Applied Intelligence (2023) 53:11439–11473
https://doi.org/10.1007/s10489-022-03964-9

Improving evolutionary algorithms with information feedback model for large-scale many-objective optimization

Yong Wang[1] · Qian Zhang[1] · Gai-Ge Wang[1]

Accepted: 4 July 2022 / Published online: 6 September 2022

Abstract

Recently, many evolutionary algorithms have been proposed. Compared to other algorithms, the core of the many-objective evolutionary algorithm using a one-by-one selection strategy is to select offspring one by one in environmental selection. However, it does not perform well in resolving large-scale many-objective optimization problems. In addition, a large amount of meaningful information in the population of the previous iteration is not retained. The information feedback model is an effective strategy to reuse the information from previous populations and integrate it into the update process of the offspring. Based on the original algorithm, this paper proposes a series of many-objective evolutionary algorithms, including six new algorithms. Experiments were carried out in three different aspects. Using the same nine benchmark problems, we compared the original algorithm with six new algorithms. Algorithms with excellent performance were selected and compared with the latest studies using the information feedback model from two aspects. Then, the best one was selected for comparison with six state-of-the-art many-objective evolutionary algorithms. Additionally, non-parametric statistical tests were conducted to evaluate the different algorithms. The comparison, with up to 15 objectives and 1500 decision variables, showed that the proposed algorithm achieved the best performance, indicating its strong competitiveness.

Keywords Many-objective · Information feedback model · One-by-one selection · Evolutionary algorithms · Large-scale optimization

1 Introduction

Owing to the needs of engineering applications and scientific experimental research, there are many multi-objective optimization problems (MOPs) in real life. The main optimization goal is to maximize or minimize the final value with as few resources as possible [1]. The number of objective functions of MOPs increases to two or three, unlike in the single-objective optimization problem [2]. However, the objective function for many optimization problems is not limited to three or fewer. As a result, many-objective optimization problems (MaOPs) with more than three goals arise [3]. There is often no ideal solution to achieve optimal values for all goals. At present, methods to solve optimization problems are generally divided into two categories: traditional mathematical optimization methods (e.g., gradient descent and Newton) and evolutionary algorithms [4].

Traditional optimization algorithms that use information on the first derivative and second derivatives of related functions are often unsuitable for practical application problems and cannot guarantee that the final approximate optimal solution set can be evenly distributed and converged [5]. To solve this problem, many researchers have proposed many evolutionary algorithms. These algorithms draw inspiration from biological evolution and guide the process of finding solutions by simulating the process of biological evolution, using the survival of the fittest principle in genetics [6, 7]. Such as genetic algorithm (GA) [8], particle swarm optimization (PSO) [9], artificial bee colony (ABC) [10], ant colony algorithm (ACO) [11] and differential evolution (DE) [12].

✉ Gai-Ge Wang
wgg@ouc.edu.cn

Yong Wang
markwy@126.com

Qian Zhang
zhangqaq@stu.ouc.edu.cn

[1] School of Computer Science and Technology, Ocean University of China, Qingdao, 266100, China

Many scholars have proposed numerous multi-objective optimization algorithms (MOEAs) to solve MOPs. The most famous ones are the decomposition-based multi-objective evolutionary algorithms (MOEA/D) [13], genetic algorithms based on non-dominated sorting and elite selection (NSGA-II) [14], evolutionary algorithms using neighborhood environment selection (SPEA2) [15], etc. With the advent of 5G era and the rise of deep learning, evolutionary algorithms have applications in many emerging fields [16]. They are currently widely used in economic dispatching [17], code detection [18], engineering optimization [19], face reconstruction [20], job-shop scheduling [21–23], high performance computing [24], computer network [25], and other fields. These MOEAs have been proved successful in solving most MOPs. However, the non-dominated solution set obtained by these algorithms often cannot effectively approximate the real Pareto front of MaOPs with optimization goals greater than three [26]. Therefore, considering the improvement of the convergence and diversity of the algorithms, this paper examines the many-objective evolutionary algorithm using a one-by-one selection (1by1EA) [27], aiming to improve its optimization performance in large-scale MaOPs. The method uses a convergence index to select the solutions in the current population one by one, adopts a distribution index based on cosine similarity to evaluate the similarity between solutions, and proposes the boundary maintenance mechanism of corner solutions to maintain diversity.

However, few researchers have focused on the utility of prior knowledge and historical information. Many have adopted the method of retaining the best individual in the process of each iteration and discarding all other individuals. This is not conducive to the development of population diversity and the preservation of information. More importantly, most MOEAs focus on solving optimization problems with less than 100 decision variables. The problem of more than a thousand decision variables in engineering application research may also exist. Therefore, we have to enhance 1by1EA to improve the result. Therefore, this paper introduces the information feedback model (IFM) [28] into the framework of 1by1EA and proposes a new optimization algorithm called 1by1EA-IFM, which solves large-scale MaOPs. Unlike most existing algorithms, this algorithm retains the historical information of individuals in the population before selecting offspring individuals. The experiment was carried out in three aspects. First, the performance of the improved six algorithms was compared with the original algorithm. Second, algorithms with excellent performance were selected and compared with current IFM research from two levels. Finally, we chose the one with the best performance and compared it with the other six state-of-the-art algorithms from a longitudinal perspective.

The main contributions of this work can be summarized as follows:

(1) A novel framework, called 1by1EA-IFM is proposed, where individuals from historical populations are retained in two ways: either in a fixed location or randomly retained. The individual performance is determined by the fitness function value. In each retaining approach, k individuals are selected ($k \in\{1,2,3\}$). Based on the value of k, IFM includes six models (i.e., M-F1, M-F2, M-F3, M-R1, M-R2, and M-R3), which are combined with 1by1EA to present six algorithms, namely 1by1EA-F1, 1by1EA-F2, 1by1EA-F3, 1by1EA-R1, 1by1EA-R2, and 1by1EA-R3. An advantage of IFM is that it can efficiently enhance the diversity of the population by reusing some information from historical populations.

(2) Experiments were performed on large-scale MaOPs, where the number of decision variables reaches up to 1500. By controlling the number of decision variables, we can improve the scale of the problem, so as to test whether the algorithms can effectively target large-scale problems. In all the validation experiments, the comparison of the indicator values involving the different algorithms was carried out using the Mann-Whitney-Wilcoxon rank-sum test, which can determine whether one algorithm has statistical difference with other algorithms. In addition, with the intention of obtaining rigorous conclusions, two different statistical tests, Friedman's non-parametric test and Holm's post-hoc test, were conducted.

The rest of this paper is organized as follows. In Section 2, we introduce the work related to this research. Section 3 provides some basic concepts about multi/many-objective optimization, as well as the original 1by1EA and IFM. The proposed 1by1EA-IFM is described in detail in Section 4. The settings and results related to the experiment are described in Section 5. Section 6 provides a summary of this paper.

2 Related work

There are a variety of optimization problems with more than three objectives in social life, and the corresponding solution is the many-objective evolutionary algorithms (MaOEAs). For most evolutionary algorithms that rely on Pareto dominance, the non-dominant solution sets often cannot effectively approximate the real Pareto front of MaOPs. Therefore, redefining the dominance relationship is an effective measure to solve MaOPs for MOEAs. The main studies were ε domination [29], fuzzy Pareto domination

[30] and so on. Chhabra et al. [31] proposed a fuzzy Pareto-driven artificial swarm algorithm (FP-ABC) to better solve for MaOPs. In the FP-ABC algorithm, two external files are combined into the artificial bee colony algorithm to improve the performance of the ABC algorithm. In 2022, Wu et al. [29] introduced a new ε dominance relation and proposed a new MaOEAs (ε-Two_Arch2) to update the individuals in diversity archive. The common problem with these algorithms is that they may cause the population to converge to a sub-region of the Pareto optimal front.

Unlike the above algorithms based on improving dominance relationship, decomposition-based MaOEAs decompose MaOPs into several sub-problems for simultaneous optimization, and the goal of the problem is aggregated into different scalar functions. These algorithms guide the individual to search in the direction near the Pareto front by minimizing these scalar function values. In 2018, Zheng et al. [32] redesigned the weight vectors used in the sub-problem and proposed a new weighted mixture-style method to enhance MOEA/D. Lucas et al. [33] introduced a new MOEA/D with uniformly randomly adaptive weights (MOEA/D-URAW) to deal with the limitation of algorithm independent of problem geometry. However, when the Pareto front of the problem is too complex and inconsistent with the distribution of the weight vector, the distribution of the population obtained by MOEAs is often poor.

Different from the above algorithms, MaOEA/IGD [34] is a many-objective evolutionary algorithm based on the IGD index. This algorithm first used a single-objective evolutionary algorithm to estimate the range of Pareto front, and then established a hyper-plane in the target space according to this range. Bader et al. [35] used Monte Carlo approximation based hypervolume indicators for environmental selection, and proposed a hypervolume estimation algorithm (HypE). One of the biggest problems with the above indicator-based methods is that the computational complexity of the indicator can be too high, thus, the algorithm speed may be generally slower than that of ordinary algorithms.

In addition to MaOEAs covered in the above classifications, the many-objective evolutionary methods have shown some new characteristics in recent years. The selection phrase plays a key role in most algorithms, taking into account both the convergence and the distribution of the solution. However, the increase in the number of objectives and decision variables has brought huge challenge to MaOEAs, such as Pareto dominance ineffectiveness and the conflict problem of convergence and diversity [36].

2.1 Selection strategy

Many scholars have studied MaOEAs based on selection strategies. In the first few years of evolutionary algorithms' development, in order to avoid loss of external solutions leading to reduced diversity, Zitzler et al. [15] proposed a strength Pareto evolutionary algorithm (SPEA2), which used density estimation, archive truncation, and fine-grained fitness allocation as environmental selection to solve MaOPs. Deb et al. [37] used a reference point strategy-based non-dominated sorting algorithm (NSGA-III) to improve the selection process of NSGA-II to choose individuals with good convergence by using a set of pre-defined references point as the standard. Recently, many novel studies have emerged one after another. In 2021, Liu et al. [38] proposed a many-objective evolutionary algorithm based on decomposition with correlative selection mechanism (MOEA/D-CSM) to find its correlative individuals for each reference point as soon as possible to maintain the diversity of the population. In the same year, Palakonda et al. [39] proposed an ensemble framework (ENMOEA) in which mating and environmental selections of diverse MOEAs are combined. The framework demonstrates the scalability of the algorithm with the addition of a selection strategy. In this year, in order to solve discrete MaOPs, Zhao et al. [40] adopted an adaptive selection strategy to improve the convergence performance of decomposition-based ACO by using different reference points. In the end, this algorithm can effectively improve the quality of optimization.

Although the improvement of the selection strategy makes the overall performance of the algorithm better, trying more strategies can further improve the universality of the algorithm. However, the majority of researchers have not considered the scale of the problem, but have only solved the problem of the number of variables in small- and medium-scale decisions. Therefore, in-depth study of large-scale problems is needs.

2.2 Large-scale optimization

In order to deal with the problem of a large number of decision variables in practical, many researchers have made relevant studies. Chen et al. [41] introduced a scalable small subpopulations based covariance matrix adaptation evolution strategy (S3-CMA-ES). The algorithm used a series of small subpopulations to approximate the Pareto optimal solution, and introduced a variety of diversity improvement strategy to solve the MaOPs of large-scale decision variables. He et al. [42] embedded adaptive offspring generation method in a MOEA framework (DGEA) and proposed a pre-selection strategy to select parents and used them to construct a direction vector in the decision space to propagate offspring. Tian et al. [43] combined a competitive swarm optimizer with large-scale multi-objective optimization (LMOCSO), the proposed algorithm used a new particle update strategy based on two stages to update the particles.

The focus of previous research has been on the scale of the problem, but the operation of the specific details of the algorithm has not been studied in-depth, such as how to better coordinate the environmental selection and mating selection, or how to use historical information more effectively. However, the emergence of the above algorithms provides an idea for the development of this paper.

Considering the influence of selection strategy, this paper introduces IFM into 1by1EA. Instead of selecting the best individual each time, IFM selects individuals from the previous iteration in a fixed or random manner, and then, they are used for the generation of offspring, thus increasing the diversity of the population. Prior to this, researchers have conducted research on IFM. For example, Gu et al. [44] and Zhang et al. [45] used IFM to improve NSGA-III and MOEA/D, respectively. The proposed algorithms are used to solve large-scale MaOPs. Therefore, the new algorithm is compared with those in the two above mentioned studies. For large-scale research, we chose DGEA and LMOCSO which are also used to solve large-scale problems, and the remaining four state-of-the-art algorithms, which have been proposed in the last 3 years. IFM improves the competitiveness and persuasiveness of 1by1EA to solve problems with large-scale decision variables.

3 Preliminaries

3.1 Basic definitions

MOPs are a common problem in many areas of the real world. Assuming they are a minimization problem, the optimization goal is to minimize all objective functions as much as possible. MOPs can be defined as [46]:

$$\begin{aligned} &minF(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^T \\ &s.t. x \in \Omega, f \in R^m \end{aligned} \tag{1}$$

where m is the number of objective functions and Ω is decision space. x is an n-dimensional decision variable, that is, $x = [x_1, x_2, \ldots, x_n]^T \in \Omega$, it includes possible solutions to the problem. For the set of m-dimensional objective functions $F : f \rightarrow R^m$, it matches the n-dimensional decision space and the m-dimensional objective space. When $m > 3$, this problem can be defined as an MaOP.

The following concepts have been widely popularized:

- **Pareto dominance**: For any two solutions x_1, x_2 in (1), x_1 Pareto dominates x_2 if they satisfy the following conditions, denoted as $x_1 \prec x_2$.

$$\begin{aligned} &f_i(x_1) \leq f_i(x_2), \forall i \in \{1, 2, \ldots, m\} \\ &f_j(x_1) < f_j(x_2), \exists j \in \{1, 2, \ldots, m\} \end{aligned} \tag{2}$$

- **Pareto optimal set**: For $x^* \in \Omega$ in (1), if there is no solution $x_1 \in \Omega$ satisfying $x_1 \prec x^*$, then x^* is known as the Pareto optimal solution. All of these solutions come together to form the Pareto optimal set (PS).
- **Pareto optimal front**: The set of objective value vectors corresponding to each solution in PS is called Pareto optimal front (PF).

$$PF = \{F(x) \mid x \in PS\} \tag{3}$$

3.2 1by1EA

In order to balance the convergence and diversity of solutions in the high-dimensional target space, 1by1EA was proposed [27] and used to solve MaOPs.

The main contribution of this algorithm is that it uses a convergence index to select the solutions in the current population one by one, and proposes a distribution index instead of the Euclidean distance to evaluate the distance between the solutions in the high-dimensional space. It can be used to choose the neighboring solution of the selected solution, so as to weaken it by using niche technology. Unlike most MaOEAs, a boundary maintenance mechanism ensures that the corner solution is not easily discarded. A corner solution can be defined as the individual with the smallest scalar value aggregated by k objectives in the current population, and the number of k is less than the total number of objectives. The general operating framework of the algorithm is given in Algorithm 1. First, N individuals are randomly generated to form the initial parent population (P). Then, mating selection is performed on P to select parents for producing offspring, which iterates until N offspring are produced. The offspring form the offspring population Q. After the variation operations are performed on Q, Q_1 is obtained, and P and Q_1 are combined to form K. The convergence index and distribution index of each solution are calculated in K, and then, the one-by-one selection method is used to select N individuals from K to constitute the initial population of the next generation.

Algorithm 1 General framework of 1by1EA.

```
Require: $P$ (population), $N$ (population size)
Ensure: $P$
Initialize ($P$)
while the terminal condition is not met do
    $Q$ = Mating selection ($P$)
    $Q_1$ = Variation ($Q$)
    $K = P \cup Q_1$
    Calculate the convergence indicator $c(x_i)$ and distribution $d(x_i)$ of each $x_i \in K$, $i$=1, 2, ..., $N$
    $P$ = one-by-one selection ($K$)
end while
```

When calculating the convergence index, each target in the high-dimensional space is treated equally and aggregated into a scalar. The equation is as follows [27]:

$$c(x) = agg(f_1(x), f_2(x), \ldots, f_m(x)) \tag{4}$$

the distribution index takes the form of a vector, where each element represents the distance between a solution and other solutions in the population. Based on the fact that Euclidean distance is unsuitable for the distribution in high-dimensional space, cosine similarity can use the cosine value of the angle between two vectors to measure the similarity. The distribution index can be expressed as follows [47]:

$$d(x_i) = (d_1(x_i), \ldots, d_{|k|}(x_i)), i = 1, \ldots, |k| \tag{5}$$

where

$$d_j(x_i) = 1 - cos(\theta_{ij}), j = 1, \ldots, |k| \tag{6}$$

is the distance between x_i and x_j. The smaller the value, the more similar the distribution between the two solutions. This indicator can effectively remove points close to the coordinate axis instead of on the PF.

As the key operation of 1by1EA, the execution process of the one-by-one selection strategy in environment selection can be described as follows:

Step 1 Boundary maintenance. Corner individuals are selected from K in set K_s.

Step 2 Determine the set K_s.

Step 2.1 The convergence index is calculated for the remaining individuals in K and they are put into K_s one by one according to the one-by-one selection strategy.

Step 2.2 The distribution index is used to measure which solutions are close to the individuals in K_s, and then, the solutions and those dominated by these solutions are de-emphasized. Thus, the non-dominated solutions are retained.

Step 2.3 Stopping criterion. If K is not null, repeat Step 2.1 and Step 2.2, and if K is null, go to Step 2.4.

Step 2.4 Selection. If the number of solutions in K_s is greater than N, then the first N individuals are the initial parents of the next generation. If the number of solutions in K_s is less than N, the de-emphasized and dominated solutions will be selected and filled into K_s according to the principle of survival of the fittest until the number is N.

Step 3 $P = K_s$. K_s is the next generation of parents.

Before calculating the two index values, the normalization of each individual in K ensures that only corner solutions are retained in K. Previous studies have shown that the normalization operation can effectively deal with the problem of dimensionality curse [37]. For other details about the algorithm, please refer to the original paper [27].

3.3 Information feedback model

As mentioned before, IFM can use a simple fitness weighting method to extract and fully use the information in the previous iteration. In the previous iteration, a fixed position or random method can be used to select k individuals from the population. The method of selecting individuals in a fixed manner is classified as F, and selecting individuals in a random manner is classified as R. If k is 1, the model can be defined as M-F1, and six IFMs can be obtained.

In M-F, individuals are selected in a fixed manner, which means that for individuals who need to be updated at the current generation, individuals are selected at the same position at the current and previous generations. Here are three models for selecting individuals based on fixed methods: M-F1, M-F2, and M-F3. In M-R, individuals are selected in a random manner, which means that for individuals who need to be updated at the current generation, the location of individuals selected from the previous generation is random. Therefore, three models for selecting individuals based on random methods are M-R1, M-R2, and M-R3.

Suppose that the current generation is t, the position of the next generation individual is i, x_i^t is the i-th individual of the t-th generation, f_i^t is the fitness value corresponding to the i-th individual of the t-th generation, and y_i is the individual generated by the basic algorithm. The corresponding fitness value is F^{t+1}. λ and μ_k are the weight vectors, satisfy , λ >0, and μ_k >0, where the value of k is 1, 2, or 3, $\lambda + \Sigma\mu_k = 1$, λ >0, and μ >0. j_m is a randomly selected individual position, i and j are not equal, and then, m is between 1 and the population size N. These six models can be defined as follows:

- For the individual to be generated at the $(t + 1)$-th generation, we select an individual from the random position j of the previous generation and combine with the individual at the current $(t + 1)$-th generation to update individual at the next generation. The model is expressed as follows [28]:

$$\begin{aligned} x_i^{t+1} &= \lambda y_i^{t+1} + \mu x_j^t \\ \lambda &= \frac{f_j^t}{F^{t+1} + f_j^t} \\ \mu &= \frac{F^{t+1}}{F^{t+1} + f_j^t} \end{aligned} \tag{7}$$

  when $i = j$, this model can be defined as M-F1.
- We randomly select an individual from the t-th generation and the $(t$ - 1)-th generation. A total of two individuals are selected to update the next-generation

individuals. The model is expressed as follows [44]:

$$\begin{aligned} x_i^{t+1} &= \lambda y_i^{t+1} + \mu_1 x_{j_1}^t + \mu_2 x_{j_2}^{t-1} \\ \lambda &= \frac{1}{2} \cdot \frac{f_{j_2}^{t-1}+f_{j_1}^t}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}} \\ \mu_1 &= \frac{1}{2} \cdot \frac{F^{t+1}+f_{j_2}^{t-1}}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}} \\ \mu_2 &= \frac{1}{2} \cdot \frac{F^{t+1}+f_{j_1}^t}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}} \end{aligned} \tag{8}$$

where $x_{j_1}^t$ and $x_{j_2}^{t-1}$ are individuals selected from random positions at the t-th and (t - 1)-th generations. The corresponding fitness function values are $f_{j_1}^t$ and $f_{j_2}^{t-1}$. When $i = j_1 = j_2$, this model can be defined as M-F2.

- We randomly select three individuals from the t-th, (t - 1)-th, and (t - 2)-th generations to update individuals at the next generation. The model is expressed as follows [45]:

$$\begin{aligned} x_i^{t+1} &= \lambda y_i^{t+1} + \mu_1 x_{j_1}^t + \mu_2 x_{j_2}^{t-1} + \mu_3 x_{j_3}^{t-2} \\ \lambda &= \frac{1}{3} \cdot \frac{f_{j_3}^{t-2}+f_{j_2}^{t-1}+f_{j_1}^t}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}+f_{j_3}^{t-2}} \\ \mu_1 &= \frac{1}{3} \cdot \frac{F^{t+1}+f_{j_3}^{t-2}+f_{j_2}^{t-1}}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}+f_{j_3}^{t-2}} \\ \mu_2 &= \frac{1}{3} \cdot \frac{F^{t+1}+f_{j_3}^{t-2}+f_{j_1}^t}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}+f_{j_3}^{t-2}} \\ \mu_3 &= \frac{1}{3} \cdot \frac{f_{j_2}^{t-1}+f_{j_1}^t+F^{t+1}}{F^{t+1}+f_{j_1}^t+f_{j_2}^{t-1}+f_{j_3}^{t-2}} \end{aligned} \tag{9}$$

where $x_{j_1}^t$, $x_{j_2}^{t-1}$, and $x_{j_3}^{t-2}$ are individuals selected from random positions at the t-th, (t - 1)-th and (t - 2)-th generations. The corresponding fitness function values are $f_{j_1}^t$, $f_{j_2}^{t-1}$, and $f_{j_3}^{t-2}$. When $i = j_1 = j_2 = j_3$, this model can be defined as M-F3.

3.4 Motivation

As mentioned earlier, MOEAs have difficulty in dealing with MaOPs. The main reason is that a sharp increase in the number of non-dominated solutions in high-dimensional objective spaces can lead to greater convergence pressure; so, traditional MOEAs tend to stagnate in terms of convergence. Moreover, with the explosion of dimensions, the order of magnitude of MaOPs has become large. Although many researchers have tried to adopt MOEAs such as NSGA-II and MOEA/D to solve MaOPs, experiments show that owing to the curse of dimensions, MOEAs cannot balance the relationship between convergence and diversity. Recently, many studies have been conducted to modify the dominance relationship so as to enhance the ability to distinguish solutions [29–31]. Convergence is an important property of solutions, and many researchers have used this property to make distinguishing solutions easier by improving a specific stage of the evolutionary process. In this improvement, the excellent solutions are selected one by one, thus increasing the selection pressure toward the Pareto front. The convergence indicator used in the 1by1EA is driven by this idea, which logically resembles the aggregate function in MOEA/D, but without a pre-defined weight vector. MOEA/D can optimize N standard quantum problems simultaneously, rather than by directly solving MOPs as a whole. The same idea applies to solving MaOPs, with 1by1EA being an example.

However, if only the effect of convergence is considered, the algorithm may be more likely to fall into local optimization. To solve this problem, the distribution of solutions in MaOPs also needs to be considered; generally, density estimation and niche techniques are more popular methods. The density estimation method estimates the neighborhood density value for each individual, selects individuals with smaller density value in the next evolution, and then deletes individuals with a larger density value. The characteristic of niche techniques is to form several stable sub-groups, namely niches, and then let individuals evolve in a specific environment. Jiang et al. [48] improved SPEA by introducing an efficient reference direction-based density estimator, which can maintain the distribution of the population. The series of NSGA proposed by Deb et al. is a concrete embodiment of this technology, but the difficulty lies in determining the scope of the niche. Motivated by the above ideas, a distribution threshold is defined in the 1by1EA. Once a solution is selected by the convergence indicator, all solutions with distance to the selected one is less than the threshold are abandoned.

Based on the above description, 1by1EA can effectively balance the convergence and diversity of the solution, and it is theoretically feasible to select this algorithm for research in this paper. But this algorithm still has the following two limitations:

- 1by1EA has a certain absoluteness in the process of iteration, that is, to retain the best individuals at each iteration and discard all remaining individuals.
- 1by1EA does not perform as well when dealing with large-scale problems.

One of the main reasons why 1by1EA has difficulty in solving large-scale many-objective problems is that it only considers the relationship between different solutions. In the one-by-one selection strategy of the algorithm, once a solution is selected, all solutions with distribution distance from the selected solution less than the specified threshold are not valued. When the number of decision variables increases significantly, although this method can select the solutions one by one according to the convergence and distribution indicators, because the distribution of the solutions is very random, the effect of removing a lot of solutions within the threshold range of each selected solution may still be relatively large.

Therefore, to resolve the issues discussed above, this paper uses IFM to enhance 1by1EA and applies the enhanced algorithm to solve large-scale MaOPs. We preserve part of the historical information in the population in a fixed or random way, so that the transfer of information between different generations promotes the inheritance of population diversity by 1by1EA. While the individuals that are preserved are not necessarily the best, they can be used for the next renewal of the population, which allows for the continuous use of much useful information. Furthermore, this inheritance is of great significance for the solution set approaching the Pareto optimal front of large-scale MaOPs. After theoretical analysis, follow-up experiments also verified our idea. Thus, while ensuring the good coverage of Pareto optimal front, 1by1EA-IFM enhances the ability of the algorithm to solve large-scale problems. In addition, given the NSGA-III-IFM and MOEA/D-IFM proposed by Gu et al. [44] and Zhang et al. [45], the role of IFM is more effectively reflected in our research results than in previous research. The similarity between these several algorithms is that they all adopt the information retention mechanism of IFM, the difference is that our proposed 1by1EA-IFM is more convincing in the selection of candidate solutions. The details of 1by1EA-IFM are introduced in the next section.

4 1by1EA-IFM

In this section, in order to save the information in the historical iteration, we integrated IFM into 1by1EA. First, we introduce the overall framework of the proposed 1by1EA-IFM, and then explain the details. Finally, we choose one from the six algorithms as an example to describe the execution flow of the new algorithm in detail.

4.1 General framework of 1by1EA-IFM

To use the historical information in the previous iteration, we introduce the information feedback mechanism after producing new offspring individuals by the mutation operator before environmental selection and then updating these individuals. The approximate operation process of the algorithm based on 1by1EA is as follows. First, initialize the entire population and ideal points randomly, and then update the individuals in the population; this mainly includes three steps: parent mating selection to produce new offspring, operation operators for variation operations, and IFM to update individuals after variation. Finally, in the environment selection stage, the next generation parent population (P) is selected according to the one-by-one selection strategy. For more details about the algorithm, see Section 4.3.

4.2 Operators

A binary tournament selection strategy is proposed in mating selection. Density estimation based on distribution index and information ranking based on one-by-one selection are used as tournament strategies. Promising individuals can then be produced to form a mating pool. First, two individuals are randomly selected from the parent population. The lower ranked individual is selected from the two individuals, whose ranking value rank(x) can be obtained from the one-by-one selection results of previous generations. If the two individuals are equally ranked, individuals with lower density estimates tend to be preferred, and density values can be calculated in conjunction with (5) and (6) according to the following formula [27]:

$$d_k(x) = \frac{1}{\Sigma_{i=1}^{k} d_i^{min}(x) + 1} \tag{10}$$

where $d_i^{min}(x)$, $i = 1, \ldots, k$ is one of the k smallest values in $\{d_1(x), \ldots, d_{|k|}(x)\}$. Finally, if no amount of density estimation can distinguish between the two individuals, then a random selection is made between them.

The crossover operator used in this algorithm is a simulated binary crossover, which inherits useful genetic information from two or more offspring in the form of a single point crossover [49]. In this operator, p_1 and p_2 are two parents, c_1 and c_2 are two children, they satisfy $(p_1 + p_2)/2 = (c_1 + c_2)/2$. β is a spread factor, which is defined as $|(c_2 - c_1)/(p_2 - p_1)|$. If there is a random number ε in the range [0,1], then the range of value of ε will determine which of the following formulas will solve for β.

$$\begin{aligned} c_1 &= (p_1 + p_2) - 0.5\beta(p_2 - p_1) \\ c_2 &= (p_1 + p_2) + 0.5\beta(p_2 - p_1) \\ c(\beta) &= \begin{cases} 0.5(\varepsilon + 1)\beta^{\varepsilon}, & \varepsilon \le 0.5 \\ 0.5(\varepsilon + 1)\frac{1}{\beta^{\varepsilon+2}}, & \varepsilon > 0.5 \end{cases} \end{aligned} \tag{11}$$

After the crossover operator is executed, the individual is updated by the polynomial mutation [50]. In this operator, p_m is defined as a parent, and q_m is the offspring produced by mutation operator. σ is a random number in the range [0,1]. η is a distribution index. u_m and l_m are the upper and lower bounds of m-dimensional decision variables. Thus, the form of this mutation operator is $q_m = p_m + \delta(u_m - l_m)$. The following formula can be referred to calculate δ:

$$
\begin{aligned}
\delta &= \begin{cases} [2\sigma + (1-2\sigma)(1-\delta_1)^{\frac{1}{\eta+1}}] - 1, & \sigma \leq 0.5 \\ 1 - [2(1-\sigma) + 2(\sigma - 0.5)(1-\delta_2)^{\eta+1}]^{\frac{1}{\eta+1}}, & \sigma > 0.5 \end{cases} \\
\delta_1 &= (p_m - l_m)/(u_m - l_m) \\
\delta_2 &= (u_m - p_m)/(u_m - l_m) \qquad (12)
\end{aligned}
$$

4.3 1by1EA-F1

Here, we describe in detail how 1by1EA is combined with IFM and give detailed workflow. We mainly chose the model of selecting an individual in a fixed way from the previous generation as an example. The new algorithm after their combination was named 1by1EA-F1. The remaining five algorithms (1by1EA-F2, 1by1EA-F3, 1by1EA-R1, 1by1EA-R2, and 1by1EA-R3) all performed the same as this algorithm. The main execution of 1by1EA-F1 can be interpreted as the following process:

Step 1 Initialization. Generate random population P and ideal point Z_{min}. Rank of each solution in one-by-one selection and initialize distribute threshold.
Step 2 Update. Supposing the current generation is t and $i = 0$ (i is the subscript of the individual).
Step 2.1 Generate offspring Y^{t+1} by using the crossover and mutation operator of the original 1by1EA. y_i^{t+1} produced by the above process and constitute Y^{t+1}, and then, their fitness value will be calculated by functions.
Step 2.2 Calculate x_i^{t+1} by combining y^{t+1} with x_i^t according to (7). K^t is the set of x_i^{t+1}.
Step 2.3 Determine whether the calculation has been completed for each individual. If it is satisfied, perform Step 2.4. If not, $i = i + 1$ and repeat Step 2.1 and Step 2.2.
Step 2.4 Environmental selection.
Step 2.4.1 Normalize each solution in K^t.
Step 2.4.2 Calculate the convergence indicator $c(x)$ and distribution indicator $d(x)$ of each solution $x \epsilon K^t$, select some excellent one by one and put them into K_s.
Step 2.4.3 The corner solution is chosen to form set K_s.
Step 2.4.4 Guarantee the number of individuals in K_s to be N, which is the population size.
Step 2.4.5 Update the distribution threshold.
Step 2.4.6 Verify whether termination condition is met. The termination condition is the maximum number of iterations defined in advance. If it is satisfied, go to Step 3. If not, $t = t + 1$ and repeat Step 2.
Step 3 Output. Output P^{t+1} to be equal to K_s.

The flow chart of 1by1EA-F1 can be shown in Fig. 1.

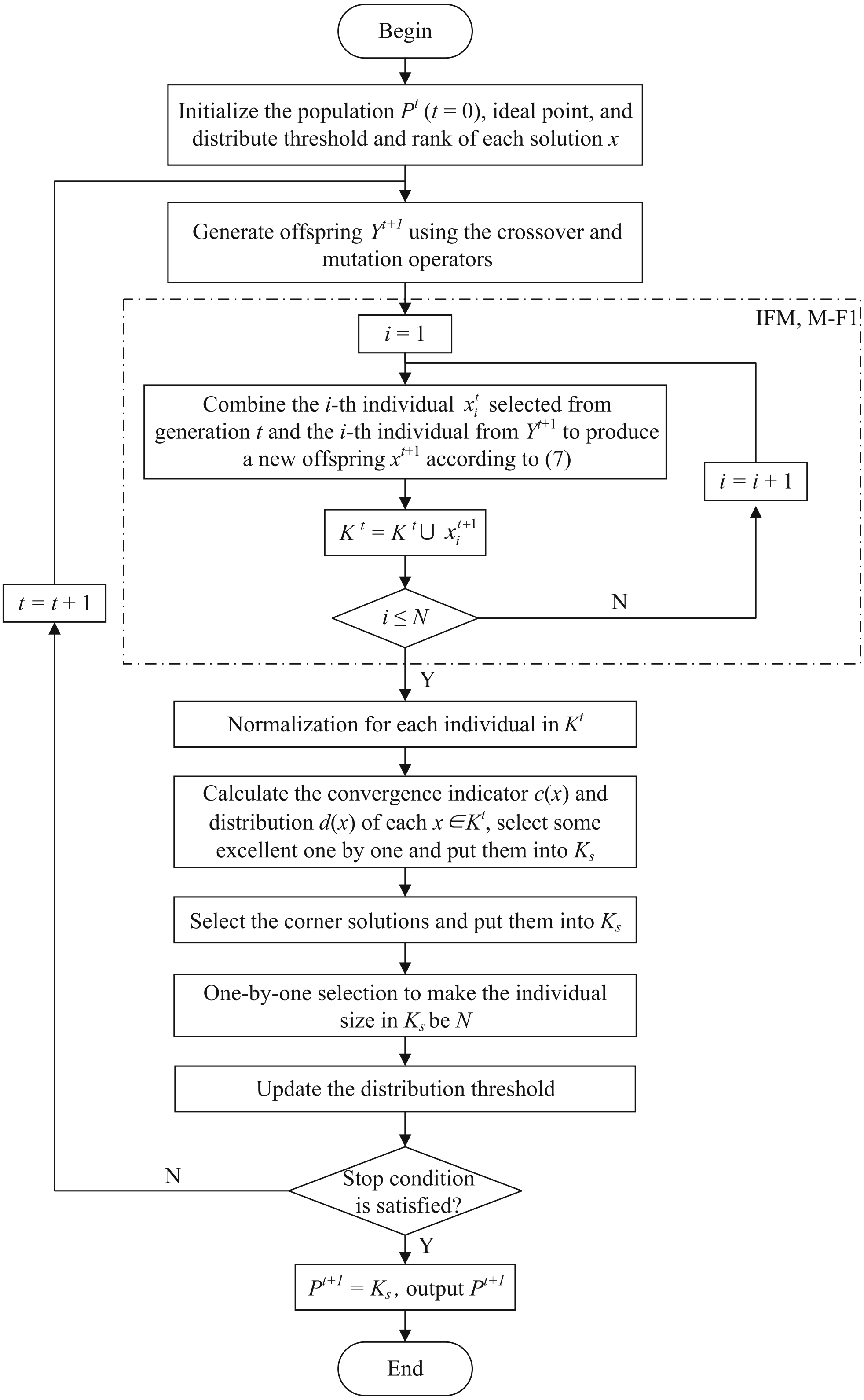
4.4 Complexity

In this subsection, the complexity of the algorithm is analyzed. For convenience, assuming that the problem to be optimized has m objectives and n decision variables, the decision-maker requires n solutions. Its complexity is analyzed according to Algorithm 1 and the above steps. In addition, it is assumed that the size of the population is set to N, and the generation is set to be t. Therefore, in the initialization process (Step 1), the computational complexity of generating the ideal point is O(mN). This is because in the process of generating the ideal point, an aggregation function is used to aggregate all the objectives, and each component in the ideal point corresponds to the minimum value of each sub-objective. Furthermore, generating a random population and ranking of each solution require O(nN) and O(mN) computations, respectively. As the convergence indicator and the distribution indicator need to be calculated according to the ideal point and the solution needed by the decision maker, the computational complexity in Step 2.4.2 is O(mN^2) and O(nN^2), respectively. Then, generation of offspring (Step 2.1) requires O($\frac{N}{2}(n + n)$), because the size of the mating pool is set to N. In Step 2.2 and Step 2.3, IFM begins to work, because each generation in the iterative process needs information from the historical generation, the computational complexity is about O($\frac{N}{2}nmt$). During environmental selection, owing to the introduction of one-by-one selection strategy, the best case of computational complexity is O(N^2), while the worst case is O(N^3), and N individuals are linearly assignment to the ideal points. Furthermore, it is fact that generally $N \gg m$ and $N \gg n$ in MaOPs. Therefore, when the iteration termination condition is reached, the complexity of the proposed algorithm is O(tN^3).

5 Experiments

In this section, we describe the verification process of the proposed 1by1EA-IFM in detail. The experiment was divided into three stages. First, we compared the performance of the proposed 1by1EA-IFM with 1by1EA in dealing with large-scale multi-objective test problems (LSMOPs) with different decision variables. Then we focused on the work of previous researchers based on IFM and compared the outstanding algorithms in 1by1EA-IFM (selected in the first stage) with the excellent algorithms in NSGA-III-IFM and MOEA/D-IFM on the same indicators and test issues. In addition, we focused on MOKP and selected two other comparison algorithms to obtain the verification results. In the last stage, we chose an algorithm specifically designed to solve the large-scale problem and

Fig. 1 The flow chart of 1by1EA-F1


Begin
Initialize the population P^t (t = 0), ideal point, and distribute threshold and rank of each solution x
Generate offspring Y^{t+1} using the crossover and mutation operators
IFM, M-F1
i = 1
Combine the i-th individual x_i^t selected from generation t and the i-th individual from Y^{t+1} to produce a new offspring x^{t+1} according to (7)
$K^t = K^t \cup x_i^{t+1}$
$i \le N$
N
$i = i + 1$
Y
$t = t + 1$
Normalization for each individual in K^t
Calculate the convergence indicator $c(x)$ and distribution $d(x)$ of each $x \in K^t$, select some excellent one by one and put them into K_s
Select the corner solutions and put them into K_s
One-by-one selection to make the individual size in K_s be N
Update the distribution threshold
Stop condition is satisfied?
N
Y
$P^{t+1} = K_s$, output P^{t+1}
End

the other five state-art-of MaOEAs to measure our work from a vertical perspective. In order to avoid contingency, all algorithms had been verified to deal with LSMOPs [51] with different numbers of decision variables. In these problems, the parameters that had to be set included the number of objectives (m), the dimension of the decision variable (n), and the number of subcomponents in each variable group (n_k). Here, we took $n = m * 100$, $n_k = 5$. These settings proved valid when LSMOPs were proposed as test functions [51].

As a test suite, LSMOPs were composed of optimization problems of linear problems, convex problems, multi-model problems, and the disconnected Pareto optimal front, which were defined by specified functions. The different characteristics of these fronts are shown in Table 1. In this paper, the objective number m did not exceed 15, and most experiments used m values of 3, 5, 8, 10, and 15. The inverted generational distance (IGD) [52] and the generational distance (GD) [53] were adopted as performance indicators in most experiments. GD was used to evaluate the convergence of the algorithm. It was mainly used to express the separation distance between the approximate PF and the real PF. Its calculation formula is as follows [54]:

$$GD = \frac{(\Sigma_{i=1}^{n} d_i^m)^{\frac{1}{m}}}{n} \tag{13}$$

where n is the number of points on the approximate PF, m is the number of dimensions of the objective, and d_i represents the average value of the nearest Euclidean distance between each point on the approximate PF and the reference point on the true PF. The smaller the value, the better the convergence of the algorithm.

However, as GD can only evaluate the convergence of the algorithm, in order to simultaneously evaluate the convergence and diversity of the algorithm, IGD was used, it is the solution obtained by mapping the reference point on the real PF to the algorithm. The main idea was to take points uniformly from the real PF, find the closest point on the known PF to the point on the real PF for each point on the real PF, and then add the distances between these points and average them. The calculation formula is as follows [55]:

$$IGD = \frac{\Sigma_{i=1}^{n} \mid d_i \mid}{n} \tag{14}$$

where n is the number of points on the true PF. The smaller the value is, the better the overall performance of the algorithm is.

All experiments in this paper were implemented in Matlab on PlatEMO with two Intel G3204 CPUs. Considering the characteristics of evolutionary algorithms, the IGD and GD values shown in all the following tables were the results of indicators after 20 independent iterations on specific problems, taking the average and standard deviation. More importantly, in terms of statistical method, the Mann-Whitney-Wilcoxon rank-sum test [56] was used to determine whether there is a statistically significant difference in indicators between one algorithm and another algorithm, and the null hypothesis was rejected at a significance level of 5%. Note that the datasets on which statistical tests relied contain a set of values that the algorithm calculates indicators for on the LSMOPs' test suite with different objectives. All the above settings were used in all subsequent experiments.

Table 1 The characteristics of LSMOPs

Problem	Characteristics	Modality
LSMOP1	Linear	Unimodal
LSMOP2	Linear	Mixed
LSMOP3	Linear	Multi-modal
LSMOP4	Linear	Mixed
LSMOP5	Convex	Unimodal
LSMOP6	Convex	Mixed
LSMOP7	Convex	Multi-modal
LSMOP8	Convex	Mixed
LSMOP9	Disconnected	Mixed

5.1 Comparison of 1by1EA with 1by1EA-IFM

5.1.1 Parameter settings

For the fairness of the results, in the simulated binary crossover and polynomial mutation, the two distribution index values were set to 20. The probability of crossover and mutation were 1.0 and $1/n$, respectively, where n was the number of decision variables. The parameter settings of these evolutionary operator were the same as those in recent studies, and these settings proved to be able to generate good offspring in [34, 42, 45, 57, 58]. From the historical research on evolutionary algorithms, we know that when the above parameters are set to other values, the results are more insensitive to the parameters; so, the values of these parameters were selected in this work to mitigate the impact of parameter settings on the results. The maximum number of iterations of the algorithm was 10000, and their population sizes were the same. In all algorithms, the value of k in (10) was set to $1/N$ to balance the computational cost and accuracy in density estimation, and the threshold of the number of individuals (R) selected in advance was set to 1 to enable the population to balance the distribution and convergence of the population. The setting of these two experimental parameters in the original algorithm was proved in their own experiments. When the threshold was 1, the convergence and diversity of the solution could be properly balanced. The population size was set from 100 to 200 according to the different dimensions of the objective space.

5.1.2 Experimental results and analysis

Table 2 lists the average and standard deviation of the IGD values for each algorithm running on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives. The best result in each row of this table is bolded, which applies in all the similar rest of tables below. The symbols "+" and "-" indicate whether the null hypothesis was accepted by the Mann-Whitney-Wilcoxon rank-sum test at the significance of 5% level, that is, whether the IGD and GD results of 1by1EA-IFM were better or worse than those of 1by1EA. Further, the symbol "=" indicates that the corresponding null hypothesis was rejected. Note that the number of "+", "-", and "=" in the last row respectively indicates how many times 1by1EA-IFM was obviously better, obviously worse, and similar to 1by1EA over each test problem on different objectives. The best results in each row of the table are emphasized.

In all 45 test instances, the sum of the best results achieved by the six algorithms in 1by1EA-IFM was 32, while the best results of 1by1EA could only reach 13. In the test example with a lower objective of less than 10 targets, 1by1EA-F1 and 1by1EA-F2 performed extremely well. The sum of the best results obtained by the two was 15, which accounts for almost 50% of the number of best results. Among the three-objective test problems, 1by1EA-F1 could obtain the best results of the average IGD value on LSMOP1/3/5/6/8/9, and the other three best results were respectively obtained by 1by1EA-R1 and 1by1EA-F3. Among the five-objective test functions, 1by1EA-F1 and 1by1EA-R1 could obtain the best results of six out of LSMOP1-9. Among the eight-objective test functions, 1by1EA-F1 and 1by1EA-F2 had a similar performance. In the 10- and 15-objective test functions, the best results were obtained more by 1by1EA-F2 and later algorithms. The last row of the table shows that the values that were superior to 1by1EA were 29, 26, 25, 22, 24, and 23, respectively. This shows that our proposed algorithm could beyoud 1by1EA on most test functions from the perspective of the number of objectives. From the level of the test suite, motivated by the IGD results, it can be seen from the last column in the table that the best results obtained by 1by1EA algorithm were concentrated on LSMOP2-4, whose PF were linear. Therefore, to some extent, it can be considered that 1by1EA is more suitable for the linear PF, and our proposed algorithm has wider application scope, which can be interpreted that the sampled points from the Utopian PF for the proposed algorithm are the Pareto-optimal solutions owing to the mixed feature of the PF, and the information retaining in IFM has good effects.

The findings obtained above are more obvious in the results of the GD value. As shown in Table 3, in all 45 test examples, 43 best results were achieved, and 1by1EA could only achieve two. Unlike the result obtained by the IGD value, the best result of the GD value was evenly distributed among the six algorithms proposed. As can be seen from the last row in the table, the sum of the numbers of different objective problems obtained by the six algorithms with significantly superior performance was 36, 38, 37, 31, 40, and 38 respectively. Compared with the IGD, GD could be further improved. As an indicator to consider the convergence, GD could not measure the quality of the solution set more comprehensively than IGD, but it could also reflect the performance of the algorithm. We previously mentioned that IFM can increase the diversity of solutions in the population at the next iteration through information feedback mechanism. The experimental results of the GD value confirm the conclusion that some information of the solution is retained the next time through the information reuse strategy, which can improve the convergence of the algorithm by increasing the diversity of the solution set. There is no relevant strategy in 1by1EA; so, 1by1EA-IFM had better convergence. Moreover, when the dimensionality of the problem was enhanced, when the decision variable increased to 1500, the performance of our proposed algorithm was not affected. Moreover, excellent results could be more evenly distributed in various LSMOPs with different characteristics, that is, the algorithm could effectively face complex PF, and 1by1EA can hardly solve diverse problems. This finding still relies on the role of information retention mechanisms; in complex environmental situations that are simulated, the continuous reuse of historical information can make the evolution of the population more suitable for environmental changes.

In order to show the results more clearly, here, we used 10-objective problem as an example to illustrate the performance of the six algorithms and the 1by1EA. Figure 2 shows the performance of the seven algorithms on LSMOP1-9 when $m = 10$ along with the iteration. It can be clearly found that on LSMOP1/5/7/8/9, the index values of 1by1EA were all significantly higher than those of the proposed algorithm, which shows that in most LSMOPs, our algorithm was capable of solving these problems.

5.1.3 Statistical analysis of the results

We performed two statistical tests with the above obtained results. It should be emphasized that these two tests can further illustrate the superiority of our algorithms from a statistical point of view. As we knew, the results obtained

Table 2 IGD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-F3, 1by1EA-R1, 1by1EA-R2, 1by1EA-R3 and 1by1EA on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-F3	1by1EA-R1	1by1EA-R2	1by1EA-R3	1by1EA
LSMOP1	3	300	**1.0338e+0 (5.48e-2) +**	1.1533e+0 (5.68e-2) +	1.1741e+0 (6.75e-2) +	3.0781e+0 (3.08e-1) =	1.1776e+0 (5.58e-2) +	1.2250e+0 (6.99e-2) +	3.1226e+0 (3.37e-1)
	5	500	**9.2042e-1 (5.56e-2) +**	9.4811e-1 (3.14e-2) +	9.7153e-1 (3.84e-2) +	9.3521e-1 (4.87e-2) +	9.6153e-1 (4.27e-2) +	9.7997e-1 (3.13e-2) +	5.2587e+0 (2.59e-1)
	8	800	7.7874e-1 (2.04e-2) +	**6.4976e-1 (9.53e-3) +**	6.6798e-1 (1.32e-2) +	7.8274e-1 (2.11e-2) +	6.5987e-1 (1.36e-2) +	6.9235e-1 (1.78e-2) +	6.4569e+0 (3.64e-1)
	10	1000	9.4274e-1 (1.37e-2) +	**6.9391e-1 (9.50e-3) +**	7.0551e-1 (1.29e-2) +	9.4284e-1 (1.35e-2) +	7.0847e-1 (1.28e-2) +	7.6649e-1 (2.68e-2) +	6.9196e+0 (2.05e-1)
	15	1500	2.4771e+0 (8.21e-2) +	1.6312e+0 (5.19e-2) +	**1.5036e+0 (6.11e-2) +**	2.5257e+0 (8.00e-2) +	1.6890e+0 (3.85e-2) +	1.6433e+0 (5.17e-2) +	7.5775e+0 (3.32e-1)
LSMOP2	3	300	1.8597e-1 (1.35e-2) -	1.9270e-1 (1.63e-2) -	1.7934e-1 (2.43e-2) -	**1.0071e-1 (8.85e-3) =**	12.0539e-1 (1.09e-2) -	2.1681e-1 (2.59e-2) -	1.0614e-1 (1.22e-2)
	5	500	3.0754e-1 (1.55e-2) -	2.7528e-1 (2.65e-2) -	2.6730e-1 (2.62e-2) -	2.9869e-1 (2.39e-2) -	3.0022e-1 (2.13e-2) -	3.0569e-1 (1.78e-2) -	**1.3502e-1 (1.09e-2)**
	8	800	3.3945e-1 (1.24e-2) -	3.3845e-1 (1.76e-2) -	3.4014e-1 (1.52e-2) -	3.4026e-1 (1.48e-2) -	3.4347e-1 (1.72e-2) -	3.4553e-1 (1.26e-2) -	**1.9258e-1 (1.93e-3)**
	10	1000	3.5871e-1 (1.50e-2) -	3.6491e-1 (1.29e-2) -	3.7479e-1 (1.39e-2) -	3.5787e-1 (1.21e-2) -	3.7726e-1 (1.50e-2) -	3.7446e-1 (1.66e-2) -	**2.1617e-1 (6.26e-3)**
	15	1500	4.1941e-1 (8.40e-3) -	4.2292e-1 (1.08e-2) -	4.2030e-1 (1.06e-2) -	4.1942e-1 (9.53e-3) -	4.2729e-1 (8.25e-3) -	4.2675e-1 (1.26e-2) -	**3.6543e-1 (7.92e-3)**
LSMOP3	3	300	**9.3862e+0 (3.16e-1) +**	1.1776e+1 (1.84e+0) =	1.3122e+1 (2.71e+0) =	1.1642e+1 (6.27e-1) =	1.1913e+1 (2.91e+0) =	1.7030e+1 (5.00e+0) -	1.1574e+1 (4.35e-1)
	5	500	5.0855e+1 (1.98e+1) -	5.1978e+1 (1.66e+1) -	7.4839e+1 (1.73e+1) -	5.9899e+1 (2.30e+1) -	5.8700e+1 (1.55e+1) -	7.8453e+1 (2.13e+1) -	**1.4147e+1 (9.40e-1)**
	8	800	1.0433e+2 (4.98e+1) -	1.1194e+2 (2.29e+1) -	1.0470e+2 (1.13e+1) -	1.0249e+2 (4.88e+1) -	1.1540e+2 (3.78e+1) -	1.0811e+2 (1.11e+1) -	**3.1168e+1 (3.28e+0)**
	10	1000	1.2422e+2 (5.91e+1) -	9.9376e+1 (1.03e+1) -	8.8069e+1 (7.16e+0) -	1.2119e+2 (6.81e+1) -	9.8291e+1 (1.72e+1) -	8.1721e+1 (1.40e+1) -	**3.4054e+1 (2.05e+0)**
	15	1500	7.2553e+1 (3.59e+1) -	1.0864e+2 (5.35e+0) -	1.0764e+2 (1.41e+1) -	9.1578e+1 (2.59e+1) -	1.1181e+2 (6.39e+0) -	1.1453e+2 (5.24e+0) -	**4.4652e+1 (2.31e+1)**
LSMOP4	3	300	2.3934e-1 (2.64e-2) +	2.4527e-1 (1.79e-2) =	**2.3895e-1 (4.10e-3) +**	2.5631e-1 (2.68e-3) =	2.5714e-1 (1.58e-2) =	2.5614e-1 (1.13e-2) =	2.5614e-1 (2.79e-3)
	5	500	**2.4540e-1 (7.82e-3) +**	2.5057e-1 (7.75e-3) +	2.5234e-1 (8.98e-3) =	2.4713e-1 (8.85e-3) +	2.6073e-1 (1.04e-2) -	2.7602e-1 (1.32e-2) -	2.5395e-1 (1.17e-3)
	8	800	3.7210e-1 (1.32e-2) -	3.7791e-1 (1.46e-2) -	3.7616e-1 (1.44e-2) -	3.7284e-1 (1.17e-2) -	3.7779e-1 (1.50e-2) -	3.7455e-1 (1.37e-2) -	**2.2761e-1 (3.04e-3)**
	10	1000	3.7987e-1 (1.16e-2) -	3.9084e-1 (1.18e-2) -	3.8933e-1 (1.54e-2) -	3.8409e-1 (1.21e-2) -	3.9396e-1 (1.29e-2) -	3.9714e-1 (1.49e-2) -	**2.6511e-1 (1.19e-2)**
	15	1500	4.2120e-1 (9.45e-3) -	4.2783e-1 (1.17e-2) -	4.2242e-1 (1.36e-2) -	4.2315e-1 (1.01e-2) -	4.3324e-1 (9.33e-3) -	4.3380e-1 (8.24e-3) -	**3.7028e-1 (7.99e-3)**
LSMOP5	3	300	**2.2007e+0 (1.08e-1) +**	2.4859e+0 (1.26e-1) +	2.4906e+0 (1.37e-1) +	3.7659e+0 (4.09e-1) =	2.5629e+0 (1.23e-1) +	2.5281e+0 (9.52e-2) +	3.8292e+0 (4.57e-1)
	5	500	**1.7835e+0 (1.07e-1) +**	2.1119e+0 (1.03e-1) +	2.1844e+0 (1.10e-1) +	1.7903e+0 (9.23e-2) +	2.2747e+0 (1.16e-1) +	2.4572e+0 (9.95e-2) +	3.5114e+0 (3.40e-1)
	8	800	1.9562e+0 (9.67e-2) +	**1.9194e+0 (9.49e-2) +**	1.9245e+0 (1.02e-1) +	1.9844e+0 (9.49e-2) +	2.1289e+0 (8.64e-2) +	2.1406e+0 (6.85e-2) +	7.2977e+0 (6.37e-1)
	10	1000	2.2840e+0 (7.22e-2) +	1.9922e+0 (8.33e-2) +	**1.8768e+0 (7.57e-2) +**	2.3027e+0 (7.98e-2) +	2.1124e+0 (7.05e-2) +	2.0663e+0 (4.61e-2) +	1.0651e+1 (3.85e-1)
	15	1500	3.8207e+0 (1.29e-1) +	2.8275e+0 (5.29e-2) +	**2.5776e+0 (6.86e-2) +**	3.7610e+0 (1.17e-1) +	2.8623e+0 (8.56e-2) +	2.7031e+0 (5.48e-2) +	1.0675e+1 (3.87e-1)
LSMOP6	3	300	**2.5526e+2 (8.48e+1) +**	4.5985e+2 (7.07e+1) +	5.7889e+2 (6.29e+1) +	1.9910e+3 (8.03e+2) =	5.3001e+2 (8.08e+1) +	5.8742e+2 (6.69e+1) +	1.5802e+3 (6.86e+2)
	5	500	1.2303e+2 (3.92e+1) +	2.7936e+2 (3.41e+1) +	3.9208e+2 (6.27e+1) +	**1.1769e+2 (4.71e+1) +**	3.2334e+2 (5.15e+1) +	4.3785e+2 (4.52e+1) =	4.8320e+2 (1.10e+2)
	8	800	**1.7078e+0 (1.90e-2) +**	2.0875e+1 (8.54e+1) =	6.4428e+0 (2.06e+1) -	1.7092e+0 (2.78e-2) +	2.0157e+1 (8.18e+1) -	1.9354e+0 (3.77e-2) -	1.7450e+0 (2.76e-2)

Table 2 (continued)

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-F3	1by1EA-R1	1by1EA-R2	1by1EA-R3	1by1EA
	10	1000	1.4710e+0 (8.01e-3) +	1.4832e+0 (3.25e-2) +	1.5476e+1 (6.26e+1) -	**1.4649e+0 (1.69e-2) +**	1.5201e+0 (2.31e-2) =	1.5355e+0 (1.92e-2) -	1.5130e+0 (7.08e-3)
	15	1500	1.2848e+3 (9.61e+1) +	6.3194e+2 (5.52e+1) +	**5.4861e+2 (8.28e+1) +**	1.2818e+3 (1.20e+2) +	7.2170e+2 (7.65e+1) +	6.7062e+2 (5.91e+1) +	9.5794e+3 (5.00e+2)
LSMOP7	3	300	4.4012e+0 (1.40e+1) -	1.7704e+1 (2.61e+1) -	1.9465e+1 (2.15e+1) -	**1.1481e+0 (6.46e-2) =**	1.0583e+1 (2.24e+1) -	9.3278e+0 (1.62e+1) -	1.1650e+0 (6.72e-2)
	5	500	2.5984e+0 (1.04e-1) -	2.9855e+0 (1.38e-1) -	7.1308e+0 (1.84e+1) -	2.5803e+0 (9.14e-2) -	3.0441e+0 (1.13e-1) -	3.0937e+0 (1.26e-1) -	**2.2271e+0 (1.17e-1)**
	8	800	**2.8503e+2 (4.77e+1) +**	3.0267e+2 (4.29e+1) +	3.7294e+2 (4.49e+1) +	2.8961e+2 (3.67e+1) +	3.3557e+2 (3.23e+1) +	4.2732e+2 (3.30e+1) +	2.8213e+3 (5.84e+2)
	10	1000	4.2675e+2 (5.20e+1) +	**3.3053e+2 (3.53e+1) +**	3.5007e+2 (3.57e+1) +	4.0537e+2 (3.98e+1) +	3.8526e+2 (3.58e+1) +	4.6505e+2 (2.56e+1) +	9.0175e+3 (6.68e+2)
	15	1500	1.7883e+0 (9.15e-3) -	1.7886e+0 (8.55e-3) -	1.7924e+0 (1.54e-2) =	1.7859e+0 (7.19e-3) -	1.8093e+0 (1.13e-2) -	1.8103e+0 (1.03e-2) -	**1.7818e+0 (7.23e-3)**
LSMOP8	3	300	**5.5526e-1 (3.48e-3) +**	5.5542e-1 (3.37e-3) +	5.5962e-1 (4.01e-3) +	6.1425e-1 (2.21e-2) =	5.5695e-1 (3.51e-3) +	5.5992e-1 (3.42e-3) +	6.0865e-1 (1.53e-2)
	5	500	9.3762e-1 (3.16e-2) =	9.3305e-1 (2.67e-2) =	9.3573e-1 (1.58e-2) =	**9.2962e-1 (2.25e-2) =**	9.3801e-1 (1.93e-2) =	9.3617e-1 (1.53e-2) =	9.4237e-1 (5.88e-2)
	8	800	**1.2055e+0 (2.39e-2)+**	1.2139e+0 (4.13e-2) +	1.2559e+0 (4.15e-2) +	1.2136e+0 (3.09e-2) +	1.3321e+0 (3.55e-2) +	1.4231e+0 (3.75e-2) +	3.7112e+0 (3.97e-1)
	10	1000	1.4596e+0 (3.13e-2) +	**1.3086e+0 (2.14e-2) +**	1.3263e+0 (2.56e-2) +	1.4574e+0 (3.32e-2) +	1.3772e+0 (2.60e-2) +	1.4452e+0 (2.97e-2) +	6.2624e+0 (3.39e-1)
	15	1500	1.1898e+0 (9.19e-3) +	1.1892e+0 (8.56e-3) +	1.1968e+0 (1.02e-2) +	1.1938e+0 (1.35e-2) +	**1.1857e+0 (1.04e-2) +**	1.1978e+0 (1.57e-2) +	1.3192e+0 (3.43e-3)
LSMOP9	3	300	**1.3207e+1 (9.74e-1) +**	1.5052e+1 (6.69e-1) +	1.5310e+1 (9.31e-1) +	1.9691e+1 (1.77e+0) =	1.5620e+1 (9.68e-1) +	1.6826e+1 (9.06e-1) +	1.8801e+1 (1.19e+0)
	5	500	3.7268e+1 (1.67e+0) +	3.9640e+1 (2.17e+0) +	4.2101e+1 (1.16e+0) +	**3.7110e+1 (1.96e+0) +**	4.2634e+1 (1.63e+0) +	4.5612e+1 (1.44e+0) +	1.1661e+2 (4.40e+0)
	8	800	7.9048e+1 (2.73e+0) +	**7.2088e+1 (1.37e+0) +**	7.4597e+1 (2.03e+0) +	8.0226e+1 (2.06e+0) +	7.5784e+1 (1.85e+0) +	8.0862e+1 (1.73e+0) +	4.6196e+2 (1.30e+1)
	10	1000	1.4372e+2 (3.71e+0) +	**1.1405e+2 (2.27e+0) +**	1.1765e+2 (2.31e+0) +	1.4127e+2 (3.07e+0) +	1.1943e+2 (2.53e+0) +	1.2467e+2 (2.95e+0) +	8.7442e+2 (1.61e+1)
	15	1500	8.5936e+2 (1.38e+1) +	6.3449e+2 (9.47e+0) +	**6.0050e+2 (1.33e+1) +**	8.5620e+2 (1.81e+1) +	6.4654e+2 (1.28e+1) +	6.1505e+2 (1.20e+1) +	2.5094e+3 (4.50e+1)
+/-/=			29/15/1	26/15/4	25/16/4	22/13/10	24/17/4	23/19/3	

Table 3 GD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-F3, 1by1EA-R1, 1by1EA-R2, 1by1EA-R3 and 1by1EA on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	M	N	1by1EA-F1	1by1EA-F2	1by1EA-F3	1by1EA-R1	1by1EA-R2	1by1EA-R3	1by1EA
LSMOP1	3	300	**1.6787e-1 (3.95e-2) +**	2.5575e-1 (4.44e-2) +	2.7272e-1 (5.60e-2) +	5.0295e-1 (3.23e-2) =	2.6053e-1 (5.54e-2) +	2.7906e-1 (6.48e-2) +	5.0106e-1 (3.93e-2)
	5	500	3.8119e-1 (8.63e-2) +	4.1140e-1 (8.54e-2) +	3.7617e-1 (9.87e-2) +	4.0146e-1 (1.25e-1) +	3.5139e-1 (6.21e-2) +	**3.4712e-1 (6.24e-2) +**	4.4940e-1 (3.69e-2)
	8	800	2.5392e-1 (5.71e-2) +	2.1248e-1 (9.49e-2) +	1.9930e-1 (7.73e-2) +	2.4375e-1 (5.69e-2) +	1.8874e-1 (7.67e-2) +	**1.6720e-1 (5.52e-2) +**	3.9445e-1 (3.12e-2)
	10	1000	1.3873e-1 (4.50e-2) +	**8.1891e-2 (4.17e-2) +**	1.3121e-1 (6.09e-2) +	1.4242e-1 (4.79e-2) +	9.0216e-2 (4.25e-2) +	1.2197e-1 (5.82e-2) +	4.6261e-1 (1.49e-1)
	15	1500	1.2279e-1 (2.80e-2) +	**8.9880e-2 (3.19e-2) +**	1.1377e-1 (4.05e-2) +	1.2739e-1 (2.50e-2) +	9.0100e-2 (3.00e-2) +	1.0814e-1 (4.03e-2) +	3.8610e-1 (5.00e-2)
LSMOP2	3	300	2.6954e-3 (6.57e-5) +	2.8146e-3 (8.21e-5) +	2.9379e-3 (1.20e-4) +	4.7213e-3 (9.48e-5) =	2.5308e-3 (7.35e-5) +	**2.3769e-3 (7.22e-5) +**	4.7345e-3 (1.03e-4)
	5	500	2.3211e-3 (9.86e-5) +	2.2881e-3 (1.67e-4) +	2.3853e-3 (1.44e-4) +	2.3807e-3 (1.35e-4) +	**2.2691e-3 (1.56e-4) +**	2.3679e-3 (7.63e-5) +	3.9148e-3 (1.07e-4)
	8	800	4.4543e-3 (3.92e-4) +	4.5092e-3 (4.35e-4) +	4.4390e-3 (2.53e-4) +	**4.3371e-3 (1.87e-4) +**	4.6317e-3 (4.58e-4) +	4.4823e-3 (2.64e-4) +	5.4325e-3 (8.73e-4)
	10	1000	**3.9669e-3 (1.49e-4) +**	4.0429e-3 (9.77e-5) +	4.2019e-3 (9.26e-5) +	3.9714e-3 (1.51e-4) +	4.1052e-3 (8.20e-5) +	4.2519e-3 (9.41e-5) +	5.7612e-3 (1.31e-3)
	15	1500	4.9863e-3 (4.36e-5) +	4.9306e-3 (4.32e-5) +	4.8991e-3 (5.36e-5) +	5.0043e-3 (3.23e-5) +	4.9305e-3 (4.04e-5) +	**4.8973e-3 (4.72e-5) +**	5.3709e-3 (6.67e-4)
LSMOP3	3	300	4.9135e+2 (6.59e+2) +	**1.8384e+2 (4.63e+2) +**	4.9278e+2 (6.35e+2) +	1.0926e+3 (3.92e+2) =	4.6814e+2 (1.42e+3) +	6.8739e+2 (1.03e+3) =	8.9633e+2 (4.55e+2)
	5	500	2.1514e+2 (1.61e+2) =	3.2567e+2 (2.46e+2) =	5.9974e+2 (2.16e+2) -	**2.0294e+2 (1.50e+2) +**	3.8893e+2 (2.41e+2) =	6.2333e+2 (2.39e+2) -	2.8158e+2 (1.50e+2)
	8	800	8.4507e+2 (3.05e+2) =	**5.8044e+2 (1.39e+2) =**	6.9422e+2 (1.98e+2) =	8.0945e+2 (2.00e+2) =	9.5822e+2 (1.65e+3) =	6.6112e+2 (1.31e+2) =	9.0516e+2 (9.73e+2)
	10	1000	7.9500e+2 (7.35e+2) +	7.4126e+2 (1.01e+3) +	5.9800e+2 (1.40e+2) +	9.0685e+2 (1.01e+3) +	**4.8707e+2 (9.17e+1) +**	6.2922e+2 (1.18e+2) +	1.1988e+3 (8.15e+2)
	15	1500	3.9962e+2 (9.70e+1) +	2.9749e+2 (6.07e+1) +	3.1796e+2 (5.97e+1) +	4.1002e+2 (1.41e+2) +	**2.8563e+2 (5.04e+1) +**	3.5357e+2 (7.17e+1) +	6.0577e+2 (2.41e+2)
LSMOP4	3	300	1.4750e-2 (1.49e-3) +	1.5740e-2 (1.38e-3) =	1.5574e-2 (1.29e-3) =	1.6337e-2 (1.44e-3) =	1.4262e-2 (1.81e-3) +	**1.3304e-2 (1.46e-3) +**	1.6145e-2 (1.59e-3)
	5	500	1.2698e-2 (1.31e-3) +	1.2555e-2 (1.78e-3) +	1.1782e-2 (1.99e-3) +	1.2916e-2 (1.32e-3) +	1.1079e-2 (1.37e-3) +	**9.8109e-3 (1.24e-3) +**	1.3837e-2 (8.23e-4)
	8	800	4.9191e-3 (2.07e-4) +	4.6639e-3 (1.91e-4) +	**4.5475e-3 (1.66e-4) +**	5.0148e-3 (1.96e-4) +	4.6198e-3 (2.47e-4) +	4.5722e-3 (1.90e-4) +	9.0210e-3 (7.90e-4)
	10	1000	4.3462e-3 (7.35e-5) +	4.3584e-3 (5.48e-5) +	4.4790e-3 (7.75e-5) +	**4.3292e-3** (7.31e-5) +	4.4136e-3 (6.15e-5) +	4.5539e-3 (8.05e-5) +	6.2578e-3 (4.15e-4)
	15	1500	5.1910e-3 (5.18e-5) +	5.0781e-3 (7.64e-5) +	5.0985e-3 (5.32e-5) +	5.2086e-3 (4.11e-5) +	**5.0691e-3 (4.50e-5) +**	5.0967e-3 (5.87e-5) +	6.6928e-3 (1.50e-3)
LSMOP5	3	300	**9.3852e-1 (7.11e-1) +**	1.2434e+0 (6.92e-1) +	1.2755e+0 (6.82e-1) +	9.5780e+0 (1.47e+0) =	1.2037e+0 (7.61e-1) +	1.5312e+0 (6.11e-1) +	8.6105e+0 (2.22e+0)
	5	500	7.7674e-1 (1.75e-1) +	8.7321e-1 (4.28e-1) -	1.0411e+0 (5.39e-1) -	7.4475e-1 (2.07e-1) +	7.8453e-1 (4.38e-1) -	**6.3027e-1 (2.31e-1) =**	7.7947e-1 (9.55e-1)
	8	800	4.2194e-1 (1.96e-1) +	6.7163e-1 (3.83e-1) +	5.1959e-1 (3.19e-1) +	4.9705e-1 (2.79e-1) +	3.2808e-1 (2.28e-1) +	**2.0585e-1 (9.67e-2) +**	1.1502e+0 (8.15e-1)
	10	1000	2.7792e-1 (2.71e-1) +	1.8992e-1 (1.59e-1) +	2.1371e-1 (1.44e-1) +	2.4181e-1 (1.24e-1) +	1.7229e-1 (1.37e-1) +	**1.0617e-1 (5.23e-2) +**	9.1536e-1 (2.33e-1)
	15	1500	2.7792e-1 (2.71e-1) +	1.8992e-1 (1.59e-1) +	2.1371e-1 (1.44e-1) +	2.4181e-1 (1.24e-1) +	1.7229e-1 (1.37e-1) +	**1.0617e-1 (5.23e-2) +**	9.1536e-1 (2.33e-1)
LSMOP6	3	300	1.9943e+4 (4.29e+4) =	1.0080e+4 (1.56e+4) +	1.2175e+4 (2.04e+4) =	2.0011e+4 (2.09e+4) =	4.9833e+3 (9.26e+3) +	**3.8285e+3 (7.97e+3) +**	1.0694e+4 (1.27e+4)

Table 3 (continued)

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-F3	1by1EA-R1	1by1EA-R2	1by1EA-R3	1by1EA
	5	500	5.7787e+3 (6.87e+3) =	2.6037e+3 (2.80e+3) +	**2.1422e+3 (6.95e+2) +**	8.5398e+3 (1.13e+4) =	2.1429e+3 (1.36e+3) +	3.3243e+3 (4.72e+3) +	8.1248e+3 (7.37e+3)
	8	800	3.6011e+3 (9.51e+2) -	3.2073e+3 (1.11e+3) -	1.8900e+3 (8.06e+2) -	3.7473e+3 (1.21e+3) -	1.8925e+3 (6.47e+2) -	1.9133e+3 (7.61e+2) -	**6.1731e+2 (5.76e+2)**
	10	1000	3.4243e+3 (7.71e+2) -	1.6880e+3 (4.84e+2) +	**1.2229e+3 (3.71e+2) +**	3.7069e+3 (1.13e+3) -	1.4818e+3 (4.32e+2) +	1.3480e+3 (2.92e+2) +	2.0448e+3 (3.91e+2)
	15	1500	5.8416e+2 (3.98e+2) +	5.9486e+2 (6.04e+2) +	4.2651e+2 (3.00e+2) +	6.2690e+2 (3.08e+2) +	3.8622e+2 (3.40e+2) +	**2.6423e+2 (1.21e+2) +**	2.8316e+3 (6.17e+2)
LSMOP7	3	300	3.4049e+4 (3.54e+4) =	1.6897e+4 (1.37e+4) =	1.4278e+4 (1.07e+4) =	3.2672e+4 (3.95e+4) =	**1.0402e+4 (1.05e+4) +**	1.7763e+4 (1.75e+4) =	2.5902e+4 (2.98e+4)
	5	500	3.3647e+3 (1.35e+3) -	5.0840e+3 (2.98e+3) -	2.3083e+3 (1.18e+3) -	3.8043e+3 (2.43e+3) -	4.3450e+3 (1.98e+3) -	2.7719e+3 (1.95e+3) -	**4.6889e+2 (1.61e+3)**
	8	800	4.5968e+3 (3.16e+3) =	2.9021e+3 (1.32e+3) +	2.6893e+3 (1.05e+3) +	3.7004e+3 (3.29e+3) +	2.3415e+3 (9.63e+2) +	**2.1987e+3 (6.19e+2) +**	5.1140e+3 (1.15e+3)
	10	1000	1.7609e+3 (7.64e+2) +	1.6038e+3 (6.63e+2) +	1.6797e+3 (5.45e+2) +	1.8121e+3 (6.61e+2) +	1.3630e+3 (9.78e+2) +	**1.2117e+3 (5.13e+2) +**	3.8828e+3 (5.63e+2)
	15	1500	1.7609e+3 (7.64e+2) +	1.6038e+3 (6.63e+2) +	**1.6797e+3 (5.45e+2) +**	1.8121e+3 (6.61e+2) +	1.3630e+3 (9.78e+2) +	1.2117e+3 (5.13e+2) +	3.8828e+3 (5.63e+2)
LSMOP8	3	300	4.4026e-1 (1.55e-1) +	**4.3905e-1 (1.97e-1) +**	6.0657e-1 (2.28e-1) +	9.7661e-1 (3.34e-1) =	4.5621e-1 (1.67e-1) +	6.1260e-1 (1.98e-1) +	9.0227e-1 (2.67e-1)
	5	500	3.4017e-1 (7.00e-2) +	3.9484e-1 (1.38e-1) +	3.9061e-1 (4.94e-2) +	**2.9779e-1 (5.59e-2) +**	3.8080e-1 (5.00e-2) +	3.8907e-1 (5.12e-2) +	7.4394e-1 (2.57e-1)
	8	800	2.0540e-1 (1.72e-1) +	2.3745e-1 (1.85e-1) +	2.4405e-1 (2.39e-1) +	1.6325e-1 (1.21e-1) +	1.3366e-1 (8.29e-2) +	**8.4860e-2 (3.44e-2) +**	8.2973e-1 (4.19e-1)
	10	1000	7.6333e-2 (3.64e-2) +	7.2296e-2 (4.03e-2) +	1.1897e-1 (1.15e-1) +	7.3928e-2 (2.28e-2) +	7.1928e-2 (7.48e-2) +	**5.5156e-2 (8.46e-3) +**	6.5400e-1 (1.71e-1)
	15	1500	1.7657e-1 (4.88e-2) +	1.1682e-1 (5.85e-2) +	7.6335e-2 (9.99e-3) +	2.1518e-1 (6.12e-2) +	8.7877e-2 (1.85e-2) +	**7.5662e-2 (1.10e-2) +**	2.8437e-1 (3.29e-2)
LSMOP9	3	300	**1.8206e+0 (3.24e-1) +**	1.9062e+0 (3.62e-1) +	1.9030e+0 (2.64e-1) +	4.0250e+0 (8.10e-1) =	2.1170e+0 (4.71e-1) +	2.1479e+0 (2.58e-1) +	3.9186e+0 (5.28e-1)
	5	500	3.3376e+0 (4.56e-1) +	3.7337e+0 (5.96e-1) +	3.7683e+0 (5.42e-1) +	**3.1708e+0 (4.09e-1) +**	3.8100e+0 (6.57e-1) +	3.9746e+0 (4.55e-1) +	1.2706e+1 (1.15e+0)
	8	800	3.9968e+0 (1.36e-1) +	**3.5840e+0 (2.10e-1) +**	3.6752e+0 (1.60e-1) +	4.0492e+0 (1.73e-1) +	3.5993e+0 (1.85e-1) +	3.7856e+0 (1.82e-1) +	2.5476e+1 (1.90e+0)
	10	1000	5.9201e+0 (1.74e-1) +	**4.6529e+0 (1.01e-1) +**	4.9218e+0 (1.41e-1) +	5.8918e+0 (1.40e-1) +	4.6859e+0 (1.35e-1) +	4.9170e+0 (1.01e-1) +	3.6124e+1 (1.05e+0)
	15	1500	2.6547e+1 (1.99e-1) +	1.9781e+1 (1.30e-1) +	1.9087e+1 (1.46e-1) +	2.6570e+1 (1.65e-1) +	1.9736e+1 (1.44e-1) +	**1.9046e+1 (1.57e-1) +**	7.2772e+1 (4.16e-1)
+/-/=			36/3/6	38/3/4	37/4/4	31/3/11	40/3/2	38/3/4	

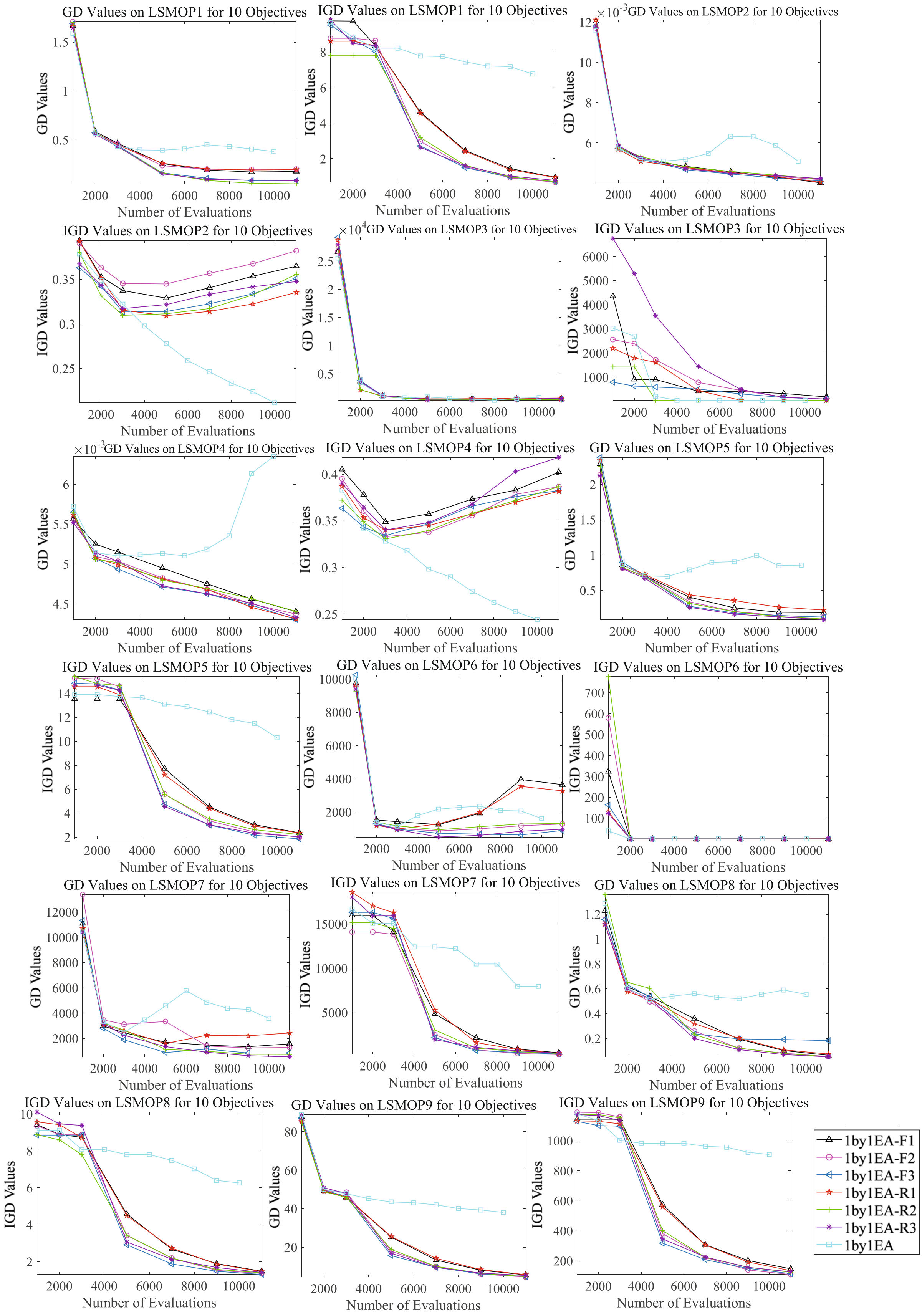


Fig. 2 The mean of IGD and GD values on LSMOP1-9 with 10 objectives

from our experiments were the mean and standard deviation of IGD and GD. We chose the standard deviation of metrics as the statistic to be measured because the magnitude of the standard deviation directly measures the degree of dispersion of the data, so that all data characteristics are adequately reflected rather than making the positive and negative extremes cancel each other out as in the case of the mean.

First, after integrating the data in tables before, Friedman's non-parametric test was applied to check whether there were difference among all algorithms. The results in Tables 2 and 3 are further analyzed using Friedman test, as shown in Table 4. Since the confidence interval is 95%, in a χ^2 distribution with 6 degrees of freedom, we gained a p-value of 0.000006, which was significantly below the significance level (α=0.05). The results of this statistical test showed that our proposed six algorithms were significantly different from 1by1EA. The best rank was obtained by 1by1EA-F1, however, 1by1EA reached the worst rank.

After obtaining the above results, Holm post-hoc test was performed to verify the statistical significance of the six new algorithms. Since the proposed 1byEA-F1 obtained the best ranking, it was considered as control algorithm, and by pairwise and multiple comparisons, we obtained the adjusted p-values after using Holm post-hoc procedure, as shown in Table 5. In this table, the algorithms were assigned an ordinal number i that indicated the degree of excellence of the algorithm, with the first column of the table representing from the best to the worst from top to bottom. The null hypothesis was rejected on the premise that the p-value obtained by Holm post-hoc procedure need to be less than the adjusted $\alpha(\alpha = \alpha/i)$. Therefore, we can conclude with a high degree of certainty that there is not significant difference among our proposed six algorithms, and they are significantly better than 1by1EA.

5.2 Comparison with NSGA-III-IFM and MOEA/D-IFM

In the first phase of the experiment, we verified the performance of 1by1EA-IFM. As there are many models, here, we counted the sum of the number of GD and IGD indicators that were significantly better than the original algorithm in Experiment 1. As shown in Table 6, the best algorithm was 1by1EA-F1, because the final results of 1by1EA-F2 and 1by1EA-R2 were both 64. Thus, we chose these three algorithms for subsequent experiments.

Table 4 Friedman test ranking for 1by1EA-F1, 1by1EA-F2, 1by1EA-F3, 1by1EA-R1, 1by1EA-R2, 1by1EA-R3, and 1by1EA

Algorithm	Average ranking
1by1EA-F1	2.68
1by1EA-F2	2.97
1by1EA-F3	3.90
1by1EA-R1	4.25
1by1EA-R2	3.12
1by1EA-R3	3.75
1by1EA	5.10

Table 5 Results obtained using Holm's post-hoc test

i	Algorithm	α/i	p-value of Holms	Null hypothesis
1	1by1EA-F2	0.05/1 = 0.0500	1.000	Not reject
2	1by1EA-R2	0.05/2 = 0.0250	1.000	Not reject
3	1by1EA-R1	0.05/3 = 0.0166	0.022	Not reject
4	1by1EA-R3	0.05/4 = 0.0125	0.013	Not reject
5	1by1EA-F3	0.05/5 = 0.0100	0.012	Not reject
6	1by1EA	0.05/6 = 0.0083	0.003	Reject

1by1EA-F1 used as control algorithm

5.2.1 Comparison of 1by1EA-IFM with NSGA-III-IFM

In order to avoid the contingency of the results, we also used the same combination as in the original paper. The rest of the parameter settings, the number of iterations, and indicator selection were also the same as those in Experiment 1. We chose the best two algorithms, NSGA-III-F1 and NSGA-III-R1, for comparison to reflect the accuracy of our results.

Table 7 shows the statistical results from the Mann-Whitney-Wilcoxon rank test. Among all the best results highlighted, 1by1EA-IFM accounted for 39, and only six of them appeared in NSGA-III-IFM. It can be seen from Table 7 that the number with better results by 1by1EA-F1, 1by1EA-F2, and 1by1EA-R2 on LSMOPs was 35, 25, and 23, respectively. 1by1EA-F1 still had the best performance among these five algorithms. 1by1EA-F2 was second, while 1by1EA-R2 did not seem to have much of an effect. Furthermore, as listed in Table 8, all the best results were obtained by our proposed algorithm. Unexpectedly, 1by1EA-R2 performed the best on the GD indicator, and the results with superior performance could cover all test example. The performance of the remaining two algorithms on this indicator was the same. Therefore, in general, the performance of 1by1EA-IFM was better. The introduction of the one-by-one selection strategy may have made 1by1EA-IFM superior to NSGA-III-IFM based on reference point selection in terms of convergence. In addition, when we examined the role of the same model, the results of this horizontal comparison were more likely to show the scalability of IFM. Although 1by1EA makes a significant effort to balance convergence and distribution, the retention of information can further improve the utilization of

Table 6 The GD and IGD values

Metric	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-F3	1by1EA-R1	1by1EA-R2	1by1EA-R3
GD	3	300	7	7	6	0	9	7
	5	500	6	6	6	7	6	6
	8	800	6	7	7	7	7	7
	10	1000	8	9	9	8	9	9
	15	1500	9	9	9	9	9	9
IGD	3	300	7	5	6	0	5	5
	5	500	5	5	4	5	4	3
	8	800	6	5	5	6	5	5
	10	1000	6	6	5	6	5	5
	15	1500	5	5	5	5	5	5
Total			65	64	62	53	64	61

Table 7 IGD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, NSGA-III-F1, and NSGA-III-R1 on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	NSGA-III-F1	NSGA-III-R1
LSMOP1	3	300	**1.0273e+0 (4.88e-2) =**	1.1435e+0 (6.29e-2) -	1.1825e+0 (7.86e-2) -	1.0453e+0 (4.69e-2) =	1.0485e+0 (4.76e-2)
	5	500	**9.2042e-1 (5.56e-2) +**	9.4811e-1 (3.14e-2) +	9.4932e-1 (3.49e-2) +	1.0693e+0 (1.01e-1) +	1.1699e+0 (1.27e-1)
	8	800	6.6122e-1 (1.50e-2) +	**6.4347e-1 (1.60e-2) +**	6.6271e-1 (1.59e-2) +	1.0583e+0 (1.16e-1) =	1.0376e+0 (8.24e-2)
	10	1000	6.2582e-1 (1.52e-2) +	**6.1395e-1 (1.92e-2) +**	6.3015e-1 (1.65e-2) +	1.1957e+0 (2.34e-1) =	1.1639e+0 (2.31e-1)
	15	1500	**7.4998e-1 (3.45e-2) +**	7.5195e-1 (4.90e-2) +	7.7725e-1 (5.80e-2) +	2.2478e+0 (4.22e-1) =	2.1306e+0 (7.61e-1)
LSMOP2	3	300	1.9083e-1 (1.25e-2) -	1.8251e-1 (1.58e-2) -	2.1225e-1 (1.40e-2) -	**6.8109e-2 (5.18e-4) =**	6.8110e-2 (6.07e-4)
	5	500	3.0197e-1 (2.37e-2) -	2.8183e-1 (2.76e-2) -	3.1779e-1 (1.89e-2) -	**1.2029e-1 (1.86e-3) =**	1.2152e-1 (2.32e-3)
	8	800	3.9283e-1 (1.65e-2) -	3.7473e-1 (2.18e-2) -	4.0209e-1 (1.49e-2) -	**3.0726e-1 (1.45e-2) =**	3.0994e-1 (1.87e-2)
	10	1000	4.2611e-1 (1.59e-2) -	4.3228e-1 (1.95e-2) -	4.3854e-1 (2.00e-2) -	4.0550e-1 (2.29e-2) -	**3.9116e-1 (1.73e-2)**
	15	1500	**4.8316e-1 (1.88e-2) +**	4.9747e-1 (2.07e-2) +	5.0454e-1 (1.61e-2) +	5.3309e-1 (3.51e-2) =	5.3142e-1 (2.38e-2)
LSMOP3	3	300	**9.5898e+0 (5.54e-1) -**	1.2575e+1 (3.49e+0) -	1.1988e+1 (2.66e+0) -	8.9814e+0 (3.46e-1) =	9.0812e+0 (4.27e-1)
	5	500	**4.8262e+1 (2.34e+1) +**	5.1596e+1 (9.89e+0) +	6.6021e+1 (1.40e+1) +	9.0602e+1 (2.69e+1) =	8.6216e+1 (3.33e+1)
	8	800	**4.6693e+1 (2.18e+1) +**	7.9864e+1 (1.77e+1) +	9.1538e+1 (2.08e+1) +	4.7459e+2 (1.36e+2) =	5.5250e+2 (1.44e+2)
	10	1000	**3.4609e+1 (1.60e+1) +**	6.7181e+1 (1.15e+1) +	8.3233e+1 (1.53e+1) +	4.5996e+2 (2.06e+2) =	5.1145e+2 (1.53e+2)
	15	1500	**6.6509e+1 (3.26e+1) +**	1.0632e+2 (2.32e+1) =	1.0508e+2 (3.26e+1) =	1.1549e+2 (4.77e+1) =	1.1344e+2 (1.02e+1)
LSMOP4	3	300	**2.4401e-1 (2.15e-2) -**	2.4535e-1 (1.74e-2) -	2.5523e-1 (1.78e-2) -	2.2612e-1 (2.00e-3) =	2.2528e-1 (2.46e-3)
	5	500	**2.3851e-1 (1.02e-2) +**	2.4559e-1 (1.02e-2) +	2.5619e-1 (9.29e-3) =	2.5620e-1 (6.54e-3) =	2.5633e-1 (5.20e-3)
	8	800	4.2701e-1 (1.45e-2) -	4.1679e-1 (2.54e-2) -	4.2712e-1 (1.87e-2) -	3.1562e-1 (1.96e-2) =	**3.1129e-1 (1.22e-2)**
	10	1000	4.4769e-1 (1.14e-2) -	4.6010e-1 (1.17e-2) -	4.6261e-1 (1.31e-2) -	**3.7820e-1 (1.58e-2) =**	3.8126e-1 (1.72e-2)
	15	1500	**4.9092e-1 (1.75e-2) +**	4.9804e-1 (1.58e-2) +	5.1110e-1 (1.62e-2) +	5.4992e-1 (3.30e-2) =	5.4695e-1 (2.95e-2)
LSMOP5	3	300	**2.2473e+0 (9.26e-2) +**	2.5014e+0 (1.09e-1) =	2.5628e+0 (1.61e-1) -	2.3765e+0 (1.28e-1) =	2.4715e+0 (1.61e-1)
	5	500	**1.7891e+0 (1.39e-1) +**	2.1628e+0 (1.23e-1) =	2.3186e+0 (1.27e-1) -	2.1866e+0 (1.58e-1) =	2.1600e+0 (2.52e-1)
	8	800	**1.6916e+0 (7.27e-2) +**	1.9175e+0 (1.25e-1) +	2.0469e+0 (1.34e-1) +	2.9444e+0 (5.61e-1) =	3.2412e+0 (7.78e-1)
	10	1000	**1.8122e+0 (1.47e-1) +**	1.9166e+0 (1.03e-1) +	2.0812e+0 (1.32e-1) +	2.9828e+0 (5.05e-1) =	2.8702e+0 (6.32e-1)
	15	1500	1.8019e+0 (1.71e-1) +	**1.7658e+0 (1.37e-1) +**	1.8532e+0 (1.18e-1) +	2.3661e+0 (3.30e-1) =	2.4879e+0 (5.56e-1)
LSMOP6	3	300	**2.5953e+2 (6.56e+1) +**	4.7076e+2 (8.37e+1) +	4.9206e+2 (5.96e+1) +	7.1614e+2 (1.54e+2) =	7.5420e+2 (1.61e+2)
	5	500	**9.6749e+1 (3.46e+1) +**	3.2791e+2 (6.85e+1) -	3.2751e+2 (6.23e+1) -	1.7423e+2 (1.09e+2) =	1.6136e+2 (9.04e+1)
	8	800	**1.6173e+0 (2.19e-2) +**	5.4252e+1 (1.13e+2) =	4.5663e+1 (9.34e+1) =	1.7930e+0 (4.37e-2) =	1.7884e+0 (5.49e-2)
	10	1000	**1.3991e+0 (9.82e-3) +**	4.1835e+1 (8.30e+1) -	8.2547e+1 (1.04e+2) -	1.4414e+0 (8.92e-3) =	3.2920e+0 (8.27e+0)
	15	1500	**1.5010e+2 (3.46e+1) +**	2.2178e+2 (3.97e+1) =	2.3648e+2 (5.17e+1) =	1.8455e+2 (6.26e+1) =	2.1850e+2 (9.49e+1)
LSMOP7	3	300	**1.2945e+0 (5.16e-2) +**	1.3644e+0 (4.28e-2) +	8.1187e+0 (2.09e+1) =	1.4939e+0 (1.58e-2) =	1.4931e+0 (1.77e-2)

Table 7 (continued)

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	NSGA-III-F1	NSGA-III-R1
	5	500	**2.5773e+0 (8.27e-2) +**	4.1687e+1 (1.02e+2) =	3.0486e+1 (6.40e+1) -	2.9911e+0 (1.01e-1) =	2.9614e+0 (7.70e-2)
	8	800	**9.2911e+1 (2.47e+1) +**	2.2766e+2 (3.78e+1) =	2.7649e+2 (4.60e+1) -	2.1632e+2 (5.98e+1) =	2.1857e+2 (7.20e+1)
	10	1000	**1.4373e+2 (2.62e+1) =**	2.6070e+2 (3.51e+1) -	2.7778e+2 (3.84e+1) -	1.8380e+2 (5.80e+1) =	1.7117e+2 (5.13e+1)
	15	1500	**1.7255e+0 (1.26e-2) +**	7.6932e+1 (8.02e+1) -	1.1294e+2 (8.30e+1) -	1.7629e+0 (1.19e-2) =	1.4130e+1 (5.53e+1)
LSMOP8	3	300	5.5532e-1 (4.53e-3) +	**5.5460e-1 (2.76e-3) +**	5.5830e-1 (3.88e-3) +	9.7049e-1 (1.63e-2) +	9.7692e-1 (9.28e-3)
	5	500	9.3379e-1 (2.97e-2) +	**9.2392e-1 (1.66e-2) +**	9.3039e-1 (1.57e-2) +	1.1704e+0 (5.29e-2) =	1.1809e+0 (3.82e-2)
	8	800	**1.1235e+0 (2.43e-2) +**	1.2413e+0 (4.84e-2) +	1.3282e+0 (4.81e-2) +	1.8535e+0 (4.39e-1) =	1.9196e+0 (3.58e-1)
	10	1000	**1.2435e+0 (4.00e-2) +**	1.2981e+0 (4.41e-2) +	1.3400e+0 (5.73e-2) +	1.8709e+0 (2.63e-1) =	1.9739e+0 (2.14e-1)
	15	1500	**1.2713e+0 (2.41e-2) +**	1.2770e+0 (1.80e-2) +	1.2824e+0 (3.73e-2) +	1.3047e+0 (1.38e-2) =	1.3023e+0 (1.75e-2)
LSMOP9	3	300	**1.3354e+1 (8.07e-1) +**	1.4615e+1 (9.20e-1) +	1.5528e+1 (1.14e+0) +	3.0398e+1 (3.97e+0) =	2.9282e+1 (5.20e+0)
	5	500	**3.8021e+1 (1.96e+0) +**	4.0998e+1 (1.68e+0) +	4.3014e+1 (1.48e+0) +	1.0937e+2 (1.42e+1) =	1.1027e+2 (1.41e+1)
	8	800	**7.1140e+1 (2.18e+0) +**	7.2537e+1 (2.38e+0) +	7.4044e+1 (2.51e+0) +	2.5767e+2 (4.68e+1) =	2.6664e+2 (4.12e+1)
	10	1000	1.0992e+2 (5.01e+0) +	**1.0806e+2 (2.94e+0) +**	1.1229e+2 (4.63e+0) +	6.0376e+2 (9.30e+1) =	5.9636e+2 (1.89e+2)
	15	1500	4.1853e+2 (2.22e+1) +	**3.9874e+2 (1.20e+1) +**	4.0824e+2 (1.44e+1) +	2.2210e+3 (7.61e+2) =	2.1716e+3 (8.94e+2)
+/-/=			35/8/2	25/13/7	23/17/5	2/1/42	

Table 8 GD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, NSGA-III-F1, and NSGA-III-R1 on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	NSGA-III-F1	NSGA-III-R1
LSMOP1	3	300	**1.6184e-1 (3.76e-2) +**	2.9020e-1 (1.27e-1) +	2.6997e-1 (4.00e-2) +	9.0971e-1 (2.09e-1) =	8.3689e-1 (1.58e-1)
	5	500	**3.8119e-1 (8.63e-2) +**	4.1140e-1 (8.54e-2) +	3.8886e-1 (1.02e-1) +	9.4192e-1 (2.65e-1) =	1.0119e+0 (2.12e-1)
	8	800	2.7851e-1 (1.15e-1) +	1.7653e-1 (7.32e-2) +	**1.5364e-1 (7.19e-2) +**	1.2041e+0 (1.27e-1) =	1.2109e+0 (1.20e-1)
	10	1000	2.1363e-1 (1.26e-1) +	**7.7347e-2 (6.99e-2) +**	1.3753e-1 (1.26e-1) +	1.6113e+0 (1.64e-1) =	1.6072e+0 (1.73e-1)
	15	1500	3.6464e-2 (2.89e-2) +	6.5306e-2 (1.25e-1) +	**3.1818e-2 (1.57e-2) +**	2.5016e+0 (4.97e-1) =	2.5979e+0 (3.30e-1)
LSMOP2	3	300	2.6948e-3 (8.60e-5) +	2.8613e-3 (7.88e-5) +	**2.5176e-3 (7.10e-5) +**	4.4770e-3 (8.34e-5) =	4.4072e-3 (1.30e-4)
	5	500	2.5967e-3 (1.24e-4) +	2.5828e-3 (1.60e-4) +	**2.5749e-3 (2.23e-4) +**	4.7158e-3 (5.60e-5) =	4.7328e-3 (5.41e-5)
	8	800	**5.9498e-3 (6.23e-4) +**	6.1259e-3 (6.01e-4) +	6.1439e-3 (8.32e-4) +	1.5432e-2 (1.05e-3) =	1.5372e-2 (9.12e-4)
	10	1000	**6.8418e-3 (8.43e-4) +**	6.9215e-3 (3.79e-4) +	7.0820e-3 (5.48e-4) +	1.2402e-2 (4.22e-4) =	1.2254e-2 (3.24e-4)
	15	1500	**1.4453e-2 (1.17e-3) +**	1.5040e-2 (1.16e-3) +	1.5368e-2 (9.44e-4) +	2.1152e-2 (8.53e-4) =	2.1250e-2 (8.11e-4)
LSMOP3	3	300	**3.1789e+2 (5.83e+2) +**	3.3676e+2 (5.15e+2) +	3.9333e+2 (4.42e+2) +	9.5366e+2 (2.60e+2) =	8.3266e+2 (2.44e+2)
	5	500	**2.6523e+2 (2.19e+2) +**	3.4093e+2 (2.08e+2) +	3.9361e+2 (2.49e+2) +	1.4103e+3 (2.16e+2) =	1.3414e+3 (2.14e+2)
	8	800	8.5257e+2 (2.79e+2) +	8.2769e+2 (2.15e+2) +	**7.9454e+2 (2.25e+2) +**	1.8104e+4 (3.02e+3) =	1.8420e+4 (2.96e+3)
	10	1000	7.3123e+2 (5.41e+2) +	7.4292e+2 (4.01e+2) +	**6.8144e+2 (3.07e+2) +**	2.4132e+4 (4.66e+3) =	2.4468e+4 (3.51e+3)
	15	1500	**3.5586e+2 (3.21e+2) +**	5.8771e+2 (4.00e+2) +	4.3635e+2 (4.34e+2) +	2.6844e+3 (4.54e+2) =	2.7774e+3 (6.31e+2)
LSMOP4	3	300	1.4892e-2 (1.40e-3) +	1.5686e-2 (9.71e-4) +	**1.4019e-2 (1.59e-3) +**	2.5877e-2 (4.84e-4) -	2.5508e-2 (4.89e-4)
	5	500	1.3676e-2 (8.81e-4) +	1.4196e-2 (1.87e-3) +	**1.2872e-2 (1.59e-3) +**	3.7311e-2 (8.85e-4) =	3.7339e-2 (8.84e-4)
	8	800	7.3241e-3 (5.18e-4) +	7.0675e-3 (6.62e-4) +	**6.6224e-3 (4.83e-4) +**	1.3433e-2 (1.58e-3) =	1.2792e-2 (1.02e-3)
	10	1000	**7.1250e-3 (5.71e-4) +**	7.7519e-3 (7.53e-4) +	7.8968e-3 (4.73e-4) +	1.1199e-2 (4.33e-4) =	1.1336e-2 (4.13e-4)
	15	1500	**1.5939e-2 (8.83e-4) +**	1.6279e-2 (8.53e-4) +	1.6288e-2 (1.17e-3) +	2.1412e-2 (6.75e-4) =	2.1245e-2 (8.65e-4)
LSMOP5	3	300	**9.6195e-1 (7.92e-1) +**	1.1971e+0 (7.77e-1) +	1.5247e+0 (1.20e+0) +	2.7926e+0 (6.41e-1) =	2.5381e+0 (3.84e-1)
	5	500	1.1530e+0 (6.78e-1) +	7.4897e-1 (2.83e-1) +	**6.4139e-1 (1.30e-1) +**	2.1483e+0 (6.24e-1) =	2.3819e+0 (7.55e-1)
	8	800	2.9728e-1 (1.99e-1) +	2.3698e-1 (1.28e-1) +	**2.3654e-1 (1.71e-1) +**	3.6491e+0 (5.67e-1) =	3.8068e+0 (8.83e-1)
	10	1000	1.2450e-1 (6.34e-2) +	1.5344e-1 (1.25e-1) +	**1.1036e-1 (4.64e-2) +**	4.2818e+0 (1.34e+0) =	3.9150e+0 (1.29e+0)
	15	1500	1.1428e-1 (2.09e-2) +	**1.0781e-1 (2.03e-2) +**	1.1878e-1 (1.72e-2) +	4.1968e+0 (2.11e+0) =	3.8565e+0 (1.84e+0)
LSMOP6	3	300	1.1403e+4 (3.21e+4) +	**8.6914e+3 (1.55e+4) +**	1.2947e+4 (1.86e+4) +	1.4837e+4 (6.64e+3) =	1.5253e+4 (7.91e+3)
	5	500	4.4019e+3 (3.58e+3) +	1.9114e+3 (6.19e+2) +	**1.5406e+3 (6.95e+2) +**	2.7382e+4 (2.01e+4) =	3.8639e+4 (2.87e+4)

Table 8 (continued)

Problem	M	N	1by1EA-F1	1by1EA-F2	1by1EA-R2	NSGA-III-F1	NSGA-III-R1
	8	800	**1.4922e+3 (5.01e+2) +**	2.1797e+3 (1.18e+3) +	2.2034e+3 (1.22e+3) +	1.9157e+4 (3.03e+3) =	1.7568e+4 (2.94e+3)
	10	1000	2.2099e+3 (2.39e+3) +	1.7140e+3 (8.87e+2) +	**1.6686e+3 (1.21e+3) +**	1.8553e+4 (3.13e+3) -	1.6380e+4 (2.53e+3)
	15	1500	1.2483e+2 (1.91e+2) +	**1.1133e+2 (1.59e+2) +**	2.0001e+2 (1.63e+2) +	6.6275e+3 (4.26e+3) =	6.5575e+3 (3.17e+3)
LSMOP7	3	300	2.7269e+4 (4.45e+4) =	1.7125e+4 (1.30e+4) =	**7.0977e+3 (6.49e+3) +**	1.6920e+4 (5.04e+3) =	1.8431e+4 (4.80e+3)
	5	500	3.7531e+3 (2.85e+3) +	**3.1387e+3 (2.31e+3) +**	3.6840e+3 (2.58e+3) +	2.4817e+4 (3.02e+3) =	2.4705e+4 (2.86e+3)
	8	800	4.1069e+3 (6.32e+3) +	2.1661e+3 (1.67e+3) +	**1.3687e+3 (7.03e+2) +**	9.4245e+3 (3.65e+3) =	1.2134e+4 (6.19e+3)
	10	1000	1.0429e+3 (7.15e+2) +	**6.2992e+2 (5.12e+2) +**	9.2407e+2 (8.55e+2) +	6.5978e+3 (2.95e+3) =	7.1982e+3 (2.27e+3)
	15	1500	**6.3652e+2 (4.68e+2) +**	9.1699e+2 (5.45e+2) +	6.8123e+2 (8.52e+2) +	1.1949e+4 (4.27e+3) =	1.3725e+4 (5.05e+3)
LSMOP8	3	300	**3.9742e-1 (1.82e-1) +**	4.1292e-1 (1.32e-1) +	4.1424e-1 (2.09e-1) +	2.5623e+0 (1.49e-1) =	2.5990e+0 (8.89e-2)
	5	500	**2.9053e-1 (4.58e-2) +**	3.2536e-1 (3.66e-2) +	3.9153e-1 (1.01e-1) +	2.3747e+0 (5.12e-1) =	2.3150e+0 (5.09e-1)
	8	800	**7.2672e-2 (8.75e-2) +**	7.8257e-2 (9.40e-2) +	7.6169e-2 (5.46e-2) +	1.8299e+0 (4.78e-1) =	1.7515e+0 (3.38e-1)
	10	1000	**4.3347e-2 (9.67e-3) +**	4.9099e-2 (6.52e-3) +	5.0076e-2 (6.17e-3) +	1.9415e+0 (5.05e-1) =	1.9976e+0 (6.66e-1)
	15	1500	2.1524e-1 (1.33e-1) +	**1.7652e-1 (1.70e-1) +**	1.9728e-1 (2.02e-1) +	2.3494e+0 (2.52e-1) -	2.1471e+0 (2.68e-1)
LSMOP9	3	300	**1.8356e+0 (2.48e-1) +**	1.8567e+0 (2.38e-1) +	2.1751e+0 (3.46e-1) +	1.1567e+1 (1.03e+0) =	1.1867e+1 (8.53e-1)
	5	500	**3.0919e+0 (2.78e-1) +**	3.8580e+0 (1.31e+0) +	3.7487e+0 (6.11e-1) +	3.2890e+1 (1.53e+0) =	3.2860e+1 (1.15e+0)
	8	800	4.1608e+0 (5.41e-1) +	**4.0801e+0 (3.11e-1) +**	4.2141e+0 (2.71e-1) +	7.2421e+1 (2.90e+0) =	7.1084e+1 (2.77e+0)
	10	1000	6.5997e+0 (3.40e-1) +	**6.5371e+0 (3.56e-1) +**	6.7200e+0 (3.50e-1) +	1.3015e+2 (6.42e+0) =	1.3077e+2 (1.01e+1)
	15	1500	3.5964e+1 (2.03e+0) +	**3.4105e+1 (1.06e+0) +**	3.5000e+1 (1.34e+0) +	3.7324e+2 (7.59e+1) =	3.5508e+2 (6.94e+1)
+/-/=			44/0/1	44/0/1	45/0/0	0/2/43	

solutions in populations. As a general information feedback framework, historical research can guide us to further optimize the evolutionary direction of algorithms.

Similarly, in order to more clearly illustrate the results, we took LSMOP5 as an example to show the changes in the IGD and GD values on different objectives with iterations. Figure 3 shows the performance of the five algorithms related to this section. Our algorithm had the best overall performance for all high-dimensional problems. The average value of LSMOP5 in 1by1EA-IFM was lower than that of NSGA-III-IFM. One research focuse of this work was to test whether the algorithm has the ability to solve large-scale MaOPs, and the high-dimensional results were very promising. The results of the comparison of our proposed algorithm with NSGA-III-IFM verify that our algorithm has strong competitiveness in solving the same large-scale MaOPs.

5.2.2 Comparison of 1by1EA-IFM with MOEA/D-IFM

In the experiment at this stage, the value of the objective number m was 5, 10, and 15. The test problem was still LSMOPs. The other parameters in our algorithm were the same as those used in previous experiments. We also used a HyperVolume (HV) [59] indicator to evaluate the results. It measures the size of the area volume value in the target space enclosed by the non-dominated solution set and the reference point obtained by the algorithm. The larger the value, the better the overall performance of the algorithm that was considered. As the best one, MOEA/D-R1 was selected for comparison.

Table 9 shows the best results in each row with emphasis. From all the statistics obtained from the Mann-Whitney-Wilcoxon rank test with the best IGD indicator values, our proposed algorithm occupied 21 of the 27 test examples, while MOEA/D-R1 only occupied six. In particular, 14 were obtained by 1by1EA-F1, which is relatively high, accounting for almost 50% of all test examples. Combined with the HV results shown in Table 10, we could still make this observation. Here, we only provided statistics on the number of HV results that were better than the MOEA/D-R1. From the table, in terms of all objective test problems, 1by1EA-F1 had the best performance, followed by 1by1EA-F2. For the nine test questions with three, five, and 10 objectives, the number of excellent results that the 1by1EA-IFM algorithm could achieve was eight, six, and nine, respectively. Similarly, the results of 1by1EA-F1 in these dimensions were seven, five, and seven, respectively. Therefore, the comprehensive performance of our proposed 1by1EA-IFM was better than that of MOEA/D-IFM. MOEA/D-IFM uses a predefined objective search to solve the problem of a slow search process in a high-dimensional space. Individual selection and retention in IFM improve the sensitivity of predefined targets, especially in a large-scale high-dimensional space. For LSMOP with different

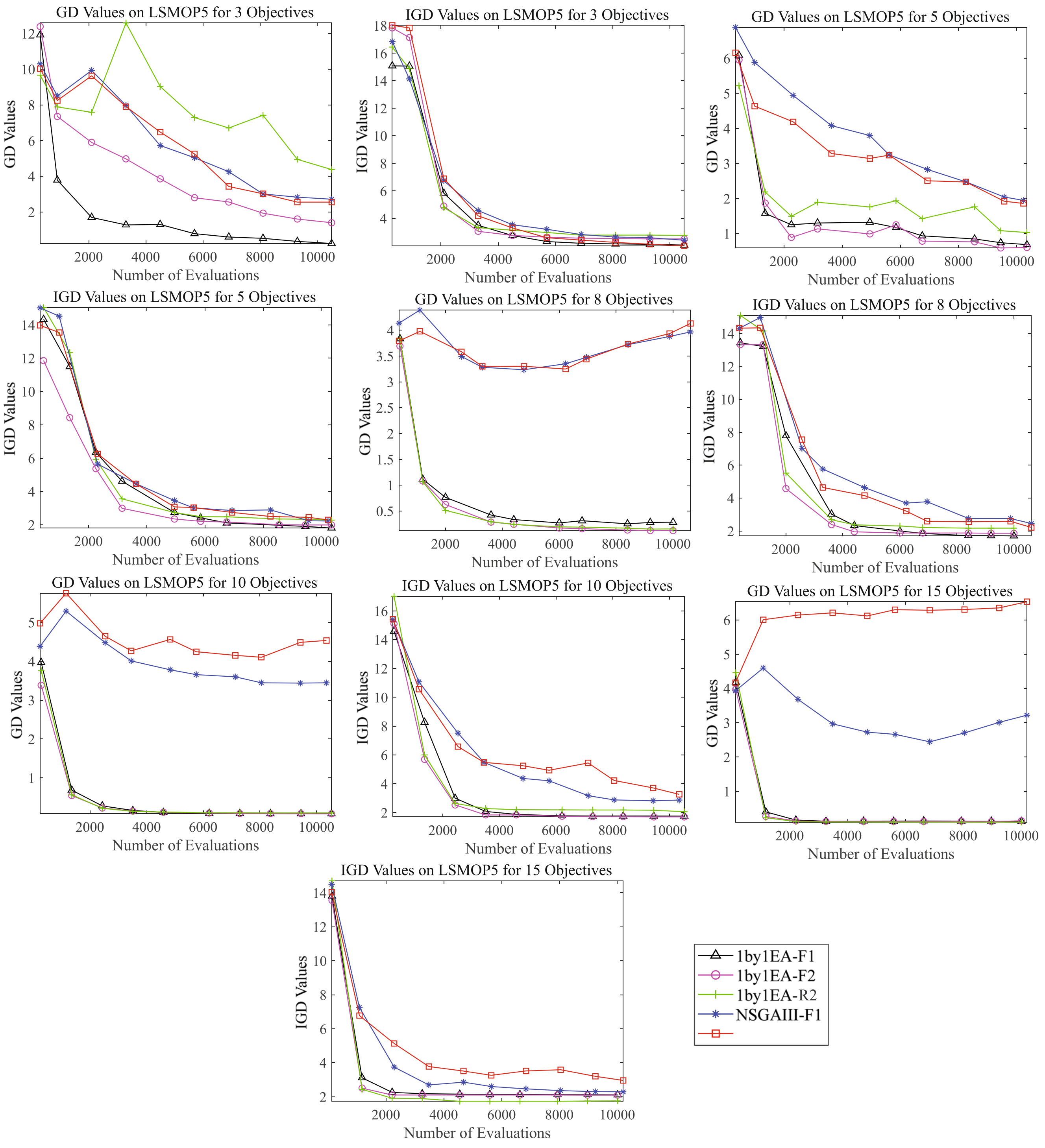


Fig. 3 The mean of IGD and GD values on LSMOP5 with 3, 5, 8, 10, and 15 objectives

fronts, 1by1EA-IFM could solve linear problems more effectively than MOEA/D-IFM, which may have been caused by target decomposition in MOEA/D. However, for the high-dimensional problems of 15 objectives, MOEA/D-IFM could also play a better role; so, the enhancement of the connection between solutions in IFM effectively improved the performance of the evolutionary algorithm.

In order to demonstrate the ability of our proposed algorithm to solve higher-dimensional problems, here, we took LSMOP1-9 with 15 objectives as an example to show the changes in the IGD values obtained by the

Table 9 IGD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, and MOEA/D-R1 on LSMOP1-9 with 5, 10, and 15 objectives

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	MOEA/D-R1
LSMOP1	5	500	**9.2042e-1 (5.56e-2) +**	9.4811e-1 (3.14e-2) +	9.4932e-1 (3.49e-2) +	1.1886e+0 (1.03e-1)
	10	1000	9.4274e-1 (1.37e-2) -	**6.9391e-1 (9.50e-3) +**	7.0879e-1 (1.25e-2) +	8.5627e-1 (7.67e-2)
	15	1500	2.4781e+0 (5.64e-2) -	1.6307e+0 (5.78e-2) -	1.6710e+0 (3.70e-2) -	**1.0461e+0 (1.70e-1)**
LSMOP2	5	500	3.0361e-1 (2.14e-2) -	2.7852e-1 (2.12e-2) -	3.0921e-1 (1.37e-2) -	**2.0992e-1 (1.57e-2)**
	10	1000	**3.6062e-1 (1.27e-2) +**	3.6484e-1 (1.37e-2) +	3.8019e-1 (1.50e-2) +	4.0387e-1 (1.67e-2)
	15	1500	**4.1813e-1 (9.98e-3) +**	4.2476e-1 (7.59e-3) +	4.2658e-1 (8.06e-3) +	4.6083e-1 (1.69e-2)
LSMOP3	5	500	5.5538e+1 (2.58e+1) +	5.5366e+1 (1.73e+1) +	**5.3944e+1 (8.61e+0) +**	1.1079e+2 (6.25e+1)
	10	1000	1.4817e+2 (6.43e+1) +	**9.8114e+1 (1.04e+1) +**	9.9901e+1 (1.17e+1) +	3.1961e+2 (8.16e+1)
	15	1500	**8.7291e+1 (3.09e+1) +**	1.0893e+2 (6.55e+0) +	1.1053e+2 (8.71e+0) +	2.4198e+2 (1.18e+2)
LSMOP4	5	500	**2.4603e-1 (1.08e-2) +**	2.5482e-1 (1.06e-2) +	2.7326e-1 (1.72e-2) +	2.9874e-1 (1.86e-2)
	10	1000	**3.7896e-1 (1.26e-2) +**	3.8820e-1 (1.35e-2) +	3.9866e-1 (1.16e-2) +	4.1017e-1 (1.64e-2)
	15	1500	**4.2418e-1 (1.04e-2) +**	4.2715e-1 (1.03e-2) +	4.3419e-1 (9.78e-3) +	4.7260e-1 (1.23e-2)
LSMOP5	5	500	**1.7703e+0 (1.18e-1) +**	2.1843e+0 (1.19e-1) =	2.2564e+0 (1.25e-1) =	2.1256e+0 (3.14e-1)
	10	1000	2.3191e+0 (7.88e-2) -	1.9803e+0 (6.32e-2) -	2.0824e+0 (6.40e-2) -	**1.7920e+0 (1.78e-1)**
	15	1500	3.8384e+0 (1.38e-1) -	2.7882e+0 (1.00e-1) -	2.8476e+0 (7.19e-2) -	**1.6023e+0 (1.24e-1)**
LSMOP6	5	500	**1.3027e+2 (4.21e+1) +**	2.8713e+2 (4.74e+1) +	3.3055e+2 (4.14e+1) +	8.3081e+2 (2.82e+2)
	10	1000	**1.4713e+0 (8.64e-3) =**	1.4783e+0 (2.14e-2) =	1.5145e+0 (2.10e-2) =	5.8191e+1 (1.02e+2)
	15	1500	1.3065e+3 (1.31e+2) -	6.3230e+2 (5.54e+1) =	7.2912e+2 (6.60e+1) -	**5.8008e+2 (1.66e+2)**
LSMOP7	5	500	**2.5512e+0 (8.76e-2) +**	1.7483e+1 (6.50e+1) =	1.1134e+1 (3.60e+1) =	7.6350e+1 (1.70e+2)
	10	1000	4.1939e+2 (2.78e+1) +	**3.2728e+2 (4.57e+1) +**	3.8178e+2 (2.71e+1) +	7.2789e+2 (1.49e+2)
	15	1500	**1.7863e+0 (8.29e-3) +**	1.7891e+0 (1.46e-2) +	1.8077e+0 (9.93e-3) +	4.2560e+1 (1.08e+2)
LSMOP8	5	500	**9.3254e-1 (2.90e-2) +**	9.4153e-1 (2.29e-2) +	9.4320e-1 (1.82e-2) +	1.0862e+0 (6.48e-2)
	10	1000	1.4679e+0 (2.99e-2) -	**1.3066e+0 (2.64e-2) +**	1.3885e+0 (3.34e-2) =	1.3662e+0 (7.26e-2)
	15	1500	1.1900e+0 (1.54e-2) +	1.1872e+0 (1.02e-2) +	**1.1807e+0 (1.22e-2) +**	1.2502e+0 (2.60e-2)
LSMOP9	5	500	**3.6495e+1 (1.96e+0) +**	4.0636e+1 (2.08e+0) +	4.2674e+1 (2.09e+0) +	5.0713e+1 (4.17e+0)
	10	1000	1.4376e+2 (2.69e+0) -	**1.1516e+2 (2.51e+0) +**	1.1990e+2 (2.38e+0) +	1.3687e+2 (7.22e+0)
	15	1500	8.6195e+2 (1.68e+1) -	6.3538e+2 (1.30e+1) -	6.4677e+2 (8.64e+0) -	**4.4587e+2 (2.18e+1)**
+/-/=			17/9/1	18/5/4	17/6/4	

Table 10 The optimal HV result of 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, and MOEA/D-R1 on LSMOP1-9 with 5, 10, and 15 objectives

M	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	MOEA/D-R1
5	500	7	0	1	1
10	1000	5	1	0	3
15	1500	7	2	0	0

above algorithm as it iterated. As shown in Fig. 4, on LSMOP2/3/4/7/8, the performance of 1by1EA-IFM was significantly better than that of MOEA/D-R1, that is, in the nine test functions, for more than half of the problems, the algorithms we proposed all outperformed those in previous research. However, it is undeniable that, MOEA/D-IFM was more suitable for solving problems characterized by disconnected of PF, as the results of LSMOP9 demonstrate.

5.2.3 Statistical analysis of the results

Based on the IGD and GD values obtained by all above comparison, we performed the same statistical analysis in two steps. As shown in Tables 11 and 12, the computed ranking by using Friedman's test for the above two comparison were displayed respectively. In two χ^2 distributions with 4 and 3 degrees of freedom respectively,

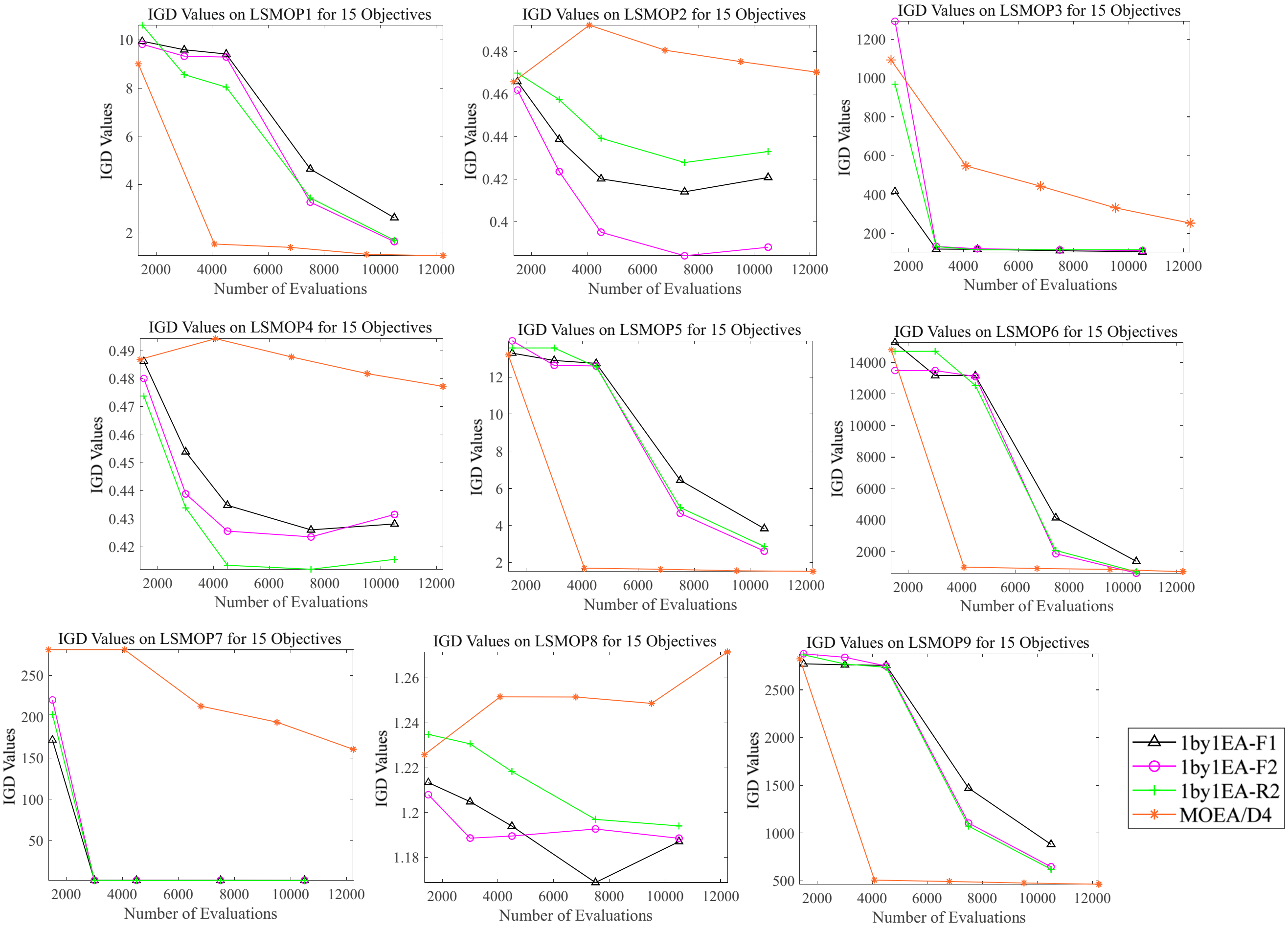


Fig. 4 The mean of IGD values on LSMOP1-9 with 15 objectives

Table 11 Friedman test ranking for 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, NSGA-III-F1, and NSGA-III-R1

Algorithm	Average ranking
1by1EA-F1	2.45
1by1EA-F2	2.59
1by1EA-R2	2.70
NSGA-III-F1	3.69
NSGA-III-R1	3.56

Table 13 Results obtained using Holm's post-hoc test

i	Algorithm	α/i	p-value of Holms	Null hypothesis
1	1by1EA-F2	0.05/1 = 0.0500	1.000	Not reject
2	1by1EA-R2	0.05/2 = 0.0250	1.000	Not reject
3	NSGA-III-R1	0.05/3 = 0.0166	0.009	Reject
4	NSGA-III-F1	0.05/1 = 0.0100	0.004	Reject

1by1EA-F1 used as control algorithm

these average ranking demonstrated that 1by1EA-IFM was different from NSGA-III and MOEA/D-IFM with a p-value = 0.000000008 under the significance level $\alpha = 0.05$. The best ranking was obtained by 1by1EA-IFM, and the performance of our proposed algorithms were close to and generally better than NSGA-III-IFM and MOEA/D-IFM. These ranking were helpful for us to determine the order number in Holm's test.

When we achieved Friedman's test, Holm's method was used to compare the best ranking algorithm with the remaing algorithms. Since the proposed 1byEA-F1 obtained the best ranking, it was considered as control algorithm. Tables 13 and 14 respectively displayed the adjusted p-values by Holm post-hoc procedure based on above two experiments. The null hypothesis was rejected when the achieved p-value lower than the adjusted α ($\alpha = \alpha/i$). After analyzing, all the p-values about NSGA-III-IFM and MOEA/D-IFM obtained by Holm post-hoc were lower than α. Thus, 1by1EA-F1 is statistically better than those algorithms in NSGA-III-IFM and MOEA/D-IFM.

5.2.4 Multi-objective knapsack problem (MOKP)

Here, we adopted MOKP and selected two recently proposed algorithms, AGE-MOEA and PREA, to compare with the results of 1by1EA-IFM and MOEA/D-IFM on this problem.

- AGE-MOEA [60] comprehensively considers the diversity of individuals in the population and the proximity to the ideal point. It does not have a pre-supposed PF, but uses a quick geometry program to estimate PF. The increase in computational complexity is more obvious.
- PREA [61] is a kind of MaOEA based on probability index. PREA introduces a ratio index with infinite norm so that the algorithm can select a promising area in the target space. Then, a population selection strategy based on parallel distance selection is used to offspring populations in this area.

In order to adapt to the problem studied in this paper, we set M to three, five, and eight, and N was set to 300, 500, and 700, respectively. The number of populations and the maximum number of iterations remained unchanged. The weight was a random number between 10 and 100. The experimental results are shown in Table 15. It can be seen that the average results of IGD values obtained by 1by1EA-IFM were significantly better than those of other MaOPs. The best part compared with other comparison algorithms is highlighted in the table. 1by1EA-F2 could obtain the optimal IGD value in the problem of 3 backpacks with 300 items and three backpacks with 500 items. 1by1EA-R2 could reach the smallest IGD value in the remaining problems. This result not only proves the advantages of 1by1EA-IFM in solving MOKP, but also shows that its performance was superior to MOEA/D-IFM.

5.3 Comparision with other many-objective algorithms

5.3.1 Compared algorithms

In order to evaluate the performance of the proposed new algorithm, we selected the best performing algorithm among the six proposed new algorithms and compared it with the following six advanced evolutionary algorithms proposed

Table 12 Friedman test ranking for 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, and MOEA/D-R1

Algorithm	Average ranking
1by1EA-F1	1.78
1by1EA-F2	2.41
1by1EA-R2	1.92
MOEA/D-R1	3.89

Table 14 Results obtained using Holm's post-hoc test

i	Algorithm	α/i	p-value of Holms	Null hypothesis
1	1by1EA-R2	0.05/1 = 0.0500	1.000	Not reject
2	1by1EA-F2	0.05/2 = 0.0250	0.439	Not reject
3	MOEA/D-R1	0.05/3 = 0.0166	0.002	Reject

1by1EA-F1 used as control algorithm

Table 15 IGD values of 1by1EA-F1, 1by1EA-F2, 1by1EA-R2, 1by1EA, AGEMOEA, PREA and MOEA/D-R1 on MOKP

Problem	*M*	*N*	1by1EA-F1	1by1EA-F2	1by1EA-R2	1by1EA	AGEMOEA	PREA	MOEA/D-R1
MOKP	3	300	1.5380e+4 (2.40e+2) -	**1.5058e+4 (1.64e+2)** +	1.5096e+4 (2.27e+2) +	1.7734e+4 (1.43e+2) -	1.7729e+4 (1.19e+2) -	1.7901e+4 (1.09e+2) -	1.5234e+4 (1.13e+2)
MOKP	3	500	2.5936e+4 (2.41e+2) -	**2.5359e+4 (2.08e+2)** +	2.5503e+4 (3.15e+2) +	2.9771e+4 (1.50e+2) -	2.9815e+4 (1.49e+2) -	3.0016e+4 (1.43e+2) -	2.5750e+4 (1.46e+2)
MOKP	3	700	3.5345e+4 (2.93e+2) -	3.4757e+4 (3.40e+2) +	**3.4661e+4 (2.51e+2)** +	4.0512e+4 (1.53e+2) -	4.0577e+4 (1.49e+2) -	4.0793e+4 (2.98e+2) -	3.5081e+4 (2.01e+2)
MOKP	5	300	1.9337e+4 (1.09e+2) +	1.9290e+4 (1.25e+2) +	**1.9267e+4 (9.17e+1)** +	2.1182e+4 (1.19e+2) -	2.0961e+4 (1.63e+2) -	2.1520e+4 (1.92e+2) -	1.9703e+4 (1.25e+2)
MOKP	5	500	3.1486e+4 (2.03e+2) +	3.1375e+4 (1.26e+2) +	**3.1315e+4 (8.51e+1)** +	3.3721e+4 (1.84e+2) -	3.3481e+4 (1.80e+2) -	3.4251e+4 (2.10e+2) -	3.1847e+4 (1.30e+2)
MOKP	5	700	4.4668e+4 (2.21e+2) +	4.4506e+4 (2.31e+2) +	**4.4445e+4 (1.70e+2)** +	4.7798e+4 (1.59e+2) -	4.7437e+4 (1.68e+2) -	4.8488e+4 (3.52e+2) -	4.5067e+4 (2.58e+2)
MOKP	8	300	2.3527e+4 (3.14e+1) +	2.3387e+4 (3.45e+1) +	**2.3328e+4 (2.08e+1)** +	2.4360e+4 (9.13e+1) -	2.3632e+4 (1.26e+2) +	2.4188e+4 (1.75e+2) -	2.3877e+4 (1.13e+2)
MOKP	8	500	3.9506e+4 (2.77e+1) +	3.9328e+4 (4.39e+1) +	**3.9255e+4 (2.66e+1)** +	4.0658e+4 (1.51e+2) -	3.9885e+4 (1.55e+2) =	4.0599e+4 (2.80e+2) -	3.9979e+4 (1.12e+2)
MOKP	8	700	5.5222e+4 (4.65e+1) +	5.5022e+4 (6.36e+1) +	**5.4906e+4 (4.86e+1)** +	5.6709e+4 (1.92e+2) -	5.5698e+4 (2.59e+2) =	5.6790e+4 (4.21e+2) -	5.5733e+4 (2.55e+2)
	+/-/=		6/3/0	9/0/0	9/0/0	0/9/0	1/6/2	0/9/0	

in recent years on LSMOP1-9: MaOEA-IGD, NSGA-II-SDR, DGEA, DEA-GNG, LMOCSO, and MSEA. Based on Section 5.1, we know that the best algorithm was 1by1EA-F1, and this result could be verified in almost all subsequent experiments. The selected comparison algorithms almost cover the main categories of MaOEAs in Section 1, and there are algorithms specifically for solving large-scale MaOPs.

- MaOEA-IGD [34] is a many-objective evolutionary algorithm based on the IGD indicator. Specifically, it uses the IGD index at each generation and adopts a computationally efficient advantage comparison method through a linear allocation mechanism from a global perspective to choose a solution with better convergence and diversity. As the algorithm was evaluated based on the IGD index in the whole process and IGD was included in all the experimental indexes in this paper, this algorithm was selected as the comparison algorithm.
- A new algorithm based on Pareto dominance relation NSGA-II-SDR [62] is proposed. It is an improvement of NSGA-II, using an adaptive niche technology based on the angle between candidate solutions to make only the convergent optimal candidate solution in the niche be the non-dominated solution. This algorithm has shown effectiveness in solving MaOPs.
- DGEA [42] is an algorithm specifically proposed to solve LSMOPs. It uses an adaptive offspring generation method to generate promising candidate solutions in high-dimensional spaces. However, the algorithm has not been verified in the dimension of many-objective problems. Based on this, we performed a comparison in high dimensions.
- DEA-GNG [63] is a decomposition-based multi-objective evolutionary algorithm, which is guided by the growing neural gas network (GNG), and uses the nodes in the GNG as reference vectors according to an adaptive strategy. The method proposed by this algorithm is relatively novel; so we chose this algorithm as a comparison algorithm.
- LMOCSO [64] combines a competitive swarm optimizer (CSO)-based with LSMOPs, adopting a new particle updating strategy that suggests a two-stage strategy to update position, so that search efficiency is improved. Similar to DGEA, the method is also used to solve LSMOPs, but when the number of objectives is increased, the performance of LMOCSO has not been proven, which is why we chose it as a comparison method.
- As a newly proposed algorithm, MSEA [65] also allows populations to evolve toward PF through diversity preservation. This method divides the optimization

process into multiple stages, and updates the population with different selection strategies at different stages. Since MSEA is also based on diversity improvement algorithm, we selected it for comparison to reflect the performance of IFM for population diversity.

5.3.2 Parameter settings

All comparison algorithms used the following parameter settings. In simulated binary crossover and polynomial mutation, both distribution indexes were set to 20. The probability of crossover and mutation was 1.0 and $1/n$, respectively, where n is the number of decision variables. The reason for these settings is discussed in Section 5.1, and the fixed setting could highlight the robustness and generality of algorithms for solving various MaOPs. The maximum number of iterations is set to 10000, and the maximum number of iterations would not exceed 13000. The population numbers of the algorithm were the same as in the settings of the original paper.

For the setting of some specific parameters in the algorithm, the performance was best when the number of direction vectors was set to 10 in DGEA. In DEA-GNG, the parameter α, used to determine the number of iterations for training GNG at each generation, was set to 0.1, and the default learning coefficient parameter eps was set to 0.314. In MaOEA-IGD, the number of evaluations for nadir point estimation DNPE was set to $100*N$. The remaining three algorithms did not involve specific parameters. These parameter settings were set empirically in the same manner as in the original studies. It was verified that the settings have achieved average performance over all the test suites in their studies.

5.3.3 Experimental results and analysis

Tables 16 and 17 show the results of different algorithms in terms of their average values of IGD and GD, where the values with better results are emphasized. When algorithm A was compared to algorithm B, "+" and "-" indicate the number of instances in which the results of A were significantly better and obviously worse than B, respectively. "=" means that there was no statistically significant number of instances between the two algorithms. Therefore, as shown in the last row in the above tables, all comparison algorithms were compared to our proposed 1by1EA-F1. We can see from Table 16 that 1by1EA-F1 achieved the 22 best IGD results among the 45 test instances, and the number of best results obtained by MaOEA-IGD, NSGA-II-SDR, DEA-GNG, DGEA, LMOCSO, and MSEA was 5, 0, 4, 4, 8, and 2, respectively. On LSMOP5/8/9, 1by1EA-F1 could perform optimally in almost all objective dimensions. On LSMOP1/3/6, our algorithm also achieved some excellent results. As shown in Table 17, the 31 best GD results among the 45 test instances were obtained by 1by1EA-F1, while the other algorithms only achieved 2, 10, 0, 0, 2, and 0. At the same time, 1by1EA-F1 fully demonstrated its superiority in the problem of objectives in all dimensions of LSMOP1/2/4/5/9. This means that 1by1EA-F1 had better performance on most test problems.

Another observation is that 1by1EA-F1 was more suitable for solving LSMOP5-9 where the hyperplane was not linear in terms of convergence and diversity of equilibrium solutions, while 1by1EA-F1 was suitable for almost all problems in terms of the convergence of the solution. One reason for this may be that for non-dominated solutions with nonlinear distributions, the algorithm can choose more randomly when selecting a single solution in a historical iteration. As the numbers of objective dimensions and decision variables increase, each single solution selected can exist more dispersed in the large-scale search space. This demonstrates the effectiveness of IFM in solving large-scale problems and in promoting population convergence. Another interesting point is that as LSMOP3 had the most complex multi-modal fitness landscape among the nine test problems, 1by1EA-F1 performed poorly on LSMOP3 with different objectives, which indicates that IFM has difficulty in always obtaining the best performance for large-scale test problems. This may be related to the shape of the PF. However, as this was an extreme case, after the above analysis, we can still conclude that 1by1EA-F1 can effectively solve large-scale MaOPs.

In the test functions for PF related to the convex landscape, the performance of MaOEA-IGD was better than the results of the test for the PF of linear. As MaOEA-IGD uses reference point-based assignment and the convex landscape has better reference point distribution, it was competitive on LSMOP5-8 with partial dimensionality. However, this also made it inferior to other algorithms on LSMOP1-4.

NSGA-II-SDR uses a niching technique based on Pareto dominance; so, it was very competitive in terms of convergence in comparison with other algorithms. However, for some specific LSMOPs, the inability to regulate the niche size affected its performance.

DEA-GNG did not seem to perform well in any LSMOPs except LSMOP2. As LSMOP2 is a partially separable problem, and PF has a mixed modality, GNG-based reference vectors in a complex problem can be adaptively adjusted; so, the population diversity is better. However, this may be because of the influence of the curvature information around the node. So, the algorithm performance could not be enhanced in large-scale problems.

While DGEA is also associated with large-scale problems, we found that it did not do well on almost all LSMOPs

Table 16 IGD values of MaOEA-IGD, NSGA-II-SDR, DEA-GNG, DGEA, LMOCSO, MSEA and 1by1EA-F1 on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	*M*	*N*	MaOEA-IGD	NSGA-II-SDR	DEA-GNG	DGEA	LMOCSO	MSEA	1by1EA-F1
LSMOP1	3	300	1.0715e+1 (9.14e-1) -	6.2340e+0 (2.85e-1) -	4.9686e+0 (2.95e-1) -	2.5561e+0 (5.15e-1) -	1.5345e+0 (9.62e-2) -	4.5645e+0 (3.62e-1) -	**1.0214e+0 (5.37e-2)**
	5	500	1.7822e+1 (1.37e+0) -	6.2701e+0 (3.98e-1) -	7.1209e+0 (3.96e-1) -	2.5302e+0 (1.06e+0) -	1.5941e+0 (1.14e-1) -	7.9858e+0 (5.05e-1) -	**9.1838e-1 (5.11e-2)**
	8	800	8.4956e+0 (9.48e-1) -	7.2094e+0 (4.62e-1) -	7.7245e+0 (3.53e-1) -	2.1779e+0 (1.08e+0) -	9.9104e-1 (6.19e-2) -	8.3818e+0 (3.45e-1) -	**7.8289e-1 (2.15e-2)**
	10	1000	7.1844e+0 (1.33e-1) -	7.3569e+0 (2.68e-1) -	7.6511e+0 (2.34e-1) -	2.1665e+0 (1.58e+0) -	**9.3946e-1 (8.64e-2) =**	8.1223e+0 (3.18e-1) -	9.4257e-1 (1.93e-2)
	15	1500	7.4402e+0 (1.99e-1) -	7.7233e+0 (3.85e-1) -	7.4021e+0 (2.71e-1) -	3.3401e+0 (2.05e+0) =	**9.8093e-1 (8.87e-2) +**	7.4767e+0 (3.59e-1) -	2.4829e+0 (1.10e-1)
LSMOP2	3	300	5.8940e-1 (1.78e-3) -	8.0859e-2 (3.03e-4) +	8.3872e-2 (3.48e-4) +	8.2866e-2 (9.99e-4) +	**7.5816e-2 (6.72e-4) +**	8.2454e-2 (1.78e-4) +	1.8084e-1 (2.01e-2)
	5	500	6.9791e-1 (1.28e-2) -	1.2265e-1 (1.24e-3) +	**1.1781e-1 (6.62e-4) +**	1.5105e-1 (4.62e-3) +	1.1995e-1 (3.93e-4) +	1.2113e-1 (8.63e-4) +	3.0375e-1 (1.65e-2)
	8	800	3.7623e-1 (2.51e-2) -	2.1214e-1 (1.35e-2) +	**1.9406e-1 (3.08e-3) +**	3.2615e-1 (1.93e-2) =	1.9667e-1 (7.26e-3) +	1.9852e-1 (6.12e-3) +	3.3154e-1 (1.38e-2)
	10	1000	2.6206e-1 (3.48e-3) +	2.2927e-1 (1.00e-2) +	**2.1143e-1 (3.08e-3) +**	3.7888e-1 (1.71e-2) -	2.7502e-1 (2.98e-2) +	2.1280e-1 (4.71e-3) +	3.6158e-1 (7.35e-3)
	15	1500	3.1881e-1 (6.88e-3) +	3.1233e-1 (6.59e-3) +	2.9876e-1 (7.41e-3) +	4.5378e-1 (4.30e-2) -	3.7806e-1 (4.01e-2) +	**2.5096e-1 (9.33e-3) +**	4.2999e-1 (7.93e-3)
LSMOP3	3	300	1.1717e+1 (4.41e-1) -	1.4765e+1 (5.27e-1) -	1.4691e+1 (8.88e-1) -	**7.7806e+0 (5.35e+0) =**	1.4410e+1 (1.85e+0) -	1.1972e+1 (9.80e-1) -	9.3536e+0 (4.27e-1)
	5	500	1.2973e+1 (5.26e-1) +	1.9059e+1 (2.55e+0) +	4.1206e+2 (4.50e+2) -	**9.2889e+0 (2.64e+0) +**	1.7328e+1 (4.14e+0) +	2.5346e+1 (7.55e+0) +	4.3360e+1 (1.85e+1)
	8	800	2.4510e+1 (1.95e+0) +	3.6692e+1 (6.20e+0) +	8.5910e+3 (3.43e+3) -	**2.0905e+1 (1.10e+1) +**	2.2432e+1 (4.86e+0) +	7.0182e+2 (8.62e+2) =	1.1191e+2 (4.75e+1)
	10	1000	2.4640e+1 (1.44e+0) +	4.2756e+1 (2.18e+1) +	7.1825e+3 (2.55e+3) -	5.0270e+1 (1.00e+2) +	**2.2568e+1 (4.34e+0) +**	1.1817e+3 (1.39e+3) -	1.2802e+2 (7.13e+1)
	15	1500	2.3997e+1 (1.89e+0) +	6.4594e+1 (3.03e+1) +	1.1066e+2 (1.10e+1) -	**1.1269e+1 (5.51e-1) +**	2.6839e+1 (6.60e+0) +	2.3698e+2 (4.11e+2) =	9.5702e+1 (2.25e+1)
LSMOP4	3	300	8.5608e-1 (3.60e-3) -	2.5404e-1 (1.64e-3) =	2.6922e-1 (1.86e-3) =	2.6121e-1 (8.59e-3) =	**2.3824e-1 (2.21e-3) =**	2.6827e-1 (2.58e-3) =	2.5082e-1 (2.83e-2)
	5	500	8.1694e-1 (1.35e-2) -	2.6897e-1 (1.07e-2) -	2.7591e-1 (3.81e-3) -	3.0666e-1 (1.42e-2) -	**2.4256e-1 (1.48e-3) =**	2.7711e-1 (2.98e-3) -	2.4661e-1 (1.35e-2)
	8	800	3.9668e-1 (3.82e-2) -	2.4218e-1 (1.15e-2) +	2.3234e-1 (3.34e-3) +	3.3878e-1 (1.26e-2) +	**2.3125e-1 (1.21e-2) +**	2.3832e-1 (4.00e-3) +	3.7278e-1 (1.07e-2)
	10	1000	2.8222e-1 (4.60e-3) +	2.5035e-1 (4.97e-3) +	**2.3884e-1 (3.32e-3) +**	3.7744e-1 (2.22e-2) =	2.6234e-1 (1.80e-2) +	2.4334e-1 (4.99e-3) +	3.8155e-1 (1.24e-2)
	15	1500	3.3446e-1 (1.15e-2) +	3.1998e-1 (6.78e-3) +	3.1171e-1 (7.59e-3) +	4.5503e-1 (3.82e-2) =	3.9427e-1 (3.44e-2) +	**2.7187e-1 (1.07e-2) +**	4.3504e-1 (7.83e-3)
LSMOP5	3	300	8.9431e+0 (8.30e-1) -	1.4949e+1 (1.19e+0) -	8.6329e+0 (6.67e-1) -	6.0649e+0 (1.67e+0) -	3.3560e+0 (1.75e-1) -	3.8007e+0 (3.61e-1) -	**2.2020e+0 (1.76e-1)**
	5	500	7.1123e+0 (1.31e+0) -	1.2078e+1 (1.76e+0) -	1.2738e+1 (1.47e+0) -	5.6856e+0 (3.51e+0) =	4.0761e+0 (4.06e-1) -	1.2915e+1 (2.29e+0) -	**1.7549e+0 (1.13e-1)**
	8	800	8.3601e+0 (4.72e-1) -	1.4143e+1 (1.76e+0) -	1.6022e+1 (1.54e+0) -	7.5795e+0 (4.28e+0) -	4.1582e+0 (3.53e-1) -	2.0077e+1 (2.86e+0) -	**1.9990e+0 (1.00e-1)**
	10	1000	1.1429e+1 (3.36e-1) -	1.6288e+1 (1.87e+0) -	1.7307e+1 (6.38e-1) -	9.9902e+0 (3.24e+0) -	3.6260e+0 (2.90e-1) -	2.0829e+1 (2.97e+0) -	**2.3168e+0 (1.15e-1)**
	15	1500	9.0215e+0 (4.97e-1) -	1.2170e+1 (6.96e-1) -	1.5062e+1 (1.16e+0) -	5.3716e+0 (3.60e+0) =	2.8648e+0 (1.63e-1) -	2.1351e+1 (2.76e+0) -	**2.5401e+0 (8.37e-2)**
LSMOP6	3	300	1.6685e+4 (9.92e+3) -	5.4293e+3 (6.78e+2) -	3.0049e+3 (5.95e+2) -	3.8587e+3 (2.97e+3) -	8.2995e+2 (2.64e+2) -	1.7638e+3 (4.90e+2) -	**2.8554e+2 (7.69e+1)**
	5	500	3.6651e+2 (1.05e+2) -	1.6597e+3 (7.46e+2) -	2.4593e+3 (9.02e+2) -	2.7900e+3 (2.40e+3) =	6.7857e+2 (2.98e+2) -	2.1216e+3 (2.13e+3) -	**1.1301e+2 (3.67e+1)**

Table 16 (continued)

Problem	*M*	*N*	MaOEA-IGD	NSGA-II-SDR	DEA-GNG	DGEA	LMOCSO	MSEA	1by1EA-F1
	8	800	**1.5438e+0 (3.15e-2) +**	1.6967e+0 (2.82e-2) =	3.1141e+4 (2.57e+4) -	9.3174e+1 (2.33e+2) -	1.8102e+0 (3.08e-2) -	1.4688e+3 (4.50e+3) -	1.7076e+0 (2.60e-2)
	10	1000	**1.4692e+0 (7.80e-3) =**	1.5097e+0 (5.99e-3) -	2.6167e+4 (2.43e+4) -	1.8525e+2 (7.12e+2) -	1.5117e+0 (1.58e-2) -	1.5452e+1 (5.38e+1) -	1.4698e+0 (6.79e-3)
	15	1500	3.9593e+3 (5.49e+2) -	1.0874e+4 (1.42e+3) -	1.5483e+4 (3.52e+3) -	6.9542e+3 (4.25e+3) -	9.2656e+2 (1.58e+2) -	8.3821e+4 (2.58e+4) -	**5.2267e+2 (6.04e+1)**
LSMOP7	3	300	**1.1935e+0 (2.79e-2) +**	1.2371e+0 (6.44e-2) +	1.5257e+0 (1.75e-2) +	1.2520e+0 (1.02e-1) +	1.4525e+0 (5.51e-2) +	1.5188e+0 (3.13e-2) +	3.0279e+0 (6.15e+0)
	5	500	**1.6442e+0 (1.14e-1) +**	1.9563e+0 (2.16e-1) +	2.7502e+4 (2.25e+4) -	2.5088e+0 (6.89e-1) =	2.6668e+0 (1.66e-1) -	5.7406e+2 (2.55e+3) -	2.5681e+0 (8.78e-2)
	8	800	1.5507e+3 (3.32e+2) -	5.8585e+3 (1.81e+3) -	1.0080e+4 (3.52e+3) -	7.0206e+3 (3.83e+3) -	1.9312e+3 (4.79e+2) -	6.1111e+4 (1.86e+4) -	**2.8800e+2 (3.83e+1)**
	10	1000	6.3825e+3 (8.30e+2) -	1.2681e+4 (1.02e+3) -	1.7218e+4 (4.34e+3) -	8.4444e+3 (5.38e+3) -	1.5958e+3 (2.27e+2) -	7.2022e+4 (1.95e+4) -	**3.9375e+2 (5.19e+1)**
	15	1500	**1.6168e+0 (1.42e-2) +**	1.7678e+0 (8.33e-3) =	1.4188e+4 (1.35e+4) -	1.6479e+0 (1.89e-1) =	1.7951e+0 (7.87e-3) -	4.0090e+3 (6.73e+3) -	1.7737e+0 (1.08e-2)
LSMOP8	3	300	9.7855e-1 (5.14e-4) -	7.3880e-1 (7.50e-2) -	8.0298e-1 (1.16e-1) -	6.5527e-1 (5.03e-2) -	5.8824e-1 (5.00e-3) -	6.5122e-1 (2.14e-2) -	**5.5358e-1 (4.63e-3)**
	5	500	1.1564e+0 (1.70e-2) -	1.1720e+0 (2.93e-2) -	7.0975e+0 (3.51e+0) -	1.1130e+0 (6.09e-2) -	9.5294e-1 (2.79e-2) =	2.5987e+0 (1.84e+0) -	**9.4830e-1 (3.26e-2)**
	8	800	4.1839e+0 (3.11e-1) -	8.3180e+0 (2.02e+0) -	8.0754e+0 (9.96e-1) -	6.7751e+0 (2.04e+0) -	3.1303e+0 (2.58e-1) -	1.2829e+1 (1.73e+0) -	**1.2102e+0 (2.64e-2)**
	10	1000	6.6061e+0 (3.10e-1) -	9.3319e+0 (1.08e+0) -	1.0329e+1 (7.45e-1) -	5.5268e+0 (3.35e+0) -	3.0234e+0 (2.59e-1) -	1.4735e+1 (2.02e+0) -	**1.4603e+0 (2.27e-2)**
	15	1500	1.3225e+0 (4.35e-4) -	1.2554e+0 (7.83e-2) =	3.9561e+0 (2.33e+0) -	1.2751e+0 (3.09e-2) -	1.2326e+0 (1.35e-2) =	7.7114e+0 (4.99e+0) -	**1.2274e+0 (1.61e-2)**
LSMOP9	3	300	2.3923e+1 (2.66e+0) -	3.3389e+1 (1.71e+0) -	3.1739e+1 (2.42e+0) -	5.0285e+1 (7.90e+0) -	**1.1106e+1 (5.05e+0) +**	2.3472e+1 (2.41e+0) -	1.3535e+1 (9.04e-1)
	5	500	9.1868e+1 (6.33e+0) -	1.5050e+2 (5.42e+0) -	2.1328e+2 (9.62e+0) -	1.4290e+2 (3.42e+1) -	7.6211e+1 (3.19e+1) -	1.4552e+2 (1.36e+1) -	**3.6816e+1 (2.16e+0)**
	8	800	3.8449e+2 (1.38e+1) -	5.1347e+2 (1.54e+1) -	7.2587e+2 (2.22e+1) -	3.4433e+2 (4.45e+1) -	2.4957e+2 (6.84e+1) -	6.6937e+2 (2.51e+1) -	**8.1108e+1 (1.72e+0)**
	10	1000	7.1737e+2 (1.52e+1) -	9.4183e+2 (1.52e+1) -	1.1672e+3 (2.34e+1) -	5.4204e+2 (6.86e+1) -	4.5485e+2 (1.09e+2) -	1.1371e+3 (2.83e+1) -	**1.4315e+2 (2.81e+0)**
	15	1500	1.7848e+3 (2.82e+1) -	2.4925e+3 (3.49e+1) -	2.9499e+3 (5.14e+1) -	1.3277e+3 (1.38e+2) -	1.3077e+3 (1.49e+2) -	2.8600e+3 (5.76e+1) -	**5.7542e+2 (8.85e+0)**
+/-/=			12/32/1	14/27/4	9/35/1	8/26/11	15/25/5	10/32/3	

Table 17 GD values of MaOEA-IGD, NSGA-II-SDR, DEA-GNG, DGEA, LMOCSO, MSEA and 1by1EA-F1 on LSMOP1-9 with 3, 5, 8, 10, and 15 objectives

Problem	*M*	*N*	MaOEA-IGD	NSGA-II-SDR	DEA-GNG	DGEA	LMOCSO	MSEA	1by1EA-F1
LSMOP1	3	300	6.8977e+0 (2.20e+0) -	1.4091e+0 (2.15e-1) -	1.1287e+0 (1.05e-1) -	1.4211e+0 (5.08e-1) -	7.1596e-1 (5.29e-2) -	1.0528e+0 (1.21e-1) -	**1.6048e-1 (9.02e-2)**
	5	500	2.5226e+0 (3.09e-1) -	8.8197e-1 (1.53e-1) -	1.0197e+0 (1.09e-1) -	1.4591e+0 (3.83e-1) -	5.2489e-1 (4.82e-2) -	1.0531e+0 (1.18e-1) -	**4.4447e-1 (1.43e-1)**
	8	800	7.5383e-1 (3.96e-2) -	4.8192e-1 (1.51e-1) -	6.1701e-1 (4.71e-2) -	1.5047e+0 (1.90e-1) -	7.5330e-1 (1.11e-1) -	8.4454e-1 (5.21e-2) -	**2.6812e-1 (5.57e-2)**
	10	1000	6.2580e-1 (2.91e-2) -	3.5878e-1 (3.39e-2) -	5.7613e-1 (3.13e-2) -	1.5326e+0 (1.34e-1) -	1.0512e+0 (1.40e-1) -	7.2014e-1 (2.13e-2) -	**1.3612e-1 (4.79e-2)**
	15	1500	6.6006e-1 (2.25e-2) -	2.9510e-1 (5.07e-3) -	5.8105e-1 (2.69e-2) -	2.3509e+0 (2.34e-1) -	1.5728e+0 (2.35e-1) -	5.7933e-1 (3.41e-2) -	**1.2261e-1 (2.55e-2)**
LSMOP2	3	300	1.5871e-2 (1.74e-3) -	4.9651e-3 (9.06e-5) -	5.3560e-3 (1.05e-4) -	5.5432e-3 (1.71e-4) -	5.2546e-3 (6.25e-5) -	5.4410e-3 (1.01e-4) -	**2.7030e-3 (1.23e-4)**
	5	500	9.8226e-3 (6.40e-4) -	4.1604e-3 (8.86e-5) -	4.0899e-3 (6.17e-5) -	6.5385e-3 (2.40e-4) -	4.6770e-3 (3.23e-5) -	4.4849e-3 (7.31e-5) -	**2.4031e-3 (1.31e-4)**
	8	800	1.5101e-2 (3.05e-3) -	6.5403e-3 (1.50e-3) -	7.5367e-3 (4.85e-4) -	1.7174e-2 (1.66e-3) -	7.7121e-3 (6.19e-4) -	9.4807e-3 (4.64e-4) -	**4.5304e-3 (4.05e-4)**
	10	1000	1.1392e-2 (7.23e-4) -	5.3394e-3 (2.41e-4) -	5.9152e-3 (1.12e-4) -	1.9228e-2 (8.76e-4) -	1.0523e-2 (1.53e-3) -	6.0050e-3 (1.51e-4) -	**3.9897e-3 (1.37e-4)**
	15	1500	1.1009e-2 (1.15e-3) -	5.8003e-3 (8.41e-5) -	8.0106e-3 (2.10e-4) -	2.8829e-2 (2.80e-3) -	1.9020e-2 (3.59e-3) -	6.4222e-3 (2.88e-4) -	**5.4311e-3 (5.61e-5)**
LSMOP3	3	300	1.1477e+2 (9.95e+1) =	8.7653e+0 (5.62e+0) +	1.0911e+3 (1.57e+2) -	1.0219e+1 (1.09e+1) +	**7.5613e+0 (1.53e+0) +**	7.0471e+2 (9.22e+1) -	2.1559e+2 (3.62e+2)
	5	500	1.5568e+2 (1.10e+2) =	6.6748e+0 (2.93e+0) +	1.9346e+3 (1.64e+2) -	1.9200e+1 (1.26e+1) +	**5.5103e+0 (4.11e+0) +**	2.5465e+3 (1.34e+2) -	1.6186e+2 (1.52e+2)
	8	800	3.2920e+3 (2.05e+3) -	**5.4708e+1 (3.35e+1) +**	7.2626e+3 (1.32e+3) -	2.2522e+3 (1.21e+3) -	8.5129e+2 (1.58e+3) -	8.8263e+3 (1.58e+3) -	8.2216e+2 (2.25e+2)
	10	1000	3.7070e+3 (2.00e+3) -	**1.2866e+2 (5.40e+1) +**	8.5654e+3 (1.15e+3) -	2.6611e+3 (1.15e+3) -	6.9459e+2 (1.26e+3) =	1.0201e+4 (9.18e+2) -	6.8537e+2 (1.50e+2)
	15	1500	2.1160e+3 (1.48e+3) -	**5.5443e+1 (3.28e+1) +**	1.4005e+3 (4.20e+1) -	1.4645e+3 (9.66e+2) -	2.0619e+2 (2.70e+2) +	1.5464e+3 (8.96e+1) -	5.3039e+2 (1.19e+2)
LSMOP4	3	300	8.3579e-2 (1.28e-2) -	2.3155e-2 (4.30e-4) -	2.7387e-2 (5.73e-4) -	2.4775e-2 (1.01e-3) -	2.3613e-2 (2.59e-4) -	2.7971e-2 (3.48e-4) -	**1.4354e-2 (1.94e-3)**
	5	500	5.5403e-2 (6.25e-3) -	2.2370e-2 (3.49e-3) -	2.8351e-2 (1.29e-3) -	3.5227e-2 (3.08e-3) -	2.0492e-2 (3.19e-4) -	3.4222e-2 (1.04e-3) -	**1.2462e-2 (9.24e-4)**
	8	800	2.5064e-2 (1.73e-3) -	9.2424e-3 (1.08e-3) -	1.2361e-2 (6.83e-4) -	2.1218e-2 (1.92e-3) -	1.1581e-2 (1.43e-3) -	1.3438e-2 (8.31e-4) -	**4.9580e-3 (1.94e-4)**
	10	1000	1.1562e-2 (1.33e-3) -	7.2756e-3 (4.57e-4) -	9.3281e-3 (4.19e-4) -	1.8571e-2 (1.22e-3) -	1.1981e-2 (1.48e-3) -	9.4688e-3 (7.81e-4) -	**4.3568e-3 (5.69e-5)**
	15	1500	1.2898e-2 (5.67e-4) -	6.1995e-3 (8.89e-5) -	8.1163e-3 (1.38e-4) -	3.0183e-2 (3.28e-3) -	1.9411e-2 (2.61e-3) -	7.1989e-3 (4.08e-4) -	**5.7548e-3 (8.87e-5)**
LSMOP5	3	300	1.4116e+1 (2.76e+0) -	6.5132e+0 (6.89e-1) -	5.0261e+0 (3.53e-1) -	8.3546e+0 (5.93e+0) -	4.2763e+0 (5.31e-1) -	5.8436e+0 (1.04e+0) -	**9.1646e-1 (8.00e-1)**
	5	500	5.5018e+0 (1.71e+0) -	2.6092e+0 (6.08e-1) -	6.0704e+0 (1.68e-1) -	1.1756e+1 (1.01e+1) -	2.5037e+0 (3.78e-1) -	6.4976e+0 (1.69e-1) -	**8.2532e-1 (3.16e-1)**
	8	800	3.1802e+0 (8.37e-1) -	2.2351e+0 (2.53e-1) -	5.2765e+0 (1.86e-1) -	9.7383e+0 (8.76e+0) -	3.0068e+0 (6.25e-1) -	5.9669e+0 (1.25e-1) -	**4.3199e-1 (2.50e-1)**
	10	1000	2.8104e+0 (4.09e-1) -	2.8661e+0 (5.96e-1) -	4.7216e+0 (1.68e-1) -	5.7755e+0 (6.21e+0) -	4.4779e+0 (6.32e-1) -	5.3795e+0 (1.44e-1) -	**1.7371e-1 (7.90e-2)**
	15	1500	1.2898e-2 (5.67e-4) -	6.1995e-3 (8.89e-5) -	8.1163e-3 (1.38e-4) -	3.0183e-2 (3.28e-3) -	1.9411e-2 (2.61e-3) -	7.1989e-3 (4.08e-4) -	**5.7548e-3 (8.87e-5)**

Table 17 (continued)

Problem	*M*	*N*	MaOEA-IGD	NSGA-II-SDR	DEA-GNG	DGEA	LMOCSO	MSEA	1by1EA-F1
LSMOP6	3	300	4.2915e+4 (2.55e+4) -	**9.9564e+3 (2.74e+3)** =	1.7047e+4 (3.85e+3) =	3.7977e+4 (5.86e+4) =	2.8646e+4 (4.63e+4) =	1.6970e+4 (4.51e+3) =	2.7408e+4 (4.74e+4)
	5	500	2.0694e+3 (2.68e+3) +	**1.8036e+3 (5.30e+2)** +	2.7860e+4 (2.58e+3) -	2.7195e+4 (3.23e+4) -	1.7022e+4 (6.83e+3) -	1.9236e+4 (6.25e+3) -	4.9488e+3 (4.28e+3)
	8	800	7.8029e-2 (2.34e-2) +	**3.6612e-2 (2.97e-3)** +	4.5196e+4 (1.83e+3) -	2.4388e+4 (1.18e+4) -	1.9036e+4 (8.77e+3) -	5.5386e+4 (3.29e+3) -	3.7887e+3 (8.51e+2)
	10	1000	4.4129e-2 (4.92e-3) +	**1.5839e-2 (9.97e-4)** +	3.8965e+4 (2.51e+3) -	2.1040e+4 (6.80e+3) -	3.1798e+4 (1.16e+4) -	4.7930e+4 (3.43e+3) -	3.6258e+3 (7.00e+2)
	15	1500	5.9545e+3 (2.88e+3) -	1.6184e+3 (5.52e+2) -	3.3721e+4 (2.36e+3) -	2.4729e+4 (3.02e+4) -	2.8429e+4 (1.65e+4) -	4.5449e+4 (2.21e+3) -	**3.9568e+2 (4.60e+2)**
LSMOP7	3	300	**1.0560e-1 (2.70e-2)** +	7.8550e-1 (1.07e+0) +	9.1859e+3 (9.16e+2) =	9.2710e+3 (1.83e+4) +	1.7416e+2 (3.71e+2) +	3.7300e+3 (4.10e+2) +	2.1504e+4 (2.27e+4)
	5	500	**1.4587e-1 (5.83e-2)** +	2.4515e-1 (5.26e-1) +	4.3130e+4 (2.45e+3) -	4.6475e+4 (3.04e+4) -	1.2809e+3 (2.92e+3) +	4.5834e+4 (3.80e+3) -	3.8671e+3 (2.47e+3)
	8	800	6.7354e+3 (5.02e+3) -	**1.8671e+3 (1.96e+2)** +	4.4173e+4 (3.03e+3) -	2.2233e+4 (2.26e+4) -	3.0215e+4 (1.47e+4) -	5.8635e+4 (2.37e+3) -	2.5519e+3 (1.29e+3)
	10	1000	9.8705e+3 (4.88e+3) -	2.5579e+3 (1.99e+2) -	4.0664e+4 (2.40e+3) -	2.5022e+4 (2.24e+4) -	4.0622e+4 (1.73e+4) -	5.3754e+4 (1.32e+3) -	**1.7851e+3 (1.30e+3)**
	15	1500	7.7432e-2 (1.21e-2) +	**2.7040e-2 (2.46e-3)** +	3.2272e+4 (1.84e+3) -	2.0480e+4 (8.47e+3) -	2.7216e+4 (1.61e+4) -	4.3156e+4 (2.55e+3) -	1.5080e+3 (5.72e+2)
LSMOP8	3	300	3.3861e+0 (8.19e-1) -	6.5171e-1 (9.00e-2) -	1.0128e+0 (9.80e-2) -	9.6646e-1 (3.70e-1) -	1.1758e+0 (1.49e-1) -	6.2476e-1 (7.95e-2) -	**3.7396e-1 (1.59e-1)**
	5	500	2.2541e+0 (3.25e-1) -	4.3581e-1 (4.79e-2) -	2.7774e+0 (8.31e-2) -	2.0398e+0 (3.98e-1) -	6.2893e-1 (6.75e-2) -	2.8982e+0 (1.14e-1) -	**3.2793e-1 (6.84e-2)**
	8	800	3.1971e+0 (4.26e-1) -	2.6563e+0 (1.59e-1) -	2.6475e+0 (8.47e-2) -	3.8165e+0 (3.61e+0) -	2.0835e+0 (2.41e-1) -	3.0498e+0 (6.22e-2) -	**1.6347e-1 (1.44e-1)**
	10	1000	2.2583e+0 (3.14e-1) -	2.7380e+0 (4.55e-2) -	2.3674e+0 (6.20e-2) -	7.7890e+0 (6.47e+0) -	2.7719e+0 (4.51e-1) -	2.7436e+0 (5.19e-2) -	**9.1363e-2 (5.67e-2)**
	15	1500	6.9160e-1 (2.24e-1) -	**2.1983e-2 (2.33e-2)** +	2.2203e+0 (7.41e-2) -	2.0097e+0 (2.63e-1) -	2.1158e+0 (5.76e-1) -	2.4295e+0 (6.17e-2) -	2.4407e-1 (8.57e-2)
LSMOP9	3	300	5.3594e+0 (8.71e-1) -	6.9058e+0 (7.88e-1) -	9.4999e+0 (1.31e+0) -	1.5737e+1 (4.39e+0) -	3.6974e+0 (2.93e+0) -	6.7413e+0 (7.02e-1) -	**1.8391e+0 (2.81e-1)**
	5	500	1.3046e+1 (1.23e+0) -	1.6413e+1 (1.53e+0) -	2.5551e+1 (1.72e+0) -	6.7938e+1 (3.34e+1) -	4.3287e+1 (2.77e+1) -	2.4788e+1 (3.38e+0) -	**3.1320e+0 (5.53e-1)**
	8	800	3.6847e+1 (1.79e+0) -	2.7828e+1 (1.26e+0) -	6.2490e+1 (1.61e+0) -	1.9573e+2 (4.43e+1) -	1.6750e+2 (7.01e+1) -	6.4544e+1 (1.56e+0) -	**4.0559e+0 (1.17e-1)**
	10	1000	6.3575e+1 (1.69e+0) -	3.8063e+1 (8.51e-1) -	9.6402e+1 (1.89e+0) -	3.4215e+2 (6.65e+1) -	3.5688e+2 (1.58e+2) -	9.2789e+1 (1.92e+0) -	**5.9105e+0 (1.55e-1)**
	15	1500	1.9456e+2 (3.90e+0) -	8.0564e+1 (4.72e-1) -	2.2802e+2 (7.24e+0) -	9.1801e+2 (2.29e+2) -	9.1983e+2 (2.04e+2) -	1.8576e+2 (2.20e+0) -	**1.9233e+1 (1.84e-1)**
+/-/=			6/37/2	13/31/2	0/43/2	3/41/1	5/38/2	1/43/1	

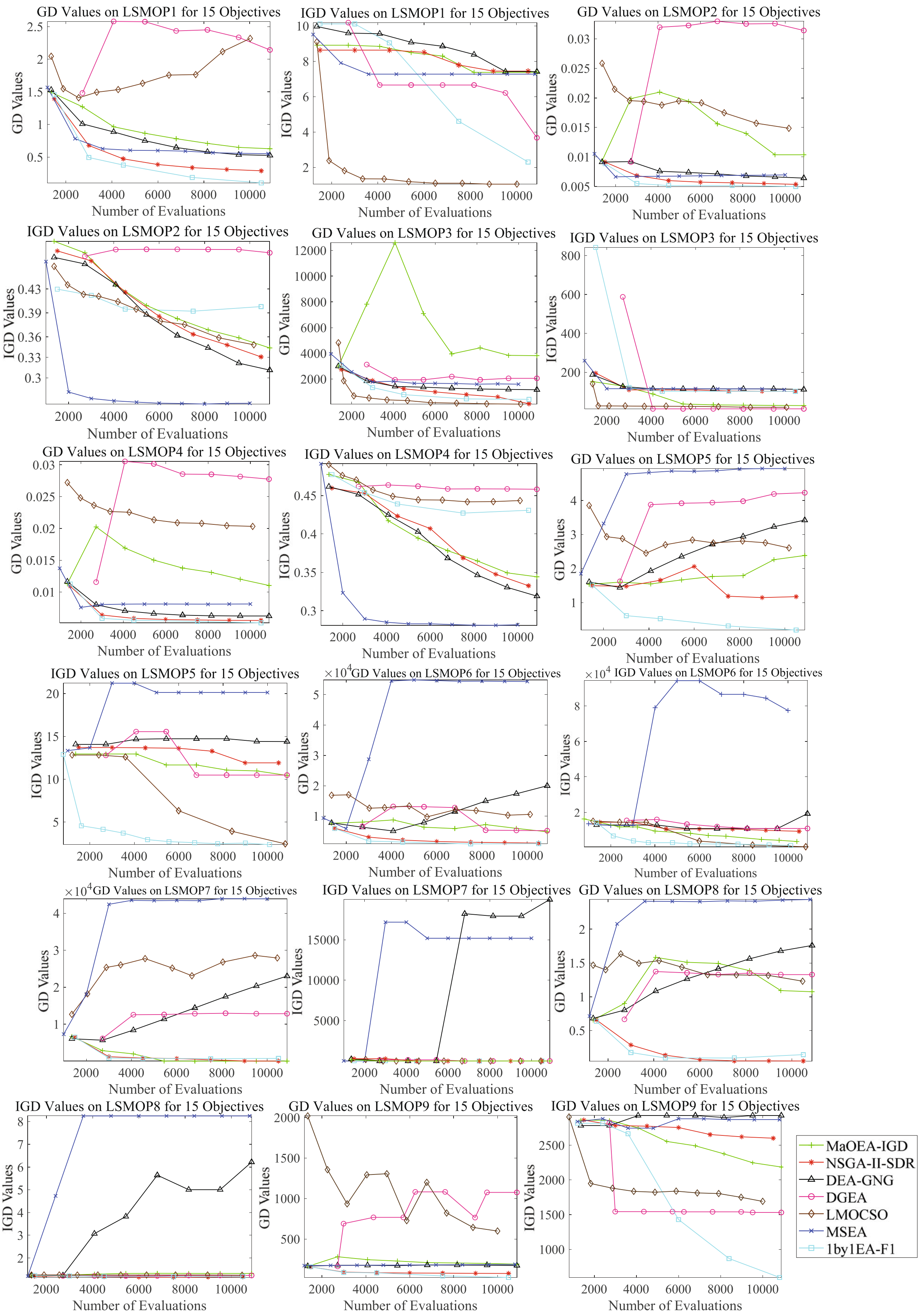


Fig. 5 The mean of IGD and GD values on LSMOP1-9 with 15 objectives

other than LSMOP3. There is a pre-selection strategy in DGEA, which means that it is difficult to always obtain better performance using the same selection strategy on different testing problems. Furthermore, from the results of IGD and GD, DGEA could effectively maintain population diversity, but the convergence in large-scale space was still insufficient.

LMOCSO was proposed to solve large-scale MOPs, and it includes a novel particle update strategy that combines all the updated particles with the original particles. However, the results of the experiment suggest that this information retention strategy may not have as much impact as IFM.

MSEA replaces a solution in the population by improving the selection strategy. From the test results for LSMOPs, the limitations of MSEA were mainly related to the diversity of selection criteria. The normalization in 1by1EA improved the convergence of the algorithm, but the diversity of solution selection criteria may have made the algorithm transform a objective into the wrong scale when solving high-dimensional large-scale problems.

Figure 5 shows the results of IGD and GD values calculated by these seven evolutionary algorithms in a clearer way when $m = 15$. Corresponding to the above table, when the objective number was 15, 1by1EA-F1 could obtain the lowest IGD value on LSMOP6/7/9/10. Then this algorithm could obtain the smaller GD value in almost all LSMOPs with different objectives. This indicates that 1by1EA-IFM can still adapt to most of the LSMOPs in a high-dimensional space, and performs excellently.

5.3.4 Statistical analysis of the results

Similar to the previous experiments, we still performed statistical analysis of the obtained experimental datasets in two steps. As shown in Table 18, the computed ranking through Friedman's test performed the difference among these algorithms. The results of this statistical test showed that 1by1EA-F1 obtained a significant difference with a p-value tending to 0 infinitely under the significance level $\alpha = 0.05$. The best ranking was obtained by 1by1EA-F1, followed by NSGA-II-SDR, DEAGNG, LMOCSO, MaOEA-IGD, MSEA, and DGEA respectively. This order was useful for the later Holm's test.

Next, we applied Holm's method to compare the best ranking algorithm with the remaing algorithms. Since the proposed 1byEA-F1 obtained the best ranking, it was considered as control algorithm. Table 19 demonstrated the adjusted p-values by Holm post-hoc procedure, where the null hypothesis was rejected when the achieved p-value less than α/i. Analyzing these results, we concluded that all the p-values obtained by Holm post-hoc were lower than the adjusted α ($\alpha = \alpha/i$), which confirmed that the proposed 1by1EA-F1 statistically outperforms the other algorithms. Thus, the effectiveness of 1by1EA-IFM is verified.

Table 18 Friedman test ranking for MaOEA-IGD, NSGA-II-SDR, DEA-GNG, DGEA, LMOCSO, MSEA, and 1by1EA-F1

Algorithm	Average Ranking
MaOEA-IGD	4.08
NSGA-II-SDR	3.15
DEA-GNG	3.75
DGEA	6.07
LMOCSO	4.05
MSEA	4.09
1by1EA-F1	2.80

Here, we use the above experimental results to explain again why the M-F1, M-F2 and M-R1 had the best performance in our proposed algorithms. First, in most evolutionary algorithms, the best individual in each environmental selection is retained. The introduction of IFM increases the uncertainty in environment selection. By retaining one, two, or three individuals from a fixed or random position in the historical iteration, the algorithm can avoid falling into the local optimum to a certain extent, thereby improving its convergence. Although the individual retained each time may not be the best, this selection method can indeed speed up the convergence. The above experimental results show that the GD values in most of the experiments were relatively low; so, 1by1EA-IFM made a contribution to the convergence.

The performance of 1by1EA-F1 was undoubtedly superior. This superiority comes from the fact that we only consider the use of information at a fixed location at the last generation, where the individuals are the closest to this generation. Thus, the value and inheritance of information were the highest. Moreover, selecting individuals at a fixed position in each iteration can increase the randomness of individual selection without making this randomness too messy; so, the 1by1EA-F1 had the best performance.

Table 19 Results obtained using Holm's post-hoc test

i	Algorithm	α/i	p-value of Holms	Null hypothesis
1	NSGA-II-SDR	0.05/1 = 0.0500	0.025	Reject
2	DEAGNG	0.05/2 = 0.0250	0.024	Reject
3	LMOCSO	0.05/3 = 0.0166	0.012	Reject
4	MaOEA-IGD	0.05/4 = 0.0125	0.002	Reject
5	MSEA	0.05/5 = 0.0100	0.001	Reject
6	DGEA	0.05/6 = 0.0083	0.000	Reject

1by1EA-F1 used as control algorithm

6 Conclusion

In previous studies, few algorithms have been proposed that can reuse individual information in historical iteration populations. Inevitably, a lot of valuable information often appears in these algorithms. In this paper, a new 1by1EA was proposed based on this mechanism. It integrates information in historical populations and reuses them in subsequent iterations. Furthermore, this paper adopted the new algorithms specifically to solve the large-scale MaOPs. According to the six models of IFM, six algorithms were proposed. After comparing these six algorithms with the original 1by1EA, we found that 1by1EA-F1, 1by1EA-F2, and 1by1EA-R2 had the best overall performance. Among them, 1by1EA-F1 performed particularly well. Therefore, it was chosen for comparison in later experiments. By comparing the algorithms with the best algorithms in NSGA-III-IFM and MOEA/D-IFM on LSMOP1-9 and MOKP, we proved that 1by1EA-IFM could achieve the best performance in most experimental settings. Finally, we selected six relatively state-of-the-art evolutionary algorithms and verified them in the same large-scale MaOPs. The results verify the strong competitiveness of 1by1EA-F1.

In future research, we will focus on exploring why the performance of several other algorithms is not as good as that of 1by1EA-F1, and will make improvements to this problem on a series of MaOPs. Then, we will combine IFM with other MaOEAs to explore its ability to balance the diversity and convergence of algorithms. In addition to MOKP, for other problems such as TSP and scheduling, which have practical application value, we will consider incorporating the improved algorithms into them and observe whether the algorithms have the ability to resolve these problems. For the use of selection strategies, we will develop more strategies based on random selection and compare them with IFM to obtain better results. In addition, allowing the algorithm to learn adaptively to achieve dynamic optimization of the problem is also a direction worthy of consideration in the future.

Author Contributions Conceptualization: Yong Wang, Qian Zhang; methodology: Gai-Ge Wang; software: Gai-Ge Wang; validation: Qian Zhang; formal analysis: Yong Wang; investigation: Qian Zhang; resources: Gai-Ge Wang; data curation: Yong Wang; writing—original draft preparation: Qian Zhang; writing—review and editing: Gai-Ge Wang, Yong Wang; visualization: Yong Wang; supervision: Yong Wang; project administration: Gai-Ge Wang. All authors have read and agreed the final manuscript.

Declarations

Ethics approval and consent to participate This paper does not contain any study with human participants or animals performed by any author.

Consent for Publication All authors agreed with the content and that all gave explicit consent to submit.

Conflict of Interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

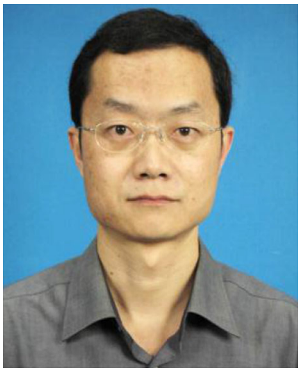
References

1. Zhang W, Hou W, Li C, Yang W, Gen M (2021) Multidirection update-based multiobjective particle swarm optimization for mixed no-idle flow-shop scheduling problem. Compl Syst Model Simul 1(3):176–197
2. Cai X, Hu Z, Chen J (2020) A many-objective optimization recommendation algorithm based on knowledge mining. Inf Sci 537:148–161
3. Lin Q, Liu S, Wong K-C, Gong M, Coello CAC, Chen J, Zhang J (2018) A clustering-based evolutionary algorithm for many-objective optimization problems. IEEE Trans Evol Comput 23(3):391–405
4. Hua Y, Liu Q, Hao K, Jin Y (2021) A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts. IEEE/CAA J Autom Sin 8(2):303–318
5. Fan Q, Ersoy OK (2021) Zoning search with adaptive resource allocating method for balanced and imbalanced multimodal multi-objective optimization. IEEE/CAA J Autom Sin 8(6):1163–1176
6. Zhao F, Di S, Cao J, Tang J et al (2021) A novel cooperative multi-stage hyper-heuristic for combination optimization problems. Compl Syst Model Simul 1(2):91–108
7. Wang G-G, Deb S, Cui Z (2019) Monarch butterfly optimization. Neural Comput Applic 31(7):1995–2014
8. Tang J, Liu G, Pan Q (2021) A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. IEEE/CAA J Autom Sin 8(10):1627–1643
9. Houssein EH, Gad AG, Hussain K, Suganthan PN (2021) Major advances in particle swarm optimization: theory, analysis, and application, vol 63
10. Agarwal SK, Yadav S (2019) A comprehensive survey on artificial bee colony algorithm as a frontier in swarm intelligence. In: Ambient Communications and Computer Systems, Springer, pp 125–134
11. Roopa C, Harish B, Kumar SA (2019) Segmenting ecg and mri data using ant colony optimisation. Int J Artif Intell Soft Comput 7(1):46–58
12. Opara KR, Arabas J (2019) Differential evolution: a survey of theoretical analyses. Swarm Evol Comput 44:546–558
13. Zhang Q, Li H (2007) Moea/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
14. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
15. Zitzler E, Laumanns M, Thiele L (2001) Spea2: improving the strength pareto evolutionary algorithm. TIK-report 103
16. Cui Z, Xue F, Cai X, Cao Y, Wang G-g, Chen J (2018) Detection of malicious code variants based on deep learning. IEEE Trans Ind Inform 14(7):3187–3196
17. Hu Z, Li Z, Dai C, Xu X, Xiong Z, Su Q (2020) Multiobjective grey prediction evolution algorithm for environmental/economic dispatch problem. IEEE Access 8:84162–84176

18. Cui Z, Zhao Y, Cao Y, Cai X, Zhang W, Chen J (2021) Malicious code detection under 5g hetnets based on a multi-objective rbm model. IEEE Netw 35(2):82–87
19. Shadkam E, Bijari M (2020) A novel improved cuckoo optimisation algorithm for engineering optimisation. Int J Artif Intell Soft Comput 7(2):164–177
20. Cai X, Cao Y, Ren Y, Cui Z, Zhang W (2021) Multi-objective evolutionary 3d face reconstruction based on improved encoder–decoder network. Inf Sci 581:233–248
21. Gao D, Wang G-G, Pedrycz W (2020) Solving fuzzy job-shop scheduling problem using de algorithm improved by a selection mechanism. IEEE Trans Fuzzy Syst 28(12):3265–3275
22. Wang G-G, Gao D, Pedrycz W (2022) Solving multi-objective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. IEEE Trans Indust Inf
23. Han X, Han Y, Chen Q, Li J, Sang H, Liu Y, Pan Q, Nojima Y (2021) Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm. Compl Syst Model Simul 1(3):198–217
24. Wang G-G, Cai X, Cui Z, Min G, Chen J (2020) High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm. IEEE Trans Emerg Top Comput 8(1):20–30
25. Thakare AN, Bhagat L, Thomas A (2017) A self-organised routing algorithm for cognitive radio-based wireless sensor networks using biologically-inspired method. Int J Artif Intell Soft Comput 6(2):148–169
26. Dutta S, Das KN (2019) A survey on pareto-based eas to solve multi-objective optimization problems. In: Soft Computing for Problem Solving, Springer, pp 807–820
27. Liu Y, Gong D, Sun J, Jin Y (2017) A many-objective evolutionary algorithm using a one-by-one selection strategy. IEEE Trans Cybern 47(9):2689–2702
28. Wang G-G, Tan Y (2019) Improving metaheuristic algorithms with information feedback models. IEEE Trans Cybern 49(2):542–555
29. Wu T, An S, Han J, Shentu N (2022) An ε-domination based two-archive 2 algorithm for many-objective optimization. J Syst Eng Electron 33(1):156–169
30. Prajapati A (2021) Two-archive fuzzy-pareto-dominance swarm optimization for many-objective software architecture reconstruction. Arab J Sci Eng 46(4):3503–3518
31. Chhabra JK (2018) Amarjeet: Fp-abc: Fuzzy-pareto dominance driven artificial bee colony algorithm for many-objective software module clustering. Comput Lang Syst Struct 51:1–21
32. Zheng W, Tan Y, Meng L, Zhang H (2018) An improved moea/d design for many-objective optimization problems. Appl Intell 48(10):3839–3861
33. Farias LR, Araújol AF (2019) Many-objective evolutionary algorithm based on decomposition with random and adaptive weights. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), IEEE, pp 3746–3751
34. Sun Y, Yen GG, Yi Z (2018) Igd indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Trans Evol Comput 23(2):173–187
35. Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76
36. Li K, Wang R, Zhang T, Ishibuchi H (2018) Evolutionary many-objective optimization: a comparative study of the state-of-the-art. IEEE Access 6:26194–26214
37. Deb K, Jain H (2013) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601
38. Liu R, Wang R, Bian R, Liu J, Jiao L (2021) A decomposition-based evolutionary algorithm with correlative selection mechanism for many-objective optimization. Evol Comput 29(2):269–304
39. Palakonda V, Mallipeddi R, Suganthan PN (2021) An ensemble approach with external archive for multi-and many-objective optimization with adaptive mating mechanism and two-level environmental selection. Inf Sci 555:164–197
40. Zhao H, Zhang C, Zheng X, Zhang C, Zhang B (2022) A decomposition-based many-objective ant colony optimization algorithm with adaptive solution construction and selection approaches. Swarm Evol Comput 68:100977
41. Chen H, Cheng R, Wen J, Li H, Weng J (2020) Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. Inf Sci 509:457–469
42. He C, Cheng R, Yazdani D (2020) Adaptive offspring generation for evolutionary large-scale multiobjective optimization. IEEE Trans Syst Man Cybern Syst 52(2):
43. Tian Y, Zheng X, Zhang X, Jin Y (2019) Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. IEEE Trans Cybern 50(8):3696–3708
44. Gu Z-M, Wang G-G (2020) Improving nsga-iii algorithms with information feedback models for large-scale many-objective optimization. Futur Gener Comput Syst 107:49–69
45. Zhang Y, Wang G-G, Li K, Yeh W-C, Jian M, Dong J (2020) Enhancing moea/d with information feedback models for large-scale many-objective optimization. Inf Sci 522:1–16
46. Saini N, Saha S (2021) Multi-objective optimization techniques: a survey of the state-of-the-art and applications. Eur Phys J Spec Top 230(10):2319–2335
47. Wei G (2017) Some cosine similarity measures for picture fuzzy sets and their applications to strategic decision making. Informatica 28(3):547–564
48. Jiang S, Yang S (2017) A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. IEEE Trans Evol Comput 21(3):329–346
49. Yi J-H, Xing L-N, Wang G-G, Dong J, Vasilakos AV, Alavi AH, Wang L (2020) Behavior of crossover operators in nsga-iii for large-scale optimization problems. Inf Sci 509:470–487
50. Abed-alguni BH, Alawad NA, Barhoush M, Hammad R (2021) Exploratory cuckoo search for solving single-objective optimization problems. Soft Comput 25(15):10167–10180
51. Cheng R, Jin Y, Olhofer M et al (2016) Test problems for large-scale multiobjective and many-objective optimization. IEEE Trans Cybern 47(12):4108–4121
52. Cai X, Xiao Y, Li M, Hu H, Ishibuchi H, Li X (2020) A grid-based inverted generational distance for multi/many-objective optimization. IEEE Trans Evol Comput 25(1):21–34
53. Liu Y, Wei J, Li X, Li M (2019) Generational distance indicator-based evolutionary algorithm with an improved niching method for many-objective optimization problems. IEEE Access 7:63881–63891
54. Chakkarapani K, Thangavelu T, Dharmalingam K, Thandavarayan P (2019) Multiobjective design optimization and analysis of magnetic flux distribution for slotless permanent magnet brushless dc motor using evolutionary algorithms. J Magn Magn Mater 476:524–537
55. Ishibuchi H, Imada R, Setoguchi Y, Nojima Y (2018) Reference point specification in inverted generational distance for triangular linear pareto front. IEEE Trans Evol Comput 22(6):961–975
56. Sun Y, Xue B, Zhang M, Yen GG (2018) A new two-stage evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 23(5):748–761

57. Tian Y, Cheng R, Zhang X, Su Y, Jin Y (2018) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Trans Evol Comput 23(2):331–345
58. Jiao R, Zeng S, Li C, Ong Y-S (2021) Two-type weight adjustments in moea/d for highly constrained many-objective optimization. Inf Sci 578:592–614
59. Rostami S, Neri F (2017) A fast hypervolume driven selection mechanism for many-objective optimisation problems. Swarm Evol Comput 34:50–67
60. Panichella A (2019) An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 595–603
61. Yuan J, Liu H-L, Gu F, Zhang Q, He Z (2020) Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions. IEEE Trans Evol Comput 25(1):75–86
62. Tian Y, Cheng R, Zhang X, Su Y, Jin Y (2019) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Trans Evol Comput 23:331–345
63. Liu Y, Ishibuchi H, Masuyama N, Nojima Y (2020) Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts. IEEE Trans Evol Comput 24:439–453
64. Tian Y, Zheng X, Zhang X, Jin Y (2020) Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. IEEE Trans Cybern 50:3696–3708
65. Tian Y, He C, Cheng R, Zhang X (2021) A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization. IEEE Trans Syst Man Cybern Syst 51:5880–5894

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Yong Wang received the Ph.D degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2009. He is currently an associate professor of software technology in the College of Computer Scicence and Technology, Ocean University of China, where he is also the Deputy Director of the College of Computer Science and Technology. He has authored or coauthored more than 30 referred papers in the area of artificial intelligence and software engineering. His current research interests include intelligent computing and operations research.

Qian Zhang was born in November 1998 in Shandong Province, China. She successfully entered Qufu Normal University in 2016 and majored in computer science and technology. In June 2020, she successfully obtained a bachelor's degree from Qufu Normal University, and continued her graduate study at Ocean University of China. Her major is software engineering. Her main research direction is large-scale many-objective optimization.

Gai-Ge Wang is an associate professor in Ocean University of China, China. His entire publications have been cited over 5200 times (Google Scholar). Thirteen and thirty-nine papers are selected as Highly Cited Paper by Web of Science, and Scopus (till November 2019), respectively. One paper is selected as ”Top Articles from Outstanding S&T Journals of China-F5000 Frontrunner”. The latest Google h-index and i10-index are 40 and 78, respectively. He is senior member of SAISE, SCIEI, a member of IEEE, IEEE CIS, ISMOST. He served as Editors-in-Chief of ***OAJRC Computer and Communications***, Editorial Advisory Board Member of ***Communications in Computational and Applied Mathematics (CCAM)***, Associate Editor of ***IJCISIM***, an Editorial Board Member of ***IEEE Access, Mathematics, IJBIC, Karbala International Journal of Modern Science,*** and ***Journal of Artificial Intelligence and Systems***. He served as Guest Editor for many journals including ***Mathematics, IJBIC, FGCS, Memetic Computing*** and ***Operational Research***. His research interests are swarm intelligence, evolutionary computation, and big data optimization.