

Research Article

A Multistrategy-Based Multiobjective Differential Evolution for Optimal Control in Chemical Processes

Bin Xu ¹, Xu Chen ², Xiuhui Huang,³ and Lili Tao⁴

¹School of Mechanical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

²School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China

³School of Energy and Power Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

⁴College of Engineering, Shanghai Second Polytechnic University, Shanghai 201209, China

Correspondence should be addressed to Bin Xu; bxu@sues.edu.cn

Received 17 May 2018; Accepted 25 July 2018; Published 16 September 2018

Academic Editor: Zhile Yang

Copyright © 2018 Bin Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Optimal control problems with multiple conflicting objectives in chemical processes are quite challenging. To solve such problems, we put forward a multistrategy-based multiobjective differential evolution, in which (1) a hybrid selection strategy is incorporated from the motivation of no single strategy outperforming all other ones in every stage; (2) a multipopulation strategy is applied to represent the main population and current optimum, and a cyclic crowding estimation is developed to maintain these optimum; and (3) a multimutation strategy is constructed to improve both exploration and exploitation ability. The effectiveness and efficiency of the proposed algorithm are validated by comparisons with some representative multiobjective evolutionary algorithms over 12 test instances. Moreover, the proposed algorithm is applied to solve 3 multiobjective optimal control problems in chemical processes. The obtained results indicate the efficiency and effectiveness of the proposed algorithm for solving multiobjective optimal control problems.

1. Introduction

Almost all optimal control problems (OCPs) in engineering have been considered to be single-objective optimization problems. In practical, for most of OCPs in chemical process, they involve multiple conflicting objectives [1]. For instance, some chemical processes are characterized by improving both product quality and yield with low energy consumption [2]. For such multiobjective optimal control problems (MOOCs), there does not exist a trajectory which optimizes all objectives simultaneously. On the contrary, there exist many optimal trajectories, called Pareto optimal solutions in a multiobjective optimization community, which represent some trade-off between the conflicting objectives.

One common and straightforward manner to solve a given MOOC is transforming the original problem into a single-objective optimal control problem (SOOC) using some weighted sum methods and then optimizing it [3–5]. One intrinsic characteristic common in the single-objective-based methods is that they aim at locating just one optimal

solution rather than multiple optimal solutions in a single run. Thus, many runs should be conducted to obtain the multiple optimal solutions. Furthermore, the relation between a uniform grid of weights and the spread on the Pareto front is unclear.

With the development of evolutionary algorithms (EAs), multiobjective evolutionary algorithms (MOEAs) have been widely used in plenty of fields [6]. One advantage of using MOEAs to solve such problems is that they can find a set of representative optimal solutions in a single run [7]. Recently, some researchers have demonstrated that MOEAs not only are effective for common multiobjective problems (MOPs) but also can be extended to solve MOOCs. Sarkar and Modak used the enhanced version of a nondominated sorting genetic algorithm (NSGA-II) to solve two fed-batch bioreactors [8]. Logist et al. used the freely available toolkit named ACADO, to solve different types of MOOCs appearing in chemical engineering with different objective numbers [1]. Chen et al. improved the performance of MODE by using a ranking-based mutation operator and then applied the new

MODE to solve 4 MOOCs in chemical process [9]. Fan et al. proposed a multiobjective differential evolution with performance metric-based self-adaptive mutation operator for typical MOOCs in chemical and biochemical processes [10]. Results showed that the proposed new approach is suitable for solving actual problems and can obtain some Pareto optimum for decision makers.

When solving the MOOC by MOEAs, the principle of it, as shown in Figure 1, can be summarized as follows: firstly, transforming the MOOC into common MOPs and then MOEAs can be applied to solve the transformed problems. Therefore, questions arising naturally in solving MOOC are twofold: how to perform transforming and how to design effective MOEAs. In this paper, we focus on the second issue and take the most used control vector parameterization (CVP) method to meet the first one.

The last two decades have witnessed major developments in EA-based algorithms for MOPs. Differential evolution (DE) [11], a simple yet effective nature-inspired stochastic algorithm, has been widely and successfully applied for solving various complex problems [9, 12–14]. However, it must be emphasized that DE was originally proposed for single-objective problems, while the main goal of a multiobjective optimization algorithm is to find a set of trade-off solutions. Thus, to solve MOPs effectively via DE, the following two issues should be considered:

- (1) How to select and/or retain the best solutions? That is, how to perform elitism.
- (2) How to promote diversity into the population.

In fact, the basic DE can be directly applied to solve MOPs by replacing the selection component [6]. Thus, one key issue in designing a MODE is designing a proper selection component. In general, the selection component includes assigning a fitness value for each candidate solution and selecting the promising ones for the new population. As pointed out in [15], different selection components in DE makes an algorithm differ from others. During the last three decades, plenty of selection components with different strategies have been developed for solving MOPs. One notable difference among these strategies is how the Pareto dominance concept is used. From this aspect, we classify the current selection strategies into three categories:

1.1. No Pareto Dominance Used. It combines all the multiple objectives into a single one, such as weighted sum approach, and then the original EAs can be applied, such as [16–19] and so on. Among all these algorithms, MOEA/D is a recent multiobjective evolutionary algorithmic framework which is based on conventional aggregation approaches [18]. MOEA/D has a fast execution speed and some improved MOEA/D-based algorithms are proposed [7, 20]. However, they need to provide a set of uniformly distributed weights and a decomposition method. This extra information highly affects MOEA/D’s performance.

1.2. Use Pareto Dominance as a Ranking Method. In this group, Pareto ranking, also called the nondominated sorting

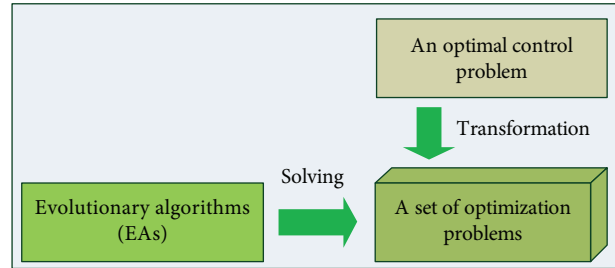


FIGURE 1: The schematic graph to illustrate the principle of solving an OCP by EAs.

method [21, 22], is used for fitness assignment for the conflicting objectives first and then the selection based on the ranking numbers can be performed. Much research uses this method as a selection strategy [21–29].

1.3. Use Pareto Dominance as a Filter. Here, a filter means finding a set of better solutions from a given set. In this situation, Pareto dominance is often used in tournament selection to find a better solution among several candidates or to identify a set of nondominated solutions [30–37]. Since this group of strategies puts more efforts on the elitists, it has a very fast convergence speed. However, it easily gets the search stuck at local optima for multimodal problems.

According to the no free lunch theorem [38], it is impossible for a single-selection strategy to outperform all other strategies on all problems at different search stages. In other words, depending on several selection strategies can be effective for different problems during different search stages. Motivated by these observations, we propose a new MOEA named DEHC using DE with a hybrid selection strategy (HSS) and cyclic crowding estimation (CCE) for MOOCs. For the hybrid selection strategy, the decomposition-based strategy [18], nondominated sorting strategy [22], and nondominated neighbor-based strategy [33] are adopted as the selection strategy. As for the diversity estimation, in this paper, we introduce an external population to store all the found nondominated solutions and then use a cyclic crowding estimation to maintain this population. Furthermore, a multiple-mutation strategy-based mutation component is used as the main reproduction operator.

The remainder of this paper is arranged as follows. Section 2 briefly presents a basic concept of MOOC and conventional DE. Section 3 presents our DEHC in detail. Section 4 presents simulation and comparative results of DEHC with other competing algorithms. In Section 5, DEHC is applied to solve 3 MOOCs in chemical processes. Section 6 draws conclusions.

2. Basic Concepts

2.1. MOOC: Multiobjective Optimal Control Problem. Consider the system $\dot{x} = f(x(t), u(t), t)$ with state variables $x(t) \in \mathbb{R}^D$, initial condition $x(t_0) = x_0$, and control variables $u(t) \in \mathbb{R}^K$. Then the main goal of solving the problem is to find the optimal control variables $u^*(t), t \in [t_0, t_f]$ that drive the plant along the trajectory $x^*(t), t \in [t_0, t_f]$ such that all

the cost functions are minimized where the final time t_f is fixed. The structure of a typical MOOCP for a continuous process can be generally described as follows:

$$\min \{J_1, J_2, \dots, J_M\}, \quad (1)$$

subject to

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), u(t), t), \quad t \in [t_0, t_f], \\ L &\leq u \leq U. \end{aligned} \quad (2)$$

Each of the objective functions $J_i (i = 1, 2, \dots, M)$ in (1) can be formulated as follows:

$$J_i(u(t)) = \phi_i(x(t_f), t_f) + \int_{t_0}^{t_f} \psi_i(x(t), u(t), t) dt, \quad (3)$$

where M is the objective number, $J_i(\cdot)$ is the total cost function, $\phi_i(\cdot)$ is the final time performance index, $\psi_i(\cdot)$ is the integrated performance during the operation, t_0 is the initial time, and t_f is the final time; $L \in \mathbb{R}^K$ and $U \in \mathbb{R}^K$ are the lower and upper boundary of control variables, respectively.

2.2. Multiobjective Optimization. Multiobjective optimization algorithms are aimed at optimizing conflicting objectives simultaneously. To aid descriptions, some definitions regarding multiobjective optimization are introduced.

Definition 1. Solution u_1 is said to dominate solution u_2 , denoted as $u_1 \succ u_2$, if and only if $\forall_i \in \{1, \dots, M\}$, $J_i(u_1) \leq J_i(u_2)$, and $\exists j \in \{1, \dots, M\}$, $J_j(u_1) < J_j(u_2)$. Reciprocally, solution u_2 is said to be dominated by solution u_1 , denoted as $u_2 \prec u_1$.

Definition 2. Solution u^* is said to be a Pareto optimum only if $\neg \exists u \in S, u \succ u^*$. All Pareto optima constitute a Pareto optimal set, denoted as X^* .

Definition 3. Pareto optimal front is defined as $PF = \{J(u^*) \mid u^* \in X^*\}$, that is, the mapping of the Pareto optimal set in the objective space.

2.3. Differential Evolution Algorithm. Like other EAs, DE begins with a randomly initialized population in the search space and then adopts trial vector generation and selection operators sequentially at each generation to move the main population toward the global optimum.

The trial vector generation comprises two operators: mutation and crossover operators. The following are the five frequently used mutation strategies in the literature:

(1) DE/rand/1

$$v_i^g = \mathbf{x}_{r_1}^g + F \cdot (\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) \quad (4)$$

(2) DE/best/1

$$v_i^g = \mathbf{x}_{\text{best}}^g + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (5)$$

(3) DE/current-to-best/1

$$v_i^g = \mathbf{x}_i^g + F \cdot (\mathbf{x}_{\text{best}}^g - \mathbf{x}_i^g) + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (6)$$

(4) DE/best/2

$$v_i^g = \mathbf{x}_{\text{best}}^g + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) + F \cdot (\mathbf{x}_{r_3}^g - \mathbf{x}_{r_4}^g) \quad (7)$$

(5) DE/rand/2

$$v_i^g = \mathbf{x}_{r_1}^g + F \cdot (\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) + F \cdot (\mathbf{x}_{r_4}^g - \mathbf{x}_{r_5}^g) \quad (8)$$

where indices $r_i, i = 1, \dots, 5$ are mutually exclusive integers randomly generated within $[1, N]$, respectively, that are also different from index i ; $\mathbf{x}_{\text{best}}^g$ is the best vector in the population at generation g ; F is the scale factor and is usually chosen between 0 and 1.

The most frequently used binomial crossover operation is performed as follows:

$$\mathbf{u}_{i,j}^g = \begin{cases} \mathbf{v}_{i,j}^g, & \text{if } (\text{rand} \leq \text{CR}) \text{ or } j = \text{sn}, \\ \mathbf{x}_{i,j}^g, & \text{otherwise,} \end{cases} \quad (9)$$

where sn is an arbitrary number in $\{1, 2, \dots, n\}$ and $u_{i,j}$ is the j th element of the i th new trial vector.

After generating the trial vectors, a greedy selection operator is performed as follows:

$$\mathbf{x}_i^{g+1} = \begin{cases} u_i^g, & \text{if } (f(u_i^g) \leq f(x_i^g)), \\ x_i^g, & \text{otherwise,} \end{cases} \quad (10)$$

where \mathbf{x}_i^{g+1} is the i th solution in the main population of the next generation.

3. Our Proposed DEHC

Because of the conflicting objectives, the DE can not be applied to solve MOPs directly. To overcome this limitation, the proposed DEHC has the following features:

(1) In the mutation step, (4) and (5) constitute a strategy candidate population pool and a random strategy is determined from this candidate pool for each solution.

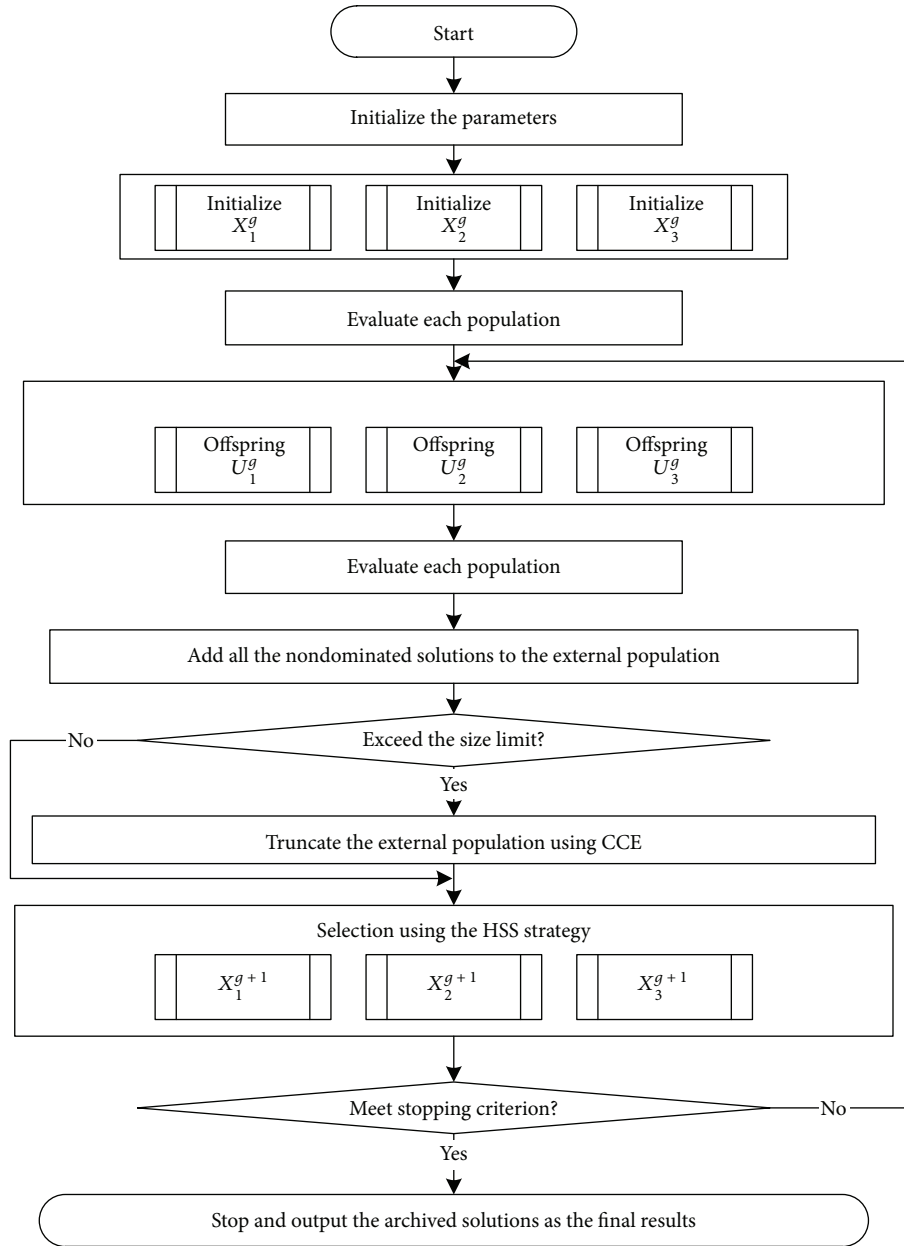


FIGURE 2: The flowchart of DEHC.

- (2) In the selection step, the greedy strategy-based selection component is replaced by a new selection component based on hybrid selection strategy.
- (3) An external population is introduced to store the best solutions found so far and to provide best solutions when strategy (5) is adopted.

The overall flowchart of DEHC is shown in Figure 2. Below, we elaborate on some of the main steps in detail.

3.1. Mutation Component with Two Strategies. For any MOPs, there is always more than one best solution, which are also called nondominated solutions. In our approach,

an external population, denoted as E , is introduced and the best solutions found are stored in it. This has two purposes:

- (1) The solutions in the external population are the final results reported to the performer(s).
- (2) The external population is the best solution candidate pool for the best solution based on mutation strategy.

In DEHC, (4) and (5) are selected to constitute a strategy candidate pool. In the mutation step, a strategy is randomly selected from the pool for each solution. To perform the best solution-based strategy (5), a best solution must be determined. Inspired by Coello et al.'s method [39], the best

solution is determined as follows: first, the crowding distance of every solution in E is calculated [22], and then two solutions are randomly selected from the population. Their crowding distances are compared and the greater will be the final best solution for participating in the mutation operator. Note that a new best solution needs to be selected for each solution if the mutation strategy is (5). The purposes for this extension are twofold:

- (1) Strategy (4) has advantage of putting more effort into improving the search engine's exploration ability.
- (2) Strategy (5) has advantage of putting more effort into improving the search engine's exploitation ability. Furthermore, it has advantage of putting more effort into exploiting the less crowded area.

3.2. Selection Component Based on Hybrid Selection Strategy. Selection component is the most essential step for any MOEAs because it is responsible for guiding the selection process at the various stages of the algorithm toward a Pareto optimal front. Many selection components with different selection strategies have been proposed for solving MOPs. Here, in DEHC, we use a new selection component called hybrid selection strategy (HSS) from the view of multipopulation. The HSS consist of three selection strategies and each strategy has its own subpopulation. These three strategies are called decomposition-based strategy [18], nondominated sorting strategy [22], and nondominated neighbor-based strategy [33]. These three strategies fall into the above three groups in Sections 1.1, 1.2, and 1.3, respectively. The reasons for hybridizing these three selection strategies are as follows:

- (1) Decomposition-based strategy has high execution speed; however, it needs providing a set of uniformly distributed weight. Although determining a set of weight is an easy job for problems with 2 objectives and 3 objectives, it becomes a tough thing as the number of objectives increases. Furthermore, it needs a decomposition approach. Most important of all, these weight values and decomposition approach highly affect the final results.
- (2) Nondominated sorting strategy can maintain the diversity of the main population. But it has a $O(mN^2)$ computational complexity (where N is the population size). So, less population size means lower computational time.
- (3) Nondominated neighbor-based strategy has a very fast convergence speed because it puts more efforts on the best solutions. This high selection pressure may mislead the search into local optima.

The proposed HSS works as follows: generating the offspring for each subpopulation and then assigning the fitness value of each solution in the combination of its subpopulation and offspring subpopulation according to the corresponding selection strategy and finally selecting a proper

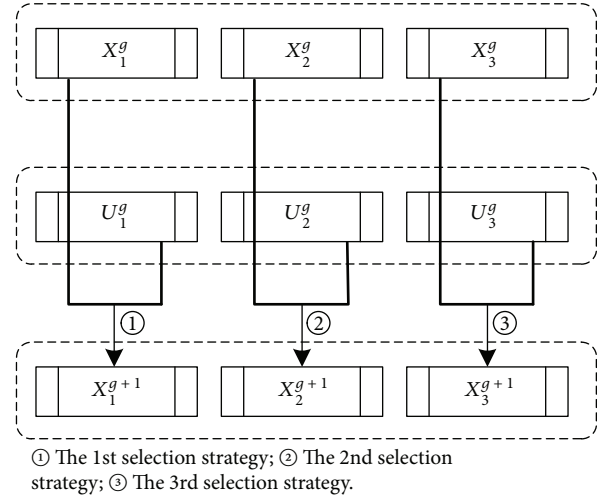


FIGURE 3: The main selection process using HSS.

number of best solutions to form its subpopulation of next generation. This process is described in Figure 3.

The main characteristic of HSS is that it utilizes the advantage of every selection strategy such as the high execution speed of the decomposition based-strategy, high population diversity of the nondominated sorting strategy, and fast convergence speed of the nondominated neighbor-based strategy. So, different selection strategies have different trade-offs between the conflicting objectives and the hybrid strategy can take advantage of all these trade-off. Although each strategy has its own subpopulation, these subpopulations are not independent. In the mutation step, all 3 subpopulations are combined together to generate new offspring individuals. Furthermore, all the best solutions found are stored in an external population and this population provides guiding information in performing the best solution-based mutation strategy in Section 3.1.

3.3. Maintenance of External Population Based on Cyclic Crowding Estimation. Maintaining an external population is beneficial [29, 30], but two issues arise in the maintenance. One is how to add the currently found best solutions into the population and the other is how to remove the inferior ones. Denote the external population as E and the currently found best solutions in the offspring as Z . Then we can add Z into E by Algorithm 1.

After adding Z into E , we need to check whether its size exceeds the size limit since the number of nondominated solutions can be huge and the computational complexity of maintaining all of them is high. Hence, we truncate the population to a predefined number. To do this, the crowding distance is the most commonly applied mechanism as for its simplification and free of parameters [22, 29, 33]. This method removes all redundant solutions at once. However, the removing of a solution will lead to the change of its neighborhood's crowding distance value. From this perspective, the one-time operator does not provide good result in all cases.

To overcome this drawback, CCE is presented. The main characteristic of CCE is to remove the most crowded


```

1: for each  $z_i \in Z$  do
2:   if  $z_i$  is dominated by any member of  $E$  then
3:     discard  $z_i$ 
4:   end if
5:   if  $z_i$  dominates a set of members  $D(z_i)$  from  $E$  then
6:      $E = E \setminus D(z_i)$ ;
7:      $E = E \cup z_i$ 
8:   end if
9:   if AHC and  $z_i$  are non-dominated with each other then
10:     $E = E \cup z_i$ 
11:   end if
12: end for

```

ALGORITHM 1: Pseudocode of CCE.

```

Input: The external population  $E$  with  $Q$  solutions
Output: The truncated external population  $E$  with  $N$  solutions
1: Let  $K = Q - N$ ;
2: for each  $x_i \in E$  do
3:   Calculate the Euclidean distance  $d(i, j)$  between  $x_i$  and  $x_j$ , where  $j = i, \dots, Q$ ;
4:   Sort  $d(i, j)$ ,  $j = i, \dots, Q$ , in ascending order;
5:   Add the first  $K + 1$  values and the corresponding solutions as  $K + 1$ 
   records into  $D$ ;
6: end for
7: while  $K > 0$  do
8:   Find the record with minimal distance in  $D$  and denote this value as
    $V_{ab}$  and the corresponding two solutions as  $P_a$  and  $P_b$ , respectively;
9:   Remove this record from  $D$ ;
10:  Find the record with minimal distance values associated with  $P_a$  and
    $P_b$  in  $D$ , respectively; denote the values as  $V_a$  and  $V_b$ , respectively;
11: if  $V_a \leq V_b$  then
12:   Remove  $P_a$  from  $E$ 
13:   Remove all records associated with  $P_a$  in  $D$ 
14: else
15:   Remove  $P_b$  from  $E$ 
16:   Remove all records associated with  $P_b$  in  $D$ 
17: end if
18:    $K = K - 1$ 
19: end while

```

ALGORITHM 2: The main truncating process by the CCE process.

solution one by one or cyclically instead of retaining the least crowded solutions at a time. This process can be described by Algorithm 2.

3.4. Computational Complexity of DEHC. As pointed above, the DEHC differs from other MOEAs in three main features, that is, multiple-mutation operators, hybrid selection strategy, and cyclic crowding estimation. To use multiple-mutation strategies, we need to find the current best vector in each generation loop. This procedure depends solely on M -independent sortings at N solutions, which has a $O(MN \log(N))$ computational complexity.

For the hybrid selection strategy, each selection strategy has a subpopulation with size $N_s = (1/3)N$. The decomposition-based strategy has a time complexity of $O(MN_s T)$, where

T is the neighborhood size. For the nondominated sorting strategy, the computational complexity is $O(M(N_s)^2)$. As for the nondominated neighbor-based strategy, the worst computational complexity is $O(N_s \lg(N_s))$. So, the worst total computational complexity of HSS is

$$O(MN_s T) + O(M(N_s)^2) + O(N_s \log(N_s)). \quad (11)$$

As for the CCE, in the worst case, the time complexity for calculating the distances is $N(Q^2)$; the time complexity for adding the $K + 1$ values is $O(KQ)$; the time complexity for removing K solutions from E is $O((KQ)^2)$.

Based on the above analysis, the procedure of HSS does not increase any burden on the runtime complexity than

TABLE 1: The IGD comparison results of DEHC with 4 MOEAs on 12 problems.

	MOEA/D Mean \pm Std	NSGA-II Mean \pm Std	NNIA Mean \pm Std	MODE Mean \pm Std	DEHC Mean \pm Std
ZDT1	$6.15E-3 \pm 4.40E-3$	$5.19E-3 \pm 2.10E-4$	$4.57E-3 \pm 2.68E-4$	$3.56E-2 \pm 3.48E-3$	$3.87E-3 \pm 5.75E-5$
ZDT2	$4.37E-3 \pm 1.39E-3$	$5.16E-3 \pm 2.25E-4$	$4.66E-3 \pm 2.42E-4$	$6.78E-2 \pm 7.49E-3$	$3.92E-3 \pm 4.55E-5$
ZDT3	$1.71E-2 \pm 1.14E-2$	$1.30E-2 \pm 1.29E-2$	$5.19E-3 \pm 2.05E-4$	$5.02E-2 \pm 5.22E-3$	$4.61E-3 \pm 9.56E-5$
ZDT4	$9.74E-3 \pm 3.71E-3$	$6.36E-3 \pm 1.15E-3$	$5.61E-3 \pm 1.44E-3$	$3.88E-1 \pm 3.12E-1$	$3.83E-3 \pm 5.70E-5$
ZDT6	$5.52E-3 \pm 8.22E-4$	$7.94E-2 \pm 6.84E-3$	$3.64E-3 \pm 8.87E-5$	$2.30E-1 \pm 2.02E-2$	$3.054E-3 \pm 2.17E-5$
DTLZ1	$1.53E-2 \pm 1.58E-4$	$1.79E-2 \pm 1.70E-3$	$2.10E-2 \pm 2.22E-2$	$6.99E-1 \pm 5.62E-1$	$1.13E-2 \pm 7.71E-5$
DTLZ2	$3.84E-2 \pm 8.70E-5$	$4.00E-2 \pm 9.06E-4$	$4.03E-2 \pm 8.82E-4$	$6.34E-2 \pm 2.08E-3$	$3.04E-2 \pm 1.39E-4$
DTLZ3	$3.85E-2 \pm 2.88E-4$	$4.31E-2 \pm 4.10E-3$	$1.04E+0 \pm 4.99E+0$	$3.45E+0 \pm 1.43E+0$	$3.51E-2 \pm 1.51E-3$
DTLZ4	$2.51E-1 \pm 3.05E-1$	$3.89E-2 \pm 6.71E-4$	$3.94E-2 \pm 8.05E-4$	$6.48E-2 \pm 1.87E-3$	$3.03E-2 \pm 1.06E-4$
DTLZ5	$4.61E-3 \pm 1.21E-5$	$1.78E-3 \pm 4.75E-5$	$1.86E-3 \pm 8.18E-5$	$5.48E-3 \pm 2.94E-4$	$1.48E-3 \pm 2.12E-5$
DTLZ6	$5.51E-3 \pm 3.68E-4$	$4.50E-2 \pm 2.23E-2$	$6.07E-2 \pm 2.37E-2$	$3.44E+0 \pm 1.90E-1$	$1.36E-3 \pm 6.58E-6$
DTLZ7	$1.65E-1 \pm 1.49E-1$	$5.36E-2 \pm 5.94E-2$	$4.24E-2 \pm 1.56E-3$	$7.72E-2 \pm 3.32E-3$	$3.14E-2 \pm 2.96E-4$

using the single-selection strategy. As for the multiple-mutation strategies, it needs higher computational complexity, but it does not dramatically impose any serious burden on the runtime complexity than the single-mutation strategy-based operator. As for the CCE procedure, it needs much higher computational complexity than the complexity of crowding distance, which is only $O(MQ \log Q)$.

4. Experimental Study on Test Functions

This section presents the numerical results that we have conducted. First, we describe a set of MOPs and state the quality indicators for evaluating DEHC's performance. Then we evaluate DEHC by comparison with some other algorithms.

4.1. Test Problems and Performance Indicators. Twelve frequently used test MOPs from the literature are used here to evaluate the performance of DEHC. The first 5 are ZDT problems with two objectives and the next 7 are DTLZ problems with three objectives [22, 40]. Among all the test problems, ZDT1, ZDT2, ZDT3, and ZDT6 have 30 decision variables and the others have 10.

In order to make a quantitative assessment on the performance of DEHC, two performance indicators are employed to evaluate the obtained approximation sets. The first is an inverted generational distance indicator (IGD) [41]. Given an approximation set A and a reference set R , IGD is defined as follows:

$$\text{IGD}(A, R) = \frac{\sum_{v \in R} d(v, A)}{|R|}, \quad (12)$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in R . If R is large enough to represent the PF very well, $\text{IGD}(A, R)$ could measure both the diversity and convergence of in a sense. To have a low value of $\text{IGD}(A, R)$, A must be very close to the PF and cannot miss any part of the whole PF.

The second is a spread indicator (IS), which measures the distribution of an approximation set [22]. However, this

indicator works well only for bi-objective problems and can not be applied directly to problems with more than two objectives. Inspired by the work in [34, 35, 42], an extended IS is used here

$$\text{IS} = \frac{\sum_{i=1}^M (d_i^e) + \sum_{i=1}^N \sum_{j=1}^{M-1} |d_{ij} - \bar{d}|}{\sum_{i=1}^M (d_i^e) + (M-1)N\bar{d}}, \quad (13)$$

where M is the objective number and d_i^e is the Euclidean distance between the i th extreme solutions in A and R and $d_{i,j}$ is the Euclidean distance between the i th solution and its j th nearest solution in A . This one is a harmonic distance-based indicator because it takes all $M-1$ nearest neighbors around one solution into consideration. So, this works especially effectively for problems whose Pareto fronts consist of curved surfaces. A smaller value of this indicator means a better distribution. Particularly, a value of zero indicates that all the Pareto optima are equidistantly spaced.

4.2. Comparative Study. In order to assess DEHC, the final results are compared with those obtained by MOEA/D [18], NSGA-II [22], and NNIA [33] since they can be treated as algorithms with a single-selection strategy. Moreover, a recently proposed multiobjective DE (MODE) for dynamic optimization in chemical engineering is also selected as competing algorithm [43]. For all the 5 algorithms, the population sizes are set to 100 for the 2-objective and 300 for 3-objective problems, respectively. The maximal FFEs (fitness function evaluations) are set to 250.00 and 750.00 for the 2-objective and 3-objective problems, respectively. In DEHC, $F = 0.5$ and $\text{CR} = 0.1$. As for the other algorithms, their other parameters are kept the same as those in the corresponding references.

For each test problem, 25 independent runs are conducted and the statistical values are analyzed in detail. Table 1 shows the mean and standard deviation (Std) of IGD by 5 algorithms. It can be observed in Table 1 that DEHC is able to achieve the lowest values on most problems,

TABLE 2: The IS comparison results of DEHC with 4 MOEAs on 12 problems.

	MOEA/D Mean \pm Std	NSGA-II Mean \pm Std	NNIA Mean \pm Std	MODE Mean \pm Std	DEHC Mean \pm Std
ZDT1	$3.78E-1 \pm 1.23E-1$	$4.90E-1 \pm 3.20E-2$	$3.52E-1 \pm 2.89E-2$	$4.23E-1 \pm 1.34E-1$	$1.43E-1 \pm 1.39E-2$
ZDT2	$2.43E-1 \pm 1.68E-1$	$4.62E-1 \pm 3.33E-2$	$3.42E-1 \pm 3.21E-2$	$5.81E-1 \pm 3.71E-2$	$1.49E-1 \pm 1.13E-2$
ZDT3	$8.86E-1 \pm 1.48E-2$	$6.08E-1 \pm 3.97E-2$	$5.30E-1 \pm 1.94E-2$	$7.47E-1 \pm 6.26E-2$	$4.50E-1 \pm 1.34E-2$
ZDT4	$6.28E-1 \pm 3.27E-1$	$3.60E-1 \pm 2.70E-2$	$3.28E-1 \pm 2.56E-2$	$1.33E+0 \pm 3.33E-1$	$1.23E-1 \pm 1.18E-2$
ZDT6	$1.98E-1 \pm 3.10E-2$	$5.87E-1 \pm 2.92E-2$	$3.23E-1 \pm 2.51E-2$	$7.17E-1 \pm 2.65E-1$	$1.37E-1 \pm 1.27E-2$
DTLZ1	$6.14E-1 \pm 6.95E-3$	$4.63E-1 \pm 3.54E-2$	$4.92E-1 \pm 8.08E-2$	$9.80E-1 \pm 4.17E-1$	$8.65E-2 \pm 5.43E-3$
DTLZ2	$6.47E-1 \pm 4.90E-3$	$4.42E-1 \pm 2.54E-2$	$4.55E-1 \pm 2.15E-2$	$5.85E-1 \pm 2.41E-2$	$9.50E-2 \pm 3.22E-3$
DTLZ3	$6.57E-1 \pm 1.01E-2$	$4.53E-1 \pm 3.24E-2$	$4.81E-1 \pm 2.95E-2$	$1.23E+0 \pm 3.24E-1$	$1.94E-1 \pm 2.94E-2$
DTLZ4	$7.82E-1 \pm 1.60E-1$	$4.16E-1 \pm 2.03E-2$	$4.34E-1 \pm 1.82E-2$	$5.93E-1 \pm 2.53E-2$	$9.54E-2 \pm 3.85E-3$
DTLZ5	$1.15E+0 \pm 3.99E-3$	$3.54E-1 \pm 1.24E-2$	$3.68E-1 \pm 1.82E-2$	$8.59E-1 \pm 1.01E-2$	$1.66E-1 \pm 7.53E-3$
DTLZ6	$8.43E-1 \pm 1.00E-1$	$4.86E-1 \pm 3.79E-2$	$5.35E-1 \pm 4.98E-2$	$3.73E-1 \pm 2.63E-2$	$1.59E-1 \pm 7.99E-3$
DTLZ7	$1.04E+0 \pm 4.34E-2$	$5.15E-1 \pm 3.58E-2$	$4.87E-1 \pm 2.47E-2$	$4.65E-1 \pm 4.13E-2$	$2.39E-1 \pm 8.57E-3$

which means that DEHC converges better than any other algorithms.

Table 2 shows the mean and Std of IS by 5 algorithms. It can be observed in Table 2 that DEHC is able to obtain a better distributed Pareto optima set than any other algorithms on most problems. Although MOEA/D achieves better Std values on DTLZ1, DTLZ3 and NNIA achieve better Std values on DTLZ3 and DEHC achieves better mean values than MOEA/D and NNIA.

In order to determine whether DEHC is significantly better than other algorithms, statistical significance tests of IGD and IS are conducted between DEHC and MOEA/D, NSGA-II, NNIA, and MODE via Mann-Whitney rank sum test, respectively. Table 3 gives the comparison results. $h = 1$ indicates a significant difference between the compared algorithms with a level of significance $\beta = 0.05$, whereas $h = 0$ indicates that the performance is not significantly different. It can be observed from the table that DEHC performs significantly better on almost all of the problems except for ZDT1 in terms of IGD with a level of significance β . As for ZDT1, the results do not show a significant difference between DEHC and NSGA-II in terms of IGD indicator. As for IS, the results show that DEHC performs better than other algorithms with a level of significance β .

To provide more information about the convergence performance of the 5 MOEAs, we calculate IGD and IS values of found optimal solutions every 25 generations. Because in our study, all the 5 algorithms employ the same initialized population to avoid difference at the initialization stage, we do not calculate the indicators for the initial population but simply assume that they are at 1. Figures 4–7 illustrate the 2 indicators of the above 5 algorithms versus the evolution time over 25 runs in all the 12 instances.

For illustration, we also show some typical fronts obtained by the 5 algorithms. Figure 8 presents the fronts on DTLZ1. Please note that these fronts correspond to the run with a median value with respect to IS. This figure clearly demonstrates that for DTLZ1, the 5 algorithms can visually converge to Pareto optimal fronts. However, the distributions have an

TABLE 3: The distribution of IGD and IS using Mann-Whitney rank sum test.

	(DEHC, MOEA/D)		(DEHC, NSGA-II)		(DEHC, NNIA)		(DEHC, MODE)	
	IGD	IS	IGD	IS	IGD	IS	IGD	IS
ZDT1	1	1	0	1	1	1	1	1
ZDT2	1	1	1	1	1	1	1	1
ZDT3	1	1	1	1	1	1	1	1
ZDT4	1	1	1	1	1	1	1	1
ZDT6	1	1	1	1	1	1	1	1
DTLZ1	1	1	1	1	1	1	1	1
DTLZ2	1	1	1	1	1	1	1	1
DTLZ3	1	1	1	1	1	1	1	1
DTLZ4	1	1	1	1	1	1	1	1
DTLZ5	1	1	1	1	1	1	1	1
DTLZ6	1	1	1	1	1	1	1	1
DTLZ7	1	1	1	1	1	1	1	1

obvious difference. DEHC provides the most uniformly distributed fronts among the four.

Figure 9 presents the fronts on DTLZ6 and these fronts correspond to the run with a median value with respect to IGD. DTLZ6's Pareto front is a line in three-dimensional space. Figure 9 demonstrates that NSGA-II and NNIA have some difficulties in converging to the optimal front and MOEA/D gets a point far from the real Pareto optimal front. Among all the 5 algorithms, DEHC seems to perform best on DTLZ6 because it converges to a curve exactly.

From the above comparisons and analyses, we can draw the conclusion that in terms of IGD and IS, DEHC generates approximation sets with competitive diversity, uniformity, and proximity to the Pareto front. Our proposed DEHC can deal with MOPs with different types of Pareto front, that is, continuous, and disjoint, having 2 and 3 objectives.

4.3. *Some Studies on Scalability of DEHC.* To study how the performance of DEHC varies as the number of decision

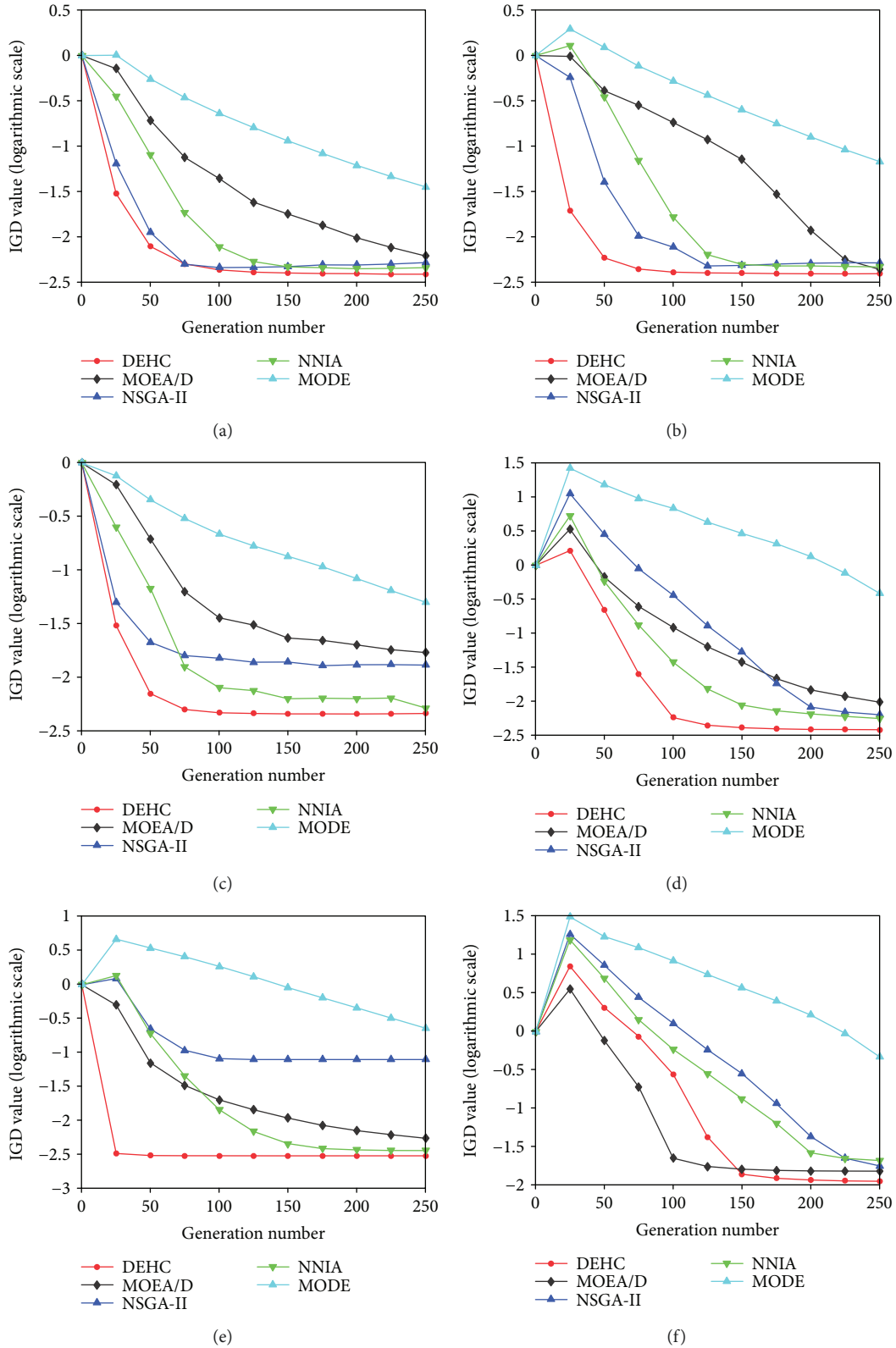


FIGURE 4: Median IGD values versus the evolution time. (a) ZDT1, (b) ZDT2, (c) ZDT3, (d) ZDT4, (e) ZDT6, and (f) DTLZ1.

variable increases, we have tried on ZDT1 with different numbers of decision variables, that is, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500,

2000, 2500, and 3000. In these experiments, MOEA/D, NSGA-II, NNIA, and MODE are also conducted with the above numbers of decision variable. All the parameters are

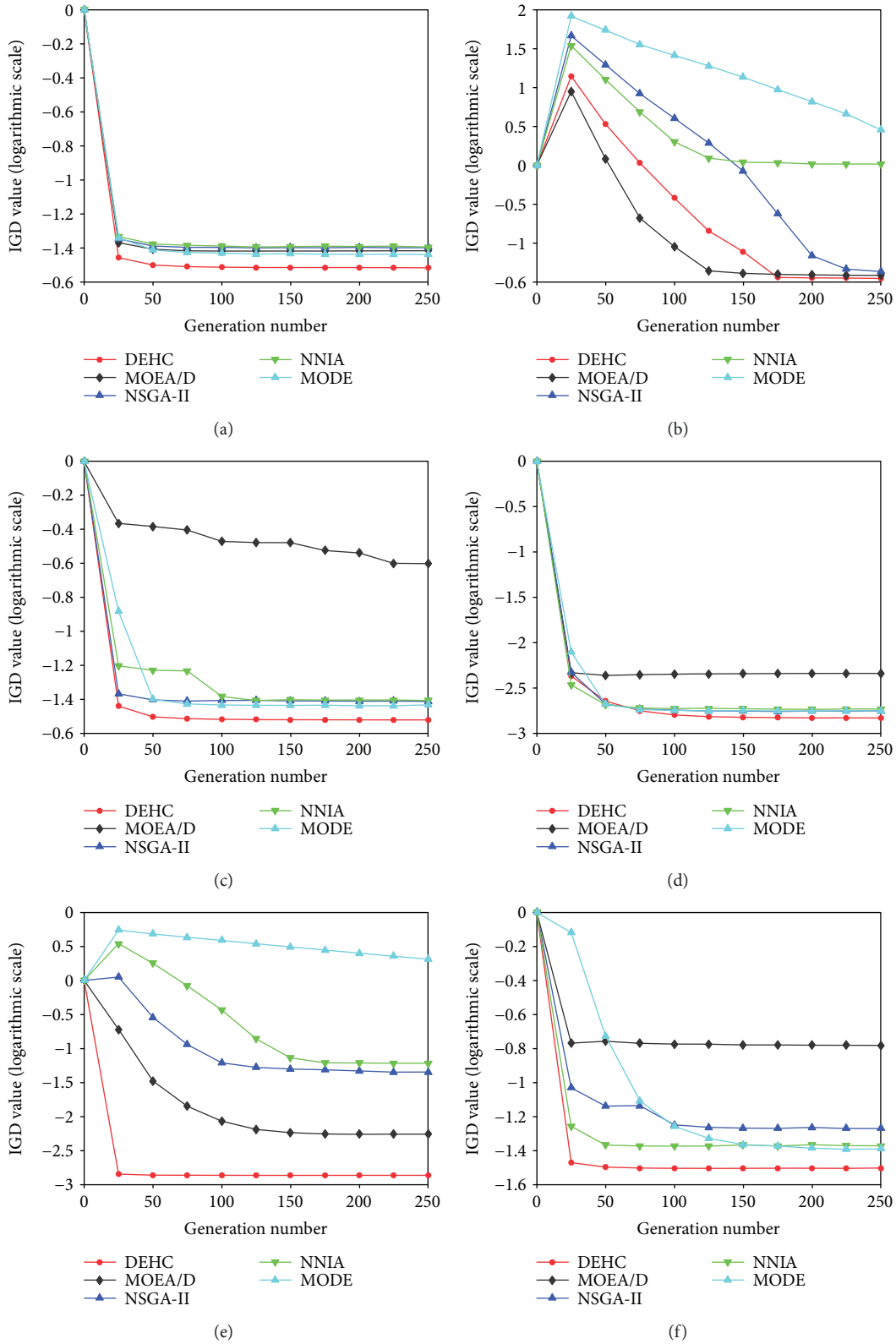


FIGURE 5: Median IGD values versus the evolution time. (a) DTLZ2, (b) DTLZ3, (c) DTLZ4, (d) DTLZ5, (e) DTLZ6, and (f) DTLZ7.

kept the same as in Section 4.2. Figure 10 gives the mean IGD values versus the numbers of decision variables. We found that among the 25 independent runs for each number of

decision variables, DEHC achieves lower IGD values on ZDT1 with different numbers of decision variable, especially from 60 to 1000. Furthermore, the mean IGD values almost

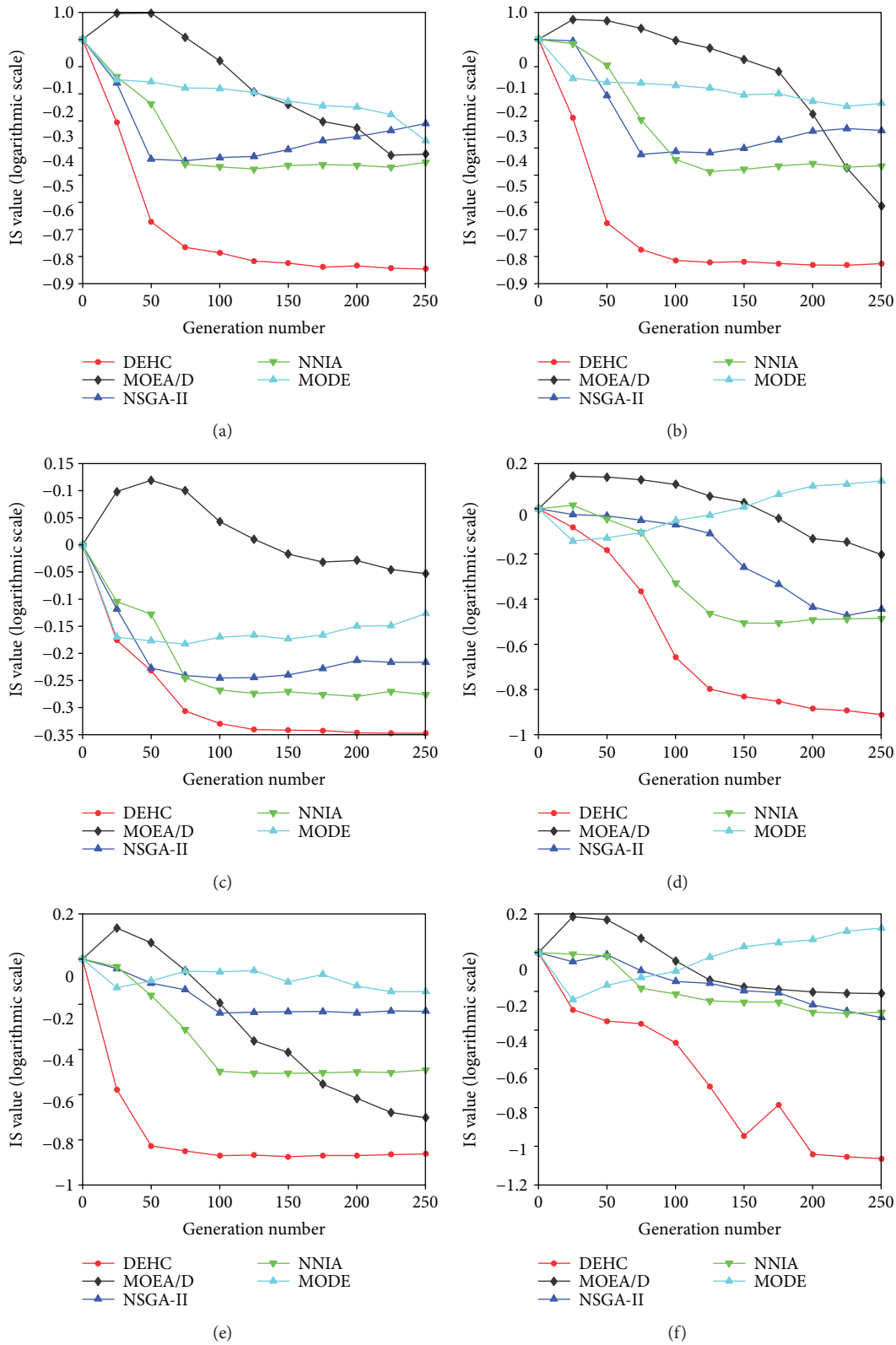


FIGURE 6: Median IS values versus the evolution time. (a) DTLZ2, (b) DTLZ3, (c) DTLZ4, (d) DTLZ5, (e) DTLZ6, and (f) DTLZ7.

linearly scale up as the number of decision variables increases. This experiment indicates that DEHC is able to handle large-scale MOPs efficiently and this ability may

profit from HSS because a different selection strategy is effective for different problems during different stages of the search.

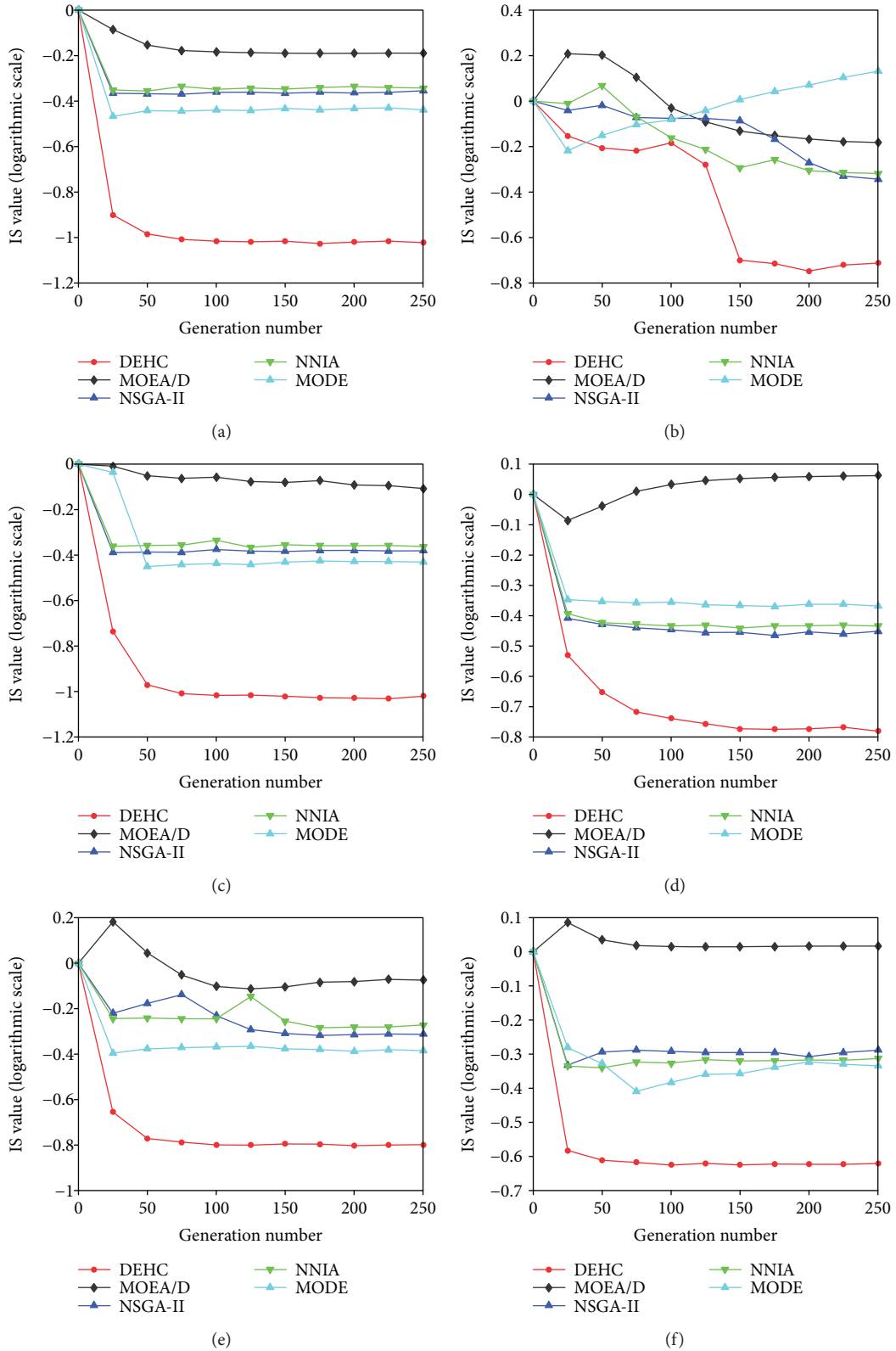


FIGURE 7: Median IS values versus the evolution time. (a) DTLZ2, (b) DTLZ3, (c) DTLZ4, (d) DTLZ5, (e) DTLZ6, and (f) DTLZ7.

4.4. *Discussion about CCE.* In order to demonstrate an improved performance of CCE, we replace CCE in DEHC with other four truncating methods and then compare them

with DEHC. These four methods are crowding distance [22], an improved version of crowding distance [44], crowding entropy [35], and harmonic distance [31]. We denote the

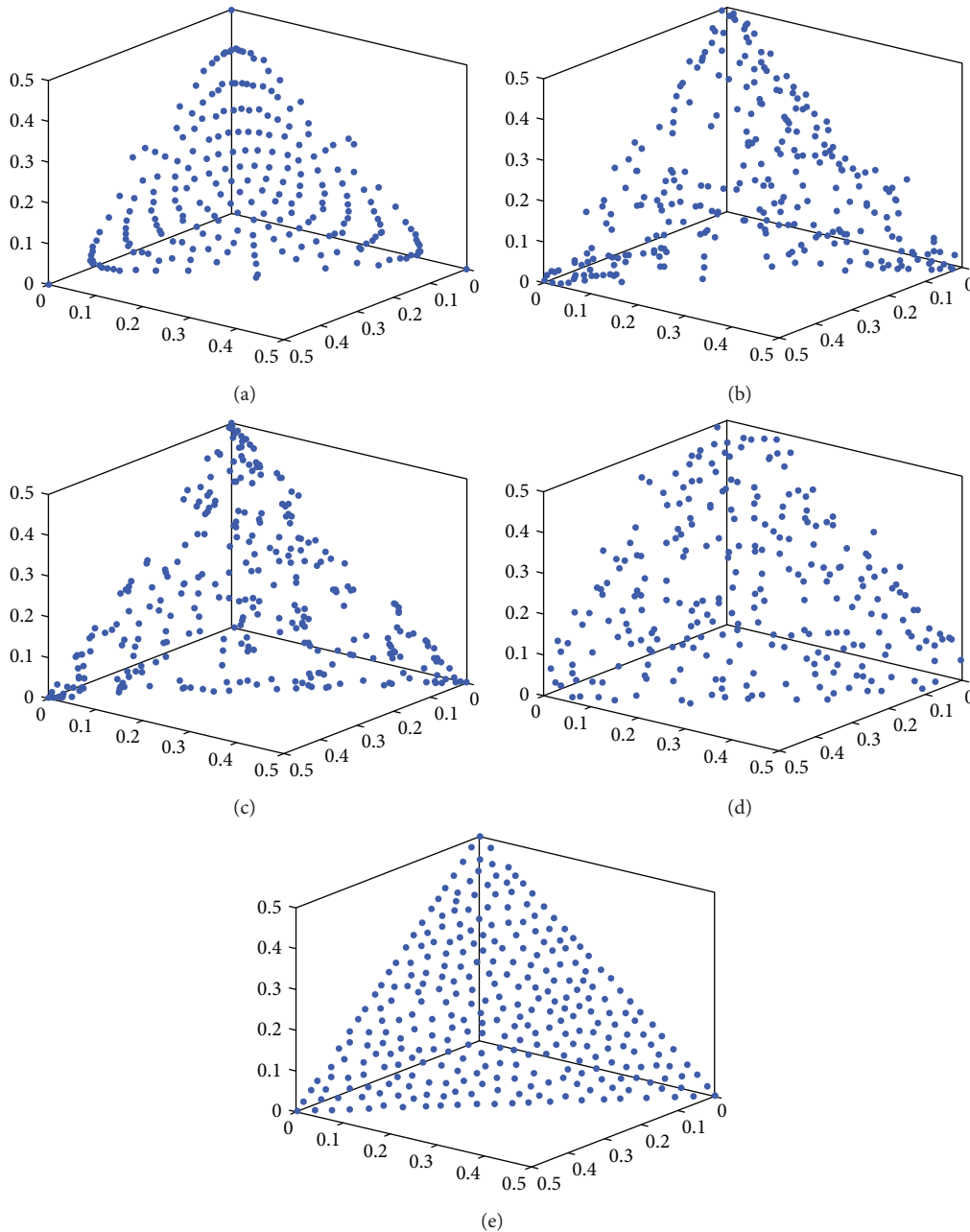


FIGURE 8: The typical Pareto front obtained by 5 algorithms on DTLZ1. (a) MOEA/D, (b) NSGA-II, (c) NNIA, (d) MODE, and (e) DEHC.

algorithms with the four methods as DEHCD, DEHCD2, DEHCE, and DEHHD, respectively. For all the five algorithms, 25 independent runs are conducted to see the consistency of the algorithms.

It can be observed from Table 4 that for the five 2-objective problems, MOECD2 performs very well because MOECD2 achieves the best values for all problems except ZDT4. As for DEHC, it follows DEHCD2 because it has the best IS on ZDT4 and ranks 2nd for the other four 2-objective problems. As for the seven 3-objective problems, DEHC generates the most uniformly distributed Pareto front among all the five algorithms. From this experiment, we can deduce that CCE is able to maintain a uniformly distributed result especially for many-objective problems.

To confirm that CCE is really suited for MOPs with many objectives, here, we expand DTLZ2 to 5, 7, and 9 objectives and then study IS. Figure 11 gives the mean IS on DTLZ2 with 3, 5, 7, and 9 objectives. It can be observed that for many objectives, CCE still works very well and it provides the most competitive results.

5. Applications to Solve MOOCs

In this section, DEHC is applied to solve three MOOCs in chemical processes taken from the literature. To apply DEHC for MOOCs, the MOOC is first transformed into common MOPs by the commonly used CVP approach. For the CVP method, the time range $[t_0, t_f]$ is divided into T equal

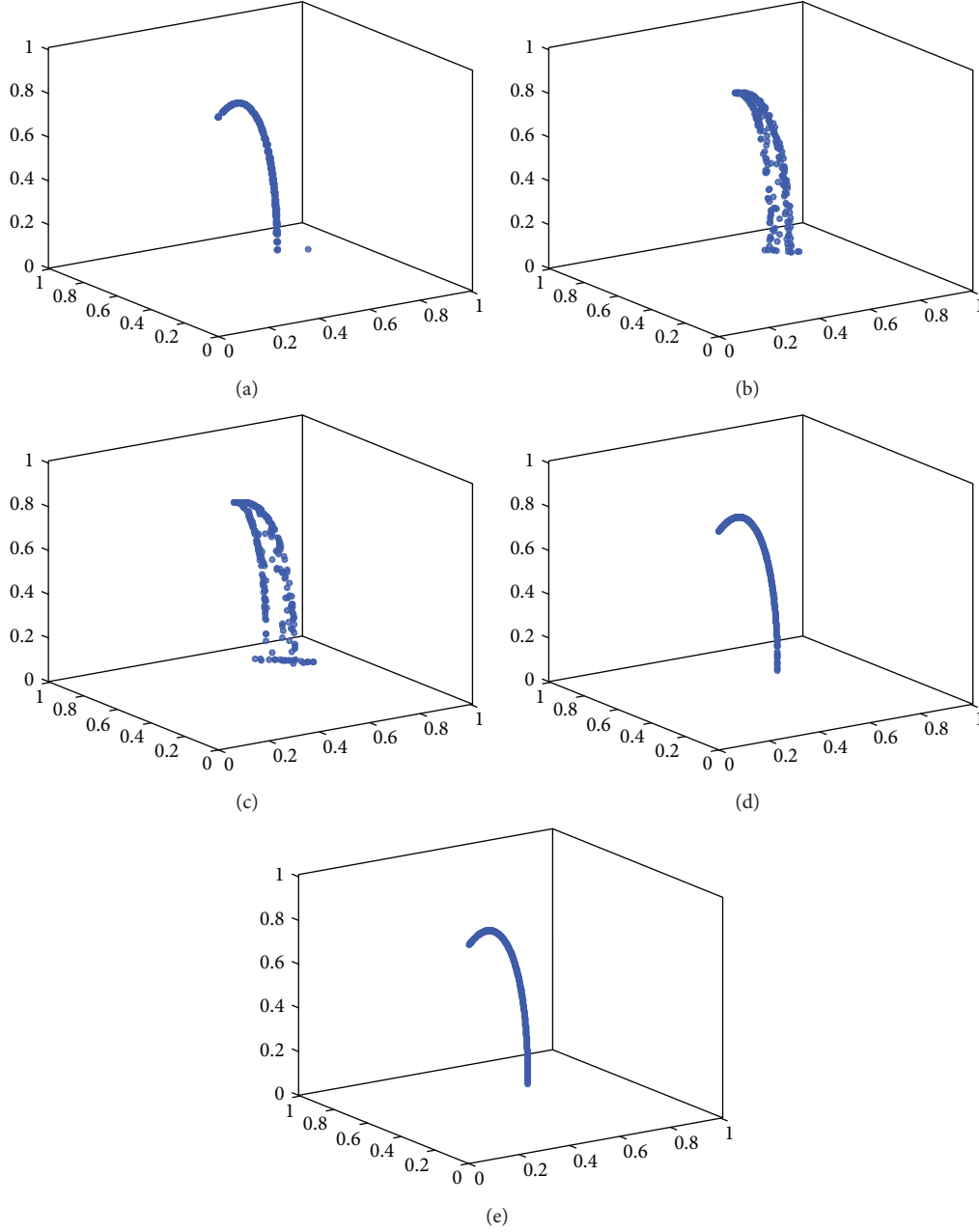


FIGURE 9: The typical Pareto front obtained by 5 algorithms on DTLZ6. (a) MOEA/D, (b) NSGA-II, (c) NNIA, (d) MODE, and (e) DEHC.

segments, that is, $[t_0, t_f] = [t_0, t_1] \times [t_1, t_2] \times \dots \times [t_{T-1}, t_T]$, $t_T = t_f$, and the control variables can be approximated by basic functions, such as piece-wise constant function and piece-wise linear functions [43, 45]. Here, we assume $u(t)$ as piecewise constant function. As a result, $T \times K$ parameters determine $u(t)$ over $[t_0, t_f]$. In terms of the j th segments; the control variables over the entire time span are as follows:

$$c_k^t = \begin{cases} u_k, & t \in [t_{j-1}, t_j], \\ 0, & \text{otherwise,} \\ j = 1, 2, \dots, T. \end{cases} \quad (14)$$

So, the total control variables over the entire time span can be described as the summation below:

$$u(t) = \sum_{j=1}^T c_j(t), \quad t \in [t_{j-1}, t_j]. \quad (15)$$

Once the control variables are discretized, DEHC can be applied to find the best control profile of the discrete time system as an approximation of the continuous problem. So, the major steps to optimize the MOOCP are outlined as the following steps:

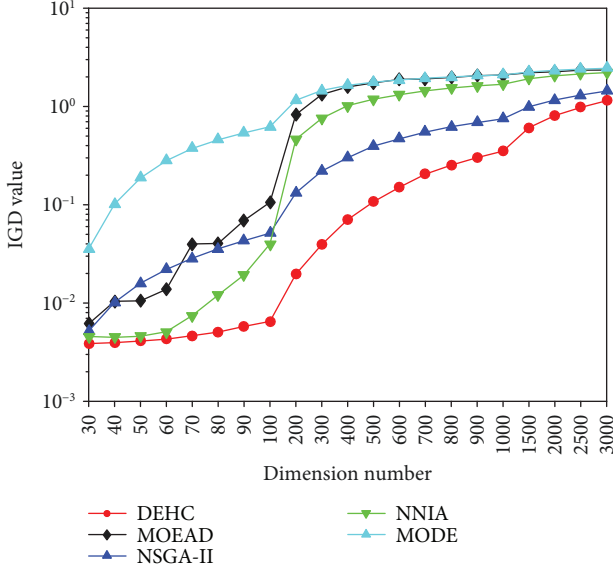


FIGURE 10: Mean IGD on ZDT1 as the number of decision variable increases.

Step 1. Divide the time range $[t_0, t_f]$ into T equal segments and denote the j th segment as $[t_{j-1}, t_j]$, where $j = 1, 2, \dots, K$ and $t_T = t_f$.

Step 2. Assume that the value of the i th input variable within the j th segment is u_i^j , where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, T$.

Step 3. Once the value of $u(t)$ is determined within the time range $[t_0, t_f]$, a numerical integral method, such as the standard Runge-Kutta method, can be employed to obtain the objective value, that is, $J(u(v))$.

Step 4. Apply DEHC to obtain the optimal u_i^j and $J(u(t))$ as an approximation of the continuous problem, where $i = 1, 2, \dots, K$ and $j = 1, 2, \dots, T$.

5.1. Case I: Catalyst Mixing Problem in a Tubular Reactor.

This problem considers a steady-state plug flow reactor of fixed length t_f . The reactor is packed with two catalysts which are required to stimulate a series of reactions (one reversible and one irreversible $S1 \leftrightarrow S2 \rightarrow S3$). These assumptions give rise to the following model [46]:

$$\begin{aligned} \frac{dx_1}{dt} &= u(10x_2 - x_1), \\ \frac{dx_2}{dt} &= u(x_1 - 10x_2) - (1 - u)x_2, \\ x(t_0) &= [1, 0]^T, \quad 0 \leq u \leq 1, t_f = 12, \end{aligned} \quad (16)$$

where x_1 and x_2 are the concentrations of S1 and S2, u is the fraction of catalyst A, and t is the spatial coordinate. The objective is to determine the optimal mixing policy of the two catalysts in order to maximize the production S3

at the reactor outlet and minimize the amount of the most expensive catalyst A.

$$\begin{aligned} \max J_1 &= 1 - x_1(t_f) - x_2(t_f), \\ \min J_2 &= \int_0^{t_f} u dt. \end{aligned} \quad (17)$$

For this problem, we set $T = 10$, the population size is $N = 50$, and the maximum generation is 100. The obtained Pareto optimal fronts by DEHC and MODE are illustrated in Figure 12. Figure 12 shows that both the two algorithms achieve a wide variety of solutions which also uniformly spread along the Pareto optimal front, but DEHC provides a more uniformly and widely distributed result along the front.

When considering maximizing the production of S3 only, Vassiliadis et al. obtained a best value of 0.0480238 by a single-objective optimization method [46] and Chen et al. obtained a best value of 0.04798 by a multiobjective DE with a ranking-based mutation operator [9]. The result of our obtained Pareto front is 0.04800, which is very close to 0.04798. To assist decision making, five Pareto optimal reaction temperatures, that is, u_1, u_2, u_3, u_4 , and u_5 in Figure 12, with equal distribution along the front are shown in Table 5 and the corresponding Pareto optimal trajectories of these are illustrated in Figure 13. It can be observed that the trajectories of mixing policy of the catalysts in the entire interval gradually decrease from 1 to 0, that is, from u_1 to u_5 , leading to the reduction of the use of catalyst A.

5.2. Case II: Optimal Operation of a Fed Batch Reactor. The third case is a batch reactor based on the two reaction systems [47]:



This process can be described by the following mechanistic model:

$$\begin{aligned} \frac{d[A]}{dt} &= -k_1[A][B] - \left(\frac{[A]}{V}\right)u, \\ \frac{d[B]}{dt} &= -k_1[A][B] - 2k_2[B]^2 + \left(\frac{(b_{\text{feed}} - [B])}{V}\right)u, \\ \frac{d[C]}{dt} &= k_1[A][B] - \left(\frac{[C]}{V}\right)u, \\ \frac{d[D]}{dt} &= 2k_2[B]^2 - \left(\frac{[D]}{V}\right)u, \\ \frac{d[V]}{dt} &= u, \\ 0 \leq u &\leq 0.01, t_f = 120 \text{ min}, \end{aligned} \quad (19)$$

where $[A]$, $[B]$, $[C]$, and $[D]$ are the concentrations of A, B, C, and D, respectively; V is the current reaction volume, u is the

TABLE 4: Comparison results of CCE and other truncating methods on IS.

	DEHCD Mean \pm Std	DEHCD2 Mean \pm Std	DEHCE Mean \pm Std	DEHHD Mean \pm Std	DEHC Mean \pm Std
ZDT1	$2.774E-1 \pm 2.383E-2$	$1.406E-1 \pm 1.199E-2$	$3.623E-1 \pm 3.416E-2$	$4.254E-1 \pm 3.093E-2$	$1.425E-1 \pm 1.390E-2$
ZDT2	$2.885E-1 \pm 2.474E-2$	$1.420E-1 \pm 1.063E-2$	$3.702E-1 \pm 3.298E-2$	$4.273E-1 \pm 3.389E-2$	$1.492E-1 \pm 1.134E-2$
ZDT3	$5.021E-1 \pm 1.324E-2$	$4.427E-1 \pm 1.064E-2$	$5.565E-1 \pm 2.896E-2$	$5.803E-1 \pm 3.071E-2$	$4.497E-1 \pm 1.341E-2$
ZDT4	$2.769E-1 \pm 2.869E-2$	$1.241E-1 \pm 1.099E-2$	$3.898E-1 \pm 3.178E-2$	$4.487E-1 \pm 3.238E-2$	$1.229E-1 \pm 1.175E-2$
ZDT6	$3.049E-1 \pm 2.767E-2$	$1.231E-1 \pm 1.030E-2$	$3.841E-1 \pm 3.080E-2$	$6.152E-1 \pm 2.915E-1$	$1.373E-1 \pm 1.275E-2$
DTLZ1	$3.860E-1 \pm 2.015E-2$	$3.885E-1 \pm 1.938E-2$	$3.986E-1 \pm 1.911E-2$	$2.381E-1 \pm 2.997E-2$	$8.648E-2 \pm 5.427E-3$
DTLZ2	$3.604E-1 \pm 9.635E-3$	$3.557E-1 \pm 1.431E-2$	$3.682E-1 \pm 1.879E-2$	$2.445E-1 \pm 8.816E-3$	$9.496E-2 \pm 3.217E-3$
DTLZ3	$3.707E-1 \pm 2.048E-2$	$3.718E-1 \pm 1.773E-2$	$3.732E-1 \pm 1.904E-2$	$1.065E+0 \pm 6.659E-1$	$1.939E-1 \pm 2.940E-2$
DTLZ4	$3.578E-1 \pm 1.537E-2$	$3.568E-1 \pm 1.741E-2$	$3.612E-1 \pm 1.920E-2$	$2.369E-1 \pm 1.005E-2$	$9.538E-2 \pm 3.852E-3$
DTLZ5	$3.505E-1 \pm 1.374E-2$	$1.962E-1 \pm 5.629E-3$	$3.929E-1 \pm 1.847E-2$	$3.996E-1 \pm 2.107E-2$	$1.657E-1 \pm 7.528E-3$
DTLZ6	$3.258E-1 \pm 1.225E-2$	$1.904E-1 \pm 8.794E-3$	$3.936E-1 \pm 1.432E-2$	$4.098E-1 \pm 1.460E-2$	$1.592E-1 \pm 7.995E-3$
DTLZ7	$4.400E-1 \pm 1.966E-2$	$4.440E-1 \pm 2.472E-2$	$4.468E-1 \pm 1.997E-2$	$2.575E-1 \pm 1.166E-2$	$2.395E-1 \pm 8.573E-3$

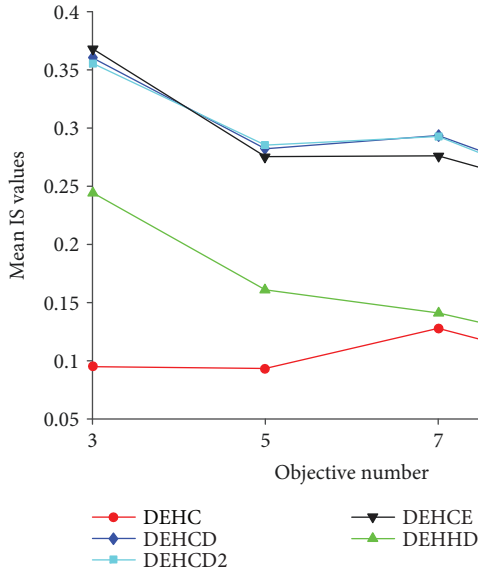


FIGURE 11: Mean IS on DTLZ2 as the number of objectives increases.

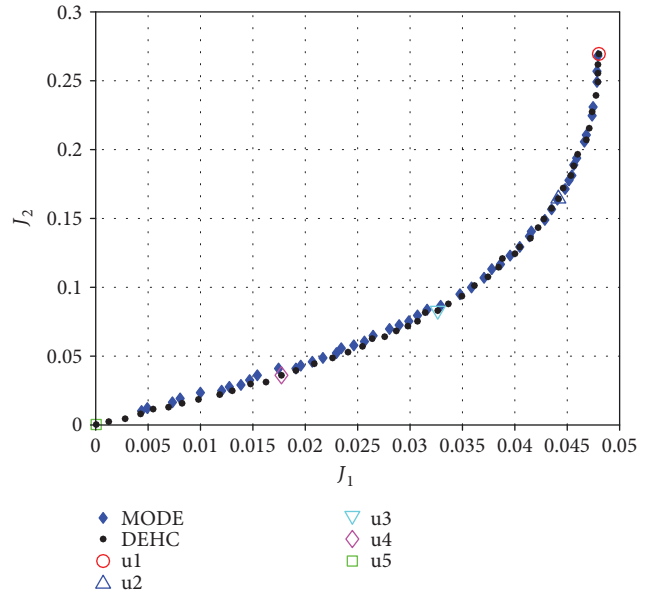


FIGURE 12: The Pareto optimal front for case I.

reactant feed rate, and $k_1 = 0.5$ and $k_2 = 0.5$ are the reaction rate constants. At the start of reaction, the reactor contains $[A](0) = 0.2$ mole/L of A ; no B ($[B](0) = 0$) and is fed to 50% ($V(0) = 0.5 \text{ m}^3$). The objective is to maximize the amount of product C and minimize the amount of by-product D .

For this problem, we set $T = 10$, the population size is $N = 50$, and the maximum generation is 100. The obtained Pareto fronts by DEHC and MODE are showed in Figure 14. It can be observed from Figure 14 that both DEHC and MODE achieve very high-quality results in approximating the true Pareto front. However, DEHC achieves a higher distribution behavior.

The obtained Pareto optimal front is illustrated in Figure 14. When the complete focus is put on maximizing the amount of product C , Chen et al. obtained a best value

TABLE 5: Five solutions selected from the front for case I.

Segment	u1	u2	u3	u4	u5
0.0–1.2	1.0000	1.0000	0.7555	0.2663	0.0000
1.2–2.4	0.5477	0.3310	0.0000	0.0392	0.0000
2.4–3.6	0.1534	0.1307	0.0447	0.0143	0.0000
3.6–4.8	0.2784	0.0953	0.0096	0.0191	0.0000
4.8–6.0	0.2354	0.0000	0.0229	0.0214	0.0000
6.0–7.2	0.2482	0.0840	0.0000	0.0000	0.0000
7.2–8.4	0.1683	0.0000	0.0000	0.0038	0.0000
8.4–9.6	0.0657	0.0043	0.0000	0.0000	0.0000
9.6–10.8	0.0011	0.0000	0.0003	0.0000	0.0000
10.8–12.0	0.0000	0.0000	0.0000	0.0000	0.0000

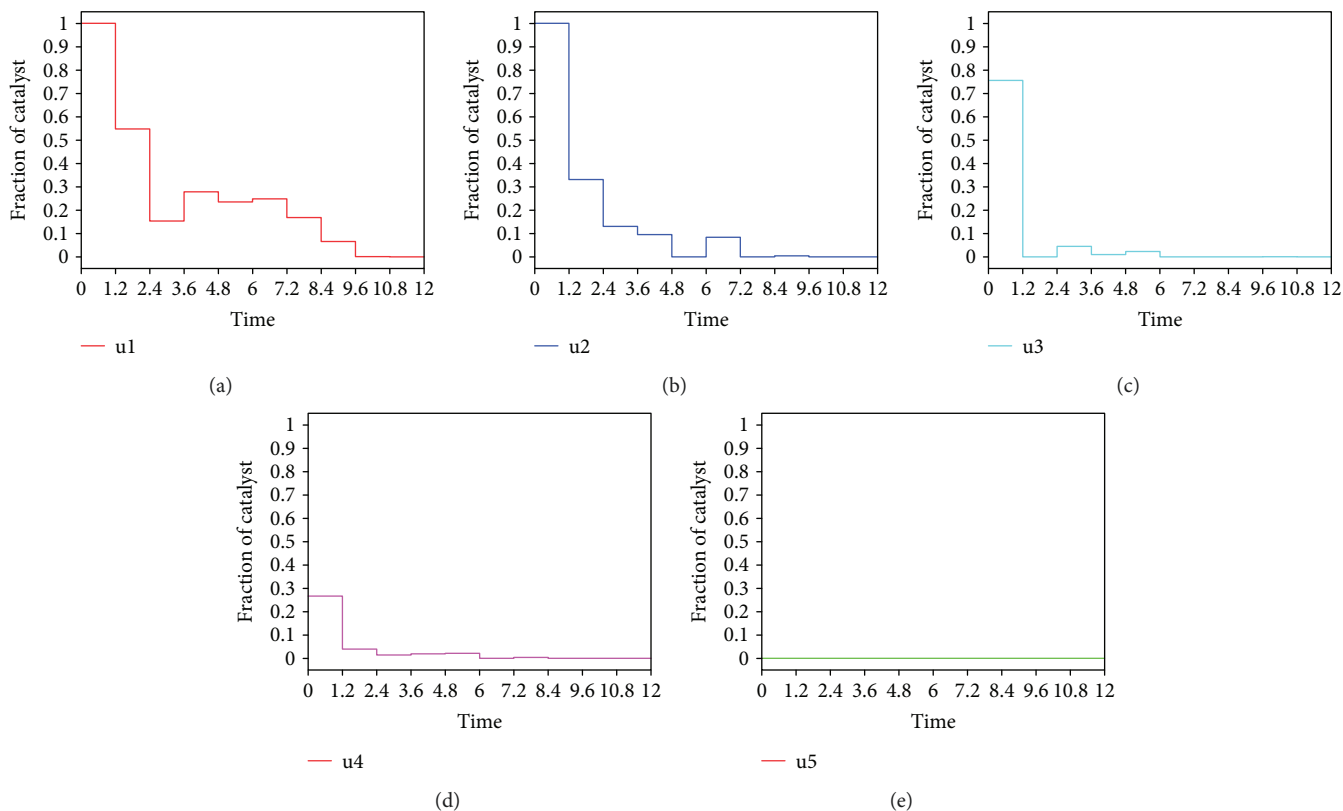


FIGURE 13: The Pareto optimal trajectories of five selected solutions for case I. (a) u1, (b) u2, (c) u3, (d) u4, and (e) u5.

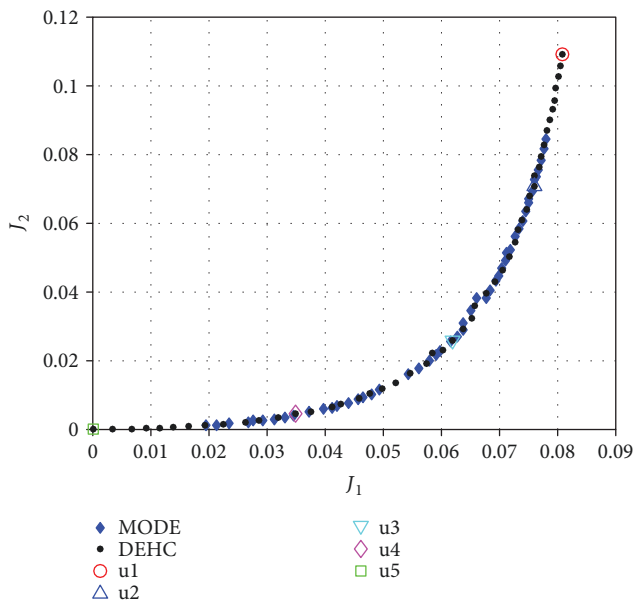


FIGURE 14: The Pareto optimal front for case II.

of 0.07936 and we obtained 0.08079 [9]. To assist decision making, five Pareto optimal reaction temperatures along the front in Figure 14 with equal distribution are shown in Table 6 and the corresponding Pareto optimal trajectories

TABLE 6: Five solutions selected from the front for case II.

Segment	u1	u2	u3	u4	u5
0.0–12.0	0.0100	0.0100	0.0096	0.0031	0.0000
12.0–24.0	0.0100	0.0100	0.0029	0.0017	0.0000
24.0–36.0	0.0100	0.0075	0.0044	0.0001	0.0000
36.0–48.0	0.0100	0.0073	0.0029	0.0027	0.0000
48.0–60.0	0.0100	0.0099	0.0045	0.0006	0.0000
60.0–72.0	0.0100	0.0029	0.0041	0.0062	0.0000
72.0–84.0	0.0100	0.0094	0.0031	0.0000	0.0000
84.0–96.0	0.0100	0.0070	0.0036	0.0020	0.0000
96.0–108.0	0.0100	0.0071	0.0072	0.0011	0.0000
108.0–120.0	0.0100	0.0002	0.0009	0.0000	0.0000

of these are illustrated in Figure 15. It can be observed from the multiple trajectories that as the reduction of the amount of by-product D , the feed rate in the entire time interval gradually reduces to the low value.

$$\begin{aligned} \max J_1 &= ([C](t_f)V(t_f)), \\ \min J_2 &= [D](t_f)V(t_f). \end{aligned} \tag{20}$$

5.3. Case III: Foreign Protein Production Using Recombinant Bacteria. The fourth problem is a model of foreign protein

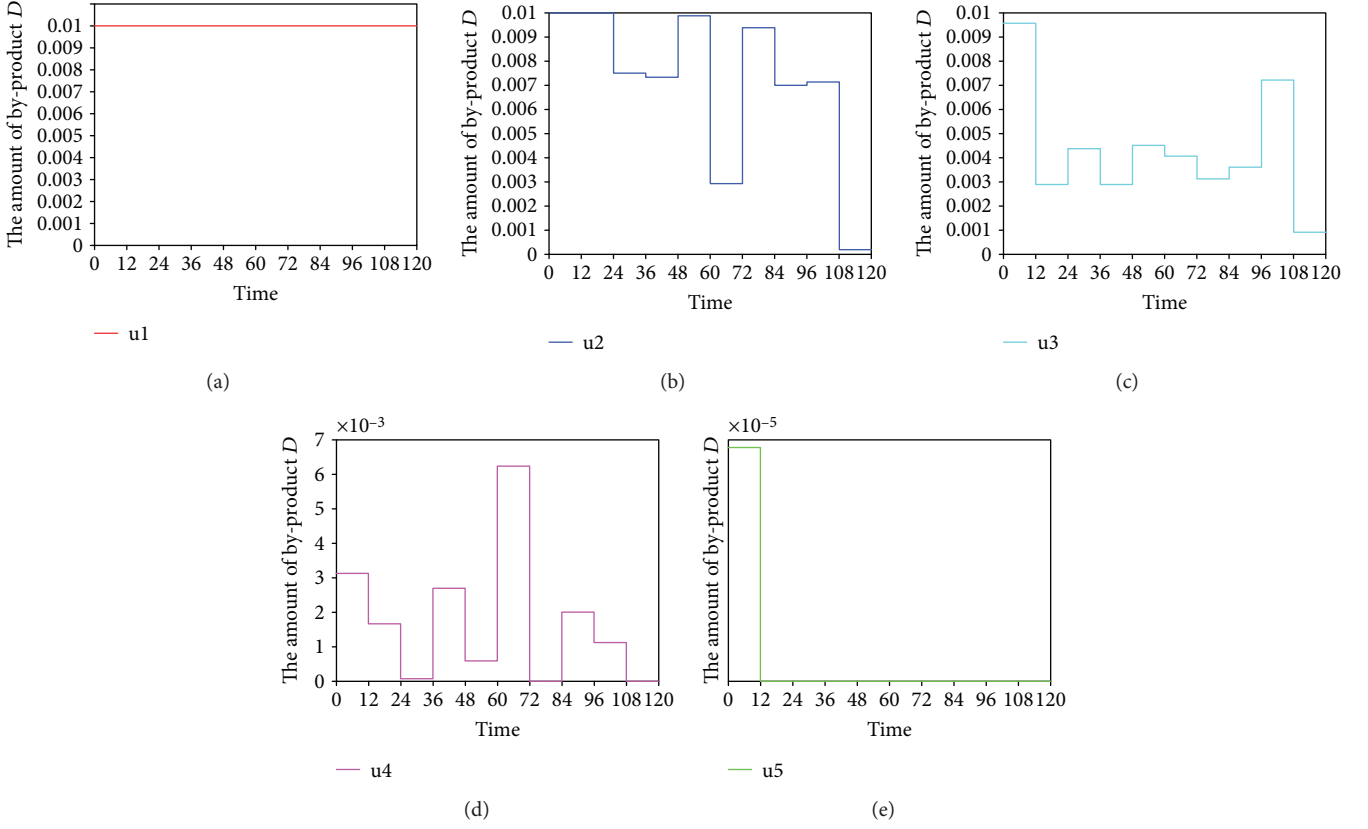


FIGURE 15: The Pareto optimal trajectories of five selected solutions for case II. (a) u_1 , (b) u_2 , (c) u_3 , (d) u_4 , and (e) u_5 .

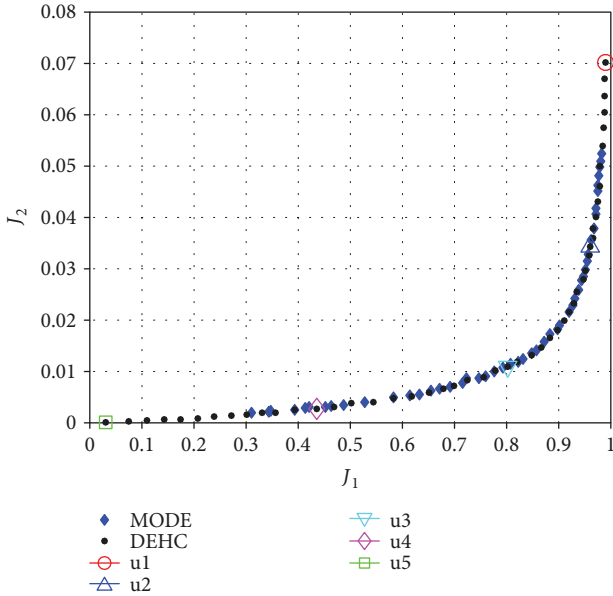


FIGURE 16: The Pareto optimal front for case III.

production using recombinant bacteria [48]. In this case, there are 7 state variables which are reaction volume (x_1 , L), cell density (x_2 , g/L), nutrient concentration (x_3 , g/L), foreign protein concentration (x_4 , g/L), inducer concentration (x_5 , g/L),

inducer shock factor on the cell growth rate (x_6), and the inducer recovery factor on the cell growth rate (x_7). The two control variables are glucose feed rate (u_1 , L/h) and inducer feed rate (u_2 , L/h). The model was described as follows [49]:

$$\begin{aligned}
 \frac{dx_1}{dt} &= u_1 + u_2, \\
 \frac{dx_2}{dt} &= ux_2 + \lambda x_2, \\
 \frac{dx_3}{dt} &= C_{nf} \frac{u_1}{x_1} + \lambda x_3 - \frac{\mu}{Y} x_2, \\
 \frac{dx_4}{dt} &= R_f x_2 + \lambda x_4, \\
 \frac{dx_5}{dt} &= C_{if} \frac{u_2}{x_1} + \lambda x_5, \\
 \frac{dx_6}{dt} &= -k_1 x_6, \\
 \frac{dx_7}{dt} &= k_2 (1 - x_7),
 \end{aligned} \tag{21}$$

$$x(t_0) = [1, 0.1, 40, 0, 0, 1, 0]^T,$$

$$0 \leq u_1 \leq 10^{-2}, 0 \leq u_2 \leq 10^{-2}, t_0 = 0, t_f = 10 \text{ h},$$

TABLE 7: Five solutions selected from the front for case III.

	u1		u2		u3		u4		u5	
	u1(1)	u1(2)	u2(1)	u2(2)	u3(1)	u3(2)	u4(1)	u4(2)	u5(1)	u5(2)
0.0-1.0	0.0000	0.0000	0.0000	0.0000	0.0072	0.0000	0.0079	0.0002	0.0040	0.0000
1.0-2.0	0.0001	0.0000	0.0053	0.0000	0.0003	0.0000	0.0002	0.0017	0.0057	0.0000
2.0-3.0	0.0039	0.0006	0.0050	0.0045	0.0002	0.0006	0.0100	0.0000	0.0098	0.0000
3.0-4.0	0.0000	0.0100	0.0011	0.0095	0.0000	0.0100	0.0000	0.0008	0.0000	0.0000
4.0-5.0	0.0012	0.0100	0.0001	0.0100	0.0063	0.0000	0.0025	0.0000	0.0094	0.0000
5.0-6.0	0.0000	0.0100	0.0020	0.0100	0.0021	0.0000	0.0042	0.0000	0.0044	0.0000
6.0-7.0	0.0004	0.0100	0.0000	0.0000	0.0045	0.0002	0.0068	0.0000	0.0055	0.0000
7.0-8.0	0.0002	0.0100	0.0061	0.0001	0.0002	0.0000	0.0060	0.0001	0.0095	0.0000
8.0-9.0	0.0003	0.0100	0.0058	0.0000	0.0002	0.0000	0.0056	0.0000	0.0094	0.0000
9.0-10.0	0.0013	0.0096	0.0055	0.0000	0.0055	0.0000	0.0096	0.0000	0.0064	0.0000

where

$$\begin{aligned}
\mu &= \frac{0.407x_3}{(0.108 + x_3 + x_3^2)/(14814.8)} \times \left(x_6 + \frac{0.22x_7}{0.22 + x_5} \right), \\
R_{fp} &= \frac{0.095x_3}{(0.108 + x_3 + x_3^2)/14814.8} \times \left(\frac{0.0005 + x_5}{0.022 + x_5} \right), \\
\lambda &= -\frac{u_1 + u_2}{x_1}, \\
k_1 = k_2 &= \frac{0.09x_5}{0.034 + x_5}, \\
C_{nf} &= 100.0, \\
C_{if} &= 4.0, \\
Y &= 0.51.
\end{aligned} \tag{22}$$

As the price of the inducer, that is, P_r , is expensive, here, the two objectives are set as maximizing foreign protein production and minimizing the consumption of the inducer by controlling the glucose feed rate and the inducer feed rate, which are modeled as follows:

$$\begin{aligned}
\max J_1 &= x_1(t_f)x_4(t_f), \\
\min J_2 &= \int_0^{t_f} u_2 dt.
\end{aligned} \tag{23}$$

For this problem, we set $T = 10$, the population size is $N = 50$, and the maximum generation is 100. The obtained Pareto fronts by DEHC and MODE are showed in Figure 16. It can be observed from Figure 16 that both DEHC and MODE generate very close approximations to the Pareto front. But the attained Pareto front with MODE cannot spread along the front as much as DEHC.

To assist decision making, five Pareto optimal reaction temperatures along the front in Figure 16 with equal

distribution are shown in Table 7 and the corresponding Pareto optimal trajectories of these are illustrated in Figure 17.

$$\max J = x_1(t_f)x_4(t_f) - P_r \int_{t_0}^{t_f} u_2(t) dt. \tag{24}$$

Roubos et al. set $P_r = 5$ and solve (24) using GA with a population size of 40 and generation of 25000 and obtained an optimum of 0.8149 [50]. We calculate the combined objective values with $P_r = 5$ and obtain an optimal value of 0.81345. Figure 18 also shows the combined objective values along the Pareto front by setting $P_r = 5$.

6. Conclusions

In this paper, we put forward an extended multiobjective differential evolution algorithm called DEHC by integrating multistrategies from three aspects, that is, hybrid selection strategy, external archive with cyclic crowding estimation, and a two-strategy based on mutation component. We have systematically studied the performance of DEHC by testing it on a set of MOPs. The numerical results show that DEHC is suitable for solving MOPs with different types of Pareto front and that it is superior to or competitive with MOEA/D, NSGA-II, and NNIA. Our proposed DEHC is able to generate very competitive results in terms of IGD and IS indicators. Finally, three well-known multiobjective optimal control problems in chemical process have been solved to further demonstrate the effectiveness and applicability of DEHC for real-world complex MOOCs.

The numerical results validate that DEHC is an advantageous approach to solving complex MOPs. This advantage comes from three aspects: (1) the hybrid selection strategy can deal with the conflicting objectives from different aspects which is advantageous to different problems during different search processes, (2) the cyclic crowding estimation is helpful to maintain a uniformly distributed nondominated solutions, and (3) the candidate pool of mutation strategies for DE mutation component is beneficial for improving DEHC's exploration and exploitation.

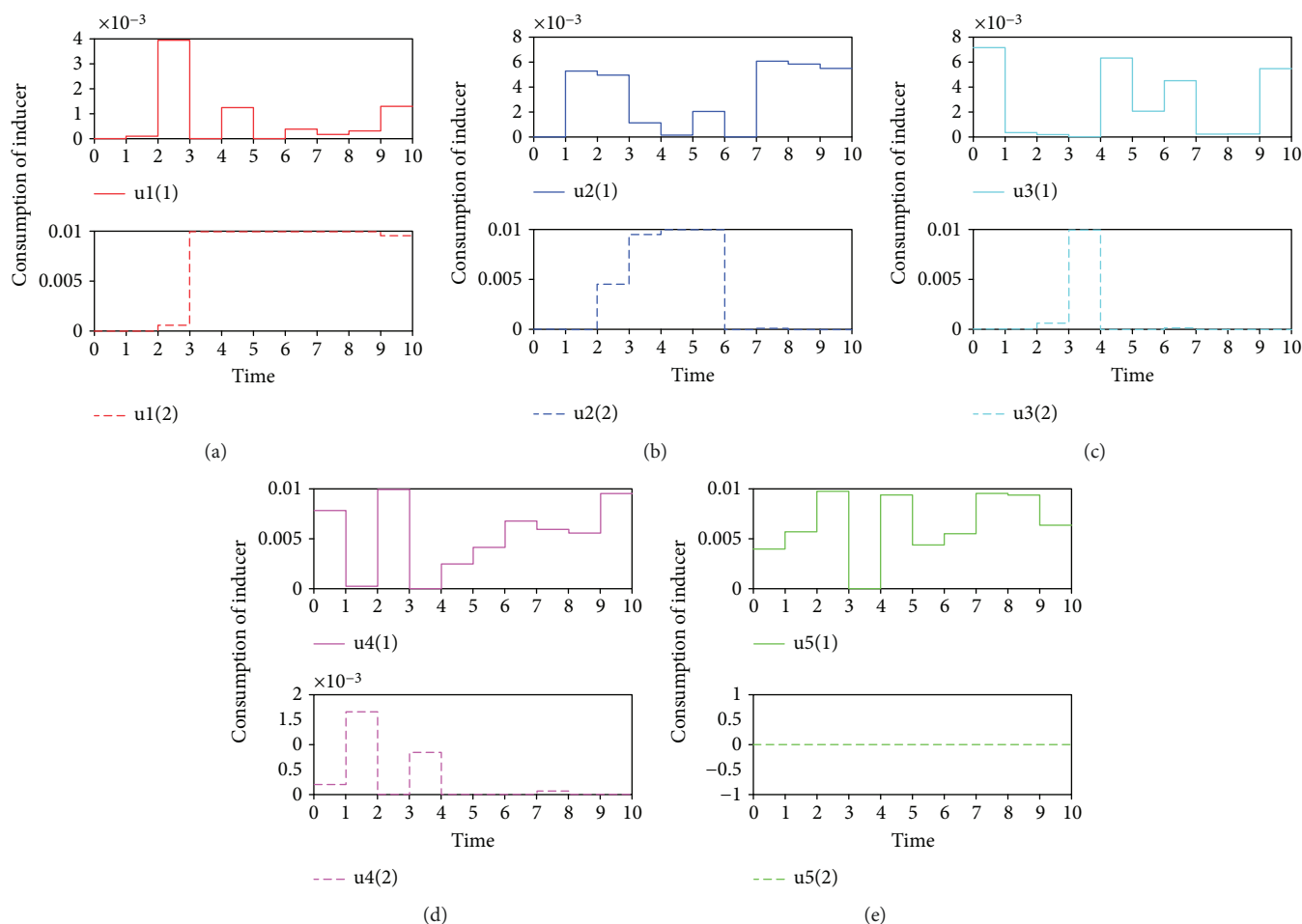


FIGURE 17: The Pareto optimal trajectories of five selected solutions for case III. (a) u_1 , (b) u_2 , (c) u_3 , (d) u_4 , and (e) u_5 .

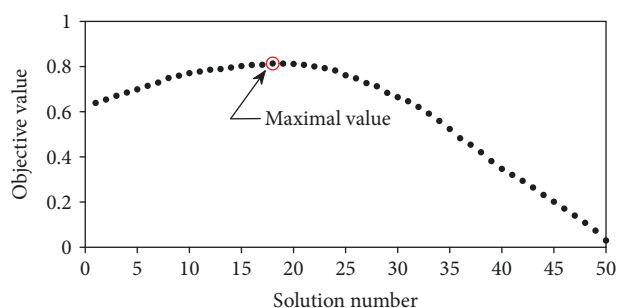


FIGURE 18: The objective value of the Pareto optimal front for case III.

What remain interesting for future work would be constrained MOOCs and MOOCs with uncertainties. Another aspect is developing effective CVP methods to convert the original MOOCs.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61703268.

References

- [1] F. Logist, M. Vallerio, B. Houska, M. Diehl, and J. van Impe, "Multi-objective optimal control of chemical processes using acado toolkit," *Computers & Chemical Engineering*, vol. 37, no. 10, pp. 191–199, 2012.
- [2] B. Xu, R. Qi, W. Zhong, W. Du, and F. Qian, "Optimization of p-xylene oxidation reaction process based on self-adaptive multi-objective differential evolution," *Chemometrics and Intelligent Laboratory Systems*, vol. 127, no. 12, pp. 55–62, 2013.
- [3] F.-S. Wang and T.-L. Shieh, "Extension of iterative dynamic programming to multiobjective optimal control problems," *Industrial & Engineering Chemistry Research*, vol. 36, no. 6, pp. 2279–2286, 1997.

- [4] F. Logist, P. M. M. van Erdeghem, and J. F. van Impe, "Efficient deterministic multiple objective optimal control of (bio)chemical processes," *Chemical Engineering Science*, vol. 64, no. 11, pp. 2527–2538, 2009.
- [5] C. Y. Kaya and H. Maurer, "A numerical method for nonconvex multi-objective optimal control problems," *Computational Optimization and Applications*, vol. 57, no. 3, pp. 685–702, 2014.
- [6] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [7] V. Ho-Huu, S. Hartjes, H. G. Visser, and R. Curran, "An improved MOEA/D algorithm for bi-objective optimization problems with complex pareto fronts and its application to structural optimization," *Expert Systems with Applications*, vol. 92, pp. 430–446, 2018.
- [8] D. Sarkar and J. M. Modak, "Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm," *Chemical Engineering Science*, vol. 60, no. 2, pp. 481–492, 2005.
- [9] X. Chen, W. Du, and F. Qian, "Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization," *Chemometrics and Intelligent Laboratory Systems*, vol. 136, no. 15, pp. 85–96, 2014.
- [10] Q. Fan, W. Wang, and X. Yan, "Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems," *Applied Soft Computing*, vol. 59, pp. 33–44, 2017.
- [11] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [12] M. Eskandari Nasab, I. Maleksaeedi, M. Mohammadi, and N. Ghadimi, "A new multiobjective allocator of capacitor banks and distributed generations using a new investigated differential evolution," *Complexity*, vol. 19, no. 5, 54 pages, 2014.
- [13] Q. Fan, X. Yan, and Y. Xue, "Prior knowledge guided differential evolution," *Soft Computing*, vol. 21, no. 22, pp. 6841–6858, 2017.
- [14] B. Xu, X. Chen, and L. Tao, "Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization," *Information Sciences*, vol. 435, pp. 240–262, 2018.
- [15] K. Deb, "A population-based algorithm-generator for real-parameter optimization," *Soft Computing*, vol. 9, no. 4, pp. 236–253, 2005.
- [16] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pp. 93–100, Pittsburgh, PA, USA, 1985.
- [17] K. Parsopoulos, D. Tasoulis, N. Pavlidis, V. Plagianakos, and M. Vrahatis, "Vector evaluated differential evolution for multiobjective optimization," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, pp. 204–211, Portland, OR, USA, 2004.
- [18] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [19] B. Y. Qu and P. N. Suganthan, "Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection," *Information Sciences*, vol. 180, no. 17, pp. 3170–3181, 2010.
- [20] J. Xiao, J. J. Li, X. X. Hong et al., "An improved MOEA/D based on reference distance for software project portfolio optimization," *Complexity*, vol. 2018, Article ID 3051854, 16 pages, 2018.
- [21] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [23] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Genetic and Evolutionary Computation — GECCO 2003*, vol. 2723 of Lecture Notes in Computer Science, pp. 37–48, Springer, Berlin, Heidelberg, 2003.
- [24] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *AI 2004: Advances in Artificial Intelligence*, vol. 3339 of Lecture Notes in Computer Science, pp. 861–872, Springer, Berlin, Heidelberg, 2004.
- [25] A. W. Iorio and X. Li, "A cooperative coevolutionary multiobjective algorithm using non-dominated sorting," in *Genetic and Evolutionary Computation - GECCO 2004*, vol. 3102 of Lecture Notes in Computer Science, pp. 537–548, Springer, Berlin, Heidelberg, 2004.
- [26] R. Angira and B. Babu, "Non-dominated sorting differential evolution (NSDE): An extension of differential evolution for multi-objective optimization," in *2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, pp. 1428–1443, Pune, India, 2005.
- [27] B.-y. Qu and P. N. Suganthan, "Multi-objective differential evolution with diversity enhancement," *Journal of Zhejiang University SCIENCE C*, vol. 11, no. 7, pp. 538–543, 2010.
- [28] H. Ghiasi, D. Pasini, and L. Lessard, "A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems," *Engineering Optimization*, vol. 43, no. 1, pp. 39–59, 2011.
- [29] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α -constrained-domination principle for constrained multi-objective optimization," *Soft Computing*, vol. 16, no. 8, pp. 1353–1372, 2012.
- [30] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [31] V. Huang, P. Suganthan, A. Qin, and S. Baskar, "Multiobjective differential evolution with external archive and harmonic distance-based diversity measure," Tech. Rep., Nanyang Technological University, Singapore, 2005.
- [32] J. Zheng, G. Xiao, W. Song, X. Li, and C. X. Ling, "A multi-objective genetic algorithm based on density," in *Advances in Computation and Intelligence*, pp. 12–25, Springer, Berlin, Heidelberg, 2007.
- [33] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.

- [34] W. Gong, Z. Cai, and L. Zhu, "An efficient multiobjective differential evolution algorithm for engineering design," *Structural and Multidisciplinary Optimization*, vol. 38, no. 2, pp. 137–157, 2009.
- [35] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Computing*, vol. 14, no. 3, pp. 193–209, 2010.
- [36] S. Tiwari, G. Fadel, and K. Deb, "AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization," *Engineering Optimization*, vol. 43, no. 4, pp. 377–401, 2011.
- [37] K. Nag, T. Pal, and N. R. Pal, "ASMiGA: an archive-based steady-state micro genetic algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 40–52, 2015.
- [38] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [39] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [40] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, pp. 105–145, Springer, London, UK, 2005.
- [41] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [42] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *2006 IEEE International Conference on Evolutionary Computation*, pp. 892–899, Vancouver, BC, Canada, 2006.
- [43] X. Chen, W. Du, H. Tianfield, R. Qi, W. He, and F. Qian, "Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1289–1299, 2014.
- [44] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *2006 IEEE International Conference on Evolutionary Computation*, pp. 1179–1186, Vancouver, BC, Canada, 2006.
- [45] J. Zhu, X. Yan, and W. Zhao, "Chemical process dynamic optimization based on the differential evolution algorithm with an adaptive scheduling mutation strategy," *Engineering Optimization*, vol. 45, no. 10, pp. 1205–1221, 2013.
- [46] V. S. Vassiliadis, E. B. Canto, and J. R. Banga, "Second-order sensitivities of general dynamic systems with application to optimal control problems," *Chemical Engineering Science*, vol. 54, no. 17, pp. 3851–3860, 1999.
- [47] P. Terwiesch, D. Ravemark, B. Schenker, and D. W. T. Rippin, "Semi-batch process optimization under uncertainty: theory and experiments," *Computers & Chemical Engineering*, vol. 22, no. 1-2, pp. 201–213, 1998.
- [48] J. Lee and W. F. Ramirez, "Optimal fed-batch control of induced foreign protein production by recombinant bacteria," *AIChE Journal*, vol. 40, no. 5, pp. 899–907, 1994.
- [49] F. Qian, F. Sun, W. Zhong, and N. Luo, "Dynamic optimization of chemical engineering problems using a control vector parameterization method with an iterative genetic algorithm," *Engineering Optimization*, vol. 45, no. 9, pp. 1129–1146, 2013.
- [50] J. A. Roubos, G. van Straten, and A. J. B. van Boxtel, "An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance," *Journal of Biotechnology*, vol. 67, no. 2-3, pp. 173–187, 1999.

