

A Fuzzy Decision Variables Framework for Large-scale Multiobjective Optimization

Xu Yang¹, Juan Zou¹, *Member, IEEE*, Shengxiang Yang, *Senior Member, IEEE*, Jinhua Zheng, *Member, IEEE*, and Yuan Liu

Abstract—In large-scale multiobjective optimization, too many decision variables hinder the convergence search of evolutionary algorithms. Reducing the search range of the decision space will significantly alleviate this puzzle. With this in mind, this paper proposes a fuzzy decision variables framework for large-scale multiobjective optimization. The framework divides the entire evolutionary process into two main stages: fuzzy evolution and precise evolution. In fuzzy evolution, we blur the decision variables of the original solution to reduce the search range of the evolutionary algorithm in the decision space so that the evolutionary population can quickly converge. The degree of fuzzification gradually decreases with the evolutionary process. Once the population approximately converges, the framework will turn to precise evolution. In precise evolution, the actual decision variables of the solution are directly optimized to increase the diversity of the population so as to be closer to the true Pareto optimal front. Finally, this paper embeds some representative algorithms into the proposed framework and verifies the framework’s effectiveness through comparative experiments on various large-scale multiobjective problems with 500 to 5000 decision variables. Experimental results show that in large-scale multiobjective optimization, the framework proposed in this paper can significantly improve the performance and computational efficiency of multiobjective optimization algorithms.

Index Terms—Evolutionary algorithms, Large-scale optimization, Multiobjective optimization, Fuzzy evolution, Decision variable

I. INTRODUCTION

MULTIOBJECTIVE optimization problems (MOPs) involve multiple conflicting objectives that need to be optimized simultaneously and are widespread among practical applications [1]–[5]. The conflict makes MOPs not have a unique solution that is optimal for all objectives, so the optimization algorithm can only obtain a set of trade-off solutions between the objectives. Multiobjective evolutionary algorithms (MOEAs) have been verified to effectively deal with this kind of problem and have undergone unprecedented development in the past two decades [6]–[9]. From the classic NSGA-II [10] to the most cutting-edge MaPSO [11], most studies on MOEAs focus on the scalability of the objective

dimension, and some MOEAs have been able to solve many-objective optimization problems with up to 15 dimensions [12]–[16]. However, these existing MOEAs rarely consider the scalability of decision variables in MOPs. In reality, in practical applications, there are many MOPs with a large number of decision variables [17]–[20]. Usually, MOPs with hundreds of dimensional decision variables are considered large-scale MOPs [21], [22].

When solving large-scale optimization problems, those sophisticated MOEAs will not achieve good performance because of a large number of decision variables. As the number of decision variables increases linearly, the size of the search space will increase exponentially, which will lead to the algorithm prematurely converging to a local optimum or converging to an oversized region [23]. In recent years, there are also some MOEAs for large-scale MOPs. According to the technical characteristics used, we can roughly divide these algorithms into four categories.

The first category is MOEAs based on cooperative coevolution (CC). This method divides the decision variables into multiple groups and then optimizes each group separately. For example, Antonio et al. applied a differential evolution algorithm (GDE3) in the cooperative coevolution framework for solving large-scale multiobjective problems [24], and then further proposed to combine MOEA/D [25] with coevolutionary techniques for decomposition in both objective and decision spaces [26]. Li et al. [27] also proposed a cooperative coevolutionary algorithm, in which a fast interdependency identification grouping method is utilized for large-scale MOPs. Meselhi et al. [28] proposed a new algorithm using multiple optimizers, along with a need-based allocation of computational budget for the sub-components.

The second category is MOEAs based on decision variable analysis. This method focuses on proposing a mechanism for analyzing decision variables and strives to get the best grouping of decision variables. Different from the first category, it is divided into different groups according to the types of decision variables and optimizes for each type of decision variable by different strategies. For example, Ma et al. [29] proposed an MOEA based on decision variable analyses (MOEA/DVA). The algorithm divides decision variables into distance variables and diverse variables by analyzing the relationship between decision variables and convergence and diversity attributes. The algorithm first optimizes the distance variable and then optimizes the diversity variable. Later, Zhang et al. [30] proposed an evolutionary algorithm based on clustering of decision variables (LMEA). First, the decision

¹ These authors have contributed equally to this work.

Xu Yang (e-mail: xuyang369369@gmail.com), Juan Zou (Corresponding author, e-mail: zoujuan@xtu.edu.cn), Jinhua Zheng, and Yuan Liu are with the Key Laboratory of Hunan Province for Internet of Things and Information Security, Xiangtan University, Xiangtan 411105, China, and also with the Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Xiangtan University, Xiangtan, 411105, China.

Shengxiang Yang are with the School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K., and the College of Computer Science and Cyberspace Security, Xiangtan University, Xiangtan 411105, China.

variable clustering method divides decision variables into two categories: convergence-related variables and diversity-related variables. Then, the convergence optimization strategy and the diversity optimization strategy are used to optimize the two types of decision variables.

The third category is MOEAs based on problem transformation. This method solves the problem by transforming the original large-scale problem into a small-scale problem through the problem transformation function. For example, Zille et al. [31] proposed a weight optimization framework (WOF), which optimizes the weight vector instead of decision variables, thus transforming the original large-scale problem into a small-scale problem. Subsequently, He et al. [32] proposed a problem reconstruction framework (LSMOF), which reconstructs the decision space through a series of reference solutions and weight variables. Then, the original large-scale MOP is transformed into a low-dimensional single-objective problem.

The last category is a new search method based on learning strategies. This method uses the learning mechanism between particles in the original decision space to improve the searchability of the algorithm. Common particle learning mechanisms are particle swarm optimization (PSO) [33], the competitive swarm optimizer (CSO) [34]. The most representative algorithm in this category is LMOCSSO proposed by Tian et al. [21]. In the algorithm, the inferior particles learn from the superior particles to produce promising offspring, thereby accelerating the global optimization search.

Although the existing large-scale MOEAs have shown encouraging performance, each algorithm has its shortcomings. For example, MOEAs based on the cooperative coevolution and grouping mechanism need to spend a lot of time analyzing decision variables to complete the grouping of decision variables. In addition, the performance of MOEAs based on the CC framework can be severely degraded due to inappropriate grouping. It is worth noting that the hypothesis of separability between decision variables is not always true. Therefore, it is not suitable for solving large-scale MOPs in which all decision variables interact. MOEAs based on problem transformation need to find a problem transformation function to ensure that the information loss is as little as possible after the original problem is transformed into a new problem. However, it is very difficult to find a perfect problem transformation function, and it is even more impossible in particularly complex problems. MOEAs based on learning strategies directly find the optimal solution in the original decision space. In a decision space of hundreds of dimensions, this type of algorithm can find a better solution by increasing the number of function evaluations of the evolutionary algorithm. As the dimensionality of decision variables increases, the size of the decision space increases exponentially. In MOPs with thousands of dimensional decision variables, it is not enough to solve the problem simply by increasing the number of function evaluations of the evolutionary algorithm.

Based on this analysis, there are two points: First, the dimensionality of the decision space can be reduced through the decision variable grouping and problem transformation mechanism so that large-scale MOPs become low-dimensional

MOPs. However, the correctness of the grouping mechanism and the correctness of the problem transformation function are difficult to guarantee. Second, the particle learning mechanism can improve the searchability of the algorithm in the search space. This type of algorithm may have poor scalability in the dimension of the decision space. As the dimension of the decision space increases, the algorithm is prone to fall into a local optimum or converge prematurely. We found that directly narrowing the search range of the algorithm in the original decision space instead of reducing the dimensionality is an effective way to obtain the advantages of both and discard their disadvantages. Therefore, we propose a fuzzy decision variables (FDV) framework for large-scale multiobjective optimization. In particular, the main contributions of this paper are summarized as follows:

- 1) A method of fuzzy evolution sub-stages division is proposed. This method divides the fuzzy evolution stage into multiple sub-stages with a gradually decreasing degree of fuzzification. The higher the degree of fuzzification in the sub-stage, the lower the accuracy of the solution obtained by FDV optimization. Therefore, in the later stage of evolution, the accuracy of the solution obtained by FDV optimization is higher.
- 2) Formulas for fuzzifying decision variables are proposed. The fuzzy formula adapts to multiple fuzzy evolution sub-stages. In a sub-stage, first, the two fuzzy target values of the decision variables are calculated. Then the degree of membership is calculated for the decision variable belonging to two fuzzy sets, and a fuzzy set is mapped to a fuzzy target value. Finally, the value of the decision variable is fuzzified into a fuzzy target value represented by a fuzzy set with a larger degree of membership.
- 3) In order to verify the effectiveness of the proposed FDV in solving MOPs, several representative MOEAs are embedded in the FDV and compared with the original algorithm on the multiobjective test suite DTLZ [35]. Experimental results show that MOEAs embedded in the FDV are significantly better than the original algorithm in all test cases, and the possibility of using existing MOEAs to solve large-scale problems is realized. Furthermore, the CSO [34] is embedded in the FDV framework (FDVCSO) and then compared with several state-of-the-art large-scale MOEAs on the large-scale multiobjective test suite LSMOP [36]. Experimental results show that FDVCSO is significantly better than other large-scale MOEAs in most test cases.

The rest of this paper is organized as follows. In Section II, we introduce the concepts and theoretical knowledge used to propose the FDV framework and explain the motivation. In Section III, the FDV framework and program implementation principle are described in detail. In Section IV, parameter analysis and experimental result analysis. Section V concludes the paper.

II. PRELIMINARIES AND MOTIVATION

A. Multiobjective Optimization Problem

MOPs can be mathematically defined as [37]:

$$\begin{cases} \min F(X) = (f_1(X), f_2(X), \dots, f_m(X)), \\ \text{subject to } X \in \Omega, \end{cases} \quad (1)$$

where $X = (x_1, x_2, \dots, x_n)$ is an n -dimensional decision variable vector from the decision space Ω ; $F(X)$ is an objective function vector that consists of m conflicting objective functions. A set of trade-off solutions, termed Pareto optimal solutions, is expected to be found for MOPs. Let $X_1, X_2 \in \Omega$; X_1 is said to dominate X_2 , denoted by $X_1 \prec X_2$, if and only if $f_i(X_1) \leq f_i(X_2)$ for each $i \in \{1, \dots, m\}$ and $f_j(X_1) < f_j(X_2)$ for at least one indicator $j \in \{1, \dots, m\}$; if all X from Ω cannot dominate X_1 , we call X_1 a nondominated or Pareto optimal solution. The set of all Pareto optimal solutions is called the Pareto optimal set (PS), and the set of all the Pareto optimal objective vectors is called the Pareto optimal front (PF). Any improvement in one objective of a Pareto optimal solution is bound to deteriorate one other objective at least.

B. Fuzzy Set and Membership Function

A fuzzy set refers to the totality of objects with the attributes described by a certain fuzzy concept. The membership function is a mathematical tool used to characterize fuzzy sets and represents an object's true degree of membership to a fuzzy set. The value interval of the membership function is $[0,1]$, and the value is called the degree of membership. A larger degree of membership indicates a higher degree of reality belonging to the fuzzy set.

If X is a collection of objects denoted generically by x , then a fuzzy set \tilde{A} in X is a set of ordered pairs [38]:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x) \mid x \in X)\}, \quad (2)$$

where $\mu_{\tilde{A}}(x)$ is called the membership function. For $x \in X$, the value $\mu_{\tilde{A}}(x)$ is called the degree of membership of x in the fuzzy set \tilde{A} . $\mu_{\tilde{A}}(x) = 0$ means that x is not a member of the fuzzy set, $\mu_{\tilde{A}}(x) = 1$ means that x is a full member of the fuzzy set. $\mu_{\tilde{A}}(x) \in (0, 1)$ means that is fuzzy member, which belong to the fuzzy set only partially [39].

The combination of fuzzy theory and the evolutionary algorithm is a very valuable research direction. At present, two papers apply fuzzy theory to large-scale optimization. For 2E-VRP, Yan et al. [40] proposed a method that integrates a graph-based fuzzy assignment scheme into an iteratively evolutionary learning process to minimize the total cost. Elsayed et al. [41] proposed a new way for two algorithms to cooperate as an effective team, in which a heuristic is applied using fuzzy rules of two complementary characteristics. the quality of solutions and diversity in the population. However, these works are mainly aimed at single-objective optimization problems.

We apply fuzzy theory to large-scale multiobjective optimization. In the proposed fuzzy evolution, the decision variable belongs to the universe of discourse of two fuzzy sets at the same time, and the degree of membership of the

decision variable belonging to the two fuzzy sets are calculated separately. The value of the decision variable is updated to the value represented by the fuzzy set with a larger degree of membership. All decision variable values of a solution vector have been updated to complete, which means that an original solution has been fuzzified into a fuzzy solution. In the subsequent evolution process, the fuzzy solution is operated instead of the original solution. The advantage of using fuzzy sets and membership functions in evolutionary algorithms is that it not only ensures that the algorithm searches in the original decision space but also reduces the search range of the decision space.

C. Motivation

The performance of existing large-scale MOEAs in MOPs with thousands of dimensional decision variables is unsatisfactory, especially the convergence. For example, using NSGA-II, MOEA/D, CMOPSO, LMOCSO, WOF-NSGA-II, and MOEA/DVA, experiments have been conducted on the DTLZ1 and DTLZ3 problems with tri-objective and 1000 decision variables. The plot of the convergence profiles of the mean IGD values achieved by these algorithms is displayed in Fig. 1. Fig. 1 shows that large-scale MOEAs (LMOCSO, WOF-NSGA-II, and MOEA/DVA) did not perform better than classic MOEAs (NSGA-II, MOEA/D, and CMOPSO), and even performed worse than classic MOEAs. Using MOEA/DVA, WOF-NSGA-II, LSMOF, and LMOCSO, experiments have been conducted on the LSMOP1 and LSMOP6 problems with tri-objective and 2000 decision variables. The plot of the convergence profiles of the mean IGD values achieved by these algorithms is displayed in Fig. 2. Fig. 2 shows that the existing large-scale MOEAs also converged poorly on large-scale testing problems.

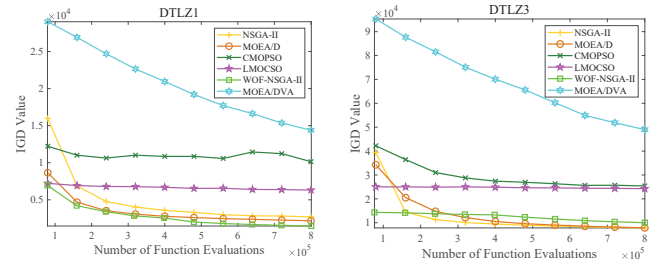


Fig. 1. Convergence profiles of several typical algorithms on tri-objective DTLZ1 and DTLZ3 with 1000 decision variables.

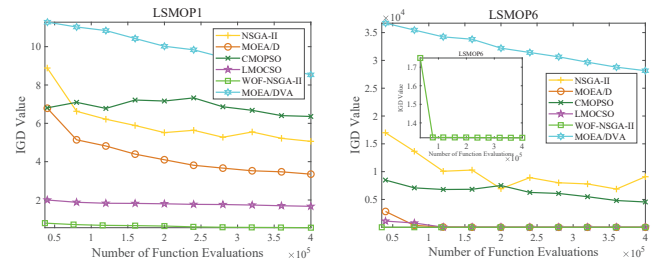


Fig. 2. Convergence profiles of several typical algorithms on tri-objective LSMOP1 and LSMOP6 with 2000 decision variables.

When the dimension of the decision space is huge, it is difficult for MOEA/DVA to guarantee the correct grouping

of decision variables. WOF-NSGA-II and LSMOF cannot guarantee that there is no information loss after problem transformation, and the huge search space makes it difficult for LMOCSSO to find valuable superior particles. Therefore, when large-scale MOEAs optimize MOPs with thousands of dimensional decision variables, there will be unsatisfactory phenomena as shown in Fig. 1 and Fig. 2.

In order to solve these problems, we propose the FDV framework. The framework does not use any decision variable grouping mechanism and problem transformation mechanism. The fuzzy evolution method is used to reduce the search range of the algorithm in the search space. Compared with other existing large-scale MOEAs, the existing MOEAs embedded in FDV show better optimization performance and computational efficiency, significantly improving the algorithm's convergence.

III. PROPOSED FRAMEWORK

The main scheme of the proposed FDV is presented in Algorithm 1. First, the population P is initialized by the population initialization method of the embedded MOEA. gen is the current number of iterations. The offspring P' are generated by the offspring generation method of the embedded MOEA. $Iter$ is the ratio of the current number of iterations to the total number of iterations. Then, the whole evolution process is divided into two stages: fuzzy evolution stage and precise evolution stage. The first stage is to add fuzzy evolution sub-stages division (line 6) and fuzzy operation (line 7) after the offspring generation operation. In the second stage, only offspring generation operations are performed. Finally, the next-generation population is screened through the embedded MOEA environmental selection method. It is worth noting that the precise evolution stage can directly evolve through the embedded MOEA. The fuzzy evolution stage is composed of two main components: fuzzy evolution sub-stages division and fuzzy operation. A detailed description is given in the following subsections. Finally, an example of fuzzy decision variables is given in the last subsection.

Algorithm 1 Main Framework of the Proposed FDV

Input: N (population size), $maxgen$ (maximum iteration number), $Rate$ (fuzzy evolutionary rate), Acc (step acceleration).

Output: P (final population).

```

1:  $P \leftarrow Initialization(N)$ ;
2: while  $gen < maxgen$  do
3:    $P' \leftarrow Operator(P)$ ;
4:    $Iter \leftarrow gen/maxgen$ ;
5:   if  $Iter \leq Rate$  then
6:      $Step \leftarrow Substages\_Division(Rate, Acc)$ ;
7:      $P' \leftarrow Fuzzy\_Operation(Step, Iter, P')$ ;
8:   end if
9:    $P \leftarrow Environmental\_Selection(P \cup P')$ ;
10: end while
11: return  $P$ ;

```

A. Fuzzy Evolution Sub-stages Division

The purpose of dividing the entire fuzzy evolution stage into multiple fuzzy sub-stages with decreasing degrees of fuzzification is to make the solution obtained by the algorithm more and more accurate. The mathematical description of the fuzzy evolutionary sub-stages division method is as follows:

$$S = \left\lceil \sqrt{\frac{2 \cdot Rate}{Acc}} \right\rceil, \quad (3)$$

$$Step(i) = \left(S \cdot i - \frac{i^2}{2} \right) \cdot Acc, i = 1, 2, \dots, S, \quad (4)$$

where S represents the number of sub-stages divided in the fuzzy evolution stage; $Step(i)$ is the cumulative step length of the first i sub-stages in the fuzzy evolution stage, the default $Step(0) = 0$; $Total = 1$ represents the total step length of the entire evolution process; $Rate$ represents the proportion of fuzzy evolution stage in the entire evolution process, namely, fuzzy evolution rate; $Acc > 0$ is the step acceleration, which can control the step length change speed between sub-stages; notably, when $Step(S) < Rate$, $Step(S + 1) = Rate$ is added.

Step acceleration (Acc) and fuzzy evolution rate ($Rate$) control the number of sub-stages in the entire fuzzy evolution stage and each sub-stage step length. Each curve in Fig. 3 represents a function image of formula (4) under certain Acc and $Rate$. The cumulative step length $Step$ of the fuzzy evolution sub-stages shows a decelerating growth trend.

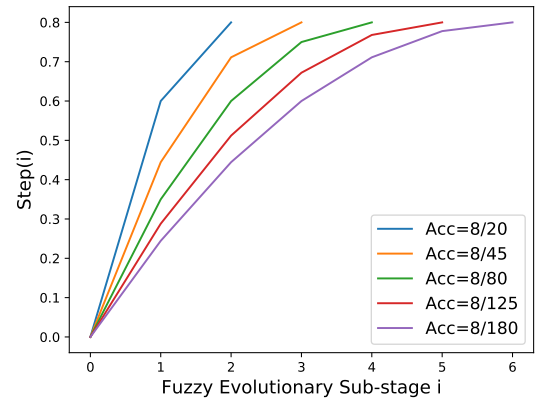


Fig. 3. The cumulative step length curve diagram corresponding to the previous i sub-stages of fuzzy evolution under different Acc when $Rate = 0.8$.

$Step(i) - Step(i - 1)$ is the step length of the i -th fuzzy evolution sub-stage. In Fig. 4, the earlier fuzzy evolutionary sub-stages have a longer step length of evolution, so sub-stages that have a higher degree of fuzzification last longer time in the evolutionary process. When the value of Acc increases under the premise that $Rate$ does not change, the fuzzy evolution is divided into fewer sub-stages. On the contrary, the fuzzy evolution is divided into more sub-stages. When FDV has poor convergence in the early stage of evolution, a larger Acc should be set to keep the algorithm in the high-fuzzification stage for a longer time. If it is found that the FDV appears to have prematurely converged during the optimization process, it may

be that the algorithm has fallen into a local optimum. Then the *Acc* should be reduced to increase the search time for searching for low-fuzzy and high-precision solutions. If it is found that the final result obtained by the algorithm is not ideal, the *Rate* of FDV should be increased appropriately. In general, in the face of different problems, different *Acc* and *Rate* parameters should be set.

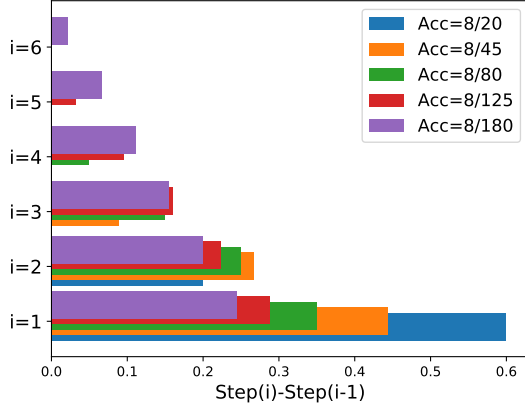


Fig. 4. The step length bar graph corresponding to the fuzzy evolution sub-stages under different *Acc* when *Rate* = 0.8.

The pseudo-code for the fuzzy evolution sub-stages division is given in Algorithm 2. To begin with, calculate the number of sub-stages of fuzzy evolution S . Next, initialize the fuzzy evolution step matrix $Step$, the value of $Step$ is obtained from lines 3-5. Finally, in order to ensure that the total step length of fuzzy evolution is equal to *Rate*, line 6 operation is required.

Algorithm 2 Fuzzy Evolution Sub-stages Division

Input: *Rate* (fuzzy evolutionary rate), *Acc* (step acceleration).

Output: $Step$ (cumulative step length of sub-stages).

- 1: $S \leftarrow \text{floor}(\text{sqrt}(2 * \text{Rate}/\text{Acc}))$; /*Formula (3)*/
 - 2: $Step \leftarrow \text{zeros}(1, S + 2)$; /*Initializes the zero matrix of $1 * (S + 2) * I$ */
 - 3: **for** $i = 0$; $i \leq S$; $i++$ **do**
 - 4: $Step(i) \leftarrow (S * i - i^2/2) * \text{Acc}$; /*Formula (4)*/
 - 5: **end for**
 - 6: $Step(S + 1) \leftarrow \text{Rate}$;
 - 7: **return** $Step$;
-

To verify the effectiveness of the proposed sub-stage division strategy, we conducted ablation studies. Control fuzzy operation and *Rate* = 0.8, change the sub-stage division strategy of fuzzy evolution, which are random division, uniform division, and the proposed division strategy. NSGA-II is respectively embedded in the FDV framework with these three division strategies, and comparative experiments are carried out on the DTLZ test suite. The algorithms corresponding to the three division strategies are Random-NSGA-II, Uniform-NSGA-II, and FDV-NSGA-II. In the setting of the test problem, the population size is 150, the objective number is 3, the number of decision variables is 1000, and the total number of evaluations is 400000. The results of the comparative experiment can be found in Table I of the supplementary file.

In Table I, FDV-NSGA-II shows an absolute advantage, and the uniform division strategy is better than the random division strategy. In general, no matter what sub-stage division strategy is used, the performance of the original algorithm can be improved, which shows that the fuzzy operation component is effective. Table I in the supplementary file shows that the proposed sub-stages division strategy has the best performance improvement ability for the algorithm, which proves that the fuzzy sub-stages division component is effective.

B. Fuzzy Operation

The entire fuzzy evolution stage is divided into multiple fuzzy evolution sub-stages with decreasing degrees of fuzzification. Therefore, how to determine the degree of fuzzification in a fuzzy evolution sub-stage and how an original solution is fuzzified will be introduced in this section. The knowledge of fuzzy sets and membership functions used in fuzzification is described in Section II. This paper proposes a series of formulas for fuzzy decision variables as follows:

$$\Gamma_n^1 = [10^i \cdot R_n^{-1} \cdot (X_n - X_n^l)] \cdot R_n \cdot 10^{-i} + X_n^l, \quad (5)$$

$$\Gamma_n^2 = [10^i \cdot R_n^{-1} \cdot (X_n - X_n^l)] \cdot R_n \cdot 10^{-i} + X_n^l, \quad (6)$$

$$\mu_{\tilde{A}_1}(X_n) = \frac{1}{X_n - \Gamma_n^1}, \quad (7)$$

$$\mu_{\tilde{A}_2}(X_n) = \frac{1}{\Gamma_n^2 - X_n}, \quad (8)$$

$$\tilde{A}_1 = \{(X_n, \mu_{\tilde{A}_1}(X_n) \mid n = 1, 2, \dots, D)\}, \quad (9)$$

$$\tilde{A}_2 = \{(X_n, \mu_{\tilde{A}_2}(X_n) \mid n = 1, 2, \dots, D)\}, \quad (10)$$

$$X'_n = \begin{cases} \Gamma_n^1, & \mu_{\tilde{A}_1}(X_n) > \mu_{\tilde{A}_2}(X_n) \\ \Gamma_n^2, & \mu_{\tilde{A}_1}(X_n) < \mu_{\tilde{A}_2}(X_n) \\ \text{rand}(\Gamma_n^1, \Gamma_n^2), & \mu_{\tilde{A}_1}(X_n) = \mu_{\tilde{A}_2}(X_n) \end{cases} \quad (11)$$

where, X represents a D-dimensional original solution vector; X' represents a D-dimensional fuzzy solution vector; X_n represents the n-th decision variable of the original solution X ; X_n^l represents the lower limit of the value of the n-th decision variable; i Represents the i-th fuzzy evolution sub-stage; R_n is the length of the value interval of the i-th decision variable. Γ_n^1 and Γ_n^2 are the fuzzy target value of the n-th decision variable; the n-th decision variable is either fuzzified into Γ_n^1 or is blurred into Γ_n^2 . \tilde{A}_1 and \tilde{A}_2 are fuzzy sets corresponding to two fuzzy target values. $\mu_{\tilde{A}_1}$ and $\mu_{\tilde{A}_2}$ are membership functions corresponding to two fuzzy sets. The following is the process of fuzzifying an original solution X into a fuzzy solution X' : Calculate the membership function values of X belonging to two fuzzy sets, and X will be fuzzified into the fuzzy target value corresponding to the fuzzy set with the larger degree of membership.

Although fuzzy operation involves related knowledge of fuzzy theory, its implementation process is simple and easy to understand. The procedure of the fuzzy operation is given

in Algorithm 3. The operations in Algorithm 3 are all matrix calculations. First, obtain the length of the decision variable value interval. Judge the current fuzzy evolution sub-stages (line2 and line3) according to the evolution rate $Iter$. Next, calculate the two fuzzy target values Γ^1 and Γ^2 of the decision variables. Γ^1 and Γ^2 correspond to fuzzy sets \tilde{A}_1 and \tilde{A}_2 respectively. Calculate the membership degree of decision variables in two fuzzy sets. Then, update the value of the decision variable according to the degree of membership. The update rule is that the value of the decision variable will be updated to the fuzzy target value corresponding to the fuzzy set with a larger membership degree. Finally, The program returns to the fuzzy solution. In particular, $logical$ is a boolean variable matrix, $find(1)$ returns true and $find(0)$ returns false.

Algorithm 3 Fuzzy Operation

Input: $Iter$ (evolutionary rate), X (original solution).

Output: X' (fuzzy solution).

```

1:  $R \leftarrow X^u - X^l$ ; /* $R$  is the range of  $X^*$ */
2: for  $i = 0$ ;  $i <= S$ ;  $i++$  do
3:   if  $Iter > Step(i)$  &&  $Iter < Step(i+1)$  then
4:     /*Formula (5) and Formula (6)*/
5:      $\Gamma^1 \leftarrow 10^{-i} * R * floor(10^i * R^{-1} * (X - X^l)) + X^l$ ;
6:      $\Gamma^2 \leftarrow 10^{-i} * R * ceil(10^i * R^{-1} * (X - X^l)) + X^l$ ;
7:     /*Formula (7) and Formula (8)*/
8:      $\mu_{\tilde{A}_1} \leftarrow 1/(X - \Gamma^1)$ ;
9:      $\mu_{\tilde{A}_2} \leftarrow 1/(\Gamma^2 - X)$ ;
10:    /*Formula (11)*/
11:    if  $\mu_{\tilde{A}_1} > \mu_{\tilde{A}_2}$  then
12:       $logical \leftarrow 1$ ;
13:    else
14:       $logical \leftarrow 0$ ;
15:    end if
16:     $X' \leftarrow \Gamma^2$ ;
17:     $X'(find(logical)) \leftarrow \Gamma^1(find(logical))$ ;
18:    return  $X'$ ;
19:  end if
20: end for

```

To describe the process of fuzzy operation more clearly, we have cited an example of a two-dimensional decision space to demonstrate the fuzzification process. In the two-dimensional decision space, the fuzzy decision variables can be regarded as the process of gridding. In Fig. 5, the process of fuzzification of the original solutions B_1 and B_2 into the fuzzy solution B' is as follows:

- 1) The fuzzy target values $\Gamma_a = [0.3, 0.1]$ and $\Gamma_b = [0.4, 0.2]$ of B_1 and the fuzzy target values $\Gamma_a = [0.4, 0]$ and $\Gamma_b = [0.5, 0.1]$ of B_2 can be obtained by formulas (5) and (6).
- 2) The fuzzy sets $\tilde{A}_1 = \{(0.38369, 11.95), (0.14369, 22.89)\}$ and $\tilde{A}_2 = \{(0.38369, 61.31), (0.14369, 17.76)\}$ corresponding to B_1 are obtained from formulas (7–10), and the fuzzy sets $\tilde{A}_1 = \{(0.43963, 25.23), (0.06963, 14.36)\}$ and $\tilde{A}_2 = \{(0.43963, 16.56), (0.06963, 32.93)\}$ corresponding to B_2 are obtained in the same way.

- 3) Using the original solution to update formula (11), B_1 and B_2 are blurred into $B' = [0.4, 0.1]$.

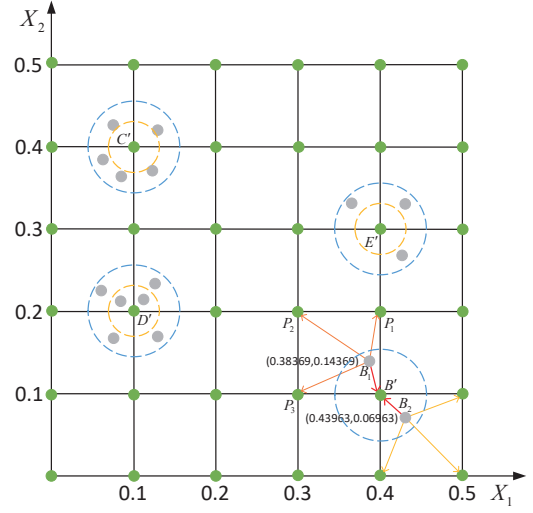


Fig. 5. The gridding process with $R_n \in [0, 1]$, $i = 1$. The blue point represents the fuzzy solution, and the gray point represents the original solution.

Each decision variable of the original solution has two fuzzy target values, so the number of target fuzzy solutions of this original solution is D^2 . Fig. 5 is an example of two-dimensional decision space, so an original solution has four objective fuzzy solutions. For example, the four target fuzzy solutions of the original solution B_1 are P_1, P_2, P_3, B' . Calculations from formulas (5–11) found that B_1 has the highest similarity with B' , so B_1 is blurred into B' . Similarly, the sixteen original solutions (the gray points in Fig. 5) in the decision space are blurred into four fuzzy solutions (B', C', D', E'). These four fuzzy solutions respectively represent an area (the blue circular area in Fig. 5). The solutions searched by FDV in the search space are these four fuzzy solutions instead of the sixteen original solutions. Therefore, FDV greatly reduces the search range of the algorithm in the decision space. It can be found from the figure that since the fuzzy solution in this area will replace the original solution in the blue circular area, the diversity of the evolutionary population will be lost. Therefore, the entire fuzzy evolution stage must be divided into multiple fuzzy evolution sub-stages with decreasing degrees of fuzzification. In the first fuzzy evolution sub-stage, D' can represent the blue area. In the second fuzzy evolution sub-stage, D' can only represent the yellow region. As the population evolves, the area represented by the fuzzy solution is getting smaller and smaller. In precise evolution, since there is no fuzzification operation, each solution can only represent itself. This paper exemplifies a two-dimensional decision space, but no image can be drawn in large-scale optimization, and it can only be abstracted into a hypercube. In a high-dimensional decision space, since the number of target fuzzy solutions of an original solution is enormous (with D^2), very few solutions will be blurred into the same fuzzy solution. Therefore, there is no need to worry about too much diversity loss caused by the obfuscation strategy. Since the degree of fuzzification gradually decreases to zero with the number of

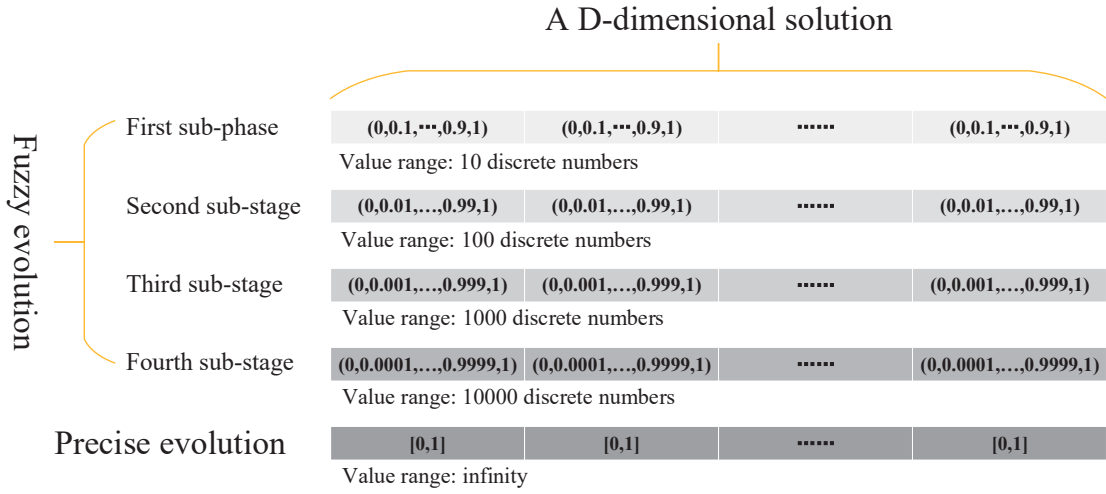


Fig. 6. The overall process of FDV solving large-scale optimization problems. The value range of decision variables $R_n \in [0, 1]$, the number of fuzzy evolution sub-stages $S = 4$.

evolutionary iterations, the diversity of the algorithm is further guaranteed.

C. An Example

Although FDV uses a small amount of fuzzy theory knowledge, its idea is very simple and effective. The overall process of solving large-scale optimization problems with FDV is shown in Fig. 6. As shown in Fig. 6, in the first sub-stage, the first decision variable is discussed. Because no matter what value the decision variable takes, it will be fuzzified into one of 10 discrete values. Therefore, the value of the decision variable is reduced from infinity to ten discrete values, which is equivalent to the algorithm only needing to search for these 10 numbers, which dramatically reduces the search range and improves the search efficiency. In the second sub-stage, no matter what value the decision variable takes, it will be fuzzified into one of 100 discrete values in the second sub-stage. Compared with the first sub-stage, the degree of fuzzification in the second sub-stage is reduced, so the value range of the decision variable is increased. Similarly, the degree of fuzzification in the third and fourth sub-stages gradually decreases, and the accuracy of the searched solution is better. Finally, it enters the stage of precise evolution and directly uses the original solution to evolve. In general, FDV focuses on convergence to search for rough solutions in the early stage and focuses on diversity to search for an accurate solution in the later stage.

IV. EMPIRICAL STUDIES

To empirically investigate whether the proposed FDV framework can improve the performance of existing MOEAs in solving large-scale MOPs. Four representatives MOEAs (i.e., NSGA-II [10], MOEA/D [25], CMOPSO [42], and LMOCSSO [21]) were embedded in FDV and compared with their original versions on the seven test problems of the DTLZ [35] test suite and the nine test problems of the LSMOP [36] test suite. In order to demonstrate that the

FDV framework has advantages compared with the current mainstream large-scale multiobjective optimization framework (i.e., MOEA/DVA [29], WOF [31], and LSMOF [32]). NSGA-II [10] was embedded in the WOF [31], LSMOF [32], and FDV frameworks and compared in the seven problems of the DTLZ test suite. Then, the CSO [34] embedded in the FDV framework (called FDVCSO) was compared with four state-of-the-art large-scale MOEAs (i.e., MOEA/DVA, WOF, LSMOF, and LMOCSSO) on the nine test problems of the LSMOP test suite. This comparative experiment can prove that FDVCSO has significant advantages over the four state-of-the-art large-scale MOEAs. Finally, in order to further study the performance of FDV on problems with irregular decision space. The CCMO [43] embedded in the FDV framework (called FDV-CCMO) was compared with four typical MOEAs (i.e., CMOPSO, MOEA/DVA, LMOCSSO, and WOF-SMPSO [44]) on the six test problems of the TREE [45] test suite.

In the remainder of this section, we present a brief introduction to the adopted performance indicator, and then give the parameter settings of the compared algorithms and FDV. Afterward, we report on the experiments in which each algorithm was run 20 times on each test problem independently, and the Wilcoxon rank-sum test [46] was used to compare the results obtained by the FDV and the other algorithms at a significance level of 0.05. Symbols “+,” “-,” and “ \approx ” indicate the compared algorithm is significantly better than, significantly worse than, or statistically tied by FDV.

A. Performance Indicator

In the experiments, a widely used performance indicator, the inverted generational distance (IGD) [47] was adopted for evaluating the performance of the algorithms.

Suppose that P^* is a set of evenly distributed reference points on the PF and Ω is the set of obtained nondominated solutions, the IGD is defined as follows:

$$\text{IGD}(P^*, \Omega) = \frac{\sum_{\mathbf{x} \in P^*} \text{dis}(\mathbf{x}, \Omega)}{|P^*|}, \quad (12)$$

where $dis(x, \Omega)$ is the minimum Euclidean distance between x and points in Ω , and $|P^*|$ is the number of elements in P^* . A smaller value of IGD will indicate a better performance of the algorithm. In this paper, the size of P^* is set to 10,000 (or a close number) for the IGD calculations.

In this paper, we use the IGD indicator instead of the HV [48] and GD [49] indicator since the PFs of DTLZ, LSMOP, and TREE test problems are relatively simple and regular. Meanwhile, the reference solution sets in PlatEMO [50] are evenly sampled, which enables the IGD indicator to well assess the qualities of the obtained solution sets.

B. Experimental Settings

We adopted the recommended parameter settings for the compared algorithms that have achieved the best performance as reported in the literature for fair comparisons.

1) *Algorithm Reproduction Operators*: In this experiment, GA operators were used in NSGA-II, MOEA/D, WOF-NSGA-II and LS-NSGA-II. The GA operator used simulated binary crossover (SBX) [51] and polynomial mutation (PM) [52]. The distribution index of the crossover was set to $n_c = 20$; the distribution index of the mutation was set to $n_m = 20$; the crossover probability p_c was set to 1.0, and the mutation probability p_m was set to $1/D$, where D is the number of decision variables. In MOEA/DVA and CCMO, the DE operator [53] and PM were used to generate offspring, with the control parameters set to $CR = 1$, $F = 0.5$, $p_m = 1/D$ and $\eta = 20$. In CMOPSO and SMPSO, the PSO [33] and PM were used. As for LMOCSO, the CSO [34] and PM were used.

2) *Algorithm Parameter Settings*: In MOEA/D, the aggregate function adopts the boundary intersection approach (PBI) [25]. In MOEA/DVA, the number of sampling solutions to recognize the control properties of the decision variables NCA was set to 20, and the maximum number of trials required to judge the interaction between two variables NIA was set to 6. In WOF, the number of evaluations for the original problem $t1$ was set to 1000. The number of evaluations for transformed problem $t2$ was set to 500; the number of chosen solutions to do weight optimization q was set to $M + 1$, and the number of groups γ was set to 4. In LSMOP, the number of reference solutions r was set to 10; the population size of the transferred problem $SubN$ was set to 30.

3) *Problem Settings*: All algorithms are tested on DTLZ, LSMOP, and TREE test suites. Table I shows the parameter settings of the test problem. N is the population size, M is the objective space dimension, D is the decision space dimension, and FES is the total number of function evaluations.

C. Parameter Analysis

Rate and *Acc* are two parameters in the fuzzy evolution sub-stages division, as well as parameters of the entire FDV framework. Different parameter selections will have an impact on the performance of the algorithm. For different MOPs, there may be different optimal parameters. Table II is the corresponding table of parameters *Rate*, *Acc* and the number of fuzzy evolution sub-stages S . In each row see that under

TABLE I
SET THE POPULATION SIZE AND THE NUMBER OF THE OBJECTIVE FUNCTION, DECISION VARIABLES, AND FES FOR EACH TEST INSTANCE.

Problems	N	M	D	FES
DTLZ 1-7	100	2	500	D*500
	—————		1500	
	150	3	2000	D*800
			500	250000
LSMOP 1-9	100	2	1000	300000
	—————		1500	350000
	150	3	2000	400000
			5000	500000
TREE 1-5	100	2	3000	60000
	—————		6000	80000
	150	3	12000	100000

TABLE II
THE IDEAL *Acc* CORRESPONDING TO *Rate* IN ACCELERATING STEP LENGTH SEGMENTATION.

Rate	Acc						S					
0.1	1/5	1/20	1/45	1/80	1/125	1/180	1	2	3	4	5	6
0.2	2/5	2/20	2/45	2/80	2/125	2/180	1	2	3	4	5	6
0.3	3/5	3/20	3/45	3/80	3/125	3/180	1	2	3	4	5	6
0.4	4/5	4/20	4/45	4/80	4/125	4/180	1	2	3	4	5	6
0.5	1	5/20	5/45	5/80	5/125	5/180	1	2	3	4	5	6
0.6	6/5	6/20	6/45	6/80	6/125	6/180	1	2	3	4	5	6
0.7	7/5	7/20	7/45	7/80	7/125	7/180	1	2	3	4	5	6
0.8	8/5	8/20	8/45	8/80	8/125	8/180	1	2	3	4	5	6
0.9	9/5	9/20	9/45	9/80	9/125	9/180	1	2	3	4	5	6

the same *Rate*, S increases with the decrease of *Acc*. It can be found from each column that in order to get the same S , *Acc* increases with increasing *Rate*. For problems that are difficult to converge, *Acc* should be increased appropriately to improve the convergence and computational efficiency of the algorithm. Faced with the problem of discontinuous PFs, *Acc* should be appropriately reduced to ensure the diversity of the algorithm. Theoretically, the setting of *Rate* increases as the dimension of decision variables rises.

In this section, the results of experiments studying algorithms on the DTLZ and LSMOP test problems are presented. In order to find suitable parameter settings, parameter analysis was conducted. In Fig. 7, the basic rule is that the larger the *Rate*, the smaller the average IGD. Careful observation found that when *Rate* was set to 0.7, 0.8, and 0.9, the performance of the algorithm was relatively good. When the *Rate* did not change, as the *Acc* increased, the average IGD had a downward trend, but there is a minimum point. In summary, the parameters of FDV on the DTLZ problem were set as $Rate = 0.8$ and $Acc = 0.4$, the parameters of FDV on the

TABLE III
STATISTICS OF IGD RESULTS OBTAINED BY EIGHT COMPARED ALGORITHMS ON 42 TEST INSTANCES FROM DTLZ TEST SUITE.
THE BEST RESULTS IN EACH TWO COLUMNS ARE HIGHLIGHTED.

Problem	M	D	NSGA-II	FDV-NSGA-II	MOEA/D	FDV-MOEA/D	CMOPSO	FDV-CMOPSO	LMOCSO	FDV-LMOCSO
DTLZ1	2	500	1.1126e+3 (1.86e+1)	2.9126e-3 (1.02e-4)	1.2031e+3 (6.32e+1)	2.0887e-2 (3.38e-2)	3.2009e+3 (2.31e+2)	1.8387e-3 (8.03e-6)	4.2730e+3 (9.37e+1)	1.0781e-2 (7.65e-6)
		1000	2.4816e+3 (4.09e+1)	2.9111e-3 (1.31e-4)	2.5270e+3 (9.81e+1)	2.7078e-3 (1.81e-3)	6.7595e+3 (1.09e+2)	1.8482e-3 (1.23e-5)	8.7660e+3 (3.71e+1)	1.7856e-3 (4.77e-8)
		1500	4.5758e+3 (1.15e+2)	6.1361e-2 (1.43e-1)	3.9498e+3 (2.53e+2)	8.6292e-3 (1.04e-2)	1.0113e+4 (3.92e+2)	1.8352e-3 (8.24e-6)	1.2200e+4 (2.55e+3)	1.7856e-3 (4.77e-8)
	3	500	1.2774e+3 (7.88e+1)	2.2882e-2 (1.27e-3)	1.2528e+3 (7.27e+1)	1.0725e-2 (1.68e-6)	3.7911e+3 (1.86e+2)	1.1228e-2 (5.14e-5)	3.3673e+3 (6.25e+2)	1.0781e-2 (7.65e-6)
		1000	2.8331e+3 (2.73e+1)	2.3138e-2 (1.80e-3)	2.4738e+3 (1.16e+2)	1.0725e-2 (2.02e-7)	8.3767e+3 (2.85e+2)	1.1261e-2 (6.70e-5)	6.8594e+3 (1.10e+3)	1.0756e-2 (4.35e-6)
		1500	4.6069e+3 (1.22e+2)	2.4035e-2 (1.59e-3)	3.6774e+3 (1.04e+2)	1.0725e-2 (1.95e-7)	1.3987e+4 (3.74e+2)	1.1291e-2 (3.65e-5)	1.0636e+4 (1.29e+3)	1.0745e-2 (3.11e-6)
DTLZ2	2	500	1.0500e-2 (3.15e-4)	5.0925e-3 (1.22e-4)	1.1981e-2 (8.72e-4)	3.9703e-3 (1.50e-5)	1.3883e-2 (5.80e-4)	4.5261e-3 (7.81e-5)	4.7919e-2 (3.04e-3)	3.9693e-3 (2.08e-6)
		1000	1.9046e-2 (8.34e-4)	5.1771e-3 (2.59e-4)	2.0984e-2 (8.84e-4)	3.9659e-3 (5.94e-9)	2.6167e-2 (7.96e-4)	4.5616e-3 (1.02e-4)	1.1191e-1 (9.31e-3)	3.9680e-3 (1.10e-6)
		1500	4.6069e+3 (1.22e+2)	5.3616e-3 (1.53e-4)	2.9533e-2 (2.73e-3)	3.9659e-3 (3.89e-9)	4.0781e-2 (2.21e-3)	4.5616e-3 (5.45e-5)	1.8137e-1 (1.78e-2)	3.9665e-3 (1.30e-7)
	3	500	7.4091e-2 (2.88e-3)	5.7812e-2 (1.93e-3)	3.6914e-2 (2.07e-3)	2.8548e-2 (3.79e-7)	1.0181e-1 (6.76e-3)	3.8454e-2 (1.01e-3)	2.3026e-1 (3.22e-2)	2.8731e-2 (6.98e-5)
		1000	1.0707e-1 (6.71e-3)	5.7528e-2 (1.40e-3)	3.5902e-2 (1.06e-3)	2.8549e-2 (1.76e-7)	2.1274e-1 (7.87e-3)	3.5652e-2 (1.53e-3)	4.1002e-1 (3.24e-2)	2.8644e-2 (3.23e-5)
		1500	1.6137e-1 (3.57e-3)	5.7059e-2 (1.68e-3)	3.6250e-2 (1.50e-3)	3.3576e-2 (7.43e-3)	1.1338e+0 (2.47e-1)	5.6536e-2 (8.80e-3)	5.4509e-1 (6.02e-2)	4.2430e-2 (4.35e-3)
DTLZ3	2	500	2.7553e+3 (7.46e+1)	5.6627e-3 (2.57e-4)	3.5709e+3 (1.94e+2)	1.1577e+0 (3.66e+0)	9.0701e+3 (4.38e+2)	4.1107e-3 (3.21e-5)	8.7863e+3 (2.77e+3)	3.9686e-3 (1.15e-6)
		1000	6.3381e+3 (1.02e+2)	5.7604e-3 (4.05e-4)	7.2571e+3 (3.03e+2)	9.7520e-2 (1.87e-1)	1.8677e+4 (5.62e+2)	4.1007e-3 (1.08e-5)	2.1949e+4 (5.58e+3)	3.9666e-3 (6.05e-8)
		1500	1.2133e+4 (2.50e+2)	5.7532e-3 (1.96e-4)	1.1216e+4 (6.83e+2)	5.8338e-3 (4.82e-3)	2.8243e+4 (1.42e+3)	4.1074e-3 (3.52e-5)	2.3260e+4 (1.49e+4)	3.9664e-3 (1.17e-7)
	3	500	3.6143e+3 (2.23e+2)	5.7928e-2 (3.19e-3)	4.3289e+3 (1.77e+2)	2.8547e-2 (8.97e-7)	1.1517e+4 (7.05e+2)	3.5364e-2 (2.09e-3)	7.8581e+3 (3.12e+3)	2.8643e-2 (3.93e-5)
		1000	8.4803e+3 (1.52e+2)	7.1604e-2 (4.18e-3)	8.5928e+3 (2.79e+2)	2.8548e-2 (3.17e-7)	2.7000e+4 (1.18e+3)	9.4845e-2 (2.04e-2)	1.1826e+4 (6.67e+3)	4.2406e-2 (4.35e-3)
		1500	1.3075e+4 (3.51e+2)	5.7706e-2 (1.20e-3)	1.2953e+4 (9.93e+2)	4.2374e-2 (4.35e-3)	4.0399e+4 (1.24e+3)	6.8595e-2 (1.17e-2)	1.6536e+4 (6.94e+3)	4.2394e-2 (4.35e-3)
DTLZ4	2	500	1.3246e-1 (2.99e-1)	5.0970e-3 (1.19e-4)	1.9447e-1 (3.30e-1)	3.9659e-3 (9.26e-9)	3.7890e-1 (3.79e-1)	6.6096e-2 (2.13e-1)	4.0057e-1 (3.57e-1)	3.9791e-3 (1.58e-5)
		1000	2.0736e-2 (9.14e-4)	5.3287e-3 (3.53e-4)	1.9809e-1 (3.63e-1)	3.7303e-1 (4.26e-1)	2.0911e-1 (3.55e-1)	1.8902e-1 (3.34e-1)	5.6626e-1 (2.05e-1)	4.1440e-3 (2.71e-4)
		1500	3.2591e-1 (3.32e-1)	5.2280e-3 (1.34e-4)	2.8945e-2 (1.26e-2)	2.1486e-1 (3.60e-1)	3.4569e-1 (3.71e-1)	1.2755e-1 (2.87e-1)	5.7564e-1 (1.97e-1)	3.9672e-3 (7.14e-7)
	3	500	7.3021e-2 (4.90e-3)	5.7157e-2 (1.58e-3)	2.1362e-1 (2.42e-1)	1.1254e-1 (2.63e-1)	4.1787e-1 (6.35e-2)	8.2826e-2 (1.63e-2)	3.1595e-1 (2.48e-1)	2.5552e-1 (1.33e-1)
		1000	1.0931e-1 (8.68e-3)	7.1074e-2 (2.40e-3)	3.2484e-1 (3.60e-1)	8.3833e-2 (1.44e-1)	2.2760e+0 (5.74e-1)	1.8548e-1 (2.44e-1)	5.4175e-1 (2.24e-1)	2.0880e-1 (1.55e-1)
		1500	1.6607e-1 (3.02e-3)	5.8364e-2 (1.66e-3)	3.0420e-1 (2.48e-1)	1.1575e-1 (2.61e-1)	2.5566e+0 (1.91e+0)	1.3225e-1 (7.04e-2)	6.6681e-1 (5.01e-1)	1.3915e-1 (1.42e-1)
DTLZ5	2	500	1.0749e-2 (4.17e-4)	5.0408e-3 (1.20e-4)	1.1977e-2 (9.26e-4)	3.9659e-3 (9.77e-9)	1.3966e-2 (6.89e-4)	4.4929e-3 (6.07e-5)	5.1290e-2 (4.64e-3)	3.9687e-3 (1.30e-6)
		1000	1.9871e-2 (3.42e-4)	5.2994e-3 (2.11e-4)	2.1276e-2 (2.21e-3)	3.9659e-3 (1.01e-9)	2.7095e-2 (1.32e-3)	4.6049e-3 (1.14e-4)	1.0362e-1 (1.12e-2)	3.9670e-3 (7.77e-7)
		1500	1.0273e-1 (4.79e-3)	5.1771e-3 (2.57e-4)	3.0189e-2 (1.98e-3)	3.9659e-3 (3.86e-9)	4.0760e-2 (1.41e-3)	4.5452e-3 (8.99e-5)	1.8404e-1 (4.51e-3)	3.9665e-3 (1.31e-7)
	3	500	1.7013e-2 (1.21e-3)	3.6950e-3 (7.36e-5)	2.6898e-2 (2.77e-4) \approx	2.7099e-2 (3.02e-5)	2.2050e-1 (2.33e-2)	3.7750e-3 (4.78e-4)	4.1423e-2 (4.40e-3)	2.6957e-2 (1.39e-3)
		1000	4.7955e-2 (3.95e-3)	3.6747e-3 (1.23e-4)	2.7832e-2 (2.45e-3) \approx	2.6355e-2 (2.72e-3) \approx	5.1090e+0 (1.48e+0)	6.346e-3 (4.2e-4)	9.8018e-2 (2.14e-2)	2.6235e-2 (2.93e-3)
		1500	9.6502e-2 (4.07e-3)	3.7351e-3 (1.99e-4)	3.1577e-2 (8.14e-4) \approx	2.7139e-2 (8.65e-6) \approx	1.8553e+1 (2.55e+0)	3.6143e-3 (4.88e-4)	2.0160e-1 (4.05e-2)	2.6703e-2 (7.25e-4)
DTLZ6	2	500	1.0763e+2 (5.73e+0)	8.9118e-1 (1.58e+0)	1.5355e+2 (1.17e+1)	3.9659e-3 (1.50e-8)	3.4185e+1 (7.04e+0)	4.1289e-3 (3.31e-5)	1.0180e-2 (2.15e-2)	4.4033e-2 (2.08e-4)
		1000	3.4192e+2 (9.45e+0)	1.1539e+1 (3.16e+0)	3.4969e+2 (2.83e+1)	3.9659e-3 (4.58e-9)	1.1801e+2 (2.68e+1)	4.1059e-3 (1.28e-5)	3.9661e-3 (1.27e-7) $+$	4.4669e-3 (1.69e-5)
		1500	7.4892e+2 (2.80e+1)	2.8312e+2 (1.83e+1)	5.7591e+2 (3.06e+1)	3.9659e-3 (7.81e-9)	3.1763e+2 (4.01e+1)	4.1067e-3 (3.19e-5)	3.9660e-3 (2.01e-8) $+$	4.4379e-3 (3.77e-5)
	3	500	2.0805e+2 (6.77e+0)	3.9179e-3 (1.68e-4)	1.5124e+2 (9.80e+0)	2.7140e-2 (5.55e-5)	2.4537e+2 (7.82e+0)	2.7986e-3 (2.81e-5)	6.2272e+1 (1.31e+1)	1.3671e-1 (3.80e-1)
		1000	6.3841e+2 (7.43e+0)	1.2113e+0 (2.63e+0)	3.4064e+2 (2.56e+1)	2.6351e-2 (3.72e-3)	5.1266e+2 (3.47e+1)	2.6650e-3 (4.03e-4)	1.0348e+2 (3.74e+1)	2.5929e-2 (3.04e-3)
		1500	9.4134e+2 (1.48e+1)	3.1147e+1 (8.70e+0)	5.3689e+2 (2.14e+1)	2.0657e+0 (3.34e+0)	7.9494e+2 (4.23e+1)	2.7867e-3 (2.64e-5)	1.1712e+2 (2.79e+1)	2.4349e+0 (6.70e+0)
DTLZ7	2	500	1.1448e-2 (8.92e-4)	5.4529e-3 (1.50e-4)	3.0657e-1 (8.30e-2)	6.3862e-2 (2.29e-2)	9.4457e-3 (5.34e-4)	4.5901e-3 (6.76e-5)	2.1131e+0 (1.78e-1)	6.5873e-3 (9.30e-6)
		1000	2.3116e-2 (1.61e-3)	5.4598e-3 (2.04e-4)	6.4505e-1 (1.09e-1)	1.1038e-1 (5.35e-2)	1.2199e-2 (6.08e-4)	4.5534e-3 (3.07e-5)	2.6980e+0 (1.97e-1)	6.5912e-3 (4.23e-6)
		1500	4.4643e-1 (4.48e-2)	7.8549e-3 (1.10e-3)	7.3424e-1 (1.16e-1)	1.0338e-1 (7.30e-2)	1.5019e-2 (1.95e-3)	4.5582e-3 (2.19e-5)	2.6604e+0 (1.44e-1)	6.5844e-3 (6.18e-6)
	3	500	9.4943e-2 (6.58e-3)	6.0166e-2 (3.61e-3)	1.0698e-1 (1.42e-4) \approx	1.0711e-1 (1.37e-4)	1.9148e-1 (2.59e-2)	8.2184e-2 (4.02e-3)	3.8368e+0 (2.69e-1)	3.5988e-1 (1.05e-1)
		1000	1.5283e-1 (1.20e-2)	6.0921e-2 (3.50e-3)	1.0506e-1 (6.72e-3)	1.0519e-1 (6.74e-3)	3.1201e-1 (3.44e-2)	1.2603e-1 (5.17e-3)	4.2590e+0 (4.49e-1)	3.5789e-1 (6.49e-2)
		1500	2.5841e-1 (2.82e-2)	5.8921e-2 (2.36e-3)	1.0696e-1 (8.14e-5) \approx	1.0713e-1 (3.44e-5)	4.1581e-1 (3.16e-2)	1.6978e-1 (8.00e-3)	4.6486e+0 (2.06e-1)	3.0810e-1 (5.43e-2)
+/ - / \approx		0/42/0		0/38/4		0/42/0		2/40/0		

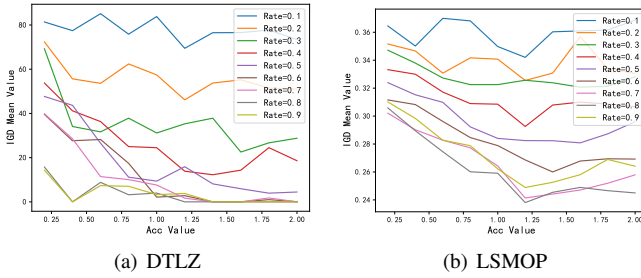


Fig. 7. Under different *Rate* and *Acc*, the IGD mean value graph obtained by the algorithm.

LSMOP problem were set as *Rate* = 0.8 and *Acc* = 1.2. It is worth noting that when *Rate* > 0.6 and *Acc* > 1.2, the parameters had little effect on the performance of FDV, which shows that FDV is robust. It is worth mentioning that even if the parameters were set arbitrarily, the original algorithm greatly improved.

D. General Performance

To study the influence of FDV on different types of MOEAs, we first confirmed that the proposed FDV framework could improve the performance of the algorithm. NSGA-II, MOEA/D, CMOPSO, and LMOCSO were embedded into the FDV framework, called FDV-*Alg*. FDV-*Alg* is pairwise compared

with the original *Alg*. The experimental results obtained by these comparative algorithms are shown in Table III.

Table III shows that FDV-*Alg* has absolute advantages over the original performance of the algorithm. FDV-NSGA-II and FDV-CMOPSO outperformed the corresponding original algorithms on all 42 test instances. FDV-MOEA/D is significantly better than MOEA/D in 38 of 42 test instances. In the DTLZ5 problem with tri-objective and 500 decision variables and the DTLZ7 problem with tri-objective, these algorithms' performance difference is minimal. In the 42 test instances, FDV-LMOCSO is significantly better than LMOCSO in 40 test instances. In the bi-objective DTLZ6 problems, when the decision variables were 1000 and 1500, the performance was inferior to LMOCSO, but the gap was minimal. In a further experiment, the statistics of IGD results obtained by eight compared algorithms on the LSMOP test suite are given in Table II of the supplementary file.

In order to clearly show that the FDV framework can significantly improve the performance of MOEAs, the nondominated solution diagrams on DTLZ1, DTLZ3, and DTLZ6 with tri-objective are given. As shown in Fig. 8, MOEAs without the embedded FDV framework have very poor convergence, resulting in a result very far from the true PF. The algorithm embedded in FDV, namely FDV-*Alg*, significantly improved the algorithm's convergence, thereby obtaining the problem's

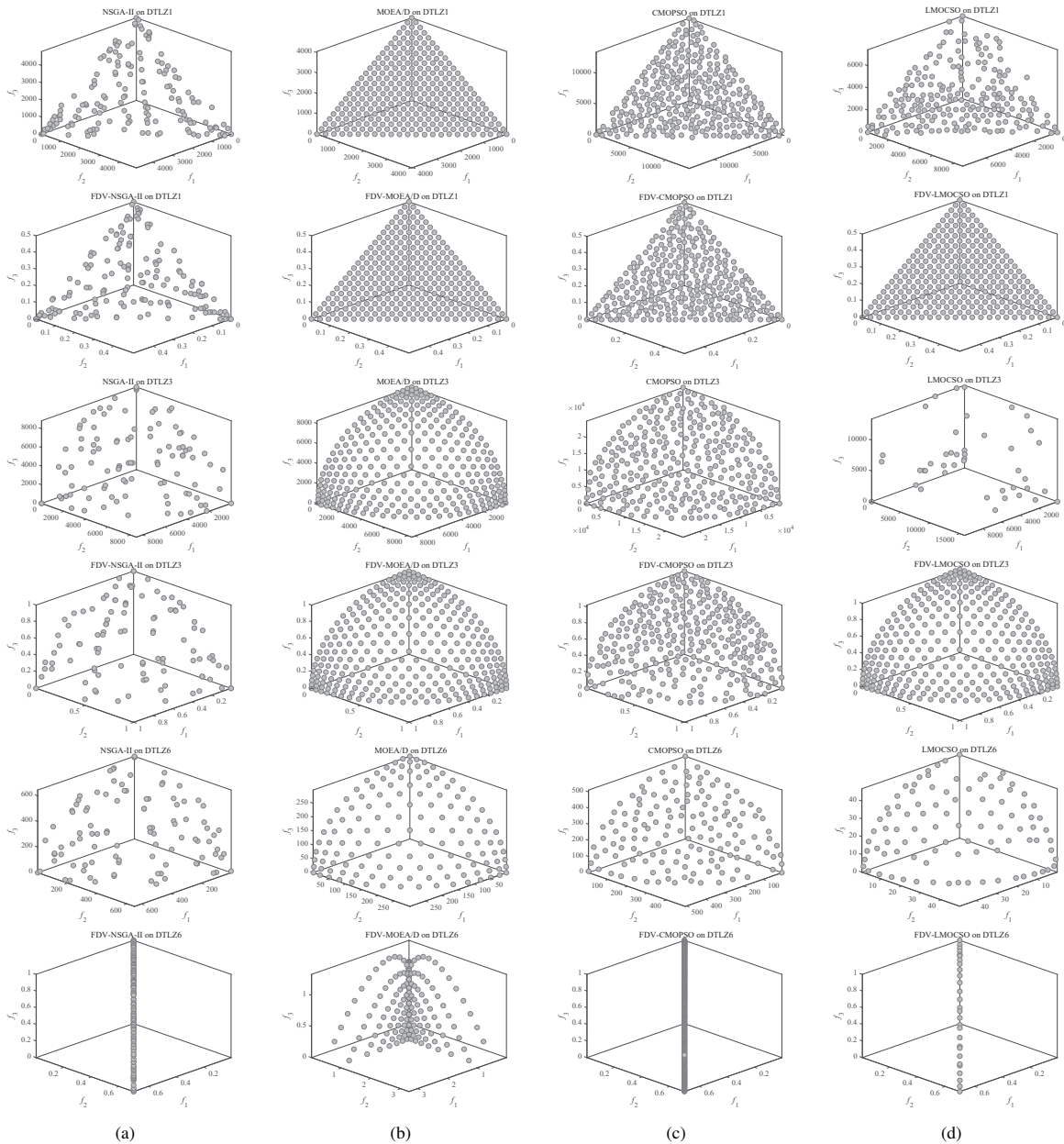


Fig. 8. Nondominated solutions obtained by NSGA-II, FDV-NSGA-II, MOEA/D, FDV-MOEA/D, CMOPSO, FDV-CMOPSO, LMOCSO, FDV-LMOCSO on tri-objective DTLZ1, DTLZ3, and DTLZ6 with 1000 decision variables.

optimal global solution. We see in Fig. 8 that *FDV-Alg* also has a huge advantage over the original algorithm. It can be noticed that the effect of FDV-MOEA/D on the DTLZ6 problem is not as good as the other *FDV-Alg*. This is because MOEA/D is a decomposition-based algorithm and relies on the weight vector to update the individual, so there will be a particular disadvantage for the problem of irregular PFs. Similarly, FDV has significant advantages in the bi-objective DTLZ (refer to Fig. 1 in the supplementary File).

E. Compared With Typical and State-of-the-art Frameworks

This section compares our proposed FDV with the three other most representative and state-of-the-art large-scale MOEA frameworks (i.e., MOEA/DVA, WOF, and LSMOF).

In WOF, LSMOF, and FDV, NSGA-II is embedded for a fair comparison. Table IV shows the statistics of IGD results obtained by MOEA/DVA, WOF-NSGA-II, LS-NSGA-II, and FDV-NSGA-II.

Table IV shows that the performance of MOEA/DVA in 42 test instances of the seven problems is relatively poor compared to the three other algorithms. This is because MOEA/DVA has a failed decision variable analysis, which leads to poor decision-making. Therefore, the variable grouping is wrong, especially the results obtained on DTLZ1, DTLZ3, and DTLZ6, which are far from the true PF. WOF-NSGA-II showed better performance in 3 of 42 test instances, mainly on the DTLZ6 problem with a higher dimension of decision variables. For LS-NSGA-II, it can be seen that the algorithm based on the problem transformation mechanism is

TABLE IV
STATISTICS OF IGD RESULTS ACHIEVED BY FOUR COMPARED ALGORITHMS ON 42 TEST INSTANCES FROM DTLZ TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	MOEA/DVA	WOF-NSGA-II	LS-NSGA-II	FDV-NSGA-II
DTLZ1	2	500	9.4384e+3 (2.36e+2) –	1.4727e+3 (4.60e+2) –	2.8880e-3 (3.06e-4) ≈	2.8844e-3 (1.41e-4)
		1000	2.0615e+4 (2.27e+2) –	2.4203e+3 (1.02e+3) –	3.0099e-3 (9.58e-5) ≈	2.8219e-3 (1.11e-4)
		1500	3.1093e+4 (5.34e+2) –	5.2054e+3 (1.18e+3) –	2.8957e-3 (1.34e-4) ≈	2.8199e-3 (1.33e-4)
	3	500	6.9543e+3 (1.99e+2) –	1.9339e+3 (5.87e+2) –	2.3692e-2 (1.13e-3) ≈	2.3532e-2 (1.02e-3)
		1000	1.4829e+4 (1.53e+2) –	2.9428e+3 (1.24e+3) –	2.3988e-2 (8.52e-4) ≈	2.3657e-2 (1.92e-3)
		1500	2.3172e+4 (2.77e+2) –	3.7675e+3 (2.74e+3) –	2.3255e-2 (1.30e-3) ≈	2.4418e-2 (1.37e-3)
DTLZ2	2	500	1.5805e+1 (3.17e-1) –	1.8259e-2 (2.10e-3) –	5.3947e-3 (1.72e-4) –	5.1181e-3 (1.72e-4)
		1000	3.4320e+1 (7.50e-1) –	3.4045e-2 (5.13e-3) –	5.2970e-3 (2.10e-4) ≈	5.1247e-3 (1.74e-4)
		1500	5.3505e+1 (7.87e-1) –	4.2799e-2 (4.42e-3) –	5.1429e-3 (1.20e-4) ≈	5.1369e-3 (1.04e-4)
	3	500	1.3020e+1 (3.47e-1) –	9.0101e-2 (1.15e-2) –	1.5509e-1 (2.26e-1) –	5.7442e-2 (1.60e-3)
		1000	2.9412e+1 (3.38e-1) –	1.3657e-1 (3.18e-2) –	6.1530e-2 (1.92e-3) –	5.5672e-2 (1.51e-3)
		1500	4.6306e+1 (2.55e-1) –	1.5913e-1 (1.89e-2) –	4.4523e-1 (9.45e-1) ≈	5.7905e-2 (1.54e-3)
DTLZ3	2	500	2.5714e+4 (8.31e+2) –	3.9358e+3 (1.41e+3) –	5.6762e-3 (3.02e-4) ≈	5.7198e-3 (2.95e-4)
		1000	5.5727e+4 (5.58e+2) –	6.9832e+3 (2.55e+3) –	5.9635e-3 (3.22e-4) ≈	5.7149e-3 (2.84e-4)
		1500	8.6068e+4 (1.09e+3) –	9.2958e+3 (4.75e+3) –	5.8083e-3 (3.13e-4) ≈	5.6973e-3 (3.43e-4)
	3	500	2.2281e+4 (4.38e+2) –	5.9001e+3 (1.70e+3) –	5.7826e-2 (1.62e-3) ≈	5.8959e-2 (2.59e-3)
		1000	4.9797e+4 (6.74e+2) –	7.3739e+3 (2.29e+3) –	5.6902e-2 (2.84e-3) ≈	5.6284e-2 (2.80e-3)
		1500	7.7417e+4 (1.38e+3) –	8.3960e+3 (1.88e+3) –	5.9823e-2 (2.76e-3) ≈	5.6269e-2 (1.82e-3)
DTLZ4	2	500	1.6134e+1 (3.40e-1) –	7.8959e-2 (2.09e-1) –	5.0968e-3 (1.34e-4) +	5.2489e-3 (1.32e-4)
		1000	3.4610e+1 (6.06e-1) –	3.3842e-2 (5.61e-3) –	5.0825e-3 (1.11e-4) ≈	5.2060e-3 (1.68e-4)
		1500	5.3556e+1 (7.50e-1) –	3.8318e-2 (6.00e-3) –	5.3357e-3 (2.82e-4) ≈	5.1227e-3 (1.52e-4)
	3	500	1.3069e+1 (5.15e-1) –	9.0918e-2 (7.64e-3) –	2.3171e-1 (1.56e-1) –	5.6393e-2 (1.46e-3)
		1000	2.9770e+1 (5.95e-1) –	9.3768e-2 (1.80e-2) –	8.2996e-1 (1.34e+0) ≈	5.6845e-2 (2.64e-3)
		1500	4.5955e+1 (5.90e-1) –	1.3997e-1 (3.06e-2) –	2.3320e+0 (3.61e+0) ≈	5.6841e-2 (1.19e-3)
DTLZ5	2	500	1.5882e+1 (3.19e-1) –	1.7684e-2 (3.12e-3) –	5.3843e-3 (2.45e-4) –	5.1269e-3 (1.60e-4)
		1000	3.4424e+1 (6.92e-1) –	3.2289e-2 (5.93e-3) –	5.1651e-3 (1.99e-4) ≈	5.1397e-3 (2.04e-4)
		1500	5.3750e+1 (6.68e-1) –	4.5238e-2 (6.37e-3) –	5.2806e-3 (3.69e-4) ≈	5.1608e-3 (1.15e-4)
	3	500	1.2930e+1 (2.03e-1) –	3.2745e-2 (8.05e-3) –	4.4177e-3 (1.94e-4) –	3.6624e-3 (1.35e-4)
		1000	2.8753e+1 (3.81e-1) –	7.7057e-2 (1.63e-2) –	4.2109e-3 (1.31e-4) ≈	3.7481e-3 (6.98e-5)
		1500	4.5553e+1 (5.68e-1) –	1.0020e-1 (2.15e-2) –	3.9968e-3 (1.87e-4) –	3.7266e-3 (1.54e-4)
DTLZ6	2	500	4.1587e+2 (1.00e+0) –	1.1328e+0 (7.00e-1) –	5.9933e-3 (2.36e-4) +	5.7777e-1 (7.66e-1)
		1000	8.4059e+2 (1.12e+0) –	1.3601e-2 (9.29e-3) +	5.9442e-3 (2.36e-4) +	1.8029e+1 (5.18e+0)
		1500	1.2668e+3 (1.43e+0) –	5.1993e-3 (2.61e-4) +	5.9150e-3 (2.49e-4) +	5.3090e+1 (1.63e+1)
	3	500	4.0900e+2 (5.00e-1) –	8.1871e-1 (6.16e-1) –	3.9915e-3 (2.25e-4) ≈	3.9764e-3 (2.25e-4)
		1000	8.2785e+2 (2.57e+0) –	2.3253e-2 (1.31e-2) ≈	4.0414e-3 (1.05e-4) ≈	3.1331e+0 (3.18e+0)
		1500	1.2500e+3 (1.67e+0) –	4.8579e-3 (1.45e-3) +	4.0295e-3 (2.23e-4) +	3.4277e+1 (9.24e+0)
DTLZ7	2	500	4.0377e+0 (9.69e-2) –	3.6996e-1 (1.70e-1) –	4.4290e-1 (3.41e-5) –	5.3154e-3 (1.82e-4)
		1000	4.2865e+0 (5.03e-2) –	3.6997e-1 (1.79e-1) –	4.4290e-1 (3.28e-5) –	5.4419e-3 (1.06e-4)
		1500	4.3835e+0 (3.49e-2) –	4.4289e-1 (3.47e-5) –	4.4289e-1 (3.36e-5) –	5.3592e-3 (1.64e-4)
	3	500	5.0510e+0 (1.85e-1) –	6.1384e-1 (3.33e-1) –	7.9770e-1 (4.66e-4) –	6.0361e-2 (2.51e-3)
		1000	5.5032e+0 (7.81e-2) –	7.9778e-1 (8.08e-4) –	7.9813e-1 (1.12e-3) –	6.0412e-2 (3.47e-3)
		1500	5.5924e+0 (1.31e-1) –	7.9828e-1 (3.89e-4) –	7.9790e-1 (6.08e-4) –	6.1651e-2 (2.82e-3)
+ / - / ≈			0/42/0	3/38/1	6/15/21	

more suitable for solving the DTLZ test problem than the algorithm based on the decision variable analysis. This is because LS-NSGA-II reduces the dimensionality of a high-dimensional multi-objective problem to a low-dimensional single-objective problem, thereby improving the performance of the algorithm. However, FDV-NSGA-II demonstrated obvious advantages in most test instances.

F. Compared With State-of-the-art Large-scale MOEAs

CSO is a new search operator used for large-scale optimization and is called FDVCSO when it is embedded in FDV. Five algorithms (MOEA/DVA, WOF, LSMOF, LMOCSO, and FDVCSO) were compared on the LSMOP test suite. Table V shows the statistics of IGD results obtained from the above five algorithms.

Table V shows that the performance of MOEA/DVA on the 54 test instances on the LSMOP test suite is not as good as other algorithms. This is because the variables' separability decreases when the dimension of the decision variable is high, leading to the wrong grouping of the decision variable. WOF had advantages in 9 of the 54 test instances, namely the LSMOP6 with bi-objective and LSMOP7. LSMOF performed well in 11 of 54 test instances, namely the LSMOP3 with tri-objective, LSMOP6, and LSMOP7. For LMOCSO, it only slightly outperformed FDVCSO on the bi-objective LSMOP3 with 1,000-dimensional decision variables and the tri-objective LSMOP9 with decision variables of 2000 and 5000 dimensions. However, FDVCSO has shown an excellent advantage in LSMOP problems.

In order to show the FDV convergence performance more

TABLE V
STATISTICS OF IGD RESULTS ACHIEVED BY FIVE COMPARED ALGORITHMS ON 54 TEST INSTANCES FROM LSMOP TEST SUITE.
THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	MOEA/DVA	WOF	LSMOF	LMOCSO	FDVCSO
LSMOP1	2	1000	6.5722e+0 (1.77e-1) –	5.4729e-1 (1.13e-1) –	6.2286e-1 (1.67e-2) –	4.7730e-1 (4.91e-2) –	8.6126e-2 (1.35e-2)
		2000	8.1932e+0 (2.01e-1) –	6.0072e-1 (4.00e-2) –	6.3739e-1 (2.17e-2) –	9.9108e-1 (9.42e-2) –	1.8275e-1 (1.34e-2)
		5000	9.9154e+0 (8.53e-2) –	5.7175e-1 (6.84e-2) –	6.5555e-1 (1.15e-2) –	1.8603e+0 (1.95e-1) –	2.7179e-1 (1.25e-2)
	3	1000	7.5217e+0 (4.03e-1) –	4.9562e-1 (6.06e-2) –	5.4413e-1 (7.99e-3) –	1.2042e+0 (4.30e-2) –	2.9260e-1 (1.67e-2)
		2000	8.8258e+0 (2.66e-1) –	5.7324e-1 (2.83e-2) –	5.7391e-1 (6.12e-3) –	1.4292e+0 (9.52e-2) –	3.1286e-1 (1.14e-2)
		5000	1.0444e+1 (2.21e-1) –	6.1706e-1 (3.03e-2) –	6.2255e-1 (1.09e-2) –	1.5576e+0 (6.27e-2) –	3.3028e-1 (6.20e-3)
LSMOP2	2	1000	3.9989e-2 (3.52e-4) –	1.8758e-2 (4.14e-4) –	1.8943e-2 (2.20e-4) –	2.3404e-2 (4.21e-4) –	8.8185e-3 (2.41e-3)
		2000	2.2828e-2 (4.99e-4) –	1.2529e-2 (2.81e-4) –	1.2722e-2 (2.65e-4) –	1.3587e-2 (2.86e-4) –	5.7267e-3 (1.22e-3)
		5000	1.2596e-2 (3.91e-4) –	9.2922e-3 (5.05e-4) –	9.8651e-3 (5.43e-4) –	7.2953e-3 (1.72e-4) –	4.0452e-3 (1.56e-4)
	3	1000	5.6399e-2 (1.16e-3) –	6.1725e-2 (3.82e-3) –	6.2447e-2 (3.11e-3) –	4.0877e-2 (2.39e-4) –	3.5866e-2 (1.90e-4)
		2000	5.2074e-2 (2.41e-3) –	5.4427e-2 (3.24e-3) –	5.3688e-2 (2.74e-3) –	3.6419e-2 (1.17e-4) –	3.4054e-2 (3.73e-5)
		5000	4.9239e-2 (2.79e-3) –	5.1123e-2 (2.85e-3) –	5.0609e-2 (2.53e-3) –	3.4380e-2 (4.15e-5) –	3.3281e-2 (2.33e-5)
LSMOP3	2	1000	7.2830e+2 (5.93e+2) –	1.3258e+0 (1.24e-1) –	1.5730e+0 (4.75e-4) –	7.3499e-1 (8.56e-3) +	8.7619e-1 (1.74e-2)
		2000	1.0337e+3 (9.27e+2) –	1.4157e+0 (6.68e-2) –	1.5770e+0 (1.41e-4) –	1.5184e+0 (1.59e+0) –	1.0009e+0 (1.96e-2)
		5000	8.3552e+2 (8.25e+2) –	1.5870e+0 (1.61e-3) –	1.5791e+0 (8.52e-5) –	3.5179e+0 (1.57e+0) –	1.3806e+0 (3.37e-2)
	3	1000	2.1983e+2 (1.58e+2) –	8.5442e-1 (1.12e-2) \approx	8.3993e-1 (1.21e-2) +	8.2870e+0 (3.12e+0) –	8.9997e-1 (1.71e-1)
		2000	2.1978e+2 (1.47e+2) –	8.6072e-1 (5.96e-8) \approx	8.5585e-1 (5.98e-3) +	1.1466e+1 (3.85e+0) –	9.8289e-1 (2.35e-1)
		5000	3.3096e+2 (2.93e+2) –	8.6071e-1 (1.07e-5) \approx	8.6062e-1 (5.56e-5) \approx	1.2734e+1 (2.88e+0) –	8.6052e-1 (1.16e-6)
LSMOP4	2	1000	6.5680e-2 (4.25e-4) –	4.3282e-2 (2.95e-3) –	3.8724e-2 (1.56e-3) –	4.8125e-2 (5.87e-4) –	1.6318e-2 (1.43e-3)
		2000	3.9985e-2 (2.67e-4) –	3.2701e-2 (2.95e-3) –	2.9578e-2 (2.35e-3) –	2.8528e-2 (1.92e-4) –	1.0947e-2 (2.17e-4)
		5000	2.1865e-2 (9.29e-4) –	1.7391e-2 (3.91e-4) –	1.7076e-2 (9.19e-4) –	1.4504e-2 (1.03e-4) –	6.2630e-3 (1.06e-4)
	3	1000	1.1876e-1 (2.90e-3) –	1.3550e-1 (3.92e-3) –	1.3712e-1 (3.29e-3) –	9.0172e-2 (5.58e-4) –	5.6488e-2 (1.56e-3)
		2000	7.8687e-2 (2.35e-3) –	8.6628e-2 (3.51e-3) –	8.9869e-2 (4.00e-3) –	5.8572e-2 (3.76e-4) –	4.2054e-2 (2.90e-4)
		5000	5.5545e-2 (2.47e-3) –	5.8138e-2 (2.36e-3) –	5.9960e-2 (3.43e-3) –	4.0752e-2 (1.05e-4) –	3.5391e-2 (9.60e-5)
LSMOP5	2	1000	1.4443e+1 (4.24e-1) –	5.2705e-1 (1.37e-1) –	7.4209e-1 (1.16e-6) –	9.8714e-1 (1.08e-1) –	5.6466e-2 (1.39e-2)
		2000	1.7637e+1 (3.15e-1) –	3.8952e-1 (1.13e-1) –	7.4209e-1 (1.16e-6) –	2.2750e+0 (3.34e-1) –	1.5802e-1 (1.11e-1)
		5000	2.1254e+1 (1.42e-1) –	6.3076e-1 (6.97e-2) \approx	7.4209e-1 (1.16e-6) –	3.8827e+0 (2.06e-1) –	5.9657e-1 (2.58e-1)
	3	1000	1.3595e+1 (5.21e-1) –	5.3685e-1 (6.50e-3) –	5.3649e-1 (8.60e-3) –	2.4392e+0 (1.53e-1) –	3.0659e-1 (1.78e-2)
		2000	1.5866e+1 (4.75e-1) –	5.3493e-1 (5.58e-3) –	5.4072e-1 (2.70e-4) –	2.9250e+0 (1.47e-1) –	3.3040e-1 (1.69e-2)
		5000	1.8767e+1 (4.26e-1) –	5.3605e-1 (4.62e-3) –	5.4082e-1 (1.24e-5) –	3.4049e+0 (1.44e-1) –	4.2293e-1 (1.15e-2)
LSMOP6	2	1000	1.2735e+3 (1.07e+3) –	4.8778e-1 (1.35e-1) +	3.1245e-1 (4.16e-4) +	7.6080e-1 (9.23e-4) –	7.5859e-1 (8.94e-5)
		2000	3.5354e+3 (3.06e+3) –	5.0593e-1 (1.45e-1) +	3.0879e-1 (4.93e-5) +	7.5379e-1 (5.59e-4) –	7.1527e-1 (8.19e-2)
		5000	2.3178e+3 (1.01e+3) –	5.7860e-1 (1.36e-1) +	3.0695e-1 (9.08e-5) +	7.4731e-1 (2.25e-4) –	7.2100e-1 (5.68e-2)
	3	1000	2.0448e+4 (3.81e+3) –	1.3080e+0 (1.26e-3) –	7.3736e-1 (3.10e-2) –	3.0475e+1 (4.62e+1) –	7.0978e-1 (2.18e-1)
		2000	2.7725e+4 (3.11e+3) –	1.3209e+0 (9.48e-4) –	7.5524e-1 (4.15e-2) +	1.5895e+2 (1.56e+2) –	9.5225e-1 (8.04e-1)
		5000	3.3334e+4 (4.04e+3) –	1.3273e+0 (1.21e-3) \approx	8.1439e-1 (7.21e-2) +	4.8817e+2 (4.20e+2) –	2.5762e+0 (5.67e+0)
LSMOP7	2	1000	4.2304e+4 (2.62e+3) –	1.5119e+0 (1.44e-3) +	1.5091e+0 (8.06e-4) \approx	2.3754e+1 (2.48e+1) –	1.5510e+0 (1.29e-1)
		2000	5.7324e+4 (2.11e+3) –	1.5166e+0 (1.39e-3) +	1.5129e+0 (3.67e-4) \approx	7.0191e+2 (4.99e+2) –	1.5092e+0 (2.53e-2)
		5000	7.5093e+4 (2.20e+3) –	1.5197e+0 (1.11e-3) \approx	1.5155e+0 (3.85e-4) \approx	2.6222e+3 (1.84e+3) –	1.5142e+0 (1.93e-3)
	3	1000	9.3063e+2 (7.33e+2) –	8.5311e-1 (1.78e-3) +	8.6090e-1 (3.09e-3) +	9.1547e-1 (1.07e-1) \approx	8.9747e-1 (5.09e-2)
		2000	1.3570e+3 (1.25e+3) –	8.4318e-1 (6.58e-4) +	8.4835e-1 (1.65e-3) +	9.4873e-1 (4.23e-2) \approx	9.3223e-1 (5.41e-2)
		5000	1.9611e+3 (1.63e+3) –	8.3811e-1 (5.65e-4) +	8.4126e-1 (4.33e-4) +	9.0053e-1 (9.11e-2) \approx	9.5515e-1 (3.49e-4)
LSMOP8	2	1000	1.2203e+1 (3.46e-1) –	5.8862e-1 (1.71e-1) –	7.4209e-1 (1.16e-6) –	9.0954e-1 (6.59e-2) –	3.3189e-2 (5.91e-3)
		2000	1.5120e+1 (2.26e-1) –	4.2282e-1 (1.15e-1) –	7.4209e-1 (1.16e-6) –	1.9347e+0 (2.40e-1) –	6.0828e-2 (7.27e-3)
		5000	1.8333e+1 (1.09e-1) –	6.6283e-1 (7.02e-2) –	7.4209e-1 (1.16e-6) –	3.2366e+0 (4.60e-1) –	2.4829e-1 (4.64e-2)
	3	1000	6.5891e-1 (5.05e-2) –	3.1831e-1 (2.66e-2) –	3.4759e-1 (1.03e-2) –	6.7676e-1 (1.14e-1) –	6.8094e-2 (1.75e-3)
		2000	6.3386e-1 (4.52e-2) –	3.2452e-1 (2.73e-2) –	3.3961e-1 (1.07e-2) –	7.1457e-1 (1.69e-1) –	6.6356e-2 (1.14e-3)
		5000	6.5344e-1 (7.85e-2) –	3.2536e-1 (1.84e-2) –	3.3986e-1 (1.10e-2) –	9.1369e-1 (7.51e-2) –	6.7543e-2 (6.00e-4)
LSMOP9	2	1000	3.3138e+1 (1.30e+0) –	8.0871e-1 (4.73e-4) –	8.0740e-1 (8.12e-4) –	3.7433e-1 (2.50e-2) –	3.5764e-2 (1.19e-2)
		2000	4.2736e+1 (1.24e+0) –	8.0746e-1 (1.31e-3) –	8.0477e-1 (3.94e-4) –	3.8973e-1 (2.05e-2) –	2.8510e-2 (3.88e-3)
		5000	5.3080e+1 (3.79e-1) –	8.0394e-1 (1.93e-3) –	8.0325e-1 (5.52e-4) –	7.8576e-1 (1.06e-1) –	1.4606e-1 (1.30e-1)
	3	1000	9.2003e+1 (1.92e+0) –	1.1776e+0 (1.13e-1) –	1.2430e+0 (1.78e-1) –	6.6576e-1 (1.01e-1) –	4.9350e-1 (1.76e-2)
		2000	1.1187e+2 (3.01e+0) –	1.1759e+0 (1.09e-1) –	1.1442e+0 (2.65e-4) –	6.3227e-1 (1.09e-1) \approx	6.5052e-1 (1.02e-1)
		5000	1.3547e+2 (1.58e+0) –	1.1443e+0 (2.65e-4) +	1.1440e+0 (1.92e-4) +	8.0977e-1 (9.58e-2) +	8.3928e+0 (1.84e+0)
+ / - / \approx			0/54/0	9/39/6	11/39/4	2/48/4	

clearly. Fig. 2 in the supplementary file shows the convergence of mean IGD values of five algorithms on the tri-objective LSMOP test suite with 2000 decision variables. In the figure, WOF-NSGA-II and LSMOF show better advantages in the LSMOP7 problem. This is because the algorithm of the problem transformation mechanism finds a good problem transformation function on this problem. On the problems of LSMOP3, LSMOP6, and LSMOP9, the performance advantages of FDVCSO were not very obvious. This is because these problems have low requirements for the algorithm's con-

vergence but pay attention to the distribution of the algorithm. On the whole, FDVCSO had a very obvious advantage in the LSMOP test suite.

To study the performance of FDV on MOPs with irregular decision space and the performance on constrained large-scale MOPs. FDV-CCMO and CMOPSO, MOEA/DVA, LMOCSO, and WOF-SMPSO are compared with five test problems on the TREE test suite. Table III in the supplementary file shows the statistics of IGD results obtained by these five algorithms. It can be found from the table that CMOPSO, MOEA/DVA,

and LMOCSO did not find any feasible solutions for TREE3 to TREE5. The possible reason is that these problems have many constraints and are more complicated than the objective function. Therefore, the decision variable grouping mechanism (MOEA/DVA) that only considers the correlation with the objective function cannot effectively solve these problems. CMOPSO and LMOCSO show poor performance because they do not have a good constraint handling mechanism. WOF-SMPSO shows better performance, which may be attributed to the ability of the PSO operator to track the global best particles, thereby performing faster convergence in a relatively simple fitness landscape. Another reason is that although the problem transformation has lost some problem information, it strengthens the ability to search for solutions. FDV-CCMO is not the best performance on TREE1, TREE3, and TREE2 with two objectives. The reason is that this type of problem has an irregular decision space, so FDV cannot uniformly fuzzy the original solution into the decision space.

V. CONCLUSION

In this paper, we have proposed a general framework for solving large-scale MOPs. The FDV framework is divided into two main stages: the first stage is the fuzzy evolution stage, aiming to search for rough solutions. The second stage is the precise evolution stage, which aims to search for precise solutions.

In fuzzy evolution, the fuzzy evolutionary stage is divided into multiple sub-stages through the proposed method of fuzzy evolution sub-stages division. Each sub-stage has a different degree of fuzzification, and the degree of fuzzification decreases gradually. The value range of decision variables is reduced from an infinite number of original values to a finite number of fuzzy values, so the FDV framework reduces the search range of the algorithm and improves the search efficiency of the algorithm. In precise evolution, the fuzzy operation is lost and the evolution method embedded in MOEA is used directly. At this stage, more attention is paid to the diversity of algorithms.

In order to verify that the proposed FDV framework enhances the ability of MOEAs to solve large-scale MOPs, our typical MOEAs were embedded in the FDV framework and paired with the original algorithm on the DTLZ test problem. In order to prove the advantages of the FDV framework in the existing large-scale optimization frameworks, the NSGA-II algorithm was embedded in several popular frameworks and compared on the DTLZ test suite. Finally, in order to demonstrate that FDVCSO also has significant advantages in several state-of-the-art large-scale MOEAs: MOEA/DVA, WOF, LSMOF, and LMOCSO, we conducted an experimental comparison on the LSMOP test suite. This experimental comparison shows that our proposed FDV framework dramatically improves the performance of existing MOEAs and has obvious performance advantages in advanced large-scale MOEAs.

The proposed FDV framework has shown good potential in large-scale multiobjective optimization, but the optimal parameters are different for different problems, and parameter analysis is usually required. Future work will develop the

parameters into self-adaptation. This part of the work involves problem analysis and automatic adjustment of parameters in the face of different problems. Designing better fuzzy formulas to solve MOPs with irregular decision spaces is also our future research work. It is also interesting to apply our proposed FDV framework to real-world, large-scale MOPs with more decision variables through parallel (for example, GPU-based) calculations.

VI. ACKNOWLEDGEMENTS

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61876164, 61772178), the Natural Science Foundation of Hunan Province (Grant No. 2020JJ4590), the Education Department Major Project of Hunan Province (Grant No. 17A212), the MOEA Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Provinces and Cities Joint Foundation Project (Grant No. 2017JJ4001), the Hunan province science and technology project funds (2018TP1036).

REFERENCES

- [1] T. O. Olowu, H. Jafari, M. Moghaddami, and A. I. Sarwat, "Multiphysics and multiobjective design optimization of high-frequency transformers for solid-state transformer applications," *IEEE Transactions on Industry Applications*, vol. 57, no. 1, pp. 1014–1023, 2020.
- [2] Y. Li, Z. Ni, T. Zhao, M. Yu, Y. Liu, L. Wu, and Y. Zhao, "Coordinated scheduling for improving uncertain wind power adsorption in electric vehicles—wind integrated power systems by multiobjective optimization approach," *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2238–2250, 2020.
- [3] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, E. Mezura-Montes, and M. Aldape-Pérez, "Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1272–1285, 2019.
- [4] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms," *IEEE transactions on cybernetics*, vol. 46, no. 3, pp. 582–594, 2015.
- [5] C.-F. Juang and T. B. Bui, "Reinforcement neural fuzzy surrogate-assisted multiobjective evolutionary fuzzy systems with robot learning control application," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 3, pp. 434–446, 2019.
- [6] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Adaptive multilevel prediction method for dynamic multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 463–477, 2021.
- [7] H. Ma, M. Fei, Z. Jiang, L. Li, H. Zhou, and D. Crookes, "A multipopulation-based multiobjective evolutionary algorithm," *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 689–702, 2018.
- [8] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Weighted pointwise prediction method for dynamic multiobjective optimization," *Information Sciences*, vol. 546, pp. 349–367, 2021.
- [9] A. Liefvooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "Landscape-aware performance prediction for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1063–1077, 2019.
- [10] 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.
- [11] Y. Xiang, Y. Zhou, Z. Chen, and J. Zhang, "A many-objective particle swarm optimizer with leaders selected from historical solutions by using scalar projections," *IEEE transactions on cybernetics*, vol. 50, no. 5, pp. 2209–2222, 2018.
- [12] Y. Liu, N. Zhu, and M. Li, "Solving many-objective optimization problems by a pareto-based evolutionary algorithm with preprocessing and a penalty mechanism," *IEEE Transactions on Cybernetics*, 2020.

- [13] Q. Lin, S. Liu, K.-C. Wong, M. Gong, C. A. C. Coello, J. Chen, and J. Zhang, "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2018.
- [14] Y. Xiang, Y. Zhou, X. Yang, and H. Huang, "A many-objective evolutionary algorithm with pareto-adaptive reference points," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 99–113, 2019.
- [15] S. Qin, C. Sun, G. Zhang, X. He, and Y. Tan, "A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems," *Complex & Intelligent Systems*, pp. 1–12, 2020.
- [16] P. Zhang, J. Li, T. Li, and H. Chen, "A new many-objective evolutionary algorithm based on determinantal point processes," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 334–345, 2020.
- [17] J. Zou, X. Yang, Z. Liu, J. Liu, L. Zhang, and J. Zheng, "Multiobjective bilevel optimization algorithm based on preference selection to solve energy hub system planning problems," *Energy*, p. 120995, 2021.
- [18] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, and Y. Jin, "A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks," *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 703–716, 2018.
- [19] X. Li and W. Wang, "Research on large-scale multi-objective optimization algorithm with irregular frontier for operation dispatching of new generation energy system integration," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, pp. 2959–2965.
- [20] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3115–3128, 2020.
- [21] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multi-objective optimization based on a competitive swarm optimizer," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3696–3708, 2019.
- [22] S. Qin, C. Sun, Y. Jin, Y. Tan, and J. Fieldsend, "Large-scale evolutionary multi-objective optimization assisted by directed sampling," *IEEE Transactions on Evolutionary Computation*, 2021.
- [23] E. T. Oldewage, A. P. Engelbrecht, and C. W. Cleghorn, "The merits of velocity clamping particle swarm optimisation in high dimensional spaces," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.
- [24] L. M. Antonio and C. A. C. Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 2758–2765.
- [25] 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.
- [26] L. M. Antonio and C. A. C. Coello, "Decomposition-based approach for solving large scale multi-objective problems," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2016, pp. 525–534.
- [27] M. Li and J. Wei, "A cooperative co-evolutionary algorithm for large-scale multi-objective optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1716–1721.
- [28] M. A. Meselhi, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Contribution based co-evolutionary algorithm for large-scale optimization problems," *IEEE Access*, vol. 8, pp. 203 369–203 381, 2020.
- [29] 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, no. 2, pp. 275–298, 2015.
- [30] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.
- [31] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 260–275, 2017.
- [32] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multiobjective optimization via problem reformulation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 949–961, 2019.
- [33] C. C. Coello and M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1051–1056.
- [34] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE transactions on cybernetics*, vol. 45, no. 2, pp. 191–204, 2014.
- [35] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.
- [36] R. Cheng, Y. Jin, M. Olhofer *et al.*, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2016.
- [37] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "An overview of population-based algorithms for multi-objective optimisation," *International Journal of Systems Science*, vol. 46, no. 9, pp. 1572–1599, 2015.
- [38] H.-J. Zimmermann, "Fuzzy set theory," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 317–332, 2010.
- [39] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy sets and systems*, vol. 1, no. 1, pp. 3–28, 1978.
- [40] X. Yan, H. Huang, Z. Hao, and J. Wang, "A graph-based fuzzy evolutionary algorithm for solving two-echelon vehicle routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 129–141, 2019.
- [41] S. Elsayed, R. Sarker, and C. A. C. Coello, "Fuzzy rule-based design of evolutionary algorithm for optimization," *IEEE transactions on cybernetics*, vol. 49, no. 1, pp. 301–314, 2017.
- [42] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Information Sciences*, vol. 427, pp. 63–76, 2018.
- [43] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2020.
- [44] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *2009 IEEE Symposium on computational intelligence in multi-criteria decision-making (MCDM)*. IEEE, 2009, pp. 66–73.
- [45] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 868–881, 2020.
- [46] W. Haynes *et al.*, "Wilcoxon rank sum test," *Encyclopedia of systems biology*, pp. 2354–2355, 2013.
- [47] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE transactions on evolutionary computation*, vol. 7, no. 2, pp. 174–188, 2003.
- [48] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using monte carlo sampling," in *Multiple criteria decision making for sustainable energy and transportation systems*. Springer, 2010, pp. 313–326.
- [49] D. A. Van Veldhuizen, G. B. Lamont *et al.*, "Evolutionary computation and convergence to a pareto front," in *Late breaking papers at the genetic programming 1998 conference*. Citeseer, 1998, pp. 221–228.
- [50] 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, no. 4, pp. 73–87, 2017.
- [51] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th annual conference on genetic and evolutionary computation*, 2007, pp. 1187–1194.
- [52] K. Deb and M. Goyal, "A combined genetic adaptive search (genaes) for engineering design," *Computer Science and informatics*, vol. 26, pp. 30–45, 1996.
- [53] 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.



Xu Yang received the B.Sc. degree in computer science and technology in 2019. He is currently pursuing the M.Sc. degree with the Department of Computer Science, Xiangtan University, Xiangtan, China.

His current research interests include large-scale optimization, bilevel optimization, and swarm intelligence.



Yuan Liu received the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University, Xiangtan, China, in 2017, and Ph.D. degree in the College of Information Science and Engineering, Hunan University, Changsha, China, in 2021. Currently, he is a lecturer at the College of Information Engineering in Xiangtan University.

His research interests include evolutionary algorithm and machine learning.



Juan Zou received the B.Sc. degree in computer application technology, the M.Sc. degree in computer application technology, and the Ph.D. degree in applied mathematics from Xiangtan University, Xiangtan, China, in 2002, 2005, and 2014, respectively.

She is currently a Professor of Computational Intelligence in the College of Computer and Cyberspace Security of Xiangtan University, Xiangtan, China. She has over 50 publications and one monograph. Her current research interests include

multiobjective optimization, swarm intelligence, artificial neural networks, and related applications.



Shengxiang Yang (M'00–SM'14) received the PhD degree from Northeastern University, Shenyang, China in 1999. He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 340 publications with an H-index of 61 according to Google Scholar. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant

real-world applications. He serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Cybernetics*, *Information Sciences*, and *Enterprise Information Systems*.



Jinhua Zheng received the B.Sc. degree in computer software and the M.Sc. degree in computer application from the Nanjing University of Science and Technology, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000.

Since 2000, he has been a Professor with the College of Information Engineering, Xiangtan University, China. He has over 140 research publications, including an academic monograph and four

textbooks. His current research interests include evolutionary computation, evolutionary multiobjective optimization, robust optimization, and evolutionary algorithms in real-world applications.