WFG1	2	11	7.9099e−1 (1.09e−1)−	6.2926e−1 (1.56e−1)−	**2.3619e−1 (1.25e−1)**
WFG2		11	3.2755e−2 (8.83e−3)−	3.2034e−2 (2.03e−2)−	**1.2549e−2 (1.28e−3)**
WFG3		11	1.4819e−2 (3.67e−3)−	1.4849e−2 (4.79e−3)−	**1.1670e−2 (5.07e−5)**
WFG4		11	4.2010e−2 (6.58e−3)−	1.7287e−2 (2.31e−3)−	**1.5845e−2 (1.60e−3)**
WFG5		11	6.7442e−2 (2.38e−3)=	6.7620e−2 (2.36e−3)=	**6.7266e−2 (2.35e−3)**
WFG6		11	6.3965e−2 (3.69e−2)−	3.8696e−2 (3.87e−2)=	**3.4654e−2 (1.67e−2)**
WFG7		11	1.6744e−2 (7.63e−3)−	1.6119e−2 (2.23e−3)−	**1.2665e−2 (1.41e−4)**
WFG8		11	1.1968e−1 (8.82e−3)−	1.3341e−1 (1.41e−2)−	**1.0855e−1 (8.64e−4)**
WFG9		11	1.1511e−1 (1.02e−1)=	**3.4173e−2 (3.79e−2)**=	1.0026e−1 (9.82e−2)
ZDT1	2	30	4.0308e−3 (2.63e−4)−	3.9771e−3 (4.20e−4)−	**3.8410e−3 (3.84e−5)**
ZDT2		30	4.5530e−3 (1.36e−3)−	3.8803e−3 (3.44e−5)−	**3.8487e−3 (3.24e−5)**
ZDT3		30	5.3723e−3 (7.34e−4)=	8.9680e−3 (1.56e−2)−	**5.2390e−3 (6.43e−4)**
ZDT4		10	**1.8366e+0 (1.86e+0)**+	3.5512e+0 (2.64e+0)=	3.6907e+0 (2.18e+0)
DTLZ1	3	7	**8.4407e−2 (1.39e−1)**+	4.5277e−1 (6.33e−1)=	4.3335e−1 (4.31e−1)
DTLZ2		12	**5.3764e−2 (3.91e−4)**=	5.3811e−2 (4.82e−4)=	5.3833e−2 (6.27e−4)
DTLZ3		12	**4.8363e+0 (9.35e+0)**+	3.6947e+1 (2.44e+1)=	4.0432e+1 (2.28e+1)
DTLZ4		12	9.7831e−2 (8.30e−2)=	7.0727e−2 (6.55e−2)=	**6.0882e−2 (3.70e−2)**
DTLZ5		12	4.8660e−3 (1.37e−3)−	4.1756e−3 (4.29e−5)=	**4.1533e−3 (3.14e−5)**
DTLZ6		12	4.7472e−3 (7.90e−4)−	4.1747e−3 (4.79e−5)=	**4.1597e−3 (4.20e−5)**
DTLZ7		22	**5.9190e−2 (1.17e−3)**+	5.9819e−2 (1.30e−3)=	5.9853e−2 (1.20e−3)
WFG1	3	12	9.2971e−1 (1.75e−1)−	6.9040e−1 (9.66e−2)−	**4.4350e−1 (9.37e−2)**
WFG2		12	1.7925e−1 (4.40e−3)=	1.8147e−1 (5.00e−3)−	**1.7893e−1 (4.36e−3)**
WFG3		12	1.6788e−1 (1.68e−2)−	1.2520e−1 (1.36e−2)−	**1.0224e−1 (1.17e−2)**
WFG4		12	2.4158e−1 (4.67e−3)−	2.1970e−1 (4.37e−3)−	**2.1593e−1 (3.27e−3)**
WFG5		12	2.2673e−1 (3.26e−3)−	2.2333e−1 (2.03e−3)=	**2.2292e−1 (2.00e−3)**
WFG6		12	2.4441e−1 (4.36e−2)−	2.3090e−1 (3.79e−2)−	**2.2150e−1 (2.37e−2)**
WFG7		12	2.1562e−1 (2.73e−3)−	2.2205e−1 (1.14e−2)−	**2.1400e−1 (1.98e−3)**
WFG8		12	3.1758e−1 (9.77e−3)−	3.0953e−1 (7.25e−3)−	**2.9979e−1 (4.77e−3)**
WFG9		12	3.4129e−1 (4.16e−2)−	**2.5408e−1 (5.96e−2)**+	3.0873e−1 (6.48e−2)
+/ − / =			7/23/6	2/17/17	–/–/–

Table 5
Description of 7 UCI datasets.

Problem	Number of patterns	Number of features	Number of class
Wine	178	13	3
Segmentation	2310	19	7
Parkinson	195	22	2
Ionosphere	351	34	2
Sonar	208	60	2
Hillvalley	606	100	2
Semeion handwritten digit	1593	256	10

and EMOSO get 0, 0, 1 and 1 the best highest HV, respectively. For the simple Wine dataset, most of the competing MOEAs have the same best HV values. The NSGA-II performs best on datasets Hillvalley and Semeion handwritten digit with larger dimension size. GDE3 achieves best performance on datasets Parkinson and Ionosphere. The proposed CMODE performs best on datasets Segmentation and Sonar.

Fig. 6 illustrates the nondominated solutions with best HV value achieved by the competing MOEAs and CMODE on seven datasets. From the subfigures (a), (b), (c) of Fig. 6, we can see that the proposed CMODE gets well distribution solutions with low error rate and small numbers of features on these three datasets. From the subfigures (d), (e) of Fig. 6, it can be observed that the proposed CMODE is competitive to the other competing MOEAs on these two datasets. For the dataset Hillvalley, the NSGA-II can get best distribution solutions, fellowed by CMOPSO, EMOSO and CMODE. For the dataset Semeion handwritten digit, the solutions achieved by MOEA/D-DE has the fewest features. The solutions achieved by NSGA-II, CMOPSO, EMOSO, GDE3 and CMODE have low error rate. Table 7 provides the lowest error rate achieved by the seven competing MOEAs. From this table, it can be observed that the number of best results got by GDE3 is the best, followed by NSGA-II. CMOPSO, EMOSO, and CMODE get the same number of lowest error rate values.

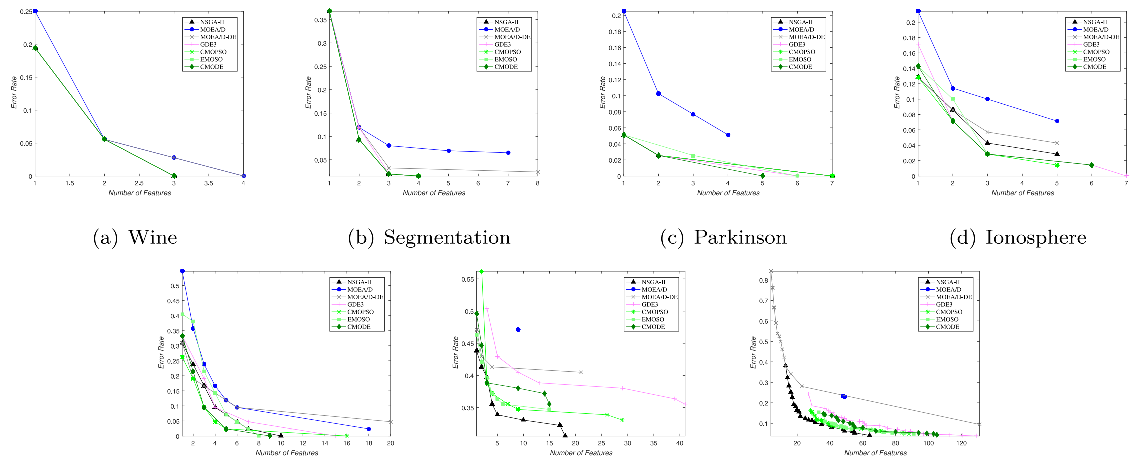
From the Tables 6 and 7, and Fig. 6, we can conclude that CMODE is competitive to other MOEAs to deal with feature selection problems.

6. Conclusions

In this work, we propose an efficient competitive mechanism based multi-objective differential evolution algorithm, called CMODE. In CMODE, a competitive mechanism based on the SDE is employed to design a novel DE mutation strategy for generating offspring, which can take advantages of the SDE to obtain a well trade-off between diversity and convergence. In addition, two variants of the CMODE using the angle competitive mechanism and the Euclidean distance competitive mechanism are proposed.

Table 6
HV value obtained by the seven competing MOEAs methods on seven datasets.

Problem	M	D	NSGA-II	MOEA/D	MOEA/D-DE	GDE3	CMOPSO	EMOSO	CMODE
Wine	2	13	**9.1418e−1 (2.34e−16)**=	9.0408e−1 (1.42e−2)−	9.1400e−1 (5.58e−4)=	**9.1418e−1 (2.34e−16)**=	**9.1418e−1 (2.34e−16)**=	**9.1418e−1 (2.34e−16)**=	**9.1418e−1 (2.34e−16)**
Segmentation	2	19	9.1978e−1 (4.38e−4)−	8.9367e−1 (1.97e−2)−	9.1671e−1 (3.88e−3)−	9.2000e−1 (3.57e−4)=	9.2003e−1 (2.68e−4)=	9.1957e−1 (1.37e−3)=	**9.2011e−1 (2.34e−16)**
Parkinson	2	22	9.4598e−1 (9.57e−3)=	9.2335e−1 (1.35e−2)−	9.3693e−1 (5.28e−3)−	**9.5415e−1 (6.50e−4)**=	9.4423e−1 (9.36e−3)−	9.5172e−1 (5.80e−3)=	9.5016e−1 (7.80e−3)
Ionosphere	2	34	9.4494e−1 (6.67e−3)=	9.0999e−1 (1.41e−2)−	9.3661e−1 (8.28e−3)−	**9.5366e−1 (5.95e−3)**=	9.4842e−1 (6.02e−3)=	9.5060e−1 (5.87e−3)=	9.4652e−1 (4.70e−3)
Sonar	2	60	9.7009e−1 (8.12e−3)=	9.1083e−1 (2.02e−2)−	9.3779e−1 (1.78e−2)−	9.6107e−1 (8.34e−3)−	9.6779e−1 (1.36e−2)=	9.6576e−1 (9.36e−3)=	**9.7317e−1 (3.27e−3)**
Hillvalley	2	100	**6.8805e−1 (1.06e−2)**=	5.1678e−1 (2.78e−2)−	6.3950e−1 (1.77e−2)−	6.3955e−1 (1.48e−2)−	6.7374e−1 (1.49e−2)=	6.7587e−1 (1.31e−2)=	6.8038e−1 (1.20e−2)
Semeion handwritten digit	2	256	**8.8729e−1 (9.66e−3)**+	6.6763e−1 (1.24e−2)−	8.3843e−1 (1.14e−2)=	8.4781e−1 (1.58e−2)=	8.6253e−1 (1.45e−2)+	8.5315e−1 (1.41e−2)+	8.3709e−1 (1.22e−2)
+/ − / =			1/1/5	0/7/0	0/5/2	0/2/5	1/1/5	1/0/6	–/–/–



(a) Wine (b) Segmentation (c) Parkinson (d) Ionosphere

(e) Sonar (f) Hillvalley (g) Semeion handwritten digit

Fig. 6. The nondominated solutions with best HV value achieved by the competing MOEAs and CMODE on seven datasets.

Table 7
Lowest error rate obtained by the seven competing MOEAs methods on seven datasets.

Problem	M	D	NSGA-II	MOEA/D	MOEA/D-DE	GDE3	CMOPSO	EMOSO	CMODE
Wine	2	13	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
Segmentation	2	19	**1.5152E−02**	6.4935E−02	2.3810E−02	**1.5152E−02**	**1.5152E−02**	**1.5152E−02**	**1.5152E−02**
Parkinson	2	22	**0.0000E+00**	5.1282E−02	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
Ionosphere	2	34	2.8571E−02	7.1429E−02	4.2857E−02	**0.0000E+00**	1.4286E−02	1.4286E−02	1.4286E−02
Sonar	2	60	**0.0000E+00**	2.3810E−02	4.7619E−02	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
Hillvalley	2	100	**3.0579E−01**	4.7107E−01	4.0496E−01	3.5537E−01	3.3058E−01	3.4711E−01	3.5537E−01
Semeion handwritten digit	2	256	4.0881E−02	2.2956E−01	9.4340E−02	**3.7736E−02**	4.7170E−02	4.7170E−02	4.4025E−02

The performance of the proposed CMODE is comprehensively evaluated by comparing it with six popular MOEAs on twenty benchmarks with different characteristics. Experiments are also done to compare the CMODE with two variants and to analysis the influences of the parameter α. Experimental results demonstrate that, on most of the twenty benchmark functions, our proposed CMODE outperforms the other six algorithms, and CMODE performs better than the two variants of CMODE. Finally, the proposed CMODE is also applied to solve the feature selection problem and results on feature selection also validate the competitive performance of our proposed CMODE. In the future, we will combine the competitive mechanism with other swarm intelligence algorithms [78–81] to design multi-objective algorithms and apply our proposed algorithms to deal with other real-world applications such as task scheduling problems [82].

CRediT authorship contribution statement

Jeng-Shyang Pan: Supervision, Investigation, Writing – review & editing. **Nengxian Liu:** Conceptualization, Investigation, Methodology, Software, Data curation, Visualization, Writing – original draft. **Shu-Chuan Chu:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.G. Herrero, A. Berlanga, J.M. Molina López, Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters, IEEE Trans. Evol. Comput. 13 (1) (2009) 151–168.

[2] L. Zhang, H. Pan, Y. Su, X. Zhang, Y. Niu, A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection, IEEE Trans. Cybern. 47 (9) (2017) 2703–2716.

[3] S.K. Goudos, J.N. Sahalos, Pareto Optimal microwave filter design using multiobjective differential evolution, IEEE Trans. Antennas and Propagation 58 (1) (2010) 132–144.

[4] W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, M.O. Cinnéide, High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III, in: GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference, 2014, pp. 1263–1270.

[5] A. Ponsich, A.L. Jaimes, C.A. Coello, A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications, IEEE Trans. Evol. Comput. 17 (3) (2013) 321–344.

[6] Y. Tian, S. Yang, X. Zhang, Y. Jin, Using platemo to solve multi-objective optimization problems in applications: A case study on feature selection, in: 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings, 2019, pp. 1710–1717.

[7] L. Abualigah, A. Diabat, M.A. Elaziz, Intelligent workflow scheduling for big data applications in IoT cloud computing environments, Cluster Comput. 24 (2021) 2957–2976.

[8] J. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: The 1st International Conference on Genetic Algorithms, no. JANUARY 1985, 1985, pp. 93–100.

[9] K. Deb, N. Srinivas, Multiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1995) 221–248.

[10] I. Fister, M. Perc, S.M. Kamal, I. Fister, A review of chaos-based firefly algorithms: Perspectives and research challenges, Appl. Math. Comput. 252 (2015) 155–165.

[11] S. Qin, C. Sun, G. Zhang, X. He, Y. Tan, A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems, Complex Intell. Syst. 6 (2) (2020) 263–274.

[12] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, F.E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, Neurocomputing 432 (2021) 170–182.

[13] L. Abualigah, A. Diabat, Advances in Sine cosine algorithm: A comprehensive survey, Artif. Intell. Rev. 54 (4) (2021) 2567–2608.

[14] S. Kukkonen, J. Lampinen, GDE3: The third evolution step of generalized differential evolution, in: 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings, Vol. 1, 2005, pp. 443–450.
[15] D. Goldberg, B. Korb, K. Deb, Messy genetic algorithms : Motivation, analysis, and first results, Complex Syst. 3 (5) (1989) 493–530.
[16] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248.
[17] K. Deb, S. Agrawal, A. Pratab, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: {NSGA-{II}}, IEEE Trans. Evol. Comput. (2002) 182–197.
[18] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evol. Comput. 10 (3) (2002) 263–282.
[19] G. Wang, H. Jiang, Fuzzy-dominance and its application in evolutionary many objective optimization, in: Proceedings - CIS Workshops 2007, 2007 International Conference on Computational Intelligence and Security Workshops, 2007, pp. 195–198.
[20] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736.
[21] X. Zhang, X. Zheng, R. Cheng, J. Qiu, Y. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, Inform. Sci. 427 (2018) 63–76.
[22] X.W. Zhang, H. Liu, L.P. Tu, J. Zhao, An efficient multi-objective optimization algorithm based on level swarm optimizer, Math. Comput. Simulation 177 (2020) 588–602.
[23] A. Got, D. Zouache, A. Moussaoui, MOMRFO: Multi-objective manta ray foraging optimizer for handling engineering design problems, Knowl.-Based Syst. 237 (2022) 1–17.
[24] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[25] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[26] H.L. Liu, L. Chen, Q. Zhang, K. Deb, Adaptively allocating search effort in challenging many-objective optimization problems, IEEE Trans. Evol. Comput. 22 (3) (2018) 433–448.
[27] R. Wang, Q. Zhang, T. Zhang, Decomposition-based algorithms using pareto adaptive scalarizing methods, IEEE Trans. Evol. Comput. 20 (6) (2016) 821–837.
[28] C. Dai, Y. Wang, M. Ye, A new multi-objective particle swarm optimization algorithm based on decomposition, Inform. Sci. 325 (2015) 541–557.
[29] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[30] Y. Zhao, C. Sun, J. Zeng, Y. Tan, G. Zhang, A surrogate-ensemble assisted expensive many-objective optimization, Knowl.-Based Syst. 211 (2021) 1–13.
[31] L. Zhu, J. Lin, Y.Y. Li, Z.J. Wang, A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem, Knowl.-Based Syst. 225 (2021) 1–18.
[32] J. Li, P. Wang, H. Dong, J. Shen, C. Chen, A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization, Knowl.-Based Syst. (2022) in press.
[33] E. Zitzler, S. Künzli, Indicator-Based Selection in Multiobjective Search, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3242, 2004, pp. 832–842.
[34] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European J. Oper. Res. 181 (3) (2007) 1653–1669.
[35] J. Bader, E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[36] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2018) 609–622.
[37] H. Wang, L. Jiao, X. Yao, Two Arch2: An improved two-archive algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (4) (2015) 524–541.
[38] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2015) 694–716.
[39] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[40] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 1–38.
[41] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. 45 (2) (2015) 191–204.
[42] N. Liu, J.-S. Pan, S.-C. Chu, A competitive learning quasi affine transformation evolutionary for global optimization and its application in CVRP, J. Internet Technol. 21 (7) (2020) 1863–1883.
[43] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Trans. Cybern. 50 (8) (2020) 3696–3708.
[44] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution - An updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[45] Z. Meng, J.S. Pan, K.K. Tseng, PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl.-Based Syst. 168 (2019) 80–99.
[46] J.S. Pan, N. Liu, S.C. Chu, A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning, IEEE Access 8 (2020) 17691–17712.
[47] K. Price, R. Storn, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. (11) (1997) 341–359.
[48] C.S. Chang, D. Xu, H. Quek, Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system, IEE Proc.: Electr. Power Appl. 146 (5) (1999) 577–581.
[49] H.A. Abbass, R. Sarker, C. Newton, PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems, in: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, Vol. 2, 2001, pp. 971–978.
[50] H.A. Abbass, The self-adaptive Pareto differential evolution algorithm, in: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, Vol. 1, 2002, pp. 831–836.
[51] J. Lampinen, DE's Selection Rule for Multiobjective Optimization, Tech. Rep., Lappeenranta University of Technology, Department of Information Technology, 2001, pp. 3–4.
[52] S. Kukkonen, J. Lampinen, An Extension of Generalized Differential Evolution for Multi-Objective Optimization with Constraints, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3242, 2004, pp. 752–761.
[53] F. Xue, A.C. Sanderson, R.J. Graves, Pareto-based multi-objective differential evolution, in: 2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings, Vol. 2, 2003, pp. 862–869.
[54] A.W. Iorio, X. Li, Solving Rotated Multi-Objective Optimization Problems Using Differential Evolution, in: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 3339, 2004, pp. 861–872.
[55] S.Z. Zhao, P.N. Suganthan, Q. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Trans. Evol. Comput. 16 (3) (2012) 442–446.
[56] T. Robič, B. Filipič, DEMO: Differential evolution for multiobjective optimization, Lecture Notes in Comput. Sci. 3410 (2005) 520–533.
[57] P. Rakshit, A. Konar, S. Das, L.C. Jain, A.K. Nagar, Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise, IEEE Trans. Syst. Man Cybern. 44 (7) (2014) 922–937.
[58] Q. Jiang, L. Wang, J. Cheng, X. Zhu, W. Li, Y. Lin, G. Yu, X. Hei, J. Zhao, X. Lu, Multi-objective differential evolution with dynamic covariance matrix learning for multi-objective optimization problems with variable linkages, Knowl.-Based Syst. 121 (2017) 111–128.
[59] N. Saini, S. Saha, A. Jangra, P. Bhattacharyya, Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm, Knowl.-Based Syst. 164 (2019) 45–67.
[60] X. Wang, Z. Dong, L. Tang, Multiobjective differential evolution with personal archive and biased self-adaptive mutation selection, IEEE Trans. Syst. Man Cybern. 50 (12) (2020) 5338–5350.
[61] M. Tian, X. Gao, X. Yan, Performance-driven adaptive differential evolution with neighborhood topology for numerical optimization, Knowl.-Based Syst. 188 (2020) 1–27.
[62] A. Jamali, R. Mallipeddi, M. Salehpour, A. Bagheri, Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system, Swarm Evol. Comput. 54 (2020) 1–14.
[63] E.V. Altay, B. Alatas, Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining, Inform. Sci. 554 (2021) 198–221.
[64] C. Yue, P.N. Suganthan, J. Liang, B. Qu, K. Yu, Y. Zhu, L. Yan, Differential evolution using improved crowding distance for multimodal multiobjective optimization, Swarm Evol. Comput. 62 (2021) 1–10.
[65] N. Schussler, K.W. Axhausen, SPEA2: Improving the strength pareto evolutionary algorithm, Transp. Res. Rec. 2105 (1) (2009) 28–36.
[66] M. Li, S. Yang, X. Liu, Shift-based density estimation for pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 348–365.

[67] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A matlab platform for evolutionary multi-objective optimization, IEEE Comput. Intell. Mag. (2017).
[68] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, Evol. Multiobjective Optim. (2005) 105–145.
[69] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, Lecture Notes in Comput. Sci. 3410 (2005) 280–295.
[70] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[71] C.A. Coello, N.C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genet. Program. Evol. Mach. 6 (2) (2005) 163–190.
[72] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.
[73] J. Manyika, M. Chui Brown, B.J. Bughin, R. Dobbs, C. Roxburgh, A. Hung Byers, Big data: The next frontier for innovation, competition and productivity, McKinsey Glob. Inst. (June) (2011) 156, URL https://bigdatawg.nist.gov/pdf/MGI{_}big{_}data{_}full{_}report.pdf.
[74] Q. Al-Tashi, S.J. Abdulkadir, H.M. Rais, S. Mirjalili, H. Alhussian, Approaches to multi-objective feature selection: A systematic literature review, IEEE Access 8 (2020) 125076–125096.
[75] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671.
[76] N. Liu, J.-S. Pan, J. Lai, S.-C. Chu, T.-T. Nguyen, An efficient differential evolution via both top collective and P-best information, J. Internet Technol. 21 (3) (2020) 629–643.
[77] A. Frank, A. Asuncion, {UCI} Machine learning repository, 2010, URL http://archive.ics.uci.edu/ml.
[78] Z. Meng, J.S. Pan, H. Xu, Quasi-affine transformation evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization, Knowl.-Based Syst. 109 (2016) 104–121.
[79] J.-S. Pan, P. Hu, S.-C. Chu, Binary fish migration optimization for solving unit commitment, Energy (2021) 120329.
[80] P.C. Song, S.C. Chu, J.S. Pan, H. Yang, Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine, in: 2nd International Conference on Industrial Artificial Intelligence, IAI 2020, 2020.
[81] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A. Al-qaness, A.H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 1–37.
[82] L. Abualigah, M. Alkhrabsheh, Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing, J. Supercomput. 78 (1) (2022) 740–765.