

## Research Article

# An Improved MOEA/D Algorithm for Complex Data Analysis

Weihoa Qian <sup>1</sup>, Jiahui Liu,<sup>1</sup> Yuanguo Lin,<sup>1</sup> Lvqing Yang <sup>1</sup>, Jianwei Zhang <sup>2</sup>, Hang Xu,<sup>3</sup>  
Minghong Liao,<sup>1</sup> Yuxuan Chen,<sup>1</sup> Yunyi Chen,<sup>1</sup> and BingShuai Liu<sup>1</sup>

<sup>1</sup>Xiamen University, China

<sup>2</sup>Zhengzhou University, China

<sup>3</sup>Putian University, China

Correspondence should be addressed to Lvqing Yang; [lqyang@xmu.edu.cn](mailto:lqyang@xmu.edu.cn) and Jianwei Zhang; [mailzjw@163.com](mailto:mailzjw@163.com)

Received 14 July 2021; Revised 30 August 2021; Accepted 1 September 2021; Published 29 September 2021

Academic Editor: Yuanpeng Zhang

Copyright © 2021 Weihoa Qian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are a large number of multiple level datasets in the Industry 4.0 era. Thus, it is necessary to utilize artificial intelligence technology for the complex data analysis. In fact, the technology often suffers from the self-optimization issue of multiple level datasets, which is taken as a kind of multiobjective optimization problem (MOP). Naturally, the MOP can be solved by the multiobjective evolutionary algorithm based on decomposition (MOEA/D). However, most existing MOEA/D algorithms usually fail to adapt neighborhood for the offspring generation, since these algorithms have shortcomings in both global search and adaptive control. To address this issue, we propose a MOEA/D with adaptive exploration and exploitation, termed MOEA/D-AEE, which adopts random numbers with a uniform distribution to explore the objective space and introduces a joint exploitation coefficient between parents to generate better offspring. By dynamic exploration and joint exploitation, MOEA/D-AEE improves both global search ability and diversity of the algorithm. Experimental results on benchmark data sets demonstrate that our proposed approach achieves global search ability and diversity in terms of the population distribution than state-of-the-art MOEA/D algorithms.

## 1. Introduction

A multiobjective optimization problem (MOP) refers to the determination of a vector composed of decision variables in a feasible domain, which satisfies all constraints and optimizes a vector composed of multiple objective functions. The multiple objective functions of these component vectors usually contain subobjectives that conflict with each other. The improvement of one subobjective may lead to the performance degradation of another or several subobjectives. That is, it is impossible to achieve the optimal values for multiple subobjectives; rather, there must be coordination and compromise among them so that each subobjective can be optimized as well as possible. MOPs are a common kind of optimization problems in the real world and often mentioned in the field of scientific research and its applications.

Many scholars have tried to solve such problems by using evolutionary algorithms. The first step is to simplify the problem of two dimensions into a single-objective prob-

lem by setting weights. However, due to the increase in the dimensions of the objective function, the simple weighting method cannot reflect the real situation of the population, and the distribution of the population is uneven. Then, scholars continued to study the strategy of using multiobjective evolutionary algorithms (MOEAs) and proposed the classic nondominated genetic sorting algorithm II (NSGA-II) [1]. This algorithm is based on the method of fast nondominated set sorting and congestion calculation. It has good performance for low-dimensional multiobjective optimization problems, but the congestion distance is not suitable for high-dimensional space, and the computational complexity is relatively high. To speed up the convergence of the algorithm, scholars proposed the classic multiobjective evolutionary algorithm based on decomposition (MOEA/D) [2] in 2007. The decomposition-based method improves the convergence speed of the algorithm, which reduces the computational complexity, and obtains more uniform solutions than NSGA-II. Thus, MOEA/D has been widely used in

complex data analysis [3, 4]. In recent years, with the development of Industry 4.0, many combinatorial optimization problems about large-scale and multiple level datasets have arisen [5, 6], while the MOEA/D algorithm has been introduced to address this problems, which is a kind of complex data analysis MOPs. For example, in the MOEA/D-Lévy algorithm [7], a Lévy flight is a short-distance hopping exploratory search strategy with occasional long-distance development search. It may be an ideal search strategy [8], since it can improve population diversity to prevent premature convergence and jump out of local optimal solutions [9]. However, the frequency of generating better offspring in the later iterations of the algorithm is very low. Besides, MOEA/D algorithms usually suffer from adapting neighborhood for the offspring generation, since the global search ability and the parent adaptive ability are insufficient.

To address the shortcomings of the above algorithms, we propose a MOEA/D with Adaptive Exploration and Exploitation, named MOEA/D-AEE, which uses dynamic search and joint development to solve global search and adaptive problems, and thereby achieves better sparse distribution and global search ability. Specifically, the proposed algorithm uses uniformly distributed random numbers to explore the solution space and introduces the joint exploitation coefficient between parents to produce better offspring by adaptively adjusting parent relationships. When two selected parents are similar, it adopts an active exploration strategy to select the parent from the initial population. In contrast, when the two selected parents are quite different, the algorithm gradually employs the exploitation strategy to select the parent from the neighbor. To verify the performance of the proposed algorithm, two experimental schemes are proposed in this study, and a comparative test is carried out on the three OR-Library public data sets in PO. The traditional MOEA/D algorithm, evolution algorithm, NSGA-II algorithm, and Lévy flight mutation algorithm and the global search performance of the algorithms are compared. The experimental results and analysis show that the performance of our algorithm is advanced in the six metrics, and the global search ability is further improved, especially in the later stage of iteration. There is still a long trajectory update, the population has a faster convergence speed in the first 20 generations and converges to the Pareto front, and the diversity performance is better on the three test platforms.

The remainder of this paper is organized as follows. Section 2 briefly reviews the research progress. In Section 3, we describe the method and implementation of the algorithm. Section 4 provides the experimental results, and the corresponding discussions are given in Section 5. Section 6 summarizes this paper and proposes possible future directions.

## 2. Related Work

**2.1. MOEA/D Algorithms.** MOPs are a class of optimization problems that contain multiple conflicting objectives. Compared with the single-objective optimization problem (SOP), the goal of an MOP is to obtain a set of optimal solutions showing the best trade-off between its multiple objectives

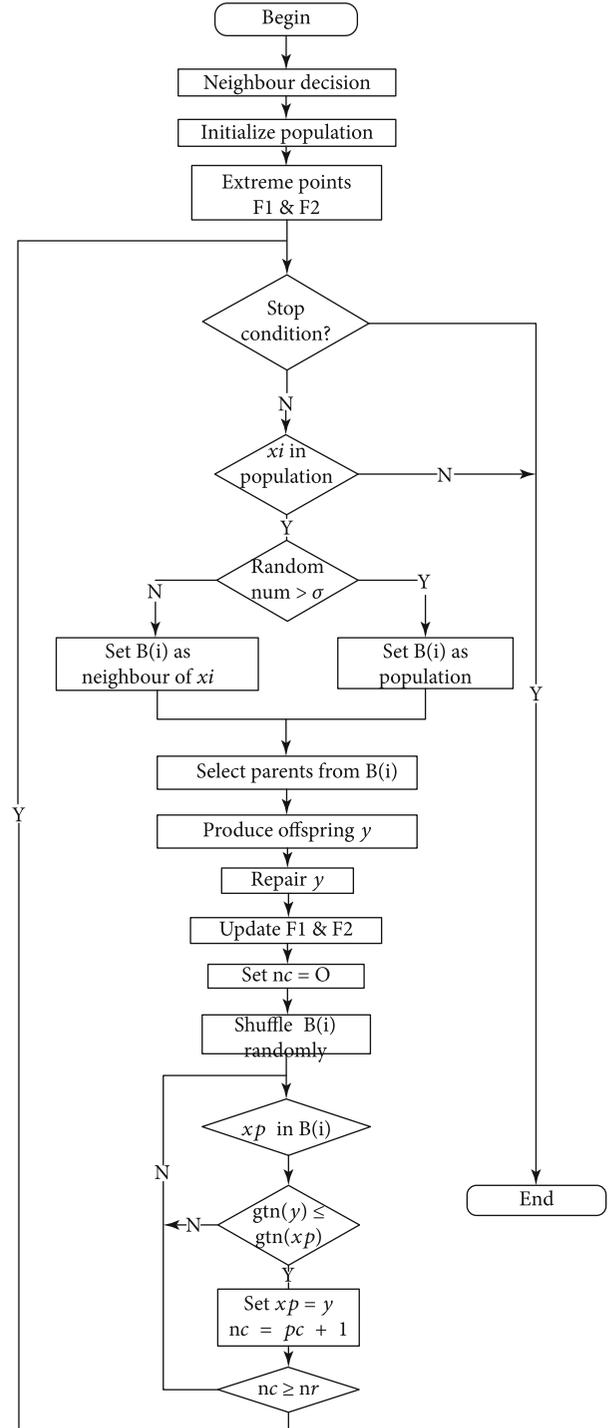


FIGURE 1: MOEA/D-AEE algorithm flow chart.

rather than an optimal solution. Such an optimal-set Pareto front contains all feasible solutions in the feasible problem domain that are not governed by other solutions. The MOP solution algorithm is called MOEA; Deb et al. and others proposed the classical NSGA-II [1], and this kind of problem was solved by sorting noncollective sets. Then, Zhang and Li proposed an MOEA/D, decomposing multiobjective problems into subproblems of multiple single-objective problems and optimizing them at the same time.

```

1 Select Neighbour B;
2 Initialize population P;
3 Compute extreme points F1 and F2;
4 while stop condition not reached do
5   for xi in opulation do
6     if random num < σ then
7       Set B(i) as neighbor of xi;
8     else.
9       Set Set B(i) as population;
10    end.
11    Select parents from B(i).
12    Use proposed method to reproduce offspring y;
13    Repair y to reasonable range;
14    Update extreme points F1 and F3;
15    Set update counter nc = 0;
16    Shuffle B(i) randomly;
17    for xp in neighbor of B(i) do
18      if gtn(y|λ, rp) < = gtn(xp|λ, rp) then
19        Set xp = y;
20        Set nc + 1;
21        if nc > = nr then
22          Break;
23        end
24      end
25    end
26  end
27 end

```

ALGORITHM 1: MOEA/D-AEE pseudocode description. MOEA/D-AEE.

Common decomposition methods in this method are the weighted sum method, Chebyshev method, and boundary crossing method [2]. The formula of the most common Chebyshev decomposition method is as follows:

$$\min_x g^{te}(x|\lambda, z^*) = \max_{m=1, \dots, M} \{\lambda_m |f_m(x) - z_m^*|\}, \quad (1)$$

$$\lambda = (\lambda_1, \dots, \lambda_M), \lambda_1 + \dots + \lambda_M = 1, \quad (2)$$

$$z^* = (\min f_1, \dots, \min f_M). \quad (3)$$

$z_m^*$  in the objective space of  $m$  dimensions represents the minimum value on the  $m$  objective dimensions,  $\lambda$  represents the weight vector, and  $z$  represents the reference point. The intersection of each dimension contour is best in the direction of the weight vector, and these points ultimately form the Pareto front.

To solve the problem of the inability to address a disproportionate scaling objective with the ordinary Chebyshev method, this paper adopts the narrowband interference-(NBI-) Chebyshev decomposition method [10] proposed by Zhang et al. The mathematical formula is as follows:

$$r^{(i)} = \alpha_i F^1 + (1 - \alpha_i) F^2, \alpha_i = \frac{N - i}{N - 1}, \quad (4)$$

$$g^{tn}(x|r^{(i)}, \lambda) = \max \left\{ \lambda_1 (f_1(x) - r_1^{(i)}), \lambda_2 (f_2(x) - r_2^{(i)}) \right\}, \quad (5)$$

$$\lambda_1 = |F_2^2 - F_2^1|, \lambda_2 = |F_1^2 - F_1^1|, \quad (6)$$

where  $F^1$  and  $F^2$  are the two extreme points of the Pareto front in the objective space, and there are  $n$  reference points  $r^i$ ,  $i = 1, \dots, N$ , which represent  $n$  reference points arranged on a line between the two extreme points.  $\lambda_1$  and  $\lambda_2$  are the weight vectors calculated by  $F^1$  and  $F^2$ , respectively, which are perpendicular to the line between the two extreme points.

In recent years, scholars have made many improvements to MOEA/D.

First of all, some of them have made improvements to the algorithm itself. For the drawback of the allocation of resources, Zhang et al. proposed an algorithm to dynamically allocate computational resources to different subproblems in MOEA/D called the MOEA/D-DRA algorithm [11]. Each subproblem uses an efficiency function to measure the improvement rate of the aggregation function and only calculates a subproblem with a higher efficiency function each time. In order to deal with the complexity of the Pareto set, Li and Zhang presented a new version of MOEA/D based on differential evolution, known as the MOEA/D-DE algorithm [12]. The test showed that MOEA/D was significantly better than NSGA-II, indicating that decomposition-based multiobjective evolutionary algorithms are very promising for addressing complex Pareto set PS shapes. To improve selection matching process, Li et al. proposed a simple and effective stable matching model (STM) [13]. Each subproblem was matched with a solution

TABLE 1: Specific parameter configuration for the algorithm.

Algorithm	Parameters	Parameters	Parameters
MOEA/D-AEE	$\alpha_0 = 1e-05$	$\beta = 0.3$	Mutation rate = $1/n$
MOEA/D-lévy	$\alpha_0 = 1e-05$	$\beta = 0.3$	Mutation rate = $1/n$
MOEA/D-GA	Crossover rate = 0.7	Mutation rate = 0.01	
NSGA-II	Crossover rate = 0.7	Mutation rate = 0.01	
MOEA/D-DEM	$F = 1.3$	Mutation rate = $1/n$	
MOEA/D-DE	$F = 1.3$		

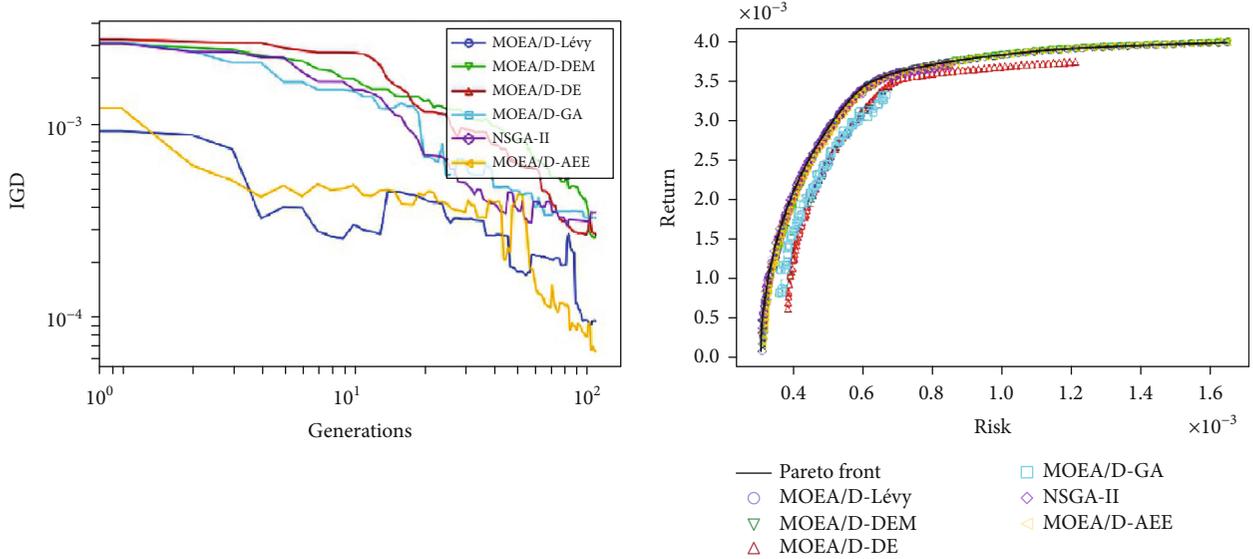


FIGURE 2: Nikkei IGD indicators and population distribution.

using an STM model, and the convergence and diversity of the algorithm were balanced.

Secondly, for the adaptive problems, Wang et al. proposed an adaptive evolutionary strategy AES-MOEA/D algorithm [14] based on an adaptive evolutionary strategy, and three improved strategies were used: the evolution strategy of simulated binary crossover (SBX) competition with the DE operator, the adaptive adjustment strategy for the mutation probability, and the boundary optimization strategy of double-sided mirror theory, improving diversity in late evolution can improve the performance of MOEA/D algorithms. Chiang and Lai also considered adaptive ability, and they proposed an algorithm based on the MOEA/D adaptive matching selection mechanism called the MOEA/D-AMS algorithm [15]. By distinguishing between resolved and unresolved subproblems, they select only unresolved subproblems to match each time. The matching pool size is dynamically adjusted according to the distance in the objective space.

Moreover, for the purpose of expanding the application of MOEA/D in high-dimensional objectives, Zheng et al. proposed an improved MOEA/D algorithm called the I-MOEA/D algorithm [16]. Compared to the original MOEA/D, the adjusted weight vector makes the distribution

of solutions more extensive and more effective in ensuring the diversity and convergence of solutions in the objective space. Tan et al. also improved the MOEA/D algorithm in high dimensions and proposed a MOEA/D-based uniform design algorithm called the UMOEA/D algorithm [17]. The weight vectors on three to five objective spaces are more uniform.

Besides, in some existing MOEAs, an external archive is usually adopted to save elite solutions found during the evolutionary search to help improve the population diversity. Inspired by the decomposition strategy, Zhang et al. propose an efficient archiving approach, called the decomposition based archiving approach (DAA) [18] to prune the external archive size. In DAA, the size of the archive depends on the number of subspaces, and they do not need to compare new solutions with all the archived solutions. This can reduce the computational cost significantly. For the disadvantages of single archive, they also study a multiobjective feature selection approach, called two-archive multiobjective artificial bee colony algorithm (TMABC-FS) [19] to balance the distribution and convergence of solutions. The leader archive and the external archive are employed to enhance the search capability of different kinds of bees to avoid miss potentially important regions.

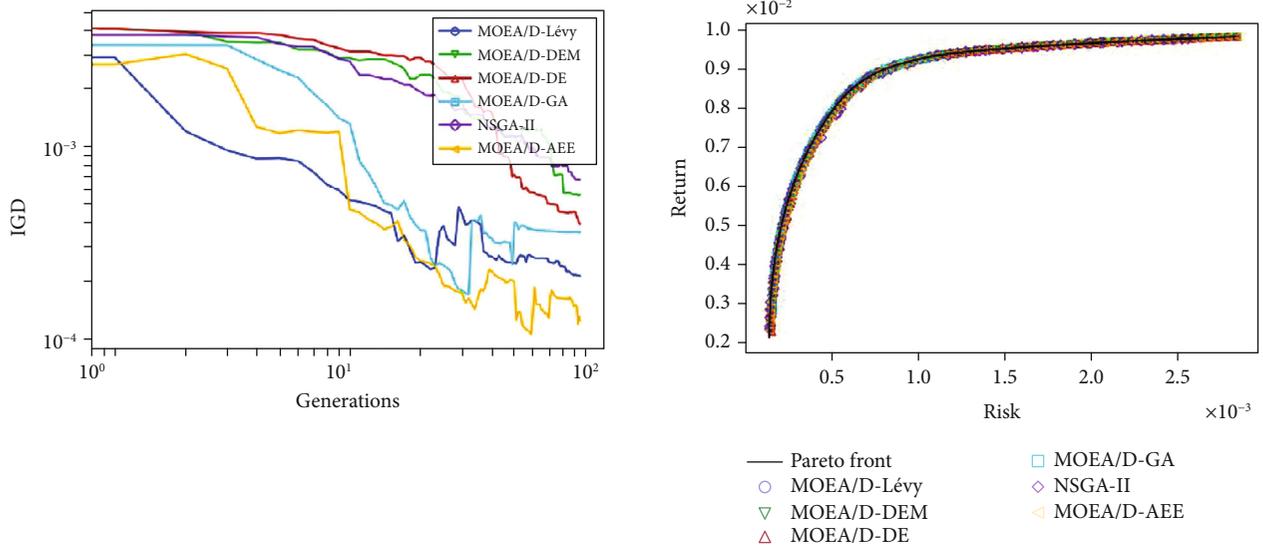


FIGURE 3: DAX 100IGD indicators and population distribution.

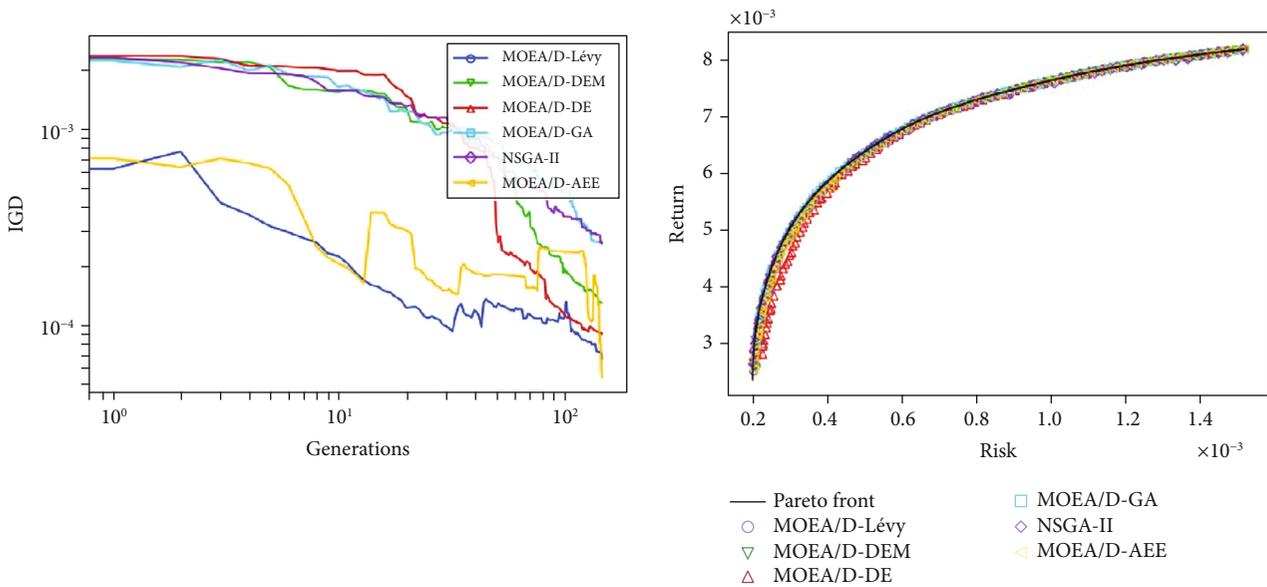


FIGURE 4: FTSE 100IGD indicators and population distribution.

2.2. MOEA/D in Industry 4.0. The self-optimization problem of cyber physical systems (CPSs) can be considered as a kind of MOPs; therefore, the MOEA/D algorithm is the need for automated and enhanced decision makings.

On the one hand, metaheuristics have been used successfully to address realistic planning and scheduling problems. Cota et al. propose multiobjective extensions of the adaptive large neighborhood search (ALNS) metaheuristic with Learning Automata (LA) to improve the search process and to solve the large scale instances efficiently. Two new algorithms are MO-ALNS and the MO-ALNS/D, and they address the unrelated parallel machine scheduling problem to maximize production [4]. Energy efficiency and timeliness

are two mutually contradictory objectives, and Shukla et al. propose a novel approach, based on the popular multiobjective evolutionary algorithm, Nondominated sorting genetic algorithm-II, to solve this problem. They introduce novel algorithms for membership function generation and calculation of fuzzy earliness [20]. Fu et al. study an integrated scheduling problem in a distributed manufacturing system [21]. They build a stochastic multiobjective model with consideration of two criteria of scheduling and uncertainties of the operation process. Also, they design a multiobjective brain storm optimization algorithm to solve the model. Fu et al. also investigate a flow-shop scheduling problem under the consideration of multiple objectives, time-dependent

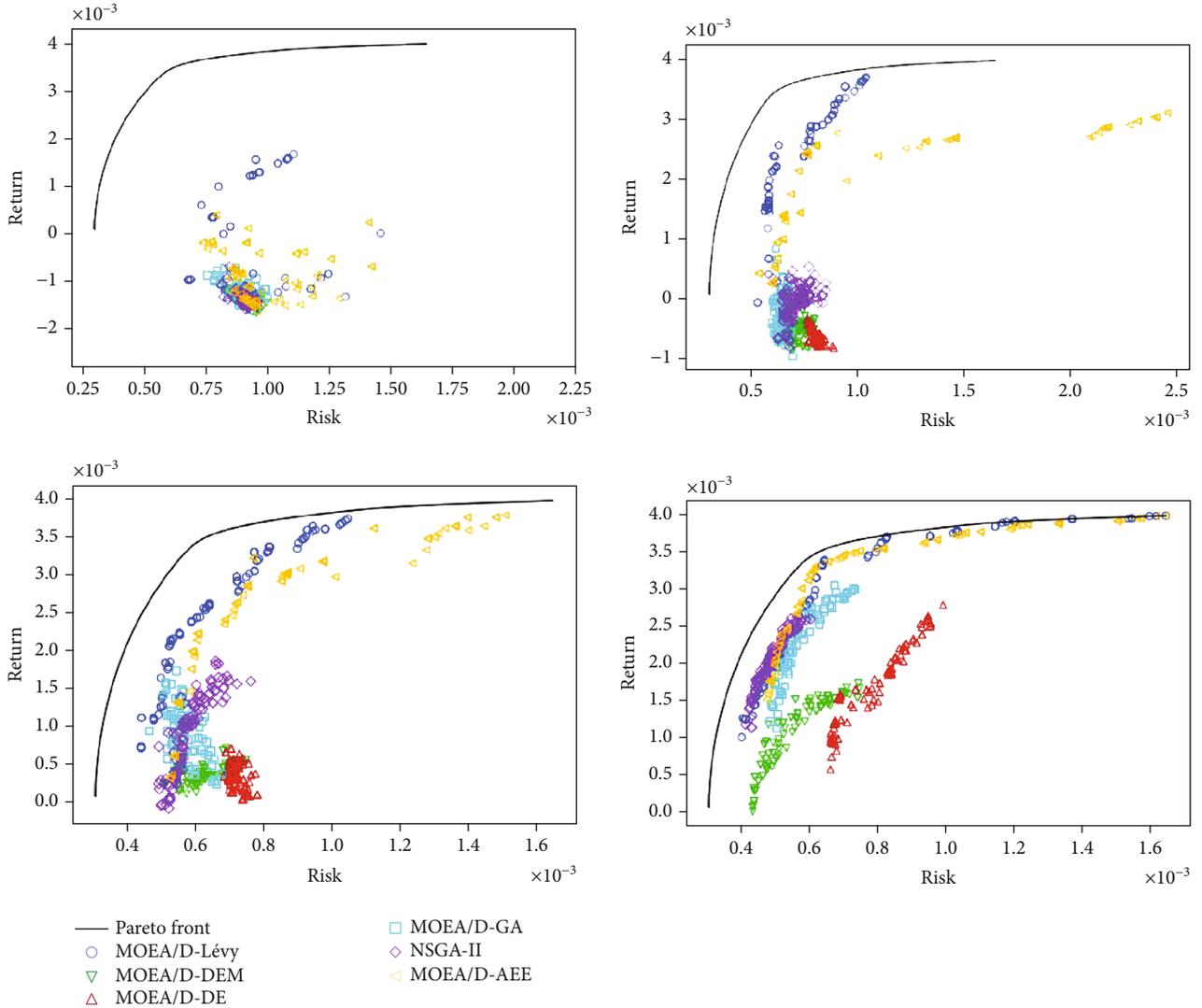


FIGURE 5: Distribution of 1, 10, 20, and 50 generations in the experiment.

processing time and uncertainty. A mixed integer programming model is formulated for this problem, and a fireworks algorithm is developed [22].

On the other hand, maintenance has always been a key activity in the manufacturing industry, and Goti et al. focus on condition-based maintenance (CBM) optimization in an industrial environment, with the objective of determining the optimal values of preventive intervention limits for equipment under corrective and preventive maintenance cost criteria. They develop a cost-benefit mathematical model. It considers the evolution in quality and production speed, along with condition based, corrective, and preventive maintenance [23]. Besides, existing big service selection and composition algorithms in industry fail to achieve the global optimum solution in a reasonable time, and Dutta et al. design an efficient quality of service-aware big service composition methodology using a distributed coevolutionary algorithm. They develop a distributed NSGA-III for finding the optimal Pareto front and a distributed multiobjective Jaya algorithm for enhancing the diversity of solutions [24].

Rather than other MOEA/D algorithms such as [25, 26], our algorithm employs both global exploration and local exploitation and thereby solves the self-optimization issue of multiple level datasets in the Industry 4.0 era. We introduce the MOEA/D-AEE algorithm in the following section.

### 3. Methods

**3.1. Framework of MOEA/D-AEE.** In this section, we describe the MOEA/D-AEE algorithm, which integrates Lévy flight algorithms with dynamic search and joint development capabilities in MOEA/D multiobjective optimization algorithms. The complete diagram of the algorithm is shown in Figure 1. In this paper, a vector is used to represent the problem variables in investment optimization. The solution of the algorithm is used with a uniform distribution in  $(0, 1)$  for initialization. For the generated progeny, this paper uses a new normal boundary intersection- (NBI-) Chebyshev decomposition method in the MOEA/D decomposition to address the return and risk [10] of the two objective

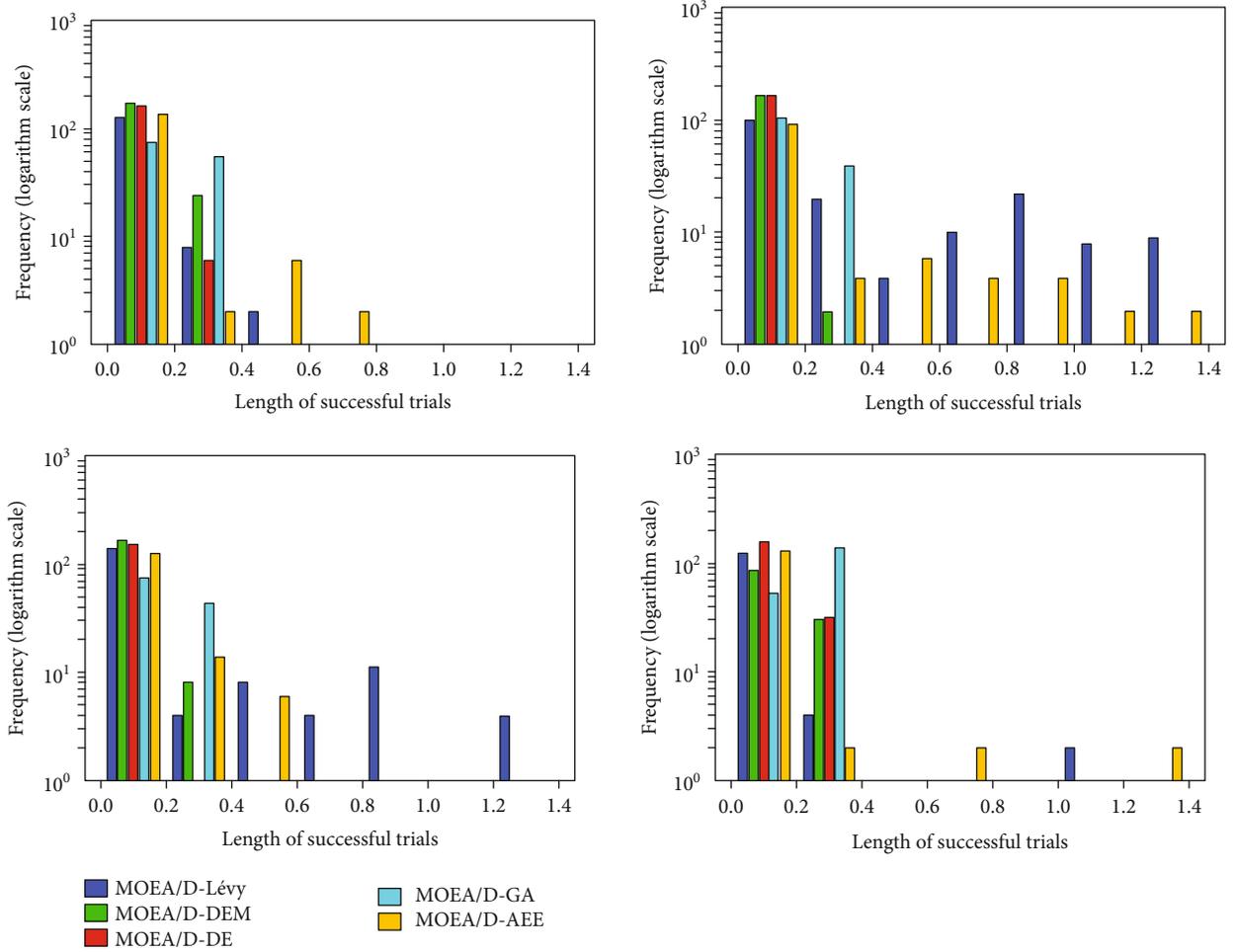


FIGURE 6: Diagram of the 1-, 3-, 5-, and 10-generation successful update trajectories.

functions. At the same time, this paper adopts the diversity maintenance strategy, the iterative mutation method of the Lévy flight, and polynomial mutation based on joint coefficients.

The Lévy flight mutation operator used in this paper is similar to the DE mutation operator and is used on different individuals. The difference is that the scaling factor used for Lévy flight mutation is generated by a heavy-tailed distribution rather than a fixed value. Additionally, Lévy flight mutation usually requires only two parents, not three. The Lévy flight mutation formula is as follows:

$$\text{Levy}(\beta) \sim \frac{u}{|v|^{1/\beta}}, \quad (7)$$

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2), \quad (8)$$

$$\sigma_u = \left[ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma(1+\beta/2)\beta 2^{\beta-1/2}} \right]^{1/\beta}, \sigma_v = 1, \quad (9)$$

where  $\beta$  is the parameter of the Lévy flight mutation algorithm,  $u$  is a random variable with a mean value of 0 and variance of  $\sigma_u^2$ , and  $V$  is a random variable with a mean

value of 0 and variance of  $\sigma_v^2$ . The expressions of  $\sigma_u$  and  $\sigma_v$  are shown in formula (9), where  $\Gamma$  is the gamma function.

The Lévy flight algorithm, because of its lack of global search ability and adaptive control ability, has a low frequency of generating better offspring in the later stage of evolution, thus reducing the effectiveness of portfolio optimization. To address these issues, we introduce a dynamic coefficient to explore the objective space globally, which improves the objective search ability. On the other hand, we utilize the joint coefficient to adaptively control exploration and exploitation for parent selection, which leads to the convergence of the algorithm.

**3.2. Global Search and Adaptive Control Ability.** The new solution generated in the subproblem in the MOEA/D algorithm replaces the old solution only in the corresponding neighborhood  $B(i)$ . The resulting new solution may not be suitable for the subproblem in the neighbourhood; so, the algorithm can easily fall into a local optimum. The objective search ability is very important for the exploration of the solution space. A strong objective search ability can make the algorithm jump out of a local optimal solution and find a better global optimal solution. We argue that the dynamic diversity of the algorithm is helpful in improving the

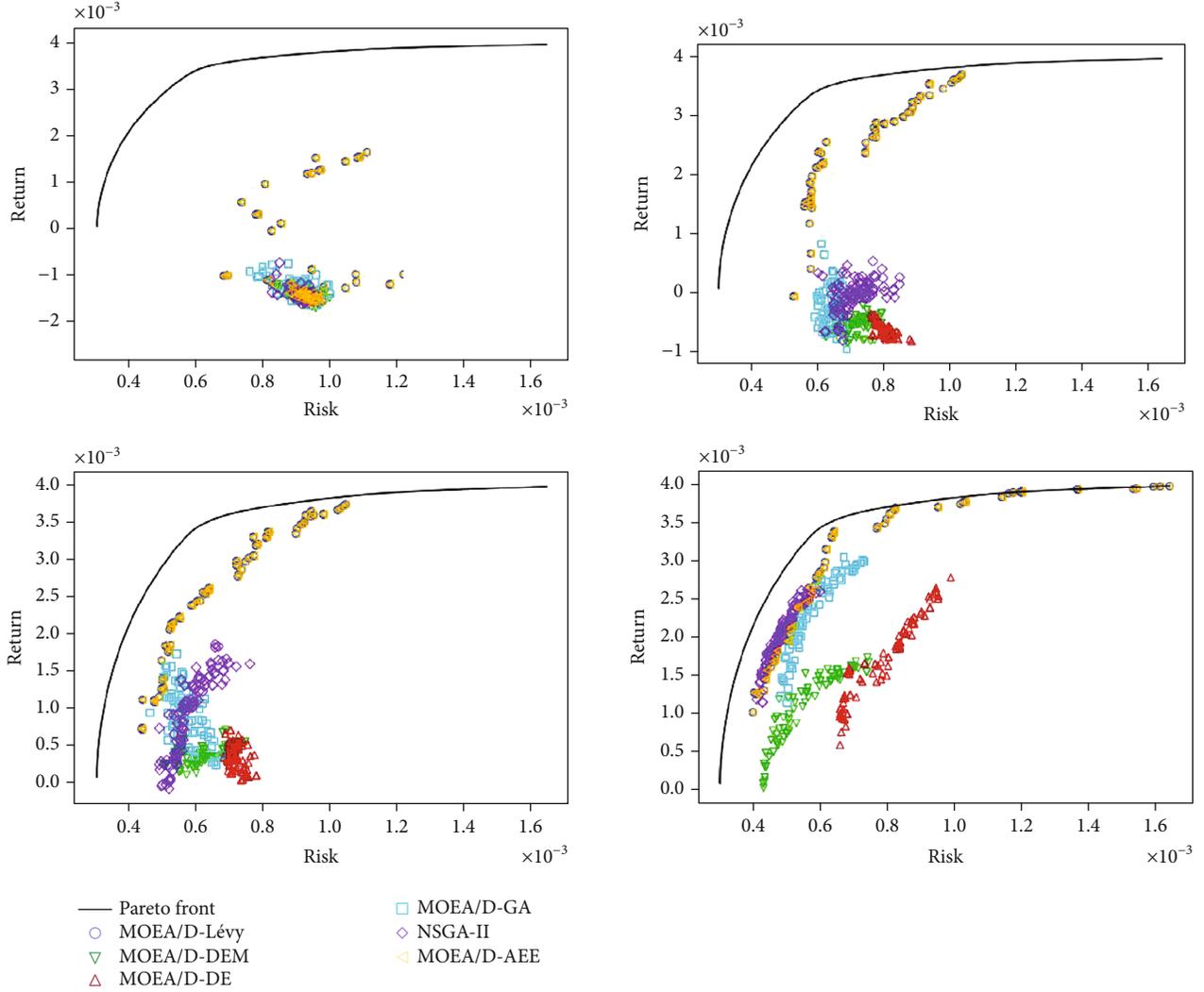


FIGURE 7: Population distribution (epsilon = 0.5).

objective search ability. To this end, we adopt a dynamic stochastic method to explore the objective space globally. That is, the amount of exploration is controlled globally by a dynamic parameter,  $\varepsilon(i)$ , that determines the randomness in parent selection. Thus, this objective search ability brings the offspring closer to the Pareto front. The mutation formula is composed of a nested polynomial mutation of the improved Lévy flight algorithm, and the formula is shown in (10), where  $x^i$  and  $x^j$  are the two parents selected in this study,  $y$  is the resulting offspring,  $\oplus$  is the method of multiplying item by item,  $\alpha_0$  is the scaling factor used, and Lévy ( $\beta$ ) is the vector generated by the MA (Mantegna's algorithm). The numbers are generated randomly under a symmetrical heavy-tailed distribution, and the algorithm data will be described in the fourth part of experiment 1. Our proposed method improves the global search ability by automatically adapting a dependent exploration probability,  $\varepsilon(i)$  at the beginning to obtain a global parameter by hand. The formula is as follows:

$$y = x^i \varepsilon + \alpha_0 (x^i - x^j) \oplus \text{Levy}(\beta). \quad (10)$$

Although the global search ability mentioned above can prevent the algorithm from falling into a local optimum, it cannot guarantee the convergence of the algorithm because it is easy to cause the algorithm to fail to converge by continuously exploring the solution space. This involves a trade-off between exploration and exploitation. The aim is to design an adaptive algorithm that can not only explore the generation of parents that are beneficial to the global optimal solution but also make full use of the current better generated parent.

An ideal solution is to automatically develop the solution space between the two parents according to the solution space under the constraint of the joint coefficient. Inspired by the value difference-based exploration (VDBE) method [27], we randomly adjust the parent relationship adaptively from uniformly distributed random variables. Specifically,

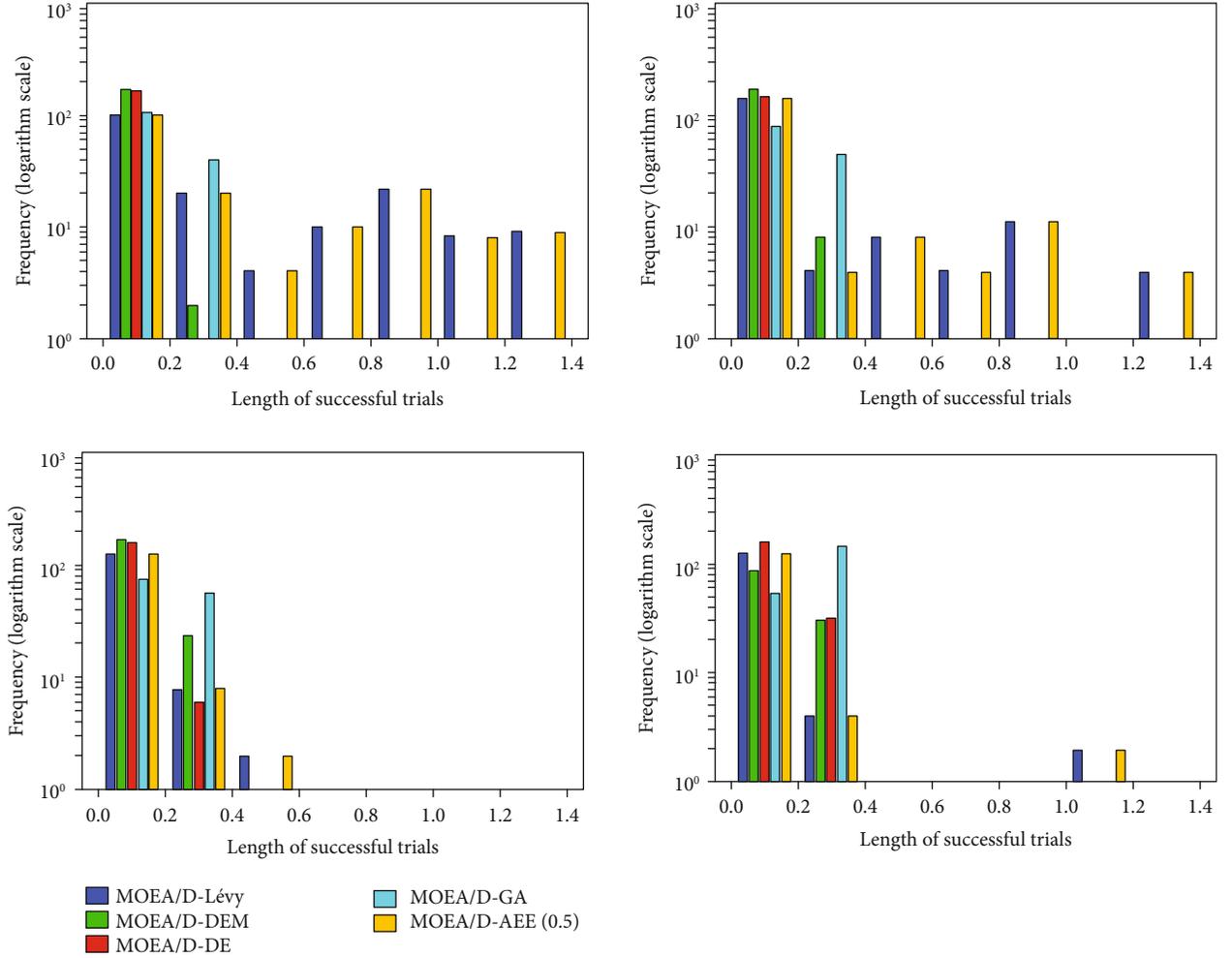


FIGURE 8: Successfully updated trajectory diagram (epsilon = 0.5).

we choose a random number epsilon between 0 and 1, and each time, we randomly select one of the parents with a probability of epsilon. If epsilon is less than  $\sigma$ , we select the parent from the neighbor; otherwise, we select the parent from the initial population. For choosing another parent in the initial population with a probability of  $(1-\text{epsilon})$ , the formula is as follows:

$$y = x^i \varepsilon + \alpha_0 (x^i - x^j) (1 - \varepsilon) \oplus \text{Levy}(\beta). \quad (11)$$

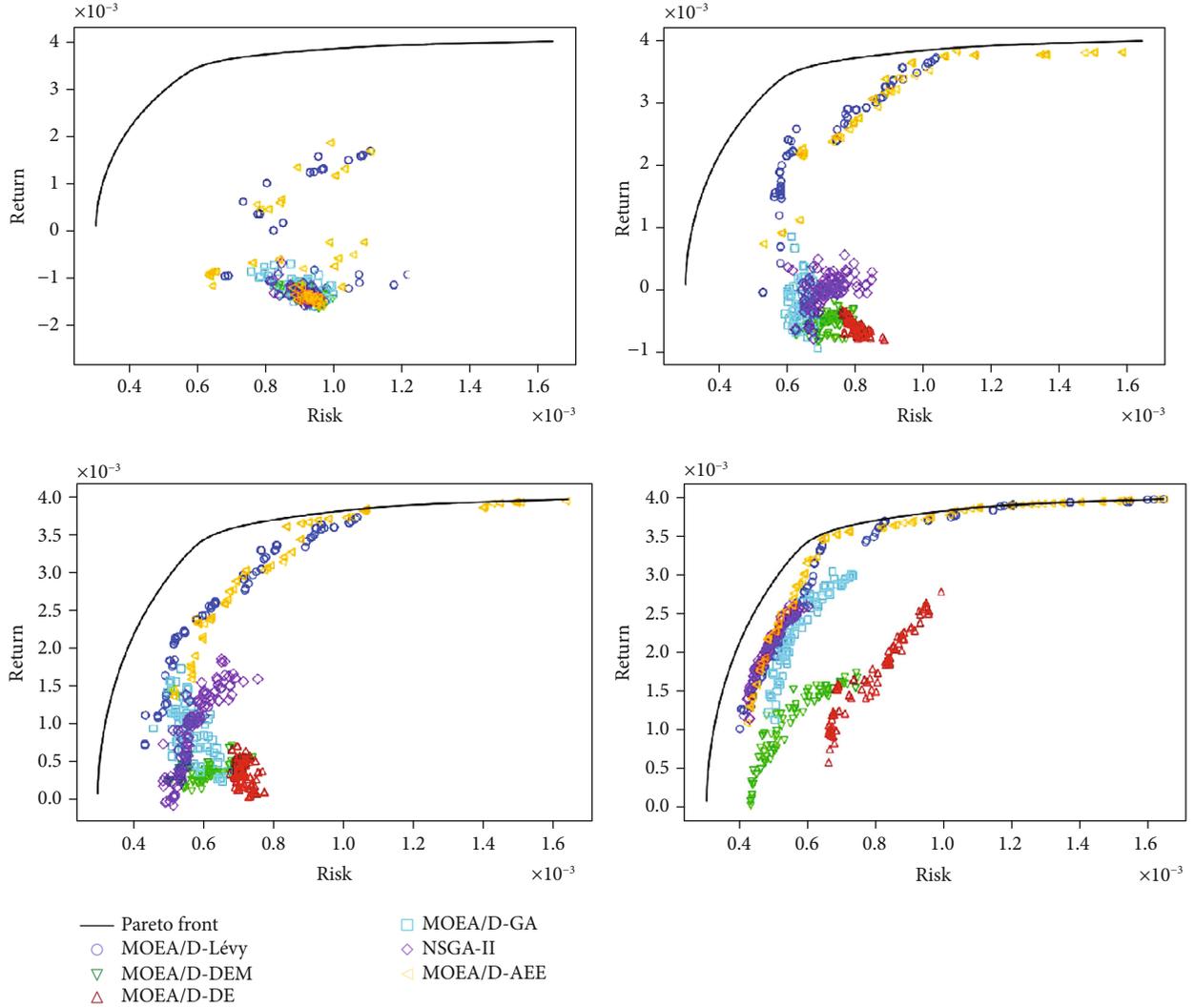
Our proposed method can achieve adaptive control of the degree of preference for exploration and exploitation with epsilon since each decision of whether to generate an offspring is explored by  $\varepsilon(i)$  availability and expanded by  $(1 - \varepsilon(i))$  availability. When two selected parents are similar, it adopts an active exploration strategy to select the parent from the initial population. In contrast, when the two selected parents are quite different, the algorithm gradually employs the exploitation strategy to select the parent from the neighbor. In this way, the algorithm can simultaneously search globally and converge. To prove the effectiveness of the algorithm, we select several sets of fixed joint coefficients

to develop and compare with the proposed method. The details are described in the fourth part of experiment 2.

**3.3. Algorithm Implementation.** This section embeds the above AEE method into the MOEA/D algorithm to form the MOEA/D-AEE algorithm. The pseudocode is shown in Algorithm 1 below.  $B$  represents the generated neighbors,  $P$  represents the generated initial population,  $F^1$  and  $F^2$  are the two extreme points of the Pareto front in the objective space, and  $\sigma$  adopts the diversity maintenance strategy in [2] by selecting the generated parent from the neighbors when it is less than  $\sigma$  and otherwise selecting the generated parent from the population. For the reproduction operation, this study uses the optimized Lévy algorithm nested polynomial mutation to generate offspring; the  $g^m$  Chebyshev discriminant method representing the NBI type introduced in part 2 is used, where  $r^p$  and  $\lambda$  can be calculated by formula (6), and  $n_r$  updates the upper bound of the neighbors.

Here are the detailed steps of the MOEA/D-AEE algorithm:

*Step 1.* Select the neighbor  $B$  based on parameters.

FIGURE 9: Population distribution ( $\epsilon = 0.2$ ).

*Step 2.* Initialize the population  $P$  based on parameters and compute objective [return, risk].

*Step 3.* Initialize the extreme points of the Pareto Front  $F^1$  and  $F^2$  based on objective [return, risk].

*Step 4.* Start the iteration of the algorithm based on parameters.

*Step 5.* Each iteration traverses each individual in the population to start execute.

*Step 6.* A random number will be generated and compared with the threshold  $\sigma$ . If the random number  $> \sigma$ , the entire population will be used as the mating pool; otherwise, the neighbors of the individual will be selected as the mating pool.

*Step 7.* According to the formula (11), use the algorithm proposed in this paper to generate the offspring  $y$  on line 12 in the pseudocode and obtain the objective [return, risk] of  $y$ .

*Step 8.* Update the extreme points of the Pareto Front  $F^1$  and  $F^2$ .

*Step 9.* Set counter  $nc$  and shuffle mating pool randomly.

*Step 10.* According to formula (5), compute the  $gtn$  value to update the individual and objective [return, risk].

*Step 11.* Exit the loop and return the updated objective [return, risk] when reach the criterion of convergence or reach the upper limit of the counter.

In addition, the offspring generated by the MOEA/D-AEE algorithm is likely to be repaired because the reproduction operation may not satisfy the constraint. Therefore, the steps of the repair method can be defined as follows:

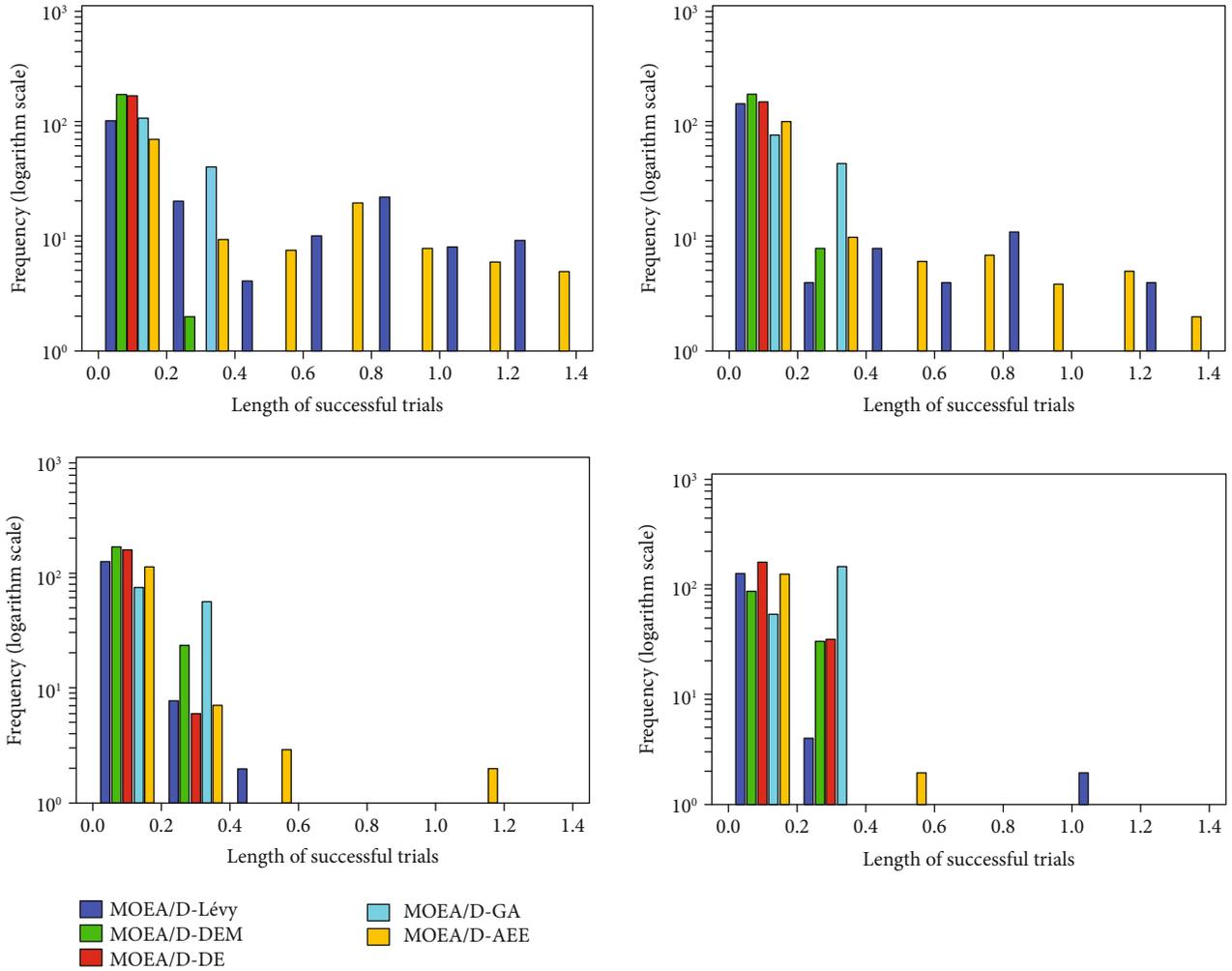


FIGURE 10: Successfully updated trajectory diagram (epsilon = 0.2).

*Step 1.* Traverse all the offspring, set all the negative offspring to 0, and keep the other offspring unchanged.

*Step 2.* Calculate the sum  $s$  of all offspring vectors.

*Step 3.* Traverse all the offspring again and scale each offspring vector; that is,  $y_i = y_i/s$ , so that the summation of all vectors equals to 1.

The algorithm we proposed needs to traverse each individual of the population in each iteration, so that the time complexity should be  $O(I \times P)$ , where  $I$  represents the number of iterations of the population, and  $P$  represents the size of the population.

It should be noted that the parameter settings of our algorithm and other comparison algorithms in the experiment 1 in chapter 4 are the same; that is, the number of iterations is 1500, and the population size is 100. It can be seen that the complexity of this algorithm is comparable to other algorithms, and it has not become significantly more complicated.

## 4. Experiments

*4.1. Preparation and Setup of Experiments.* The experiments are arranged in two parts, mainly to analyze the performance of the MOEA/D-AEE algorithm proposed in this study, and two experiments are carried out, as shown in 4.2 and 4.3. Experiment one compares several mainstream MOEA/D-based evolution algorithms, NSGA-II algorithms, Lévy flight mutation algorithms, and MOEA/D-AEE algorithms, while experiment two compares the effects of dynamic and static joint coefficients on the adaptability of MOEA/D-AEE algorithms.

*4.1.1. Data Sets.* One frequently used benchmark in assessments is the OR-library, which contains five PO data sets: Hangseng, DAX 100, FTSE 100, S&P 100, and Nikkei. The experiment was performed on three data sets disclosed by the OR-library, which contain 225, 85, and 89 test samples; the corresponding FTSE 100 data are from Japan Nikkei, Germany, and the United Kingdom; data set source: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html>.

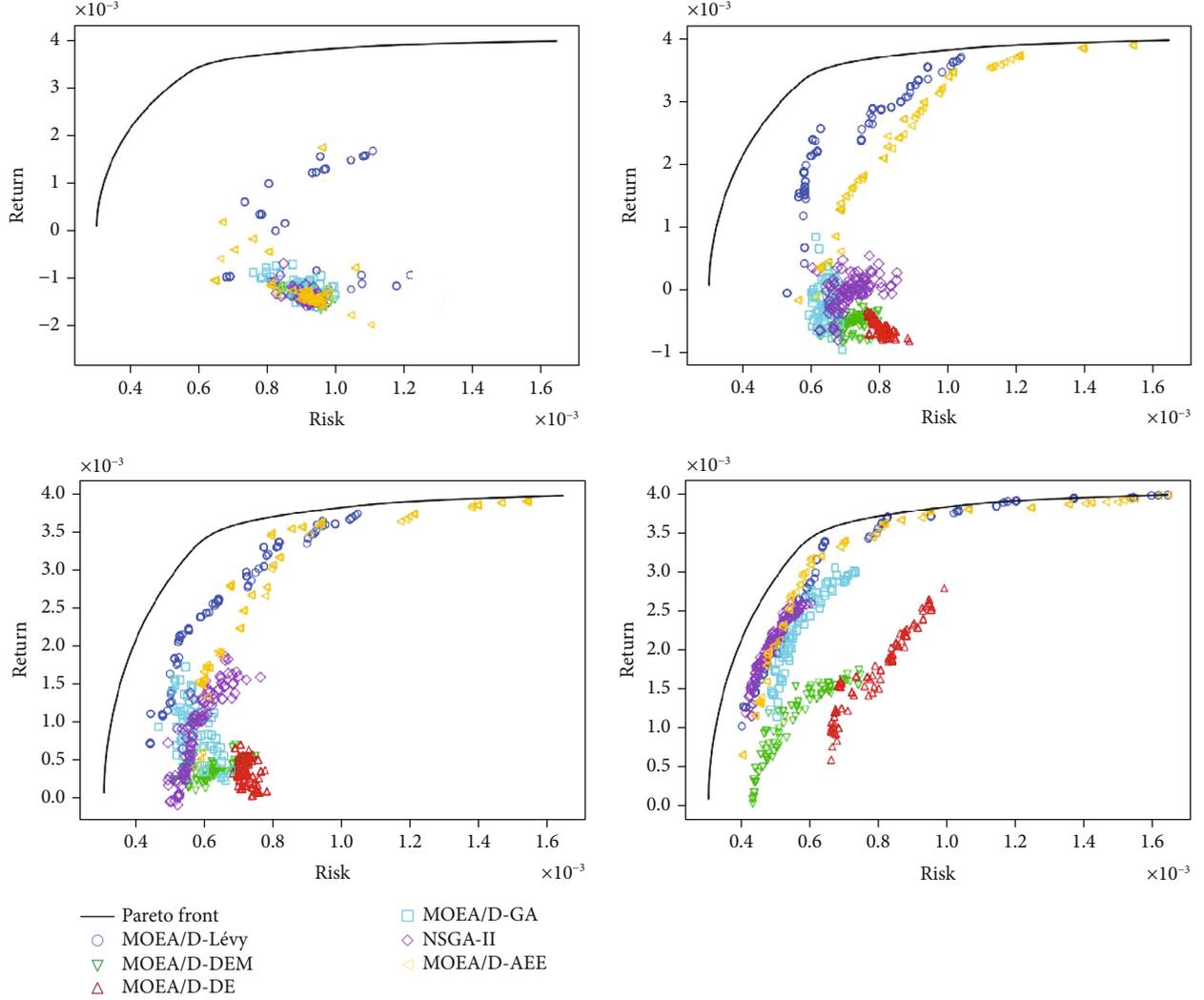


FIGURE 11: Population distribution (epsilon = 0.8).

**4.1.2. Metrics.** When measuring the performance and evaluation of the algorithm, the generated population is calculated, the statistical results of the six indexes of generational distance (GD), spacing, maximum spread, delta, inverted generational distance (IGD), and hypervolume are obtained, and the improvement of the algorithm is compared. Usually, this study uses GD to evaluate the convergence of the algorithm, delta, and maximum spread to evaluate the diversity of the algorithm, spacing to evaluate the uniformity of the algorithm, and IGD and hypervolume to comprehensively evaluate the convergence and diversity of the algorithm. The following are the formulas for the six indicators:

**Generational distance (GD):** it measures the average minimum distance from each point in the solution set  $P$  to the reference set  $P^*$ . The smaller the GD value is, the better the convergence of the algorithm.

$$GD(P, P^*) = \frac{\sqrt{\sum_{y \in P^*} \min_{x \in P} \text{dis}(x, y)^2}}{|P|}. \quad (12)$$

$P$  is the solution set obtained by the algorithm, and  $P^*$  is a set of uniformly distributed reference points sampled from the PF.  $\text{dis}(x, y)$  represents the Euclidean distance between point  $y$  in the solution set and point  $x$  in the reference set.

**Spacing(s):** This is the standard deviation to measure the minimum distance from each solution to the other solutions. The smaller the spacing value is, the more uniform the solution set of the algorithm.

$$S(P) = \sqrt{\frac{1}{|P| - 1} \sum_{i=1}^{|P|} (\bar{d} - d_i)^2}. \quad (13)$$

The  $d_i$  value of the parameters represents the minimum distance from solution  $i$  to the other solutions in  $P$ , and  $\bar{d}$  represents the mean value of all  $d_i$ .

**Maximum spread (MS):** The range of the solutions is measured by measuring the maximum range of each objective. The larger the MS value is, the greater the diversity of the representation algorithm.

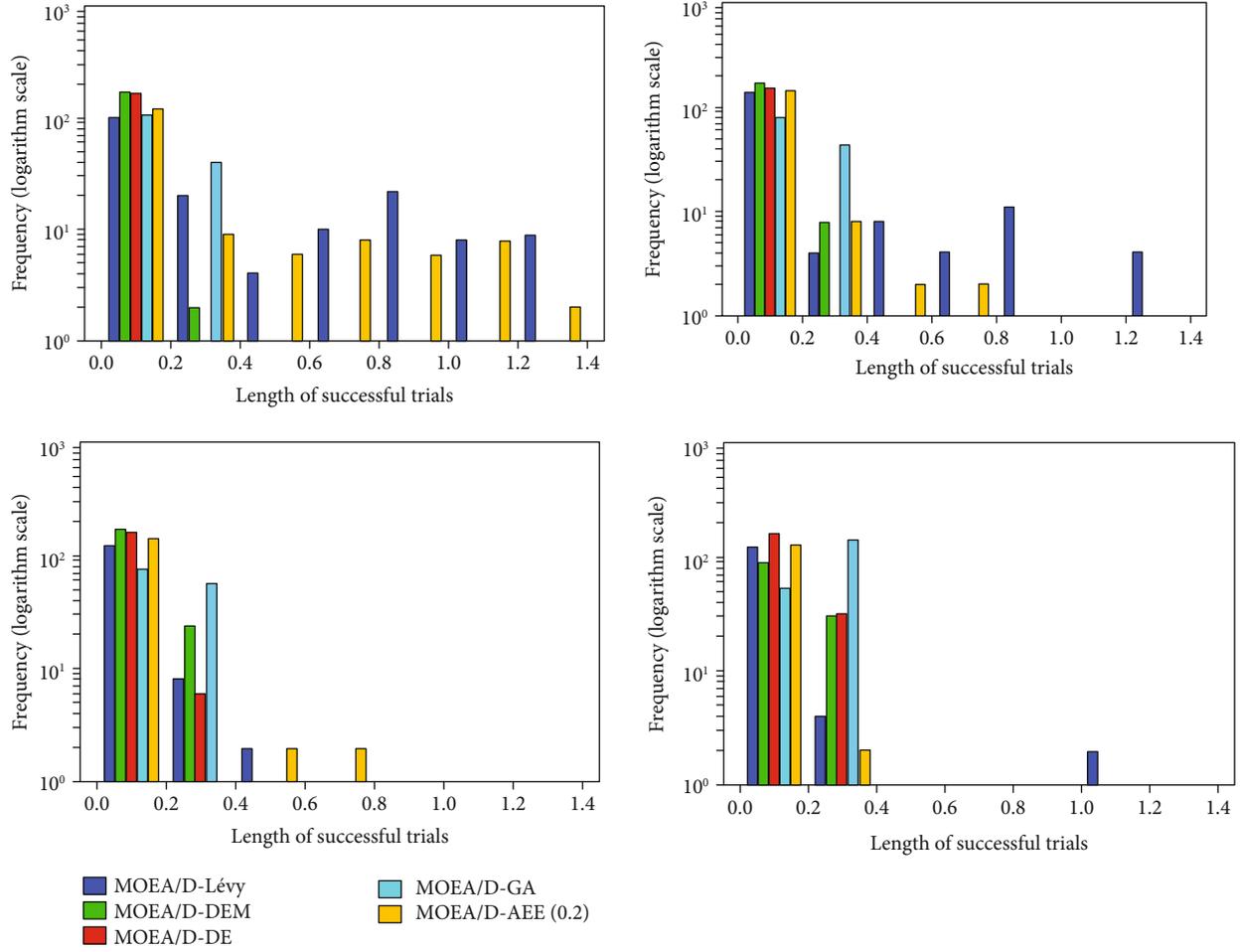


FIGURE 12: Successfully updated trajectory diagram (epsilon = 0.8).

$$MS(A) = \sqrt{\sum_{j=1}^m \max_{a, a' \in A} (a_j - a'_j)^2}, \quad (14)$$

where  $m$  represents the number of objectives, and the MS value of the nondominant solution set is the Euclidean distance of the  $m$  polar solutions.

Delta ( $d$ ): it measures the breadth of the solution set obtained. The smaller the delta value is, the greater the diversity of the representation algorithm.

$$D = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}, \quad (15)$$

where  $d_f$  and  $d_l$  are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set, and  $d_i$  is the set of Euclidean distances between the continuous solutions of the obtained nondominated set.

Inverted generational distance (IGD): the average of the distances is measured from each reference point to the nearest solution. The smaller the IGD value is, the better the

comprehensive performance of the algorithm. Recently, the IGD plus (IGD+) indicator has been proposed as a weakly Pareto compliant version of IGD by Hisao Ishibuchi et al. [28]. This indicator (called IGD+) shows some advantages over the original IGD, and we plan to use it in the future work.

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} \min_{y \in P} \text{dis}(x, y)}{|P^*|}. \quad (16)$$

$P$  is the solution set obtained by the algorithm,  $P^*$  is a set of uniformly distributed reference points sampled from the PF, and  $\text{dis}(x, y)$  represents the Euclidean distance between the points in  $P^*$  of the reference set and the points in the solution set.

Hypervolume (HV): This measures the volume of the region in the objective space between the obtained nonpersistent set and the reference point. The larger the HV value is, the better the comprehensive performance of the algorithm.

$$HV = \delta \left( \bigcup_{i=1}^{|S|} v_i \right). \quad (17)$$

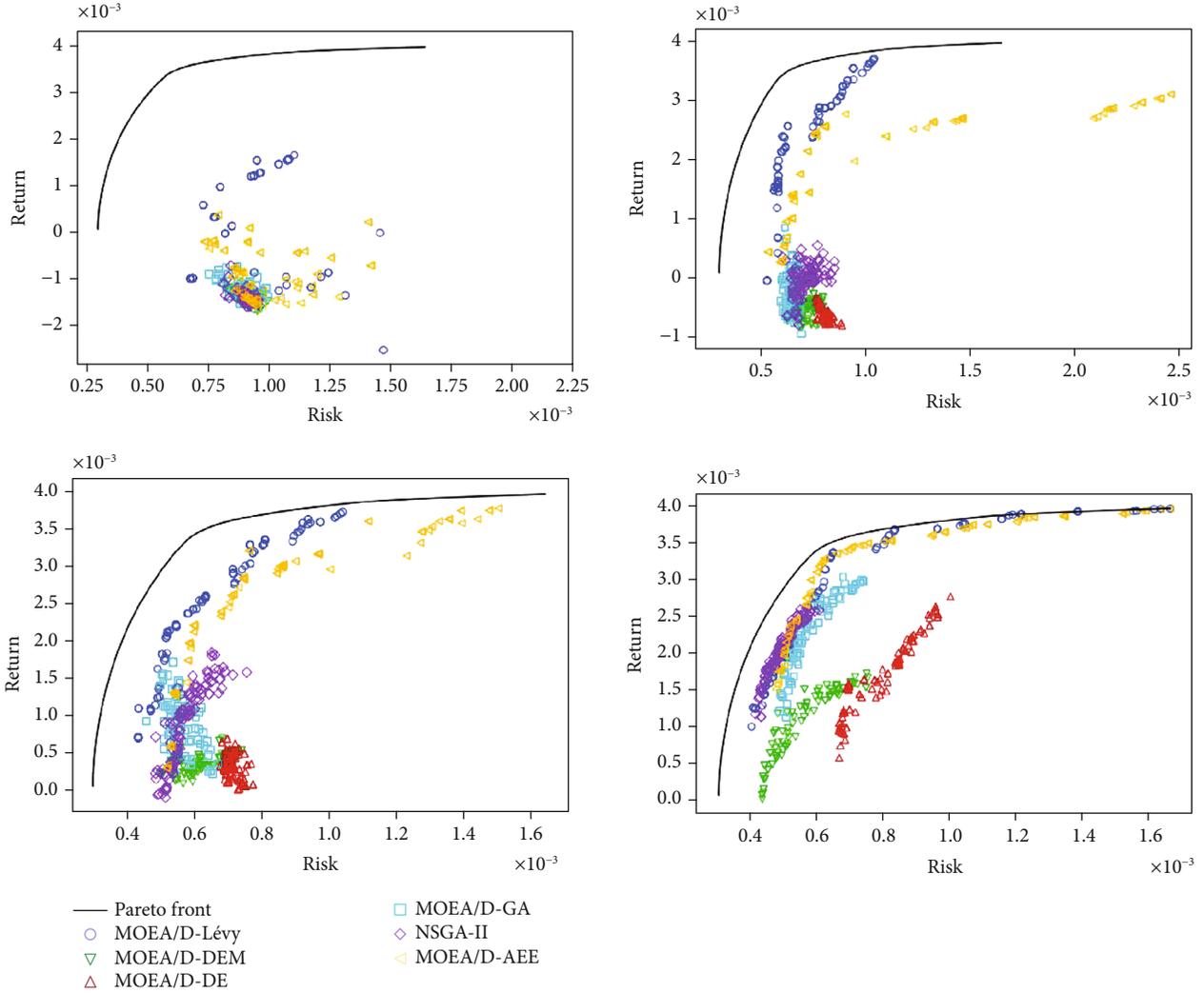


FIGURE 13: Population distribution (epsilon is a random number).

The  $I\lambda'$  parameter represents a Lebesgue measure used to measure volume.  $|S|$  represents the number of nondominant solution sets, and  $v_i$  represents the super volume of the reference point and solution  $i$  in the solution set.

**4.1.3. Parameter Settings.** All the algorithms perform 51 repetitions, the number of the population according to the common parameters is 100, the number of neighbors is 20, and each repetition needs to iterate through 1500 generations. At the same time, the threshold  $\sigma$  is 0.9 according to the selection strategy of the parent generation. If the uniformly distributed random number is greater than this number, the parent generation is selected from the whole population.  $N_r$  is still set to 2 as the upper bound of the updated neighbors.

With the exception of the epsilon correlation coefficient in the proposed MOEA/D-AEE algorithm, the settings are consistent with the parameters in the correlation algorithm paper [2]. For the specific parameters of each algorithm, the preexperiment method is used to fine tune the algorithm.

The unique parameters of each algorithm are obtained. The summary results of the parameters are shown in Table 1.

**4.2. Experiment 1-Comparison and Analysis of the Algorithm.** The MOEA/D-AEE algorithm and five related algorithms are compared in this experiment, including four MOEA/D evolution algorithms and classical NSGA-II algorithms. Other than NSGA-II, all of them use the NBI-Chebyshev decomposition method. Other than NSGA-II, the algorithms use the binary tournament method to select parents, and for the other four parents, this research setup is consistent with sigma in [2] so that the parent has a certain chance, when the random number uniformly generated between 0 and 1 is less than  $\sigma$ , of being selected from the neighbors; if the random number is greater than  $\sigma$ , then it is chosen from the whole population. DEM mutation is the DE operator plus polynomial mutation, which is different from the DE algorithm in this study. Simulated binary cross-mutation and polynomial mutation are used for MOEA/D-GA and NSGA-II.

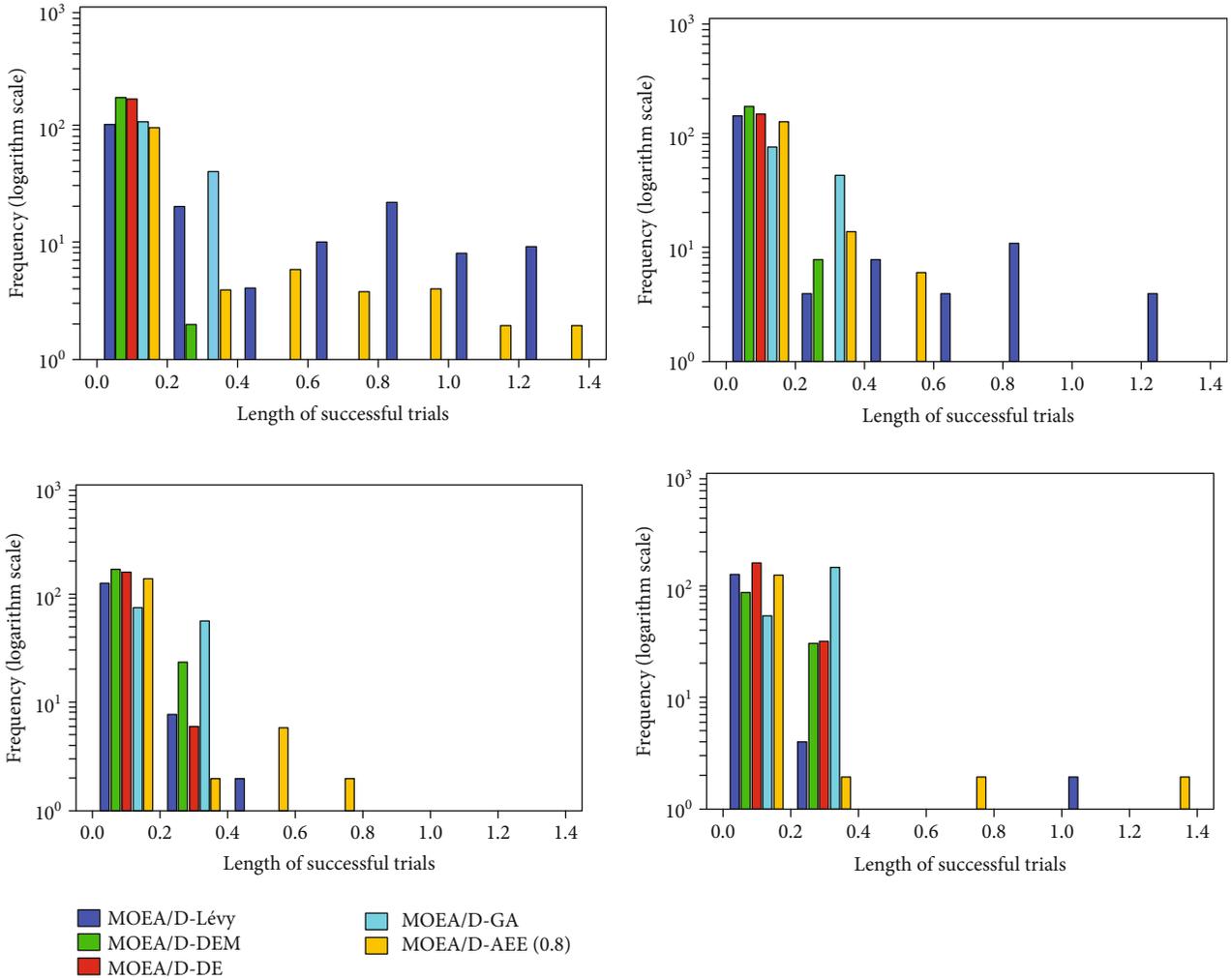


FIGURE 14: Successfully updated trajectory diagram (epsilon is a random number).

Figures 2–4 show the statistical results and population distribution of the IGD indicators. We can see from the IGD index of the data set that the MOEA/D-AEE algorithm proposed in this study has a relatively fast convergence rate in the early stage of algorithm iteration, especially on the DAX 100 data set, and the convergence is more obvious. The final population of the algorithm after iteration is very close to the Pareto front in the objective space, and the solution set in the objective space on the Pareto front is searched more completely. Compared with the incomplete search results of the MOEA/D-GA and MOEA/D-DE algorithms, the MOEA/D-AEE algorithm achieves good results. The main reason is caused by the mechanism that automatically develops the solution space between the two parents, according to the solution space under the constraint of the joint coefficient in our proposed approach. If epsilon is less than  $\sigma$ , we select the parent from the neighbor; otherwise, we select the parent from the initial population.

From the distribution of the 1-, 10-, 20-, and 50-generation populations in Figure 5, it can be seen that in the early stage of the algorithm iteration, MOEA/D-AEE

algorithms form many subsets, and the exploration of the objective space is more extensive; that is, the global search ability is stronger, and the population is closer to the Pareto front than that of other algorithms at the 20th-generation iteration, especially in high-yield and high-risk areas, showing a strong convergence trend. Better solution set distribution results can form at the later stage of iteration, and the performance is better than that of the MOEA/D-Lévy algorithm in the portfolio of high-risk assets, which verifies that the MOEA/D-AEE algorithms have strong global search ability, from the distribution of  $t$ .

In addition, in portfolio optimization, the trajectories represent the redistributed assets. Comparing the results of the successful update trajectories for 1, 3, 5, and 10 generations in Figure 6 (called successful update when the mutated offspring are better than the parents), it can be seen that the initialization is evenly distributed in the population with a low-risk objective space, and the large probability of variant offspring is successfully updated because the redistribution of the heavy-tailed distribution is concentrated on some assets rather than evenly distributed across all assets; so,

TABLE 2: Statistical results for Nikkei indicators.

Metric		MOEA/D-levy	MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II
GD	Best	5.32E-06	5.92E-06	4.71E-06	5.86E-06	9.95E-06	2.66E-06
	Median	7.36E-06	7.18E-06	9.21E-06	1.54E-04	3.31E-05	4.15E-06
	Std.	9.19E-07	1.17E-06	1.83E-04	7.86E-05	2.08E-04	1.31E-06
Spacing	Best	1.42E-05	1.27E-06	9.11E-06	5.02E-06	0.00E+00	1.02E-05
	Median	1.79E-05	1.73E-05	1.79E-05	1.15E-05	1.92E-05	1.43E-05
	Std.	4.87E-06	4.43E-06	6.54E-06	4.99E-06	9.73E-06	2.50E-06
Max spread	Best	4.17E-03	4.18E-03	4.15E-03	4.29E-03	2.63E-03	3.44E-03
	Median	3.93E-03	3.92E-03	3.89E-03	2.92E-03	2.20E-03	2.89E-03
	Std.	1.21E-04	1.19E-04	5.30E-04	6.42E-04	4.68E-04	2.86E-04
Delta	Best	3.87E-01	3.85E-01	4.04E-01	3.17E-01	8.40E-01	6.14E-01
	Median	4.42E-01	4.31E-01	4.91E-01	6.00E-01	9.36E-01	6.80E-01
	Std.	6.09E-02	5.34E-02	1.00E-01	1.34E-01	3.62E-02	2.94E-02
IGD	Best	1.72E-05	1.74E-05	2.02E-05	1.03E-04	1.77E-04	4.04E-05
	Median	2.47E-05	2.52E-05	2.99E-05	2.26E-04	2.43E-04	9.62E-05
	Std.	7.05E-06	7.55E-06	4.23E-04	1.41E-04	5.29E-04	4.30E-05
HV	Best	8.31E-06	8.31E-06	8.29E-06	7.87E-06	7.87E-06	8.22E-06
	Median	8.29E-06	8.29E-06	8.25E-06	7.12E-06	7.54E-06	7.99E-06
	Std.	1.06E-08	1.55E-08	9.65E-07	5.55E07	1.20E-06	1.17E-07

TABLE 3: Statistical results for DAX 100 indicators.

Metric		MOEA/D-levy	MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II
GD	Best	6.29E-06	6.60E-06	7.59E-06	6.75E-06	2.09E-06	4.68E-06
	Median	8.07E-06	8.06E-06	9.52E-06	2.09E-05	2.79E-06	8.18E-06
	Std.	9.03E-07	8.89E-07	2.88E-06	9.22E-05	6.44E-07	1.66E-06
Spacing	Best	2.59E-05	2.62E-05	2.28E-05	1.83E-05	1.59E-05	2.50E-05
	Median	3.38E-05	3.16E-05	3.23E-05	2.78E-05	2.43E-05	4.22E-05
	Std.	6.04E-06	6.06E-06	7.39E-06	7.99E-06	5.63E-06	5.28E-06
Max spread	Best	8.11E-03	8.16E-03	8.09E-03	8.24E-03	7.06E-03	7.84E-03
	Median	7.78E-03	7.79E-03	7.60E-03	7.33E-03	5.98E-03	7.29E-03
	Std.	2.12E-04	2.86E-04	2.29E-04	7.69E-04	3.43E-04	4.88E-04
Delta	Best	3.95E-01	3.88E-01	3.99E-01	3.95E-01	4.17E-01	5.00E-01
	Median	4.14E-01	4.07E-01	4.47E-01	4.48E-01	5.83E-01	6.61E-01
	Std.	2.18E-02	2.75E-02	3.07E-02	7.52E-02	3.96E-02	5.61E-02
IGD	Best	3.41E-05	3.32E-05	3.60E-05	3.84E-05	9.52E-05	4.16E-05
	Median	4.16E-05	4.11E-05	5.61E-05	8.83E-05	1.66E-04	5.87E-05
	Std.	1.42E-05	1.00E-05	1.95E-05	9.50E-05	3.64E-05	3.33E-05
HV	Best	1.87E-05	1.87E-05	1.87E-05	1.87E-05	1.87E-05	1.87E-05
	Median	1.87E-05	1.87E-05	1.87E-05	1.85E-05	1.80E-05	1.86E-05
	Std.	9.87E-09	9.44E-09	3.27E-08	1.10E-06	4.08E-07	4.05E-07

the solution distribution of the algorithm in the search space is more extensive than that of other algorithms. In contrast, the MOEA/D-Lévy algorithm relies more on local search to update the solution set. With the increase of iterations, the update of the long trajectories decreases, which indicates that the ability of global search is weakened. The MOEA/D-AEE algorithm introduces randomness and strengthens the performance of global search. In the process of algorithm iteration,

there is always a successful update of the long trajectories, even in the later period. To some extent, this makes up for the deficiency of updating the long trajectories after multiple iterations.

4.3. *Experiment 2: Comparison and Analysis of the Parameters.* Moreover, the experiment of adjusting the parameters is performed for the MOEA/D-AEE calculation

TABLE 4: Statistical results for FTSE 100 indicators.

Metric		MOEA/D-levy	MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II
GD	Best	5.27E-06	5.09E-06	6.13E-06	9.04E-06	2.72E-06	6.65E-06
	Median	7.05E-06	6.83E-06	8.79E-06	1.84E-05	4.61E-06	9.04E-06
	Std.	7.28E-07	1.32E-06	2.67E-06	1.70E-04	1.47E-06	1.02E-06
Spacing	Best	1.71E-05	1.70E-05	1.54E-05	7.22E-06	1.04E-05	2.25E-05
	Median	2.07E-05	2.06E-05	2.07E-05	1.71E-05	1.98E-05	2.94E-05
	Std.	3.67E-06	4.32E-06	4.00E-06	4.81E-06	5.45E-06	2.62E-06
Max spread	Best	5.96E-03	5.81E-03	5.84E-03	5.58E-03	5.35E-03	5.70E-03
	Median	5.56E-03	5.59E-03	5.45E-03	5.19E-03	4.87E-03	5.42E-03
	Std.	1.59E-04	1.22E-04	2.04E-04	5.21E-04	4.62E-04	1.52E-04
Delta	Best	4.01E-01	4.07E-01	4.13E-01	4.19E-01	4.34E-01	5.40E-01
	Median	4.33E-01	4.33E-01	4.67E-01	4.64E-01	5.06E-01	6.04E-01
	Std.	2.02E-02	2.60E-02	3.19E-02	5.03E-02	6.87E-02	3.13E-02
IGD	Best	2.30E-05	2.43E-05	2.64E-05	4.89E-05	5.30E-05	3.19E-05
	Median	3.71E-05	3.44E-05	5.09E-05	8.79E-05	8.55E-05	4.59E-05
	Std.	1.14E-05	9.27E-06	1.92E-05	1.69E-04	4.97E-05	1.22E-05
HV	Best	1.37E-05	1.37E-05	1.37E-05	1.37E-05	1.37E-05	1.37E-05
	Median	1.37E-05	1.37E-05	1.37E-05	1.36E-05	1.33E-05	1.37E-05
	Std.	5.08E-09	8.62E-09	1.73E-08	1.54E-06	6.34E-07	8.84E-08

TABLE 5: Statistical results for Nikkei indicators.

Metric		MOEA/D-AEE	MOEA/D-AEE (0.5)	MOEA/D-AEE (0.2)	MOEA/D-AEE (0.8)
GD	Best	5.92E-06	5.32E-06	6.22E-06	4.96E-06
	Median	7.18E-06	7.36E-06	7.76E-06	7.18E-06
	Std.	1.17E-06	9.19E-07	1.16E-06	1.31E-06
Spacing	Best	1.27E-05	1.42E-05	1.26E-05	1.31E-05
	Median	1.73E-05	1.79E-05	1.76E-05	1.72E-05
	Std.	4.43E-06	4.87E-06	4.75E-06	4.33E-06
Max spread	Best	4.18E-03	4.17E-03	4.09E-03	4.5E-03
	Median	3.92E-03	3.93E-03	3.91E-03	3.89E-03
	Std.	1.19E-04	1.21E-04	1.32E-04	1.63E-04
Delta	Best	3.85E-01	3.87E-01	3.93E-01	3.93E-01
	Median	4.31E-01	4.42E-01	4.35E-01	4.42E-01
	Std.	5.34E-02	6.09E-02	3.17E-02	4.47E-02
IGD	Best	1.74E-05	1.72E-05	1.93E-05	1.74E-05
	Median	2.52E-05	2.47E-05	2.59E-05	2.69E-05
	Std.	7.55E-06	7.05E-06	1.06E-05	1.49E-05
HV	Best	8.31E-06	8.31E-06	8.31E-06	8.32E-06
	Median	8.29E-06	8.29E-06	8.28E-06	8.29E-06
	Std.	1.55E-08	1.06E-08	1.21E-08	1.94E-08

method. Four groups of experimental results are obtained by taking the epsilon correlation coefficients as 0.2, 0.5, and 0.8 and random numbers between 0 and 1.

Compared with the above parameter settings, dynamic and static parameters more strongly affect the performance of the algorithm. If the joint coefficient epsilon is 0.5, Figures 7 and 8 show that the performance of the algorithm is comparable to that of MOEA/D-Lévy. At this point, the

weight of the joint coefficient applied to the two parents is the same, and this does not take advantage of automatic joint development. Figures 9 and 10 show the performance of the algorithm when epsilon is 0.2, and Figures 11 and 12 show the performance of the algorithm when epsilon is 0.8. The weight applied to the two parents is focused, the MOEA/D-AEE performance has improved, the distribution of solution sets is extensive, and the scope of exploration

TABLE 6: Statistical results for DAX 100 indicators.

Metric		MOEA/D-AEE	MOEA/D-AEE (0.5)	MOEA/D-AEE (0.2)	MOEA/D-AEE (0.8)
GD	Best	6.60E-06	6.29E-06	6.16E-06	6.25E-06
	Median	8.06E-06	8.07E-06	7.74E-06	8.23E-06
	Std.	8.89E-07	9.03E-07	7.19E-07	8.08E-07
Spacing	Best	2.62E-05	2.59E-05	2.78E-05	2.86E-06
	Median	3.16E-05	3.38E-05	3.20E-05	3.55E-05
	Std.	6.06E-06	6.04E-06	5.44E-06	5.16E-06
Max spread	Best	8.16E-03	8.11E-03	8.14E-03	8.10E-03
	Median	7.79E-03	7.78E-03	7.76E-03	7.83E-03
	Std.	1.86E-04	2.12E-04	1.63E-04	1.65E-04
Delta	Best	3.88E-01	3.95E-01	3.93E-01	3.91E-01
	Median	4.07E-01	4.14E-01	4.11E-01	4.18E-01
	Std.	2.75E-02	2.18E-02	1.93E-02	2.44E-02
IGD	Best	3.21E-05	3.41E-05	3.42E-05	3.45E-05
	Median	4.11E-05	4.16E-05	4.25E-05	4.04E-05
	Std.	1.00E-05	1.42E-05	8.58E-06	8.19E-06
HV	Best	1.87E-05	1.87E-05	1.87E-05	1.87E-05
	Median	1.87E-05	1.87E-05	1.87E-05	1.87E-05
	Std.	9.44E-09	9.87E-09	8.11E-09	19.18E-09

TABLE 7: Statistical results for FTSE 100 indicators.

Metric		MOEA/D-AEE	MOEA/D-AEE (0.5)	MOEA/D-AEE (0.2)	MOEA/D-AEE (0.8)
GD	Best	5.09E-06	5.27E-06	5.46E-06	5.62E-06
	Median	6.83E-06	7.05E-06	7.12E-06	7.65E-06
	Std.	1.32E-06	7.28E-07	1.37E-06	1.27E-06
Spacing	Best	1.70E-05	1.71E-05	1.63E-05	1.62E-05
	Median	2.06E-05	2.07E-05	2.10E-05	2.14E-05
	Std.	4.32E-06	3.76E-06	4.79E-06	4.67E-06
Max spread	Best	5.81E-03	5.96E-03	5.87E-03	5.84E-03
	Median	5.59E-03	5.56E-03	5.53E-03	5.58E-03
	Std.	1.22E-04	1.59E-04	1.85E-04	1.56E-04
Delta	Best	4.07E-01	4.01E-01	4.10E-01	4.01E-01
	Median	4.33E-01	4.33E-01	4.38E-01	4.37E-01
	Std.	2.60E-02	2.02E-02	2.01E-02	2.63E-02
IGD	Best	2.43E-05	2.30E-06	2.35E-05	2.29E-05
	Median	3.44E-05	3.71E-05	3.88E-05	3.52E-05
	Std.	9.27E-06	1.14E-06	1.62E-05	1.16E-05
HV	Best	1.37E-05	1.37E-05	1.37E-05	1.37E-05
	Median	1.37E-05	1.37E-05	1.37E-05	1.37E-05
	Std.	8.62E-09	5.08E-09	9.81E-09	8.72E-09

in the objective space is larger. At the same time, the possibility of successful updating of the long trajectories increases. As the joint coefficient epsilon changes to dynamically take random variables from a uniform distribution, as shown in Figures 13 and 14, it further improves the performance of the later iteration and adaptive allocation of weights to achieve a better global search capability. Through the experiment two, the importance of the joint coefficient adapting to the algorithm performance can be obtained, which is also

the reason why the automatic joint coefficient is introduced in this study.

## 5. Discussion

### 5.1. Experiment 1

5.1.1. *Statistical Results.* As seen from the above data analysis, the MOEA/D-AEE algorithm proposed in this study

(which uses obedience and uniformly distributed random numbers in  $(0, 1)$  to explore the objective space and the joint coefficient epsilon to establish a relationship between the two parents chosen, applies dynamic search and joint development, and solves global search and adaptive problems) performed better on multiple indicators. The above statistics show that in experiment 1, the MOEA/D-AEE algorithm obtains the top 10 indicators on the three data sets represented in Tables 2–4 (of which 4 are tied for first). Comparing the three data sets, significant progress is made in the diversity of the indicators delta and maximum spread, which indicates that there is a significant improvement in the global search ability. The convergence represented by the GD index is not as good as that based on the GA algorithm, but it is still equal to or better than that based on Lévy flight variation. At the same time, some progress is made in the comprehensive IGD indicators, and it is proven that the convergence performance of this algorithm is not reduced by the improvement in the global search ability. The potential reason is that the joint coefficient epsilon improves both global search ability and diversity of the algorithm. It is clear that MOEA/D-AEE outperforms other competitive algorithms in terms of the max spread, delta, IGD, and HV in most cases. This is because the proposed mechanism improves overall performance, which is automatically develop the solution space between the two parents according to the solution space under the constraint of the joint coefficient.

5.2. *Experiment 2.* As shown in Tables 5–7, by comparing the effects of different joint coefficients on MOEA/D-AEE algorithms in experiment two, the statistics above and an epsilon is 0.2 indicate that the MOEA/D-AEE algorithm obtained two top indexes (two of which are tied for first). When epsilon is 0.5, the MOEA/D-AEE algorithm obtains 4 top indexes (3 of which are tied for first), and when epsilon is 0.8, the MOEA/D-AEE algorithm obtained 7 top indexes (3 of which are tied for first). The standard MOEA/D-AEE algorithm obtained 13 top indexes (of which 3 are tied for first). Except for the average performance on the Nikkei data set, the other two data sets show good performance; in particular, the best value of all the indicators is achieved on the FTSE 100 data set. The adaptive ability is affected by the joint coefficient, and convergence and diversity can be better achieved when the coefficient is not fixed. It is proven that the spatial effect of using random numbers uniformly distributed in  $(0, 1)$  in this study is beneficial.

## 6. Conclusions

In this paper, we propose a MOEA/D-AEE algorithm to solve the self-optimization issue of multiple level datasets for complex data analysis. First, the proposed algorithm adopts dynamic search and joint development technology, that is, random numbers uniformly distributed in  $(0, 1)$ , to explore the solution space globally. We then introduce a joint coefficient between parents. As a result, the global search ability and adaptive ability of the algorithm are improved by automatically adjusting the parent relationship.

Comparative experiments are performed on three PO data sets in the OR-Library. The experimental results and analysis show that the MOEA/D-AEE algorithm leads in performance on six metrics. The global search ability is further improved, especially in the late iterations, in which long trajectories are still updated. The population aggregates to the Pareto front more quickly in the first 20 generations, and the method performs better on the three test platforms.

Our future research will incorporate an unsupervised clustering algorithm and reinforcement learning algorithm into machine learning, with the aim of further optimizing the joint coefficient in the mutation algorithm to improve the efficiency of the algorithm. This study may also introduce multitask deep models and continuously optimize algorithms in other application fields, employing our algorithm as a practical tool for multiobjective optimization in the Industry 4.0 era.

## Data Availability

One frequently used benchmark in assessments is the OR-library, which contains five PO data sets: Hangseng, DAX 100, FTSE 100, S&P 100 and Nikkei: data set source: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] 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.
- [2] 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.
- [3] J. G. Falcón-Cardona and C. A. Coello, "Indicator-based multi-objective evolutionary algorithms," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35, 2020.
- [4] L. P. Cota, F. G. Guimares, R. G. Ribeiro, I. R. Meneghini, and P. Siarry, "An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem," *Swarm and Evolutionary Computation*, vol. 51, no. 1, p. 100601, 2019.
- [5] A. L. Soubhia and A. L. Serpa, "Discrete optimization for positioning of actuators and sensors in vibration control using the simulated annealing method," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 2, 2020.
- [6] R. M. Khusainov, R. R. Mindiyarov, and B. I. Petrovetskiy, "Solving optimization problems in the design of fixtures for flexible manufacturing systems," *IOP Conference Series Materials Science and Engineering*, vol. 971, article 032063, 2020.
- [7] Y. He and C. Aranha, "Solving portfolio optimization problems using MOEA/D and Lévy flight," *Advances in Data Science and Adaptive Analysis*, vol. 12, p. 2050005, 2020.
- [8] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Lévy flights trajectory," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 453812, 13 pages, 2013.

- [9] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Levy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.
- [10] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-like Tchebycheff approach for Portfolio Management," in *IEEE Congress on Evolutionary Computation*, pp. 1–8, Barcelona, Spain, July 2010.
- [11] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *2009 IEEE Congress on Evolutionary Computation*, pp. 203–208, Trondheim, Norway, 2009.
- [12] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [13] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.
- [14] W. Wang, K. Li, X. Tao, and G. Fahui, "An improved MOEA/D algorithm with an adaptive evolutionary strategy," *Information Sciences*, vol. 539, pp. 1–15, 2020.
- [15] T. C. Chiang and Y. P. Lai, "MOEA/D-AMS: improving MOEA/D by an adaptive mating selection mechanism," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2011*, pp. 1473–1480, New Orleans, LA, USA, 2011.
- [16] W. Zheng, Y. Tan, L. Meng, and H. Zhang, "An improved MOEA/D design for many-objective optimization problems," *Applied Intelligence*, vol. 48, no. 10, pp. 3839–3861, 2018.
- [17] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D + uniform design: a new version of MOEA/D for optimization problems with many objectives," *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.
- [18] Y. Zhang, D. Gong, J. Sun, and Q. Boyang, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Information Sciences*, vol. 430–431, pp. 397–413, 2018.
- [19] Y. Zhang, S. Cheng, Y. Shi, D. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.
- [20] A. K. Shukla, R. Nath, P. K. Muhuri, and Q. M. D. Lohani, "Energy efficient multi-objective scheduling of tasks with interval type-2 fuzzy timing constraints in an Industry 4.0 ecosystem," *Engineering Applications of Artificial Intelligence*, vol. 87, no. 1, p. 103257, 2020.
- [21] Y. P. Fu, H. F. Wang, and M. Huang, "Integrated scheduling for a distributed manufacturing system: a stochastic multi-objective model," *Enterprise Information Systems*, vol. 13, no. 4, pp. 557–573, 2019.
- [22] Y. Fu, J. Ding, H. Wang, and J. Wang, "Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in Industry 4.0-based manufacturing system," *Applied Soft Computing*, vol. 68, pp. 847–855, 2018.
- [23] A. Goti, A. Oyarbide-Zubillaga, E. Alberdi, A. Sánchez, and P. Garcia-Bringas, "Optimal maintenance thresholds to perform preventive actions by using multi-objective evolutionary algorithms," *Applied Sciences*, vol. 9, no. 15, p. 3068, 2019.
- [24] A. Dutta, C. Jatoth, G. R. Gangadharan, and U. Fiore, "QoS-aware big service composition using distributed co-evolutionary algorithm," *Concurrency and Computation, Practice and Experience*, vol. 33, no. 19, 2021.
- [25] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: analyzing the state-of-the-art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.
- [26] Q. Xu, Z. Xu, and T. Ma, "A short survey and challenges for multiobjective evolutionary algorithms based on decomposition," in *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–5, Beijing, China, 2019.
- [27] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," in *Proceedings of the 34th Annual German Conference on Advances in Artificial Intelligence*, pp. 335–346, Berlin, Germany, 2011.
- [28] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Comparison of hypervolume," in *IGD and IGD+ from the Viewpoint of Optimal Distributions of Solutions. International Conference on Evolutionary Multi-criterion Optimization*, pp. 332–345, Springer, Cham, 2019.