

# $L_\infty$ Metric based Multi-Objective Differential Evolution Algorithm and its Industrial Application\*

ZHAN GUO<sup>1</sup>, OKAN K. ERSOY<sup>2</sup> AND XUEFENG YAN<sup>1,3,+</sup>

<sup>1</sup>Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education  
East China University of Science and Technology  
Shanghai, 200237 P.R. China

<sup>2</sup>School of Electrical and Computer Engineering  
Purdue University  
West Lafayette, IN 47907, USA

<sup>3</sup>Shanghai Institute of Intelligent Science and Technology  
Tongji University  
Shanghai, 200237 P.R. China  
E-mail: xfyan@ecust.edu.cn

In multi-objective optimization problems, the objective space of fitness functions has a close relationship with the solution space. Extracting the optimal direction and optimal parameter information are very useful for the optimization process. This paper proposes multi-objective differential evolution algorithm with a clustering based objective space division and parameter adaptation (MODECD).  $L_\infty$  metric matrix based optimal strategy is used to split the objective space into sub-spaces and to extract the optimal directions. A fitness value based parameter adaptation and mutation strategy are used to extract the optimal strategy information. The results with 20 benchmark tests show the competitiveness of the MODECD algorithm in both convergence speed and diversity of solution approximating the Pareto front. In addition, MODECD is used to optimize the fermentation process of sodium gluconate as an example of its superior performance in solving real-world problems.

**Keywords:** differential evolution algorithm,  $L_\infty$  metric, parameter adaptation, multi-objective optimization, sodium gluconate production

## 1. INTRODUCTION

A multi-objective problem (MOP) always contains two or more conflicting problems. They conflict with each other as well as interact with each other. Many engineering and scientific problems are MOPs. MOP is different from the single objective problem. In MOP, it's hard to select the superior individuals whereas the superior individuals can be easily picked up in the single objective problem. The solutions in a multi-objective problem are not single but constitute a solution set approximating the Pareto front.

The fitness values in the objective space show the quality of solutions. Each individual contains the evolution information. Some are good for evolution, and some are not. In minimization problems, the ideal solution is as small as possible. In the evolution process, the individual closest to the true Pareto front contains much more useful information, and this can guide the whole population to move towards the true Pareto front. Also, different

---

Received May 27, 2020; revised August 22, 2020; accepted October 10, 2020.

Communicated by Chia-Feng Juang.

+ Corresponding author.

\* This work was supported by National Key research and development program of China (2020YFA0908303), and National Natural Science Foundation of China (21878081).

fitness values between parent and offspring can be used to improve the quality of the parameters [1]. Finding a proper way to extract such useful information is very important. In this article, an adaptive multi-objective differential evolution algorithm (MODECD) with an objective space partition is proposed. A matrix based on  $L_\infty$  metric is adopted to implement the clustering and to divide the objective space into sub-spaces. This leads to extracting the optimal direction in each sub-space. A Gaussian mixture model based on the differences between parent and offspring fitness function values is built. This model contains the optimal parameter information in the current generation and provides a proper parameter set adaptation for the next generation. Meanwhile, a mutation adaptation strategy is developed to improve the diversity. In an evolution algorithm, the parent and offspring populations are often combined together in each generation. Here non-dominated sorting is adopted to layer the solutions based on this relationship.

The three major contributions are summarized as follows:

- 1) A  $L_\infty$  metric matrix based objective space partition method is designed. The whole objective space is split into sub-spaces, and a selection strategy has proposed to extract the optimal direction which can guide the whole evolution.
- 2) A parameter adaptation and mutation strategy based on the variety of fitness values is proposed. The information on optimal parameter and mutation strategy is extracted to build a model and to generate the next generation's parameter and mutation strategy.
- 3) The proposed MODECD is examined with 20 benchmark test functions and tested in the optimization of the fermentation industry process. Two design evaluation indices conversion rate and utilization rate of equipment have been proposed as a result to show the effectiveness of MODECD.

The succeeding sections of this paper are as follows: Section 2 describes the details of the proposed algorithm. The experimental results based on 20 benchmark functions are discussed in Section 3. Section 4 presents the application of MODECD to the fermentation process of sodium gluconate. Conclusions are given in Section 5.

## 2. RELATED WORK

The multi-objective evolutionary algorithms (MOEAs) have characteristics of robustness and powerful self-learning ability that make them effective in dealing with MOP. For instance, elitist non-dominated sorting genetic algorithm (NSGA-II) [2], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [3], multi-objective genetic algorithm (MOGA) [4], grid-based evolution algorithm (GrEA) [5], and strength Pareto evolutionary algorithm (SPEA) [6] are such algorithms. They all solve the MOP, whereas the performance of traditional evolution algorithms is not competitive with them to obtain an ideal Pareto front. To improve the performance further, new evolutionary strategies and novel parameter strategies continue to be proposed. For example, applying statistics together with the evolutionary algorithm, an algorithm called multi-objective estimation of the distribution algorithm (RM-MEDA) [7] was proposed. To improve the diversity, NSGA-III [8] used a predefined reference point set to solve many objective problems. To maximally extract the useful information in decision space, a decision space partition algorithm DSPEA [9] was developed to split the decision space into several hyperspheres

and the convergence was improved. Analyzing the influence of decision variables on diversity and convergence, MOEA/DVA was proposed to decompose the large-scale variables into low-dimensional subcomponents [10]. To implement the evolution direction dynamically adjusting, DE-RLFR was designed with a reward function which utilized the population distribution in fitness-ranking space, objective space and decision space [11]. LIBEAs [12] is one of the indicator-based MOEAs, in which a hypervolume indicator was used to reduce the computational cost. The decomposition method is an efficient method; DAA [13] applied the decomposition method to external archive, and the performance of the algorithm was improved. TriMOEA-TA&R [14] is a novel algorithm analyzing the relationship between decision variable using two archives, one for diversity, and one for convergence. In order to solve the shortcoming of harmony search algorithm, as easy to trap in local optimum and poor convergence, Dai *et al.* [15] have proposed a novel harmony search algorithm with Gaussian mutation. Two bandwidths in pitch adjustment are designed to obtain better exploration and exploitation. CSO [16] is a novel evolution algorithm which has been extended to solve multi-objective optimization problems. An integration of the archive population has used to guide the whole population towards Pareto solutions. An aggregation function has used to judge the social hierarchy. Based on the problem that some of the proposed novel MOEAs are sensitive to the Pareto front shape. Chen *et al.* [17] proposed an evolution algorithm based on diversity ranking method. Reference vector adaptation method are used to deal with the different shape of Pareto front. The performance of DE is sensitive to its mutation strategies and control parameters, a self-adaptive mutation differential evolution algorithm based on particle swarm optimization has proposed [18]. The particle swarm optimization algorithm adopted a DE mutation strategy with stronger global exploration ability and a PSO mutation strategy with higher convergence ability. MODE-FM [18] is a novel multi-objective differential evolution with fuzzy inference-based dynamic adaptive mutation factor. The fuzzy inference is dynamically tuning the mutation factor for a better exploration and exploitation. Hendra G. Harno and Ian R. Petersen proposed a differential evolution algorithm [19] based on construct an optimal linear coherent quantum controller. This method can provide a straightforward approach to deal with nonlinear and nonconvex constrains. HMODE [20] is a hybrid multi-objective DE algorithm. A controller which combine linear matrix inequality approaches with MOEA is proposed. Parvesh Kumar *et al.* [21] proposed a differential evolution technique to solve real world problem. A fractional order PID controller is design, the controller is applied in pitch control of an aircraft system. Results have shown its good robustness.

### 3. PROPOSED ALGORITHM

Similar to most differential evolution algorithms, MODECD adopts mutation and crossover operations to implement the evolution process. The proposed algorithm has a novel modification for MODE. A matrix based on L $\infty$  metric matrix generated by fitness values is used to achieve clustering used for space partitioning. This is an important step of the whole algorithm. In this section, a detailed description of the designed algorithm is given.

#### 3.1 Framework

In the beginning, an initial population  $X$  with size  $N$  is randomly generated.  $D$  is the

vector dimension,  $M$  is the number of objective functions and the boundary constraints of  $\mathbf{X}$  are written as  $[\mathbf{X}_{\min}, \mathbf{X}_{\max}]$ . The proposed algorithm adopts the mutation strategies and crossover operation to generate the offspring. Archive set is not used. Elitist solutions all participate in the evolution. Before that, a method based on  $L^\infty$  metric is proposed to cluster the solutions with similar characteristics. This step divides the objective space ( $S$ ) into several sub-spaces ( $S_1, S_2, \dots, S_k$ ). As a result, the population is divided into several sub-populations. Superior solutions are selected from each sub-space. They include the superior directions which guide the whole evolution process. Meanwhile, two important parameters  $F, CR$  are self-adapted based on the quality of each solution in the current generation. The main loop ends when the termination condition is reached.

Pseudo1 is used to show the framework of MODECD.

---

**Algorithm 1:** Framework of MODECD

---

**Input:** Problem:  $Fun$ , boundary:  $X_{boundary}, F_{boundary}, CR_{boundary}, pro, n$

**Output:** population  $X$ , fitness value  $f$

```

1       $[X, C] \leftarrow Random-Initialize(NP, X_{boundary}, F_{boundary}, CR_{boundary})$ 
2      While ( $gen < G_{max}$ )
3           $[P_{best}, S_1, \dots, S_n, C_1, \dots, C_n, f_1, \dots, f_n] \leftarrow cluster\ method(X, f, C, n)$ 
4           $[\hat{X}, \hat{S}_1, \dots, \hat{S}_n, mat] \leftarrow Update(P_{best}, S_1, \dots, S_n, pro)$ 
5           $[\hat{C}] \leftarrow Parameter-adaption(\hat{S}_1, \dots, \hat{S}_n, C_1, \dots, C_n, f_1, \dots, f_n)$ 
6           $[X, f] \leftarrow Non-dominated-Selection(X, \hat{X}, f, \hat{f})$ 
7           $[pro] \leftarrow Mutation-adaption(X)$ 
8      return  $X, fit$ 

```

---

### 3.2 Clustering Method Based on $L^\infty$ Metric

The  $L^\infty$  metric is a variant of the Minkowski distance. The distance between fitness values points to the relationship between solutions. Clustering is based on the normalization operation given by

$$\bar{f}_j(\mathbf{x}_i) = (f_j(\mathbf{x}_i) - f_j^{\min}) / (f_j^{\max} - f_j^{\min}), i = 1, 2, \dots, N; j = 1, 2, \dots, m \quad (1)$$

$$\mathbf{f}'(\mathbf{x}_i) = [\bar{f}_j(\mathbf{x}_i), \bar{f}_j(\mathbf{x}_i), \dots, \bar{f}_j(\mathbf{x}_i)]^T \quad (2)$$

where  $f_j(\mathbf{x})$  is the  $j$ th objective function value;  $\bar{f}_j(\mathbf{x})$  is the normalized  $j$ th objective function value;  $\mathbf{f}'(\mathbf{x})$  is the matrix consisting of  $m$  normalized objective function values;  $m$  is the number of objective functions;  $N$  is the number of solutions;  $f_j^{\min}$  and  $f_j^{\max}$  represent the minimum and maximum values of the  $j$ th objective function, respectively.

The objective values are normalized in the range  $[0,1]$ . Eq. (3) below is adopted to build the distance relationship between each solution:

$$d_{p,q} = \lim_{r \rightarrow \infty} \left( \sum_{j=1}^m |f'(x_{p,j}) - f'(x_{q,j})|^r \right)^{1/r} \quad (3)$$

where  $\mathbf{x}_q, \mathbf{x}_p$  are two different solutions. Matrix  $Q$  below is generated based on the distances as follows:

$$Q = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,N-1} & d_{1,N} \\ d_{2,1} & d_{2,2} & \dots & d_{2,N-1} & d_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{N-1,1} & d_{N-1,2} & \dots & d_{N-1,N-1} & d_{N-1,N} \\ d_{N,1} & d_{N,2} & \dots & d_{N,N-1} & d_{N,N} \end{bmatrix}. \tag{4}$$

As compared to the Euclidean distance, the  $L^\infty$  metric shows some advantages. For example, in Fig. 1, there are three Euclidean distances  $d_{A,B}$ ,  $d_{A,C}$ ,  $d_{A,D}$ . These distances show the relationship among solutions  $A, B, C, D$ . Suppose  $d_{A,B} = d_{A,C}$ . If solution  $A$  is the cluster center, solution  $D$  is easily assigned to  $A$ , but it is hard to tell whether  $B$  and  $C$  belong to cluster  $A$ . In Fig. 2,  $d_{A,B}$ ,  $d_{A,C}$ ,  $d_{A,D}$  represent the distances relationship between  $A, B, C, D$  based on the  $L^\infty$  metric. It is obvious that  $d_{A,D} < d_{A,C} < d_{A,B}$ . Both solutions  $C$  and  $D$  are assigned to cluster  $A$ .

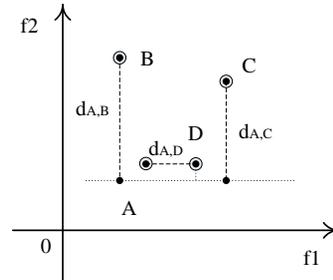
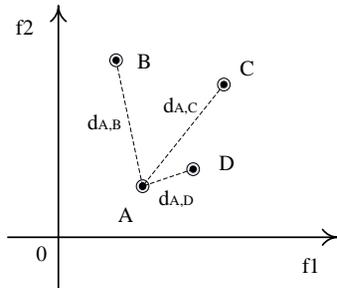


Fig. 1. Euclidean distances between  $A, B, C, D$ . Fig. 2.  $L^\infty$  metric between  $A, B, C, D$ .

The data obtained in each generation are without label. The  $K$ -means method is to be used to assign data into several clusters. With the  $K$ -means method, every solution is assigned to one cluster. The objective function  $f'(x)$  is not directly used to avoid falling into local optimum.  $Q$  is the distance relationship matrix that can reflect the location relationship between each solution  $x_i, i = 1, 2, \dots, N$ . Similarity of distances indicates the similarity of solutions. In this way, the population is divided into sub-populations  $f_1, f_2, \dots, f_n$  based on the similarities. The  $K$ -means algorithm [22] is based on minimizing  $E$ , the sum of squared errors, given by

$$E = \sum_{z=1}^k \sum_{d \in S_z} |d - \mu_z|^2 \tag{5}$$

where  $\mu_z$  is the  $z$ th cluster center given by:

$$\mu_z = \frac{1}{|S_z|} \sum_{d \in S_z} d, z = 1, 2, \dots, k. \tag{6}$$

In the first step, the number of clusters is chosen as  $k$ , and the cluster centers are selected randomly as  $\{\mu_1, \mu_2, \dots, \mu_k\}$ . The distance relationship between  $\mu_z, z = 1, 2, \dots, k$  and each row vector in  $Q$  is calculated. The minimum distance  $d_{i,z}$  is marked and its cor-

responding solution  $x_i$  is assigned to sub-population  $\hat{S}_z$ . Looping this operation, the sub-populations are updated as  $\hat{S}_z = \hat{S}_z \cup \{x_i\}$ . Thus, the population is split into  $k$  sub-populations as well as its corresponding objective space. The square distance to the center  $\mu_k$  is given by:

$$\hat{d}_{i,z} = \|d_i - \mu_k\|^2, z = 1, 2, \dots, k, i = 1, 2, \dots, |S_z|. \tag{7}$$

### 3.3 Superior Individual Selection

Each sub-population has its own superior individuals. These solutions contain the optimal direction, and they are more close to the true Pareto front as compared to other solutions. Modeling the direction of evolution is the key point of the proposed method. The data of fitness values are assumed to satisfy the linear relationship. A linear regression strategy is proposed to select the superior individuals. Based on the regression model, the evolution trend is able to form a line or a plane, and can be used as a selection standard in the proposed algorithm to choose superior individuals.

Eq. (8) expresses the distribution trend of the fitness value:

$$h_{\theta}(f') = \sum_{j=1}^m \theta_j f'_j = \theta^T f' \tag{8}$$

where  $\theta$  is the coefficient matrix, and  $h_{\theta}(f')$  represent the hypotheses function. For instance, in Figs. 3 and 4, the colored points represent the fitness values in the objective space. Different colors indicate the different clusters, for different sub-populations. Dotted line is the linear regression line. It shows the distribution trend of the fitness values in a generation. The distribution trend line is translated to make it cross the origin. The new distribution trend (guide line in Fig. 4) provides a direction to find the superior individuals in  $\hat{S}_1, \hat{S}_2, \dots, \hat{S}_k$ .

The superior individuals are defined as the ones with minimum distance to the guide line. The numbers of solution in  $k$  sub-populations are defined as  $Num = [\varphi_1, \varphi_2, \dots, \varphi_k]$ ,  $\eta_{i,z}$  is the distance of  $i$ th fitness value in the  $z$ th sub-space. Eqs. (9) and (10) below show how the minimum distances are chosen to find the superior individuals. Fig. 5 shows the details.

$$\eta_{i,z} = |\theta^T f'_{i,z} / \|\theta\|_2, i = 1, 2, \dots, \varphi_z, z = 1, 2, \dots, k \tag{9}$$

$$\eta_{\min,z} = \min(\eta_{1,z}, \eta_{2,z}, \eta_{3,z}, \dots, \eta_{\varphi_z,z}), z = 1, 2, \dots, k \tag{10}$$

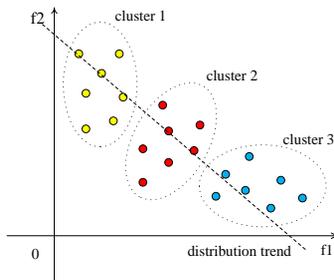


Fig. 3. Clusters and distribution in objective space.

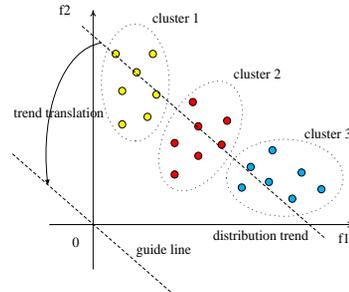


Fig. 4. Distribution trend translations in objective space.

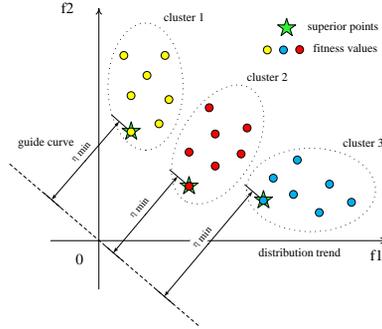


Fig. 5. Illustration of superior individual selection in the object space.

### 3.4 Update Operation

In the evolution process, three mutation strategies (DE/best/2, DE/current-to-best/bin and DE/rand-to-best/bin [23]) and crossover strategy in traditional DE are adopted. Two novel modifications of the parameters  $[F, CR]$  and self-adaptation of mutation strategy are proposed. In MODECD, every individual has its own parameters and a mutation strategy label. These labels form a label set  $mat$ . The mutation strategies are as follows:

$$\text{DE/best/2:} \quad \mathbf{v}_i = \mathbf{x}_{best,z} + F_i \bullet (\mathbf{x}_1 - \mathbf{x}_2) + F_i \bullet (\mathbf{x}_3 - \mathbf{x}_4) \quad (11)$$

$$\text{DE/current-to-best/bin:} \quad \mathbf{v}_i = \mathbf{x}_1 + F_i \bullet (\mathbf{x}_{best,z} - \mathbf{x}_1) + F_i \bullet (\mathbf{x}_2 - \mathbf{x}_3) \quad (12)$$

$$\text{DE/rand-to-best/bin:} \quad \mathbf{v}_i = \mathbf{x}_1 + F_i \bullet (\mathbf{x}_{best,z} - \mathbf{x}_i) + F_i \bullet (\mathbf{x}_1 - \mathbf{x}_2) \quad (13)$$

where  $F$  represent the mutation parameter, and  $x_1, x_2, x_3, x_4$  are randomly selected from  $S_z$ ,  $z = 1, 2, \dots, k$ .

Crossover strategy is given by:

$$u_{i,j} = \begin{cases} v_{i,j}, & rand(0, 1) \leq CR_i \\ x_{i,j}, & \text{otherwise} \end{cases}, \quad j = 1, 2, \dots, D. \quad (14)$$

Two strategies are applied to each sub-population to obtain the offsprings of each sub-population  $\bar{S} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \dots, \mathbf{u}_{\varphi_z}\}$ ,  $z = 1, 2, \dots, k$ . The parent and offspring populations are combined together; the dominance relationship and the distance among surrounding are used to select  $N$  individuals to maintain the size of the population.

### 3.5 Modification of Parameters and Update Strategy

(A) Strategy for self-adaptation of parameters

In the DE algorithm, the mutation parameter determines the magnitude of the disturbance; the crossover parameter determines the probability of preserving the characteristics of the previous generation. Different evolution phases need different parameter sets to fit for the requirement of evolution. The improvement of a solution indicates the quality of the parameters. The superior individuals selected in Section 2.2 include the information for the superior parameters as well. Dynamically adjusting the parameter set improves the

quality of the parameter set.

The change between the fitness values of the parents and the fitness values of the offsprings shows how to proceed further. For the minimum optimization problem, the minimal result is to be sought. The differences between fitness values are calculated first as

$$\Delta f_{i,z} = \sum_{i=1}^m (\hat{f}_{i,z}^j - f_{i,z}^j), \quad i = \{1, 2, \dots, \varphi_z\} \quad (15)$$

where  $m$  is the number of fitness functions,  $\hat{f}_z, z = \{1, 2, \dots, k\}$  are the fitness values of the offsprings and the parents in the  $z$ th sub-population.  $\Delta f_{i,z}$  is the difference value. When  $\Delta f_z$  is negative, it indicates the fitness value of an offspring is smaller than its parent on the average; the parameters are useful to help the individual approach the Pareto front. Each individual's corresponding parameter should be reserved. Two data sets are defined:  $T_z, z = 1, 2, \dots, k$  is used to store the negative  $\Delta f_z$ ;  $M_z, z = 1, 2, \dots, k$  is used to store the corresponding parameter vectors. To implement the self-adaptation parameter strategy, Gaussian distribution probabilistic model  $N(\mu, \sigma^2)$  is used to describe the distribution of optimal parameters based on the data set  $M$ .  $\mu$  is the mean,  $\sigma^2$  is the variance.

$$\mu_z = \frac{1}{\lambda_z} \sum_{i=1}^{\lambda_z} M_{i,z}, \quad z \in \{1, 2, \dots, k\} \quad (16)$$

$$\sigma_z^2 = \frac{1}{\lambda_z} \sum_{i=1}^{\lambda_z} (M_{i,z} - \mu_z)^2, \quad (17)$$

where  $\lambda_z$  is the size of superior parameter set  $M_z$ . The new parameter set is generated by Gaussian based modeling  $N(\mu_z, \sigma_z^2), z = \{1, 2, \dots, k\}$  and random sampling.

Different sub-populations have different parameter distributions, and have different contributions to the whole population's evolution. The data set  $T_z, z = 1, 2, \dots, k$  is used to judge the effect of  $z$ th sub-population in the current generation by generating the weight value  $w_z$  as:

$$w_z = \frac{\sum_{i=1}^{\lambda_z} T_z}{(\sum_{i=1}^{\lambda_1} T_1 + \sum_{i=1}^{\lambda_2} T_2 + \dots + \sum_{i=1}^{\lambda_k} T_k)}. \quad (18)$$

Data set  $T_z$  contains the differences of the fitness values between parents and offsprings in the  $z$ th sub-population. Bigger  $|\sum T_z|$  indicates the performance of offsprings is much better than the parents and its corresponding superior parameters should be assigned more weight to make a contribution to the whole evolution in the current generation. Thus, the Gaussian distribution is modified as

$$\hat{\mu} = \sum w_z \mu_z, \quad z = 1, 2, \dots, k, \quad (19)$$

$$\hat{\sigma}^2 = \sum w_z \sigma_z^2, \quad z = 1, 2, \dots, k. \quad (20)$$

With the new distribution, a new parameter set for the next generation is generated.

(B) Self-adaptation mutation strategy

In Section 2.4, every individual has a mutation strategy label *mat* marked 1, 2, and 3. The numbers represent three mutation strategies, respectively. Every individual has a random selection from the mutation strategies library. The significance of each mutation strategy to be selected depends on the proportion of contribution to the evolution.

$$\begin{aligned}\gamma_l &= \sum(\mathit{mat} == l), l = 1, 2, 3 \\ \xi_l &= \gamma_l / (\gamma_1 + \gamma_2 + \gamma_3)\end{aligned}\quad (21)$$

$\xi_l$  is the proportion of the *l*th mutation strategy. This proportion is used to update the mutation strategies library.

## 4. EXPERIMENTAL EVALUATION

In this section, a comparison between five state-of-the-art algorithms and the proposed algorithm MODECD is presented. The comparison algorithms are as follows: *r*-dominated based NSGA-II (rNSGA-II) [24], non-dominated sorting and local search (NSLS) [25], multi-objective evolutionary algorithm based on decomposition (MOEA/D-DE), decomposition based MOEA with covariance matrix adaptation (MOEA/D-CMA) [26], decision variable analysis based MOEA (MOEA/DVA) [10], and inverse modeling multi-objective evolutionary algorithm (IMMOEA) [27]. All the comparison algorithms are implemented by PlatEMO [28] in Matlab. The benchmark functions are bi-objectives or tri-objectives (ZDT [29], DTLZ and UF test unconstraint suites [30]). Two performance indicators: inverted generational distance (IGD), spacing (SP) and Hypervolume (HV) [31] are used as discussed below.

### 4.1 Performance Indicators

(A) **IGD**: this indicator uses the mean of minimum Euclidean distance between individuals in the population and the true PF. Smaller IGD value indicates better algorithm.

(B) **HV**: this indicator adopts a volume that dominates the solution set to measure the convergence and diversity. Meanwhile, a predefined reference point that is dominated by all Pareto front solutions is needed. For each non-dominated solution  $i \in M$  ( $M$  is a set of non-dominated solutions), a hypercube  $v_i$  is constructed with a reference point  $W$  and the solution  $i$  as the diagonal corners of the hypercube. *HV* is calculated as follows [31]:

$$HV = \text{volume} \left( \bigcup_{i=1}^{|M|} v_i \right). \quad (22)$$

Larger *HV* means the algorithm is better.

### 4.2 Experimental Setting

In the experiments, the label set is initially set as empty namely,  $\mathit{mat} = \emptyset$ . The probabilities of the three mutation strategies are selected as:  $\xi_1 = 0.35$ ,  $\xi_2 = 0.30$ ,  $\xi_3 = 0.35$ . The

population size in all the algorithm is chosen as  $NP = 100$ . Different benchmark functions have different maximum number of generations:  $G_{\max} = 250$  when the objective number is two, DTLZ1-2:  $G_{\max} = 300$ , DTLZ4-5:  $G_{\max} = 200$ , DTLZ3 and UF8-10:  $G_{\max} = 500$ . The initial mutation and crossover parameters are randomly generated in the range of  $[0.2, 0.7]$  and  $[0.1, 0.4]$ , respectively. The number of sub-spaces is chosen as  $k = 3$ . All the tests are run 30 times independently.

### 4.3 Discussions and Analysis

The mean values of IGD over 30 independent runs for the benchmark functions of rNSGA-II, NSLS, MOEADCMA, MOEA/D-DE, MOEADVA, IMMOEA and MODECD are listed in Table 1. The number in parentheses indicates the ranking of each algorithm. In the comparison algorithms, the grid is highlighted in grey if the result is best; the grid is highlighted in light grey if the result is second best.

**Table 1. IGD results of rNSGA-II, NSLS, MOEACMA, MOEA/D-DE, MOEADVA and MODECD.**

	rNSGA II	NSLS	MOEADCMA	MOEA/D-DE	MOEADVA	IMMOEA	MODECD
ZDT1	3.1620E-01(5)	7.4159E+00(7)	5.7850E-02(3)	2.9001E-02(2)	6.8561E+00(6)	1.7817E-01(4)	<b>5.6907E-03(1)</b>
ZDT2	3.5275E-01(5)	9.4764E+00(7)	8.4259E-02(3)	5.8541E-02(2)	9.3622E+00(6)	2.8968E-01(4)	<b>6.4901E-03(1)</b>
ZDT3	5.3587E-01(5)	7.6336E+00(6)	7.1110E-02(3)	6.3428E-02(2)	8.1715E+00(7)	1.7372E-01(4)	<b>5.9337E-03(1)</b>
ZDT4	3.0471E-01(3)	1.4412E+00(6)	1.1172E+00(5)	3.4401E-01(4)	8.5117E+00(7)	<b>6.2442E-03(1)</b>	1.2419E-01(2)
ZDT6	2.5782E-01(6)	9.2841E-03(3)	3.1228E-03(2)	<b>3.1119E-03(1)</b>	1.1700E-01(5)	2.1932E+00(7)	9.1069E-02(4)
DTLZ1	1.3955E-01(4)	3.3396E-01(5)	4.0777E-01(6)	5.6329E-02(2)	1.2561E-01(3)	3.5160E+00(7)	<b>2.8495E-02(1)</b>
DTLZ2	5.1425E-01(7)	<b>5.6694E-02(1)</b>	5.8786E-02(3)	7.5935E-02(5)	5.8398E-02(2)	9.4629E-02(6)	6.4944E-02(4)
DTLZ3	<b>4.2582E-01(1)</b>	5.7346E+00(5)	9.1300E+00(6)	2.9737E+00(4)	1.5988E+00(3)	4.3502E+01(7)	5.1137E-01(2)
DTLZ4	4.6845E-01(6)	3.3709E-01(5)	2.0011E-01(4)	1.7637E-01(3)	4.7517E-01(7)	8.7791E-02(2)	<b>6.7905E-02(1)</b>
DTLZ5	3.9773E-01(7)	7.0767E-03(2)	2.2434E-02(5)	1.4282E-02(4)	9.1520E-03(3)	2.4974E-02(6)	<b>5.6736E-03(1)</b>
UF1	3.3864E-01(7)	<b>3.6921E-02(1)</b>	7.2887E-02(3)	9.4215E-02(6)	6.9153E-02(2)	8.7955E-02(5)	8.7700E-02(4)
UF2	3.6486E-01(7)	4.4555E-02(3)	<b>3.9832E-02(1)</b>	6.2123E-02(6)	4.5799E-02(4)	4.1090E-02(2)	4.6349E-02(5)
UF3	3.9112E-01(6)	3.7140E-01(5)	1.7584E-01(3)	2.1897E-01(4)	4.0933E-01(7)	<b>1.5289E-01(1)</b>	1.7285E-01(2)
UF4	4.0314E-01(7)	8.5223E-02(3)	9.3318E-02(6)	9.2647E-02(4)	<b>5.6172E-02(1)</b>	9.2838E-02(5)	6.8882E-02(2)
UF5	5.9188E-01(2)	7.9700E-01(3)	1.0560E+00(5)	1.0970E+00(6)	9.5184E-01(4)	1.2239E+00(7)	<b>3.2560E-01(1)</b>
UF6	3.5800E-01(4)	4.5785E-01(6)	3.1768E-01(2)	3.2074E-01(3)	5.7378E-01(7)	3.5824E-01(4)	<b>1.4293E-01(1)</b>
UF7	4.2533E-01(7)	1.0151E-01(2)	<b>3.8767E-02(1)</b>	1.6844E-01(6)	1.4633E-01(4)	1.5678E-01(5)	1.0529E-01(3)
UF8	5.4818E-01(7)	1.9219E-01(2)	2.6591E-01(3)	2.7417E-01(5)	<b>1.1078E-01(1)</b>	2.7547E-01(6)	2.7053E-01(4)
UF9	4.8261E-01(7)	1.5165E-01(2)	2.7887E-01(4)	2.6601E-01(3)	<b>9.3860E-02(1)</b>	3.5935E-01(6)	3.0117E-01(5)
UF10	5.6427E-01(3)	6.5158E-01(5)	1.1442E+00(7)	8.6183E-01(6)	6.4381E-01(4)	<b>3.0845E-01(1)</b>	4.0695E-01(2)

The performance of the proposed MODECD achieves the best eight times and five times in second place. Especially in ZDT and DTLZ benchmark functions, MODECD has the better performance than compared algorithms. The MOEADVA is a competitive algorithm when solving UF8 and UF9, while it failed in top two with the ZDT and DTLZ benchmark functions. MOEA/D-DE always ranked in second place when dealing with ZDT benchmark functions. NSLS ranked in the second place for four times in UF benchmark functions. Fig. 6 shows the average IGD ranking of compared algorithm. The scores of rNSGA-II, NSLS, MOEADCMA, MOEA/D-DE, MOEADVA, IMMOEA and MODECD are 5.3, 3.95, 3.75, 3.9, 4.2,4.5, and 2.35, respectively. MODECD is the smallest one

compared to other algorithms. NSLS, MOEADCMA, MOEA/D-DE, MOEADVA have not too much difference. The average performance demonstrates the competitiveness of MODECD in solving multi-objectives problems.

In Fig. 7, the 30 independent runs IGD results of ZDT and DTLZ benchmark functions are present. NSLS and MOEDVA are not competitive when dealing with ZDT1-3, they are not showing in figures of ZDT1-3. The orange line represents the MODECD. It is very stable. For the 30 independent runs, the results of MODECD have no obvious fluctuations.

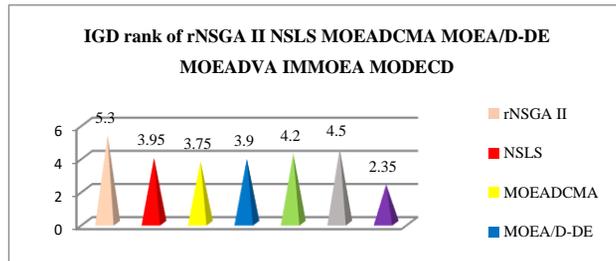


Fig. 6. Average IGD rank of rNSGA-II, NSLS, MOEADCMA, MOEA/D-DE, MOEADVA, IMMOEA and MODECD.

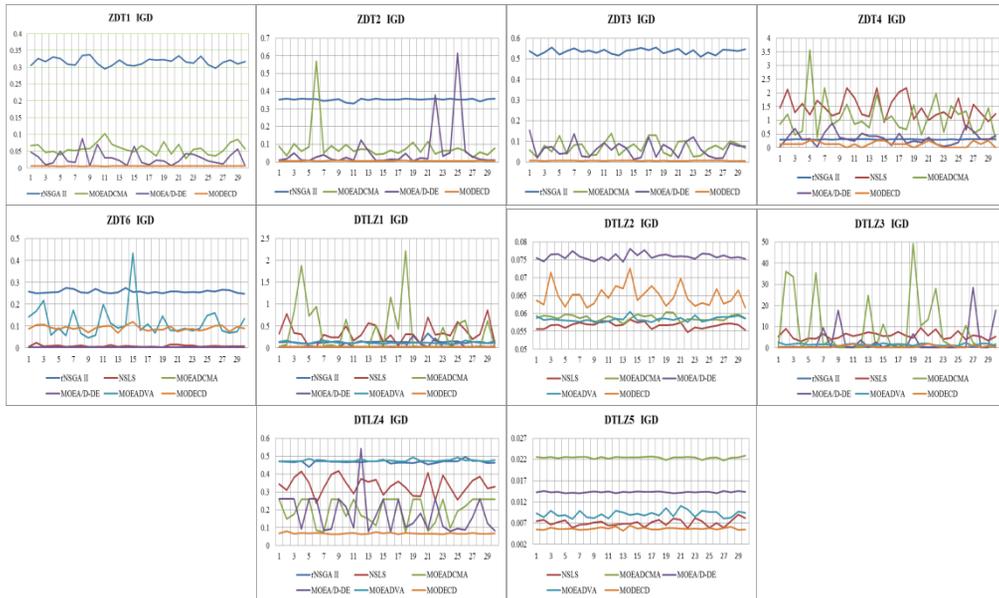


Fig. 7. IGD variance of 30 independent runs of ZDT and DTLZ benchmark functions.

In Table 2, the HV metric was adopted to evaluate the performance of the compared algorithms. The mean values of HV over 30 independent runs are given. Bigger value of HV indicates the algorithm has a better convergence and diversity. The same is valid for the IGD metric. If the result is best, the grid will be highlighted in grey; if the result is second best, the grid is highlighted in light grey. The number in parentheses shows the ranking of

each algorithm. The proposed MODECD is the best among the compared algorithms on eight out of twenty test functions and is seven times second best among the compared algorithms. rNSGA II and MOEADCMA have not ranked in the first place and ranked in the second place for three times for HV metric. NSLS and IMMOEA are two competitive algorithms when dealing with UF benchmark functions. They have ranked for four times in first place, three times in the second place and four times in first place, respectively. Fig 8 shows the average HV ranking of compared algorithm. The average ranking of MODECD is equal to 5.95. It is still in the first place. The average ranking of MOEA/D-DE is equal to 4.8. It is ranked in the second place. The entire performance shows the decomposition-based method is very effective.

**Table 2. HV results of rNSGA-II, NSLS, MOEACMA, MOEA/D-DE, MOEADVA and MODECD.**

	rNSGA II	NSLS	MOEADCMA	MOEA/D-DE	MOEADVA	IMMOEA	MODECD
ZDT1	4.4348E-01(3)	0.0000E+00(1)	6.4478E-01(5)	8.2734E-01(6)	0.0000E+00(1)	5.9299E-01(4)	<b>8.6656E-01(7)</b>
ZDT2	2.1555E-01(4)	0.0000E+00(1)	3.4143E-01(5)	4.6935E-01(6)	0.0000E+00(1)	1.6440E-01(3)	<b>5.3105E-01(7)</b>
ZDT3	3.2493E-01(3)	0.0000E+00(1)	5.2690E-01(4)	9.0091E-01(6)	0.0000E+00(1)	6.8464E-01(5)	<b>1.0166E+00(7)</b>
ZDT4	4.8351E-01(5)	1.6087E-03(2)	6.0103E-02(3)	4.2532E-01(4)	0.0000E+00(1)	<b>8.6539E-01(7)</b>	6.9316E-01(6)
ZDT6	2.0698E-01(2)	4.2729E-01(6)	3.8858E-01(5)	<b>4.3333E-01(7)</b>	3.0340E-01(3)	0.0000E+00(1)	3.1037E-01(4)
DTLZ1	7.6763E-02(3)	3.7201E-02(2)	4.4958E-01(6)	1.2001E-01(4)	<b>5.0517E-01(7)</b>	0.0000E+00(1)	1.3559E-01(5)
DTLZ2	1.6651E-01(1)	<b>7.3786E-01(7)</b>	5.5147E-01(3)	7.0074E-01(5)	5.3315E-01(2)	6.2771E-01(4)	7.1387E-01(6)
DTLZ3	3.3392E-01(5)	0.0000E+00(1)	1.5684E-01(4)	<b>4.3537E-01(7)</b>	1.7937E-03(3)	0.0000E+00(1)	4.0076E-01(6)
DTLZ4	6.8714E-04(1)	1.5120E-02(4)	5.0387E-01(6)	8.9430E-04(2)	2.4655E-01(5)	1.0029E-02(3)	<b>7.2202E-01(7)</b>
DTLZ5	3.0658E-02(1)	1.3155E-01(4)	1.8998E-01(6)	1.2919E-01(3)	<b>1.9548E-01(7)</b>	1.2124E-01(2)	1.3268E-01(5)
UF1	3.8231E-01(1)	<b>8.0694E-01(7)</b>	6.1105E-01(2)	7.1842E-01(4)	6.1609E-01(3)	7.2504E-01(5)	7.3584E-01(6)
UF2	3.6929E-01(1)	8.0020E-01(4)	6.7659E-01(3)	8.0117E-01(5)	6.5062E-01(2)	<b>8.2151E-01(7)</b>	8.1298E-01(6)
UF3	2.9374E-01(2)	3.0630E-01(3)	4.9038E-01(4)	5.1525E-01(5)	2.3208E-01(1)	<b>6.0550E-01(7)</b>	5.8456E-01(6)
UF4	2.0665E-01(1)	3.9444E-01(6)	3.1741E-01(2)	3.8538E-01(5)	3.6406E-01(3)	3.8356E-01(4)	<b>4.2979E-01(7)</b>
UF5	9.0015E-02(6)	3.2625E-03(5)	0.0000E+00(1)	1.6144E-03(4)	7.7556E-06(3)	0.0000E+00(1)	<b>2.1017E-01(7)</b>
UF6	1.8401E-01(4)	9.4248E-02(2)	1.8195E-01(3)	2.7255E-01(6)	3.1607E-02(1)	2.0639E-01(5)	<b>4.0072E-01(7)</b>
UF7	2.3977E-01(1)	5.3102E-01(6)	5.2576E-01(5)	5.1939E-01(4)	3.8382E-01(2)	4.9801E-01(3)	<b>5.7620E-01(7)</b>
UF8	1.3442E-01(1)	<b>5.3907E-01(7)</b>	3.2053E-01(2)	3.8613E-01(4)	4.4370E-01(6)	4.1483E-01(5)	3.7071E-01(3)
UF9	3.0638E-01(1)	<b>8.3185E-01(7)</b>	5.1050E-01(2)	7.0156E-01(5)	7.0574E-01(6)	5.1318E-01(3)	6.1006E-01(4)
UF10	1.2584E-01(5)	4.9623E-03(2)	2.7229E-04(1)	7.8620E-03(4)	5.4514E-03(3)	<b>3.3109E-01(7)</b>	1.3430E-01(6)

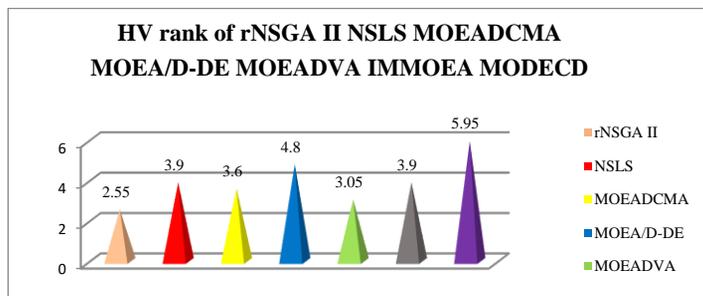


Fig. 8. Average HV rank of rNSGA-II, NSLS, MOEADCMA, MOEA/D-DE, MOEADVA, IMMOEA and MODECD.

Fig. 9 shows the HV various in UF benchmark functions for 30 independent runs. UF5-10 has obvious fluctuations, it due to the complex of benchmark functions. The results of HV obtained by MODECD are better in most cases. Meanwhile, the results of UF1-4 obtained by all compared algorithms are stable and MODECD is still competitive.

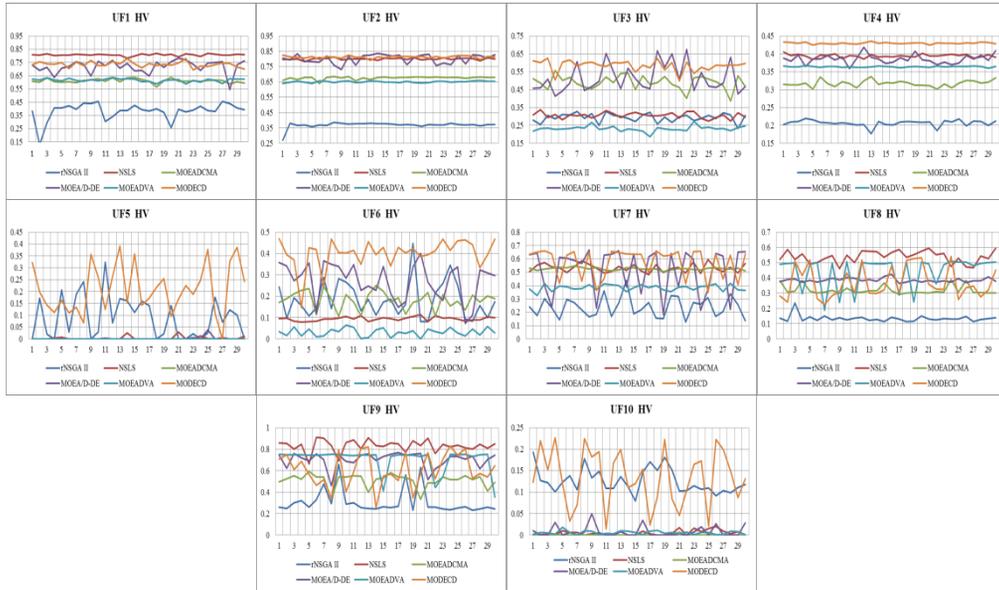


Fig. 9. HV variance of 30 independent runs of UF benchmark functions.

**Table 3. IGD and HV rankings of the ZDT, DTLZ and UF benchmark functions by fired-man test.**

	rNSGAII	NSLS	MOEADCMA	MOED/D-DE	MOEADVA	MODECD	<i>p</i> -value
HV	2.30	3.43	3.20	4.15	2.73	<b>5.20</b>	8.286e-06
IGD	4.55	3.50	3.50	3.55	3.75	<b>2.15</b>	4.340e-03

**Table 4. *p*-values obtained by IGD and HV of ZDT, DTLZ and UF benchmark functions by Wilcoxon test.**

MODECD vs		rNSGAII	NSLS	MOEADCMA	MOED/D-DE	MOEADVA	MODECD
HV	<i>p</i> -value	0.00008857	0.01374129	0.00282086	0.01374129	0.00282086	0.00008857
	Sign (0.05)	■	■	■	■	■	■
IGD	<i>p</i> -value	0.00010335	0.01374129	0.02509351	0.00220394	0.00642460	0.00010335
	Sign (0.05)	■	■	■	■	■	■

Tables 3 and 4 were obtained by Friedman test and Wilcoxon test. The *p*-value was adopted to evaluate whether the null hypothesis was rejected with the significance level equal to 0.05. The rankings of IGD metric and HV metric by Friedman test is 2.15 and 5.20, respectively. ‘■’ in Table 4 means the *p*-values is smaller than 0.05. The overall performance of MODECD is better in IGD metric and HV metric as compared to rNSGA-II, NSLS, MOEADCMA, MOEA/D-DE and MOEADVA. The properties of MODECD solution set contain good convergence and diversity.

## 5. PROCESS OPTIMIZATION FOR PRODUCTION OF SODIUM GLUCONATE BY FERMENTATION METHOD

Sodium gluconate is a deep processing product of glucose, and has a characteristic of non-toxicity, and non-volatility. It is also soluble in water. Sodium gluconate is widely used in real world industries such as food, textile and construction. The fermentation method is the mainstream approach to produce sodium gluconate. With this method, it is easy to control reaction speed, and the obtained sodium gluconate is easy to extract [32]. Usually the fungus *Aspergillus niger* is used as strain. Its secretion of glucose oxidase (GOD) is able to turn glucose into gluconic acid through a simple dehydrogenation reaction [33]. In order to maintain good growing environment of *Aspergillus niger*, the neutralizing agent sodium hydroxide is adopted to adjust the pH, and sodium gluconate is obtained by a neutralization reaction. The fermentation process is complex and contains many intermediates. Not considering the strain growth, the fermentation can be described by the following two equations:



Based on the uncertain, multi-level and nonlinear characteristics of the fermentation process, Wang *et al.* [34] proposed a mechanism model using stoichiometric equations to describe the sufficient mechanism knowledge. It includes three stages: growth of strain, consumption of substrate and generation of sodium gluconate as follows:



where  $C(\%)$  is the concentration of oxygen;  $P$ ,  $S$ ,  $X$  are the concentration of product (sodium gluconate, g/L), substrate concentration (glucose, g/L), and the amount of microbe (*Aspergillus niger*, g/L), respectively;  $r_1$ ,  $r_2$ ,  $r_3$  are the reaction rates of growth of bacteria, consumption of glucose, and production of sodium gluconate, respectively.

### 5.1 The Influence of Control Condition and Construction of MOP in Fermentation Process

*Aspergillus niger* is an aerobic fungus. The supplement of oxygen affects its growth; The supplement of oxygen also affects the transportation of oxygen between liquid and gas. There is a balance between concentration of *Aspergillus niger* and substrate to avoid the internal competition among strain and the waste of substrate. Considering the economic factor, the proper input, the proper supplement of oxygen, and proper environment the following is desirable; the more the output the better, the shorter reaction time the better. At the end of reaction, when all the fermentation liquid is poured away, the less remain of glucose the better. Paragraph a MOP for fermentation process is formed by using three

objective functions as follows: conversion rate  $f_1$ : the rate between output sodium gluconate (at end reaction time) and the difference of glucose at initial and end reaction times; remaining glucose  $f_2$ : the amount of sodium gluconate at the end of reaction time; utilization rate of equipment  $f_3$ : the rate between output sodium gluconate (at end reaction time) and whole reaction time. These three functions conflict with each other. For example, increasing  $f_1$  leads to a decrease of  $f_2$ . The mathematical description is as follows:

$$\begin{aligned} \mathbf{F}(\mathbf{x}) &= \{\max(f_1), \max(f_2), \max(f_3)\}; \\ \begin{cases} f_1 = P / (S_0 - S_e); \\ f_2 = S_e; \\ f_3 = P / t; \end{cases} & \quad (26) \\ P_{\min} \leq P \leq P_{\max}; S_{\min} \leq S \leq S_{\max}; t_{\min} \leq t \leq t_{\max}; \end{aligned}$$

where  $t$  is the reaction time;  $S_0$  represents the concentration of glucose at the initial reaction time;  $S_e$  is the concentration of glucose at the end reaction time.

## 5.2 MODECD Optimization of the Fermentation Process

Based on the power series kinetic model and the constructed optimal objective function, MODECD is used to optimize the control conditions of sodium gluconate production process. The input variable concentration of oxygen was adjusted by aeration and agitation in the lab. There was a modification of the kinetic model built by Wang *et al.* The concentration of oxygen was changed from an uncontrollable variable into a control variable. It is able to dynamically adjust the whole reaction time. The number of clusters is chosen as three. The population size was set as 100. The maximum generation  $G_{\max}$  was chosen equal to 100. The boundaries of mutation parameter  $\mathbf{F}$  were [0.2, 0.7]. The boundaries of crossover parameter  $\mathbf{CR}$  were [0.1, 0.4].

The obtained Pareto front by MODECD is shown in Fig. 10. The obtained Pareto front by NSLS is shown in Fig. 11. It is easy to see the distribution of solution set obtained by MODECD is smooth and uniform, while in the solution set obtained by NSLS is discontinuous. Some of the solutions are gathered in the same space. To be more specific, a comparison between the experimental data and a solution selected from the Pareto set is given in Table 5.

In Table 5, the growth of strain with experimental data and simulation data were quite the same. In initial reaction time, the concentration of strain was around 0.01 (g/L), and in the end reaction time, the concentration of strain was around 0.06 (g/L). At the end reaction time, the concentration of sodium gluconate increased by 2.82 g/L (from 23.03 (g/L) to 25.85 (g/L)). At the initial time, the concentration of input glucose increased, and at the end time, the remained glucose decreased. All these improvements led to a better performance of simulation data compared to experimental data. The results with the three objective functions resulted in improvements: the conversion rate increased from 91.13% to 95.31%; and the utilization rate of equipment increased from 79.25% to 91.57%. Simultaneously, the Pareto set contained many options for the experimenters to select a proper combination of input and control conditions.

**Table 5. Optimization results of fermentation process of sodium gluconate using MODECD.**

	Experimental data		MODECD	
	Initial	End	Initial	End
The biomass concentration (g/L)	0.01	0.06	0.013	0.065
The concentration of sodium gluconate (g/L)	0.84	23.03	0.51	25.85
The substrate concentration of glucose (g/L)	31.00	6.65	33.75	6.59
Time (h)	0	28	0	28
Conversion rates $f_1$ (%)	—	91.13	—	95.31
The remaining glucose $f_2$ (g/L)	31.00	6.65	33.75	6.63
Utilization rate of equipment $f_3$ (%)	—	79.25	—	91.57

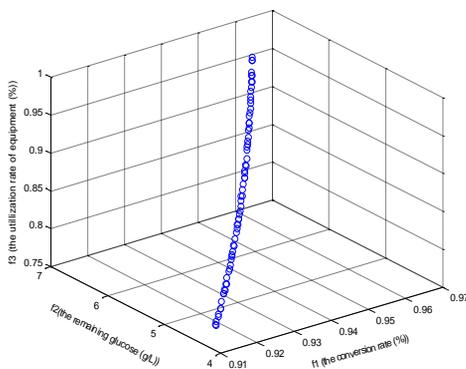


Fig. 10. Pareto front of the production of sodium gluconate via the fermentation method.

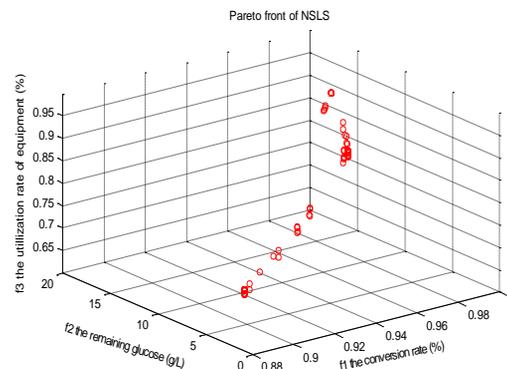


Fig. 11. NSLS Pareto front of the production of sodium gluconate via the fermentation method.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, an objective space partition based self-adaptation MODE algorithm for solving MOP is proposed. The main idea is to cluster the objective space into several subspaces based on an  $L_\infty$  metric matrix, and to extract the optimal information in each subspace to guide the evolution process. This modification increases both the diversity and the convergence of the algorithm. The proposed MODECD was compared with five state-of-the-art multi-objective evolution algorithms on 20 benchmark test functions. The statistical analysis results based on two indices IGD and HV showed the superiority of MODECD. In addition, MODECD was applied to the optimization of the production process of sodium gluconate by *Aspergillus niger*. The results demonstrated that MODECD can obtain satisfactory solutions to provided advising information to the experimenters, and improve the conversion rate, utilization rate equipment, and achieve high production efficiency. The algorithm has further scope for improvement.

The purpose of the proposed algorithm is to extract the optimal information in evolution process. The key point is division of population. In the future work, designing a more accuracy way to find the relationship between each solution and divide population at the same time is worth to study.

## REFERENCES

1. H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou, "Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey," *Swarm and Evolutionary Computation*, Vol. 44, 2019, pp. 365-387.
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, 2002, pp. 182-197.
3. S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Transactions on Cybernetics*, Vol. 46, 2016, pp. 421-437.
4. T. Murata, and H. Ishibuchi, "MOGA: Multi-objective genetic algorithms," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 289-294.
5. S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 17, 2013, pp. 721-736.
6. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, 1999, pp. 257-271.
7. Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, Vol. 12, 2008, pp. 41-63.
8. M. Tavana, Z. Li, M. Mobin, M. Komaki, and E. Teymourian, "Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS," *Expert Systems with Applications*, Vol. 50, 2016, pp. 17-39.
9. G. Yang, A. Zhang, S. Li, Y. Wang, Y. Wang, Q. Xie, and L. He, "Multi-objective evolutionary algorithm based on decision space partition and its application in hybrid power system optimisation," *Applied Intelligence*, Vol. 46, 2017, pp. 827-844.
10. X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, Vol. 20, 2015, pp. 275-298.
11. Z. Li, S. Li, C. Yue, Z. Shang, and B. Qu, "Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems," *Swarm and Evolutionary Computation*, Vol. 49, 2019, pp. 234-244.
12. S. Zapotecas-Martínez, A. López-Jaimes, and A. García-Nájera, "LIBEA: A Lebesgue indicator-based evolutionary algorithm for multi-objective optimization," *Swarm and Evolutionary Computation*, Vol. 44, 2019, pp. 404-419.
13. Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Information Sciences*, Vol. 430, 2018, pp. 397-413.
14. Y. Liu, G. G. Yen, and D. Gong, "A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, Vol.23, 2018, pp. 660-674.
15. X. Dai, X. Yuan, and L. Wu, "A novel harmony search algorithm with gaussian muta-

- tion for multi-objective optimization,” *Soft Computing*, Vol. 21, 2017, pp. 1549-1567.
16. D. Zouache, Y. O. Arby, F. Nouioua, and F. B. Abdelaziz, “Multi-objective chicken swarm optimization: A novel algorithm for solving multi-objective optimization problems,” *Computers & Industrial Engineering*, Vol. 129, 2019, pp. 377-391.
  17. G. Chen and J. Li, “A diversity ranking based evolutionary algorithm for multi-objective and many-objective optimization,” *Swarm and Evolutionary Computation*, Vol. 48, 2019, pp. 274-287.
  18. S. Wang, Y. Li, and H. Yang, “Self-adaptive mutation differential evolution algorithm based on particle swarm optimization,” *Applied Soft Computing*, Vol. 81, 2019, p. 105496.
  19. H. G. Harno and I. R. Petersen, “Synthesis of linear coherent quantum control systems using a differential evolution algorithm,” *IEEE Transactions on Automatic Control*, Vol. 60, 2014, pp. 799-805.
  20. W.-Y. Chiu, “Multiobjective controller design by solving a multiobjective matrix inequality problem,” *IET Control Theory & Applications*, Vol. 8, 2014, pp. 1656-1665.
  21. P. Kumar and J. Raheja, “Optimal design of robust FOPID for the flight control system using multi-objective differential evolution,” in *Proceedings of the 2nd International Conference on Recent Advances in Engineering and Computational Sciences*, 2015, pp. 1-4.
  22. D. Feldman, M. Schmidt, and C. Sohler, “Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering,” *SIAM Journal on Computing*, Vol. 49, 2020, pp. 601-657.
  23. H. Tang, S. Xue, and C. Fan, “Differential evolution strategy for structural system identification,” *Computers & Structures*, Vol. 86, 2008, pp. 2004-2012.
  24. L. B. Said, S. Bechikh, and K. Ghédira, “The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making,” *IEEE Transactions on Evolutionary Computation*, Vol. 14, 2010, pp. 801-818.
  25. B. Chen, W. Zeng, Y. Lin, and D. Zhang, “A new local search-based multiobjective optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, Vol. 19, 2015, pp. 50-73.
  26. H. Li, Q. Zhang, and J. Deng, “Biased multiobjective optimization and decomposition algorithm,” *IEEE Transactions on Cybernetics*, Vol. 47, 2016, pp. 52-66.
  27. R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, “A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling,” *IEEE Transactions on Evolutionary Computation*, Vol. 19, 2015, pp. 838-856.
  28. Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum],” *IEEE Computational Intelligence Magazine*, Vol. 12, 2017, pp. 73-87.
  29. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of Congress on Evolutionary Computation*, 2002, pp. 825-830.
  30. Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” Technical Report CES-487, School of Computer Science and Electronic Engineering, University of Essex, Vol. 264, 2008.
  31. E. Zitzler, D. Brockhoff, and L. Thiele, “The hypervolume indicator revisited: On the

design of Pareto-compliant indicators via weighted integration,” in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 862-876.

32. Y. Shen, X. Tian, W. Zhao, H. Hang, and J. Chu, “Oxygen-enriched fermentation of sodium gluconate by *Aspergillus niger* and its impact on intracellular metabolic flux distributions,” *Bioprocess and Biosystems Engineering*, Vol. 41, 2018, pp. 77-86.
33. H. Tan, F. Zou, B. Ma, Y. Guo, X. Li, and J. Mei, “Effect of competitive adsorption between sodium gluconate and polycarboxylate superplasticizer on rheology of cement paste,” *Construction and Building Materials*, Vol. 144, 2017, pp. 338-346.
34. X. Wang, X. Yan, F. Lu, M. Guo, and Y. Zhuang, “Power series kinetic model based on generalized stoichiometric equations for microbial production of sodium gluconate,” *Biochemical Engineering Journal*, Vol. 113, 2016, pp. 30-36.



**Zhan Guo (郭焱)** is currently pursuing the Ph.D. degree in the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shang Hai. Also, she is a joint Ph.D. student in School of Electrical and Computer Engineering, Purdue University. Her research interests include evolutionary computing, optimization and machine learning.



**Okan Ersoy** is a Professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. He received the B.S.E.E. degree from Robert College, Istanbul, Turkey, in 1967, and the M.S. Certificate of Engineering, M.S., and Ph.D. degrees from the University of California, Los Angeles, in 1968, 1971, and 1972, respectively. His current research interests include remote sensing, machine learning and pattern recognition, digital signal/image processing and recognition, transform and time-frequency methods, imaging, diffractive optics, and distant learning. He is a Fellow of the IEEE and the Optical Society of America.



**Xuefeng Yan (颜学峰)** received the B.S. degree in Biochemical Engineering and the Ph.D. degree in Control Theory Engineering from Zhejiang University, Hangzhou, China, in 1995 and 2002, respectively. He is now a Professor of East China University of Science and Technology, Shanghai, China. His research interests include complex chemical process modeling, optimizing and controlling, process monitoring, fault diagnosis, and intelligent information processing.