mathematics

MDPI

Article

NSGA-II/SDR-OLS: A Novel Large-Scale Many-Objective Optimization Method Using Opposition-Based Learning and Local Search

Yingxin Zhang [1], Gaige Wang [1,*] and Hongmei Wang [2]

[1] School of Computer Science and Technology, Ocean University of China, Qingdao 266100, China; zhangyingxin@stu.ouc.edu.cn

[2] Information Engineering College, Xinjiang Institute of Engineering, Urumqi 830023, China

* Correspondence: wgg@ouc.edu.cn

Abstract: Recently, many-objective optimization problems (MaOPs) have become a hot issue of interest in academia and industry, and many more many-objective evolutionary algorithms (MaOEAs) have been proposed. NSGA-II/SDR (NSGA-II with a strengthened dominance relation) is an improved NSGA-II, created by replacing the traditional Pareto dominance relation with a new dominance relation, termed SDR, which is better than the original algorithm in solving small-scale MaOPs with few decision variables, but performs poorly in large-scale MaOPs. To address these problems, we added the following improvements to the NSGA-II/SDR to obtain NSGA-II/SDR-OLS, which enables it to better achieve a balance between population convergence and diversity when solving large-scale MaOPs: (1) The opposition-based learning (OBL) strategy is introduced in the initial population initialization stage, and the final initial population is formed by the initial population and the opposition-based population, which optimizes the quality and convergence of the population; (2) the local search (LS) strategy is introduced to expand the diversity of populations by finding neighborhood solutions, in order to avoid solutions falling into local optima too early. NSGA-II/SDR-OLS is compared with the original algorithm on nine benchmark problems to verify the effectiveness of its improvement. Then, we compare our algorithm with six existing algorithms, which are promising region-based multi-objective evolutionary algorithms (PREA), a scalable small subpopulation-based covariance matrix adaptation evolution strategy (S3-CMA-ES), a decomposition-based multi-objective evolutionary algorithm guided by growing neural gas (DEA-GNG), a reference vector-guided evolutionary algorithm (RVEA), NSGA-II with conflict-based partitioning strategy (NSGA-II-conflict), and a genetic algorithm using reference-point-based non-dominated sorting (NSGA-III).The proposed algorithm has achieved the best results in the vast majority of test cases, indicating that our algorithm has strong competitiveness.

Keywords: evolutionary algorithm; many-objective optimization; large-scale optimization; opposition-based learning; local search

MSC: 65K10

check for updates

Citation: Zhang, Y.; Wang, G.; Wang, H. NSGA-II/SDR-OLS: A Novel Large-Scale Many-Objective Optimization Method Using Opposition-Based Learning and Local Search. *Mathematics* **2023**, *11*, 1911. https://doi.org/10.3390/math11081911

Academic Editor: Catalin Stoean

Received: 13 March 2023
Revised: 11 April 2023
Accepted: 16 April 2023
Published: 18 April 2023

1. Introduction

The multi-objective optimization problems (MOPs) have been the focus of academic and engineering fields. Many real-world problems are MOPs, such as big data [1,2], image [3,4], feature selection [5,6], community detection [7], engineering design [8,9], shop floor scheduling [10,11], and medical services [12]. Usually, the objectives in these problems are conflicting and mutually constrained, and the improvement of one objective may lead to the deterioration of another one. Therefore, there is no single solution that can optimize all objectives at the same time. Instead, one aims for an optimal compromise solution, called a Pareto optimal solution [13].

To solve above problems, some traditional methods, such as the Newton method, quasi-Newton method, and gradient descent method, can suffer from the problems [14], such as the tendency to fall into local optima and a poor convergence of approximate solutions. Evolutionary algorithms (EAs) are increasingly used to deal with MOPs because of their population-based nature and their ability to approximate the entire Pareto fronts (PFs) of MOPs in a single run [15], and these EAs are called multi-objective evolutionary algorithms (MOEAs). Among many evolutionary algorithms, some of the most representative ones include non-dominated sorting in genetic algorithms (NSGA) [16], fast and elitist multi-objective genetic algorithms (NSGA-II) [17] and genetic algorithms (GA) using reference-point-based non-dominated sorting (NSGA-III) [18,19], and multi-objective evolutionary algorithms based on decomposition (MOEA/D) [20].

However, most real-world problems often involve three or even more objectives, and such problems are informally called MaOPs. Most MOEAs have a sharp decline in effectiveness when faced with MaOPs. The main reason is that the increase in the number of objectives reduces the selection difficulty of the algorithm on the true PF, which makes it difficult to converge to the true Pareto front. In addition, the increase of the number of objectives will lead to the increase of the computational cost. Therefore, an ever-growing number of studies have started to focus on improving the MaOEAs to solve the above problems. Among them, a large proportion of MaOEAs is obtained by introducing some new strategies based on the original MOEAs, such as NSGA-III [18,19] and MOEA/DD [21]. These algorithms not only inherit the framework and advantages of the original algorithms, but the new added strategies further optimize the shortcomings of the original algorithms. Experimental results also proved that these new strategies are effective, so a growing number of researchers started to introduce the algorithms using different strategies. For example, Liu et al. [22] proposed a decomposition-based MaOEA, called MOEA/D-CSM, to solve MaOPs. They introduced a new concept related to the set of reference points, and designed a new selection mechanism based on correlation, which is called a correlation selection mechanism.

NSGA-II, proposed by Deb et al. [17] in 2002, reduced the complexity of non-inferiority-sorting genetic algorithms and had the advantages of fast operation and good convergence of the solution set, which became a benchmark for the performance of other MOEAs. However, its performance also suffers from severe dimensional catastrophe when facing MaOPs, so Tian et al. [23] proposed a new dominance relation, called SDR, for this problem in 2019. Replacing the traditional Pareto dominance relation with this new dominance relation, the new algorithm NSGA-II/SDR was proposed. The proposed SDR dominance relation can bring considerable improvements to NSGA-II and some other MOEAs for solving general MaOPs. However, it is worth further studying the improvement of the population's distribution; that is, the algorithm does not take sufficient measures to allocate the fitness of the solution. Meanwhile, crowded distance is ineffective in solving MaOPs, and the SDR is relatively dependent on the initial population. As we know, when dealing with large-scale MaOPs, the performance of NSGA-II/SDR declines significantly due to its numerous objectives, decision variables, and computational challenges. To enhance the ability of NSGA-II/SDR to handle large-scale MaOPs, this paper presents a relevant research investigation Based on this, the main research work of this paper is as follows:

1. The opposition-based learning (OBL) strategy is introduced in the population initialization stage, an opposite population will be generated according to the initial population, and the best individuals will be selected from the two populations to obtain the final initial population. In this way, the effect of optimizing population quality and convergence speed can be obtained;
2. A local search (LS) strategy is introduced in the population search process, which expands the diversity of the population by finding neighborhood solutions, which can prevent the solution from falling into the local optimum prematurely, thus ensuring a good distribution of the solution. This produces a new MaOEA, which we named NSGA-II/SDR-OLS;

3. NSGA-II/SDR-OLS and the original NSGA-II/SDR [23] are compared on nine benchmark problems in LSMOPs [24] to evaluate whether NSGA-II/SDR-OLS can effectively solve the problem of rapid performance degradation of the original algorithm in the face of large-scale MaOPs.The algorithm is then compared with PREA [25], S3-CMA-ES [26], DEA-GNG [27], RVEA [28], NSGA-II-conflict [29], and NSGA-III [18,19], and we observe its performance. The experimental results demonstrate that NSGA-II/SDR-OLS outperformed other state-of-the-art algorithms.

The remainder of this paper is as follows. In Section 2 we introduce the related work on MOEAs , MaOEAs, and Large-scale MaOPs, followed by the some preliminaries about our work in Section 3. Thereafter, the proposed algorithm NSGA-II/SDR-OLS is described in detail in Section 4. The experimental setup, test problems, and the final experimental results are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Due to its simplicity and efficiency, EAs are widely used in various types of MOPs and have been greatly developed in recent decades. With the increase of problem complexity, MOPs have gradually failed to meet the physical demand, and some algorithms started to focus on solving MaOPs. Following this, a growing number of researchers continuously improved evolutionary algorithms according to specific problems, such as dynamic multi-objective optimization problems, large-scale optimization problems, etc., to improve the performance of algorithms through various optimization strategies, and EAs with their own characteristics gradually emerged. Several classic MOEAs are introduced below.

2.1. Multi/Many-Objective Evolutionary Algorithms

2.1.1. Pareto-Dominance-Based Multi/Many-Objective Evolutionary Algorithms

In 1989, Goldberg [30] was the first to combine the Pareto domination with EA to solve MOPs. Since then, many classical MOEAs have been influenced. In 1994, Deb et al. [16] combined the non-dominated ordering method with a genetic algorithm, and proposed NSGA. Because of superiority of NSGA in dealing with MOPs, it had attracted attention, but soon, researchers also found the drawbacks of NSGA and started to focus on remedying them. In 2002, Deb et al. [17] proposed NSGA-II, which is improved by fast non-dominated ranking methods and crowded distance methods. NSGA-II was competitive in solving MOPs and became a representative algorithm for Pareto-dominance-based MOEAs. Then, many MOEA based on the original framework of NSGA-II were proposed.

The Pareto-dominance-based MOEA [31–33] is a traditional and effective algorithm to solve MOPs. It simultaneously optimizes some conflicting objectives and tries to find a set of Pareto optimal solutions according to the Pareto dominance relation. However, it suffers from a series of problems when solving MaOPs, resulting in poor performance. The main reason is that the number of objectives increases and the non-dominated space increases exponentially, which makes it difficult to distinguish between the performance of solutions only by the Pareto dominance relation. At the same time, the running time of non-dominated sorting also increases, which reduces the running efficiency. Ishibuchi et al. [34] showed that when the objective number of the optimization problem $M > 12$, all solutions in the solution set obtained by only non-dominated sorting will become non-dominated, which makes it difficult to achieve efficient convergence of the population. In addition, due to diversity-driven data, it is difficult for individuals in such algorithms to approximate the real PFs.

In view of the above problems, many solutions have been proposed in recent years. The first is a method to modify the traditional definition of Pareto dominance to adapt to high-dimensional space, so as to better decompose the solutions. For example, ε-dominance [35] and θ-dominance [31] were all improved by modifying Pareto dominance. What is more, aiming at the problem of non-dominated sorting, researchers proposed some new sorting methods, which can improve their efficiency, such as climbing sorting and deductive sorting [36]. In addition, aiming at the problem of diversity, diversity maintenance mechanisms are proposed [37].

For the poor performance of the classical NSGA-II algorithm on MaOPs, many improvement strategies have been proposed successively by researchers. Elarbi et al. [38] advanced reference point-based dominance (RP-dominance) and introduced it into NSGA-II to obtain the new RPD-NSGA-II algorithm. Pan et al. [39] suggested a rotation-based simulated binary crossover and an adaptive operator selection strategy embedded in NSGA-II. Tian et al. [23] proposed a new dominance relation called SDR. Replacing the traditional Pareto dominance relation with this new dominance relation, a new algorithm, NSGA-II/SDR, was created. The experimental results all showed that the improved strategy can bring considerable improvements to NSGA-II and some other MOEAs for solving MaOPs.

2.1.2. Preferences-Based Multi/Many-Objective Evolutionary Algorithms

As the name suggests, decision makers (DMs) take preference information as a key factor in selecting the objectives. According to the required preference information, some specific objectives are selected first, in order to achieve the goal of objective space reduction. The types of preferences are diverse, such as the reference point and reference direction, which can also be regarded as preferences. According to the timing of preference information selection, preference-based MOEAs [40,41] can be subdivided into three categories: a priori algorithms (select first, search later), interactive algorithms (search while selecting), and posteriori style algorithm (search first, select later).

The *r*-dominance-based NSGA-II (*r*-NSGA-II) was a preference-based MOEA proposed by Lamjed et al. [42]. The *r*-NSGA-II introduced the *r*-dominance relation that guides the next search of the population according to the DM's preferences, and directed the solution toward the Pareto optimal region. Experimentally, the algorithm proved to be very competitive. The MOEA/D using adaptive weight vector-guided (MOEA/D-AWV) was proposed by Wang et al. [43], and adaptively adjusted to the DM's preferences. It was demonstrated that the distribution of weight vectors can adapt well to the change of DM's preference and solved the MOPs in high-dimensional objective space.

In order to improve the ability of this class of algorithms to solve MaOPs, some preference-based MaOEAs have emerged, one after another. He and Yen [44] comprehensively analyzed the current selection strategies in MaOEAs, and then proposed a new coordinated selection strategy to improve the performance of evolutionary algorithms in many-objective optimization. The proposed MaOEA-CSS has good performance in ensuring the balance of convergence and diversity. Gong et al. [45] proposed a set-based MOEA guided by preference regions, which is called P-SEA. Preference was introduced into the set-based many-objective evolution, and the representation and utilization of preference were studied. The main idea is to dynamically determine the preferred region, and then develop the crossover operator according to the determined preferred region, and finally quickly generate the Pareto optimal set with excellent performance, according to the preferred region. Hou et al. [46] suggested that preference should be reformulated into constraints. The proposed method can stably control the degree of ROI on the problem with relatively complex PF. By comparing the proposed CP-NSGA-II with four latest preference-based MOEAs, it is proven that CP-NSAGA-II is competitive in handling MOPs and MaOPs.

2.1.3. Decomposition-Based Multi/Many-Objective Evolutionary Algorithms

Classic MOEA/D was introduced by Zhang et al. [20] in 2007, which became a representative algorithm for decomposition-based algorithms. The decomposition-based method introduces the decomposition idea, which is commonly used in mathematics, into the field of multi-objective optimization, decomposing a MOP or MaOP into multiple scalar sub-problems according to a specific method, and then optimizing them simultaneously by the optimization algorithm. The common decomposition methods include the weighted Tchebycheff approach (TCH), the weighted sum approach (WS), and the penalty-based boundary intersection approach (PBI). Although MOEA/D is very competitive for solving general MOPs, it does not perform very well for solving special MOPs and MaOPs.

As research continues, many new decomposition-based methods have emerged in the field of multi-objective optimization. Many improved versions of MOEA/D have also been used to solve special MOPs and MaOPs. The multi-objective evolutionary algorithm using decomposition and ant colony algorithm (MOEA/D-ACO) was a MOEA proposed by Ke et al. [47] in 2013, by combining ant colony optimization (ACO) [48] and MOEA/D. In this algorithm, each ant was responsible for solving a subproblem and recording the optimal solution it found for the subproblem, and then built an ant colony pheromone matrix by constructing a neighborhood matrix to select the optimal solution of itself and the colony as the better solution for updating. Jiao et al. [49] proposed a decomposition-based MaOEA, called MOEA/D-2WA, with two weight vector adjustments to solve highly constrained many-objective optimization problems (CMaOPs). It designs infeasible weights for infeasible solutions, and generates feasible weights for guiding feasible solutions. Compared with six advanced CMaOEAs, MOEA/D-2WA could better deal with highly CMaOPs. A new, miniature, multi-strategy, multi-objective artificial bee colony algorithm was raised by Peng et al. [50]. It divided the population into multiple subpopulations and generated offspring in parallel to balance exploration and exploitation.

The MOEA based on hierarchical decomposition (MOEA/HD) [51] was an improved algorithm based on MOEA/D, which was proposed by Xu et al. in 2019. To solve problems with inhomogeneous PFs, MOEA/HD divided several subproblems into different levels, and the search space of the lower-level subproblems was adaptively adjusted according to the search results of the higher-level subproblems. It was demonstrated that MOEA/HD effectively solves the problem regarding the poor performance of MOEA/D. Zhang et al. [52] introduced the information feedback models into the classic MOEA/D algorithm and proposed a MOEA/D algorithm based on the information feedback model, which is called MOEA/D-IFM. According to different IFMs, they proposed six new algorithms and classified them.

2.1.4. Indicator-Based Multi/Many-Objective Evolutionary Algorithms

In order to enhance the selection difficulty of algorithms, an indicator-based algorithm was proposed to deal with MaOPs. The indicator-based method, as the name suggests, is to take an indicator as the standard to choose better individuals. It does not rely on the Pareto dominance relation to achieve convergence of the solution set, but, rather, it guides the solution set toward the direction with better indicator values by using a specific indicator, and evaluates the optimal solution set based on the specific indicator. Jiang et al. [53] gave a detailed overview of the main indicators proposed so far. The indicator-based evolutionary algorithm (IBEA) [54] was the first to introduce an indicator into MOEA to solve MOP, proposed by Zizler et al. in 2004. The algorithm used a binary performance metric (I_{ε^+}) to calculate the minimum distance required for a solution in the optimal solution set to the Pareto front edge, where the smaller the value of the I_{ε^+} metric, the better the convergence of the solution set. The hypervolume estimation algorithm (HypE) [55] was also a classic indicator-based algorithm. HypE aimed to use Monte Carlo simulations to approximate the exact HV value, rank the optimal solutions based on the HV value, effectively balance the accuracy of the estimation and the available computational resources, and flexibly adjust the running time.

To deal with MaOPs, Liu et al. [56] introduced a MaOEA, which used a one-by-one selection strategy. In the proposed 1by1EA, the solution set with good convergence and distribution performance could be obtained by selecting according to convergence and distribution indicators. Cai et al. [57] proposed a unary diversity indicator based on the reference vector (DIR), to estimate the diversity of PF approximation for many-objective optimization. DIR was integrated into NSGA-II. Sun et al. [58] suggested an IGD indicator-based evolutionary algorithm. Each generation used the IGD indicator to select solutions with good convergence and diversity. In order to find a good balance between convergence and diversity, Liang et al. [59] introduced a two-round environment selection strategy without reference vectors and based on multiple indicators, and obtained an algorithm called 2REA. The first round of selection used the newly proposed adaptive position transformation (APT) strategy to maintain diversity, while the second round of selection aimed to enhance convergence.

2.2. Large-Scale Many-Objective Optimization Problems

In the field of multi- and many-objective optimization, current research has focused on dealing with small-scale MOPs or MaOPs. However, as the complexity of problems increases, the number of decision variables [60] in real-world problems also grows, and small-scale optimization algorithms can no longer meet the needs of solving problems, so studies on large-scale optimization problems have gradually begun. Large-scale MOPs usually refer to those complex problems with multiple objectives, and there are so many decision variables in each objective that it is difficult to achieve optimization; therefore, these problems are widely used in real engineering applications [24]. Generally speaking, the mathematical expression of a large-scale MOP can be shown below.

$$\begin{array}{l} \min G(x) = f(x_1, x_2, \ldots, x_D) \\ x_i \in [x_{\min}, x_{\max}], i = 1, 2, \ldots, D \end{array} \tag{1}$$

where $G(x)$ is the objective function of a large-scale MOP, D is the decision variable, and x_{min} and x_{max} refer to the constrained upper bound and constrained lower bound of decision variables, respectively. In general, the number of decision variables in this type of problem is more than 1000.

These problems are difficult to solve for three main reasons: (1) the computational effort of population evolution increases exponentially with the number of decision variables in the problem, (2) the number of objectives is not less than two, which makes it difficult to build mathematical models accurately, and (3) the selection environment of the problem changes continuously, which brings some uncertainty to the solution [61]. Thus, it can be seen that large-scale MOPs are usually nonlinear, non-differentiable, and characterized by the presence of at least 1000 interconnected decision variables.

Ma et al. [62] suggested the adaptive localized decision variable analysis approach evolutionary algorithm (ALDVAEA) based on the decomposition framework. The algorithm incorporated guidance on the reference vector into the analysis of the decision variables and used projection-based detection methods in the analysis of the decision variables. Wang et al. [63] proposed a large-scale optimization algorithm, called particle swarm optimization, based on reinforcement learning levels (RLLPSO). In RLLPSO, a level based population structure was constructed to improve population diversity. Aiming at the problem that NSGA-III was not effective in solving large-scale optimization problems, Gu and Wang [64] embedded six information feedback models into NSGA-III and generated six improved NSGA-III algorithms, which are collectively referred to as IFM-NSGA-III. These methods greatly improved the performance of the algorithm for large-scale optimization problems.

The above work was done to achieve the processing of large-scale problems, by analyzing decision variables and then grouping them into partitions, but the analysis process of decision variables is computationally intensive and the complexity of the algorithm solution is high. Therefore, researchers need to introduce more analytical ideas to large-scale MOPs or MaOPs in the future. In this paper, opposition-based learning (OBL) is first introduced into the process of initializing the population, and the opposite solution is obtained through the OBL method of the initial population, and then is introduced into the updating process as the final initial population, so as to accelerate the convergence of the population. Then, local search (LS) is introduced in the process of population updating. This strategy can make the solution jump out of the local optimum and continue to find the global optimal solution in the search space of the objective, which can balance the convergence and diversity of the population well.

3. Preliminaries

3.1. Basic Definitions

To deal with many objectives at the same time, it is impossible to achieve the optimal solution to meet all the objectives. Therefore, it is necessary to choose a best trade-off solution, which is called the Pareto optimal solution. Some related concepts are provided as follows.

Definition 1 (Pareto Dominance)**.** *A vector* $\boldsymbol{u} = (u_1, \ldots, u_m)^T$ *is said to dominate another vector* $\boldsymbol{v} = (v_1, \ldots, v_m)^T$*, denoted as* $u \prec v$*, if* $\forall i \in \{1, \ldots, m\}$*,* $u_i \leq v_i$ *and* $u \neq v$*.*

Definition 2 (Pareto Optimal Solution)**.** *A feasible solution* $x^* \in \Omega$ *of equation (1) is called a Pareto optimal solution, if* $\nexists y \in \Omega$ *such that* $F(y) \prec F(x^*)$*.*

Definition 3 (Pareto Set)**.** *The set of all the Pareto optimal solutions is called the Pareto set (PS), denoted as*

$$PS = \{x \in \Omega \mid \nexists y \in \Omega, F(y) \prec F(x)\} \tag{2}$$

Definition 4 (Pareto Front)**.** *The image of the PS in the objective space is called the Pareto front (PF), denoted as*

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

Definition 5 (Ideal Point)**.** *In the objective space of the minimized MOP, the ideal point* $z^I = (z_1^I, \ldots, z_M^I)$ *consists of the vector with the minimum objective function value in the solution search space* Ω*, which is mathematically represented as follows.*

$$z^I = (\min f_1(x), \ldots, \min f_M(x)), x \in \Omega \tag{4}$$

Definition 6 (Nadir Point)**.** *In the objective space of the minimized MOP, the nadir point* $z^N = (z_1^N, \ldots, z_M^N)$ *is the solution with the maximum value in the Pareto optimal solution set on each objective, which is mathematically represented as follows.*

$$z^N = (\max f_1(x), \ldots, \max f_M(x)), x \in PS \tag{5}$$

3.2. NSGA-II/SDR

MOEAs have been well proven to be efficient in solving problems with two or three objectives. However, recent studies showed that most of the individuals in MOEA are non-dominated and most of them are in a random, wandering state in the search space, so this type of algorithm faces some difficulties in dealing with many-objective problems [34].

To better balance the convergence and diversity of many-objective optimization, Tian et al. [23] proposed a new dominance relation, termed the strengthened dominance relation (SDR). In the proposed dominance relation, an adaptive niching technique was developed, based on the angles between the candidate solutions, where only the candidate solution with the best convergence in each niche was non-dominated. Experimental results showed that the proposed dominance relation was superior to the existing dominance relation in terms of balance convergence and diversity. Based on the proposed dominance relation, an improved NSGA-II algorithm (NSGA-II/SDR) was proposed, which was competitive with existing algorithms in solving MaOPs. The following will be a brief introduction to the related content of NSGA-II/SDR in two subsections.

3.2.1. SDR

The existing dominance relations can enhance the selection difficulty of MOEAs in solving MaOPs, but most dominance relations can only find a set of solutions that concentrate on a small region of the PFs. This is equivalent to modifying the existing dominance relations to be stricter than the original Pareto dominance relation, and some non-dominated solutions on the PF can be identified as dominated solutions, thus sacrificing the distribution of candidate solutions. In contrast, SDR can solve this problem. SDR re-modifies the dominance relationship, which is defined below. Specifically, a candidate solution x is said to dominate another candidate solution y in SDR (denoted as $x \prec_{SDR} y$), if and only if

$$\begin{cases} \mathrm{Con}(x) < \mathrm{Con}(y), \theta_{xy} \leq \bar{\theta} \\ \mathrm{Con}(x) \cdot \frac{\theta_{xy}}{\bar{\theta}} < \mathrm{Con}(y), \theta_{xy} > \bar{\theta} \end{cases} \tag{6}$$

where

$$\mathrm{Con}(x) = \sum_{i=1}^{M} f_1(x) \tag{7}$$

is a metric for the convergence degree of x, and is widely used in many MOEAs [56,60,65], while θ_{xy} denotes the acute angle between the objective values of the two candidate solutions, namely

$$\theta_{xy} = \arccos(f(x), f(y)), \tag{8}$$

and θ is the size of the niche to which each candidate solution belongs. The dominance relationship associated with a candidate solution x is determined mainly by considering the candidate solution in its niche.

The analysis of the SDR can be divided into two parts, corresponding to the two formulas in Equation (6).

(1) According to the first formula in Equation (6), if the angle between any x and a candidate solution y is less than $\bar{\theta}$, then x is called the dominated solution when the convergence of x is less than the convergence of y. This allows the diversity of non-dominated solution sets to be preserved.

(2) According to the second formula in Equation (6), provided that two candidate solutions x and y do not lie within the same niche (i.e., $\theta_{xy} > \bar{\theta}$), x can still control y if y converges much worse than x, where the probability of x controlling y is negatively related to the angle θ_{xy}. This ensures the convergence of the non-optimal solution set.

For further understanding, Figure 1 shows the dominance regions obtained by SDR in the dual objective space. We can see that since y_1 is located in the niche of x, and the convergence is worse than that of x, x dominates y_1. On the other hand, because y_2 is outside the niche of x and converges much less than x, x still dominates y_2. Therefore, the dominance region of x consists of two parts.

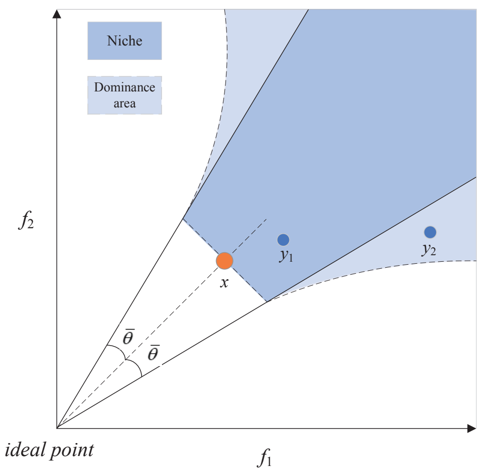

Figure 1. Dominance area of solution x obtained by SDR in bi-objective space.

As can be seen from the above description, the parameter θ is important. In SDR, θ can be estimated adaptively according to the distribution of the candidate solution set. As for NSGA-II, environmental selection always selects half of the combined population obtained at each generation, and θ is generally allowed to ensure that the ratio of non-dominated solutions in a given set of candidate solutions is around 0.5. So, θ is set to the $\lfloor |P|/2 \rfloor$-th minimum element given by

$$\left\{ \min_{q \in P \setminus \{p\}} \theta_{pq} \mid p \in P \right\} \tag{9}$$

where θ_{xy} denotes the acute angle between any pair of candidate solutions p and q.

3.2.2. Procedure of NSGA-II/SDR

Tian et al. [23] proposed SDR and embedded it in NSGA-II to obtain NSGA-II/SDR. For the specific process of NSGA-II/SDR (see Algorithm 1).

Algorithm 1: NSGA-II/SDR

```
Input: Population size N.
Output: Final population P.
P ← RandomInitialize (N);
Normalize the objective values in P;
[F_1, F_2, ..., F_n] ← Do non-dominated sorting on P by SDR;
CrowdDis ← CrowdingDistance(F);
while gen<maxgen do
    P′ ← Select N parents via binary tournament selection according to the non-dominated front and
      crowding distance of each solution in P;
    P ← P ∪ Variation(P′);
    Normalize the objective values in P;
    [F_1, F_2, ..., F_n] ← Do non-dominated sorting on P by SDR;
    CrowdDis ← CrowdingDistance(F);
    k ← Minimum value s.t. |F_1 ∪ ... ∪ F_k| ≥ N;
    if |F_1 ∪ ... ∪ F_k| > N then
        Delete |F_1 ∪ ... ∪ F_k| − N solutions from F_k with the worst crowding distance values;
    end
    P′ ← F_1 ∪ ... ∪ F_k;
    gen = gen + 1;
end
return P;
```

NSGA-II/SDR is competitive with the most improved MOEA in solving MaOPs. However, although NSGA-II/SDR performs reasonably well on MaOPs, it is well known that crowding distance is ineffective in solving MaOPs [37]. The convergence ability and convergence speed of the algorithm inherit the properties of NSGA-II, and there is room for improvement. At the same time, the balance between convergence and diversity is also a widely concern issue. Therefore, it is necessary to further improve the performance of NSGA-II/SDR on MaOPs by adding new effective policies.

4. Improved NSGA-II/SDR with Opposition-Based Learning and Local Search

In this section, the two strategies we added are first introduced in detail, namely opposition-based learning (OBL) and local search (LS), and then our proposed NSGA-II/SDR-OLS algorithm is explained.

4.1. Opposition-Based Learning

OBL was proposed by Tizhoosh [66] in 2005 and extended to genetic algorithms, reinforcement learning, and neural networks. From then on, OBL has been successfully applied in population intelligence optimization algorithms. OBL only has obvious advantages in the early stage, because, as learning continues, these advantages will turn into disadvantages. Therefore, using reverse learning at the beginning can save time and make the estimate as close as possible to the existing solution.

In the field of population-based evolutionary algorithms, population initialization often employs a purely random strategy, where the upper and lower bounds are known and a random value is taken between the upper and lower bounds during initialization. This random value allows for fast convergence if it is not far from the optimal solution. However, naturally, if this random value is very far from the existing solution, in which it is at its worst at the opposite position, then the next process will take considerable time, or, at worst, the global optimal solution cannot be explored. Without any prior knowledge, it is not realistic to make a best initial guess, so we consider looking in all directions simultaneously or, more specifically, in the opposite direction. This is what OBL does. The population initialization strategy based on OBL pronounces the death sentence on traditional, purely random strategies, in terms of convergence speed.

To obtain the global optimal solution, the OBL strategy generates solutions in the opposite direction with a random initial population, updating the quality of the optimized solution. This can, to some extent, break through the strong randomness caused by the initialization of the population, and thus speed up the convergence of the population. This is because the OBL strategy is more promising to find solutions that are closer to the PF in the initialization phase, and it is easier to find good-quality solutions in the subsequent population update process, and thus to explore the global optimal solution. The OBL is defined as follows.

Let $x \in R$ be a real number defined on a certain interval: $x \in [a, b]$. The opposite number $\tilde{x}$ is defined as follows:

$$\tilde{x} = a + b - x. \tag{10}$$

For $a = 0$ and $b = 1$, we get

$$\tilde{x} = 1 - x. \tag{11}$$

Analogously, the opposite number in a multidimensional case can be defined. Let $P(x_1, x_2, \ldots, x_n)$ be a point in a n-dimensional coordinate system with $x_1, x_2, \ldots, x_n \in R$ and $x_i \in [a_i, b_i]$. The opposite point $\tilde{P}$ is completely defined by its coordinates $\tilde{x_1}, \ldots, \tilde{x_n}$ where

$$\tilde{x}_i = a_i + b_i - x_i, i = 1, \ldots, n. \tag{12}$$

The main idea of the OBL strategy is to evaluate the fitness value of the current solution and its inverse solution by the fitness function, and then continuously adjust the convergence direction of the solution according to the fitness value, so as to choose a better individual to explore the solution space.

As shown in Figure 2, $k = 1$ represents the first application of the OBL strategy, and x_0 is the reverse solution generated by the initial solution x after the OBL strategy. Then, since x is closer to the expected solution, the OBL strategy is continued to be applied to x. At the same time, the search interval can be halved, that is, when $k = 2$, the interval is reduced from $[a_1, b_1]$ to $[a_1, b_2]$. By analogy, a new x_0 solution is generated by continuously evaluating the distance between the two solutions, x_0 and x, to the desired solution, until the estimated value is close enough to the nearest expected solution. Algorithm 2 gives the detailed steps of the OBL strategy.

Algorithm 2: Opposition-based learning

```
Input: Population P_t, fitness of population Fitness, lower bound X, upper bound Y.
Output: New population P_new.
for i = 1 to Length(P_t) do
    Calculate opposite population OP_t by equation (12);
    Calculate OP_t_Fitness;
    Evaluate OP_t_Fitness;
    Select Length(P_t) fittest individuals from {P_t, OP_t} as P_new;
end
```

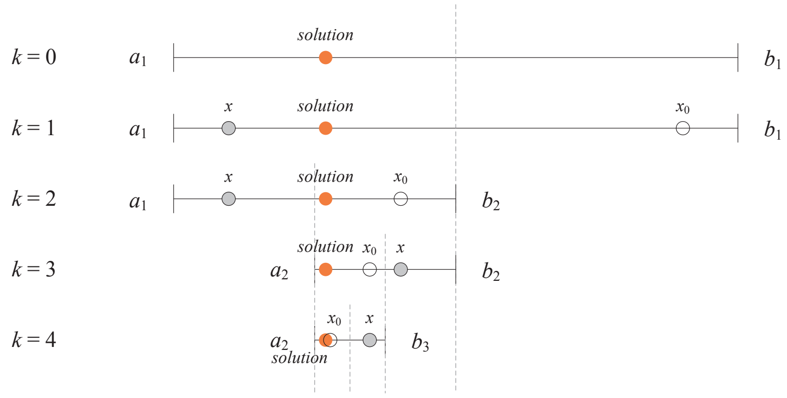

Figure 2. The process of generating the opposite solution.

4.2. Local Search

LS is a heuristic method for solving optimization problems. For some computationally complex optimization problems, such as various NP-complete problems, the time required to find the optimal solution grows exponentially with the size of the problem, so various heuristic methods are born to retreat to the next best solution, which comprise approximate algorithms, with the idea of sacrificing accuracy for time efficiency. LS is one of these methods. This method can effectively avoid the problem of premature convergence of the algorithm and enable the solution to go beyond the local optimum to obtain a better solution. It enables the solution to obtain a larger search space and thus extends the diversity of solutions [67–69].

LS selects a best neighbor from the neighborhood solution space of the current solution as the current solution for the next iteration each time, until a local optimal solution is reached. Since a solution $\boldsymbol{x}(x_1, x_2, \ldots, x_n)$ has an infinite number of neighbors in the search space, the key step of the local search strategy is to find a suitable neighboring solution. LS starts from an initial solution and then searches the neighborhood of the solution; if there is a better solution, then it moves to that solution and continues to execute the search, otherwise, we stop the algorithm to obtain the local optimal solution. The following is the local search model.

Given a population P_t with size of N solutions and a solution $\boldsymbol{x_{it}}(x_{1,i,t}, x_{2,i,t}, \ldots, x_{n,i,t})$ in P_t, where n denotes the number of variables, i denotes the i-th solution of the population and t denotes the generation to which the population belongs, define $S_{1,i,t}$ as the set of neighborhoods on the k-th variable of solution $x_{i,t}$, namely

$$S_{k,i,t} = \left\{\omega^{+}_{k,i,t}, \omega^{-}_{k,i,t}\right\}, \tag{13}$$

where $\omega^{+}_{k,i,t}$ and $\omega^{-}_{k,i,t}$ are denoted as the two neighborhoods of the solution $x_{i,t}$.

$$\omega^{+}_{k,i,t} = x_{k,i,t} + c \times (u_{k,i,t} - v_{k,i,t}) \tag{14}$$

$$\omega^{-}_{k,i,t} = x_{k,i,t} - c \times (u_{k,i,t} - v_{k,i,t}). \tag{15}$$

where $\boldsymbol{u_{i,t}}(u_{1,i,t}, \ldots, u_{k,i,t}, \ldots, u_{n,i,t})$, $k \in \{1, \ldots, n\}$, and $\boldsymbol{v_{i,t}}(v_{1,i,t}, \ldots, v_{k,i,t}, \ldots, v_{n,i,t})$, $k \in \{1, \ldots, n\}$ are two solutions randomly chosen from the population P_t, c is a perturbation factor following a Gaussian distribution $N(\mu, \sigma^2)$, and μ and σ are the mean value and the standard deviation of the Gaussian distribution, respectively. The Gaussian distribution used in the LS strategy is mainly represented by the fact that c varies with $u_{k,i,t} - v_{k,i,t}$ in the LS strategy and, in addition, the standard deviation in the LS strategy σ is a constant. Algorithm 3 gives the detailed steps of the LS strategy.

Algorithm 3: Local search

Input: Initial Population P_t, population size N, number of decision variables n.
Output: Neighborhood solutions $\omega^+_{k,i,t}, \omega^-_{k,i,t}$.

for *i = 1 to Length(P_t)* **do**
  for *k = 1 to n* **do**
    Calculate $c = N(\mu, \sigma^2)$;
    Randomly choose two solutions $u_{i,t}$ and $v_{i,t}$ from initial population P_t;
    Generate neighborhood solutions $\omega^+_{k,i,t}, \omega^-_{k,i,t}$ by equations (14) and (15), respectively;
    Replace $x_{i,t}$ with $S_{k,i,t}$;
  end
end

4.3. NSGA-II/SDR-OLS

Here, how the OBL and LS strategies are combined with NSGA-II/SDR will be described in detail and the workflow of the NSGA-II/SDR-OLS will be explained. The main process can be represented as follows.

Step 1: *Initialization.* The generated population P is randomly initialized. The OBL is applied to P to generate the initial population P_0.
Step 2: *Update.*

Step 2.1: Perform non-dominated sorting by SDR on initial population P_0.
Step 2.2: Perform LS on population P_0 to obtain population S, and merge P_0 and S to obtain population R.
Step 2.3: Perform the basic operation of GA on R to obtain R', which is merged with the parent population R to update R. The basic operation of GA is not introduced in detail here.
Step 2.4: Perform fast non-dominated sorting by SDR on population R, and perform the basic operation of GA on R to obtain R', which is merged with the parent R to update R.
Step 2.5: Determine if the algorithm has reached the maximum number of iterations or function evaluation value to control the computational workload and accuracy. If the termination condition is not fulfilled, repeat Steps 2.2–2.5, and if it is satisfied, perform Step 3.

Step 3: *Output.* Output final population R.

For a more intuitive understanding of the NSGA-II/SDR-OLS, the procedure of the algorithm can be found in Figure 3 and Algorithm 4.

Algorithm 4: NSGA-II/SDR-OLS

Input: Population size N, mean value μ, and standard deviation of the Gaussian distribution σ.
Output: Final population R.

$P \leftarrow$ *Randominitialize*(N);
$P_0 \leftarrow$ *OBL*(P) by Algorithm 2;
$[F_1, F_2, \ldots, F_n] \leftarrow$ Do non-dominated sorting on P_0 by SDR;
while *gen<maxgen* **do**
  $S \leftarrow$ *LocalSearch*(P_0) by Algorithm 3;
  $R \leftarrow P_0 \cup S$;
  $R' \leftarrow$ Do the basic operation of genetic algorithm on R;
  $[F_1, F_2, \ldots, F_m] \leftarrow$ Do fast non-dominated sorting on P_0 by SDR;
  $R' \leftarrow$ Do the basic operation of genetic algorithm on R;
  $R \leftarrow R \cup R'$;
  $gen = gen + 1$;
end
return R;

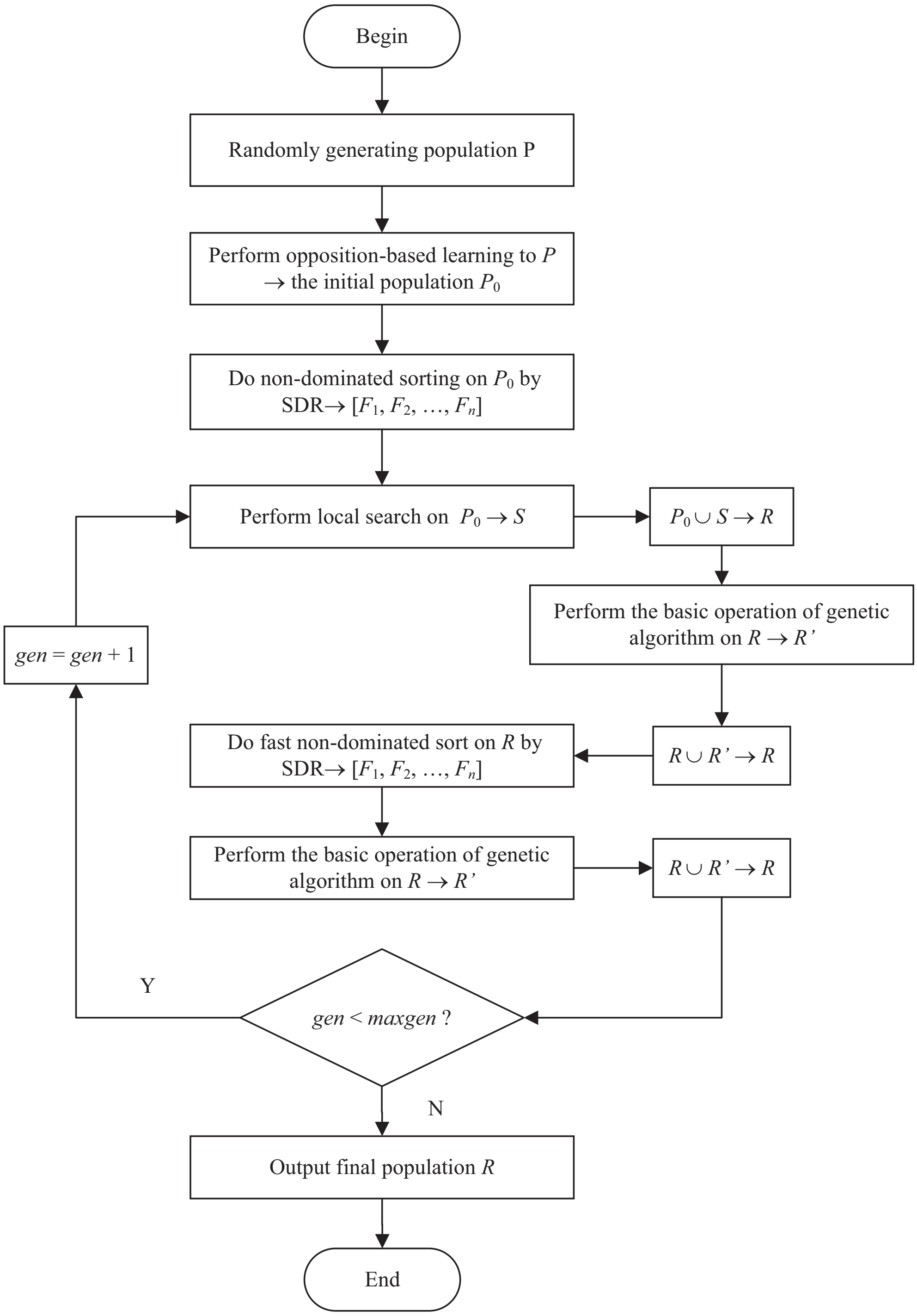

Figure 3. The main process of NSGA-II/SDR-OLS.

5. Experiments

In Section 4.3, this paper combines the opposition-based learning strategy and local search strategy with the NSGA-II/SDR and proposes the NSGA-II/SDR based on opposition-based learning and local search. In order to verify the performance of NSGA-II/SDR-OLS on large-scale MaOPs, the NSGA-II/SDR-OLS was compared with seven existing MaOEAs on the LSMOP test function set. The performance of NSGA-II/SDR-OLS was evaluated by three indicators, which are inverted generational distance (IGD), generational distance (GD), and metric for diversity (DM).

5.1. Test Problems and Performance Metrics

5.1.1. Test Problems

In earlier research in the field of multi-objective optimization, researchers started to study a series of test problems in order to evaluate the performance of various MOEAs. Deb [70] proposed the general principle for testing problems in 1999. The principle was constructed by three basic functions, namely, the distribution function f_1, the distance function g and the shape function h. Among them, the distribution function f_1 can test the diversity ability of the algorithm, the distance function g can evaluate the convergence ability of the algorithm, and the shape function can define the PFs.

Cheng et al. [24] proposed a set of large-scale MOPs or MaOPs, called LSMOP, in 2017. In the design of LSMOP test function set, only three parameters need to be set, which are the number of objectives M, the number of decision variables D, and the number of subcomponents in each variable group n_k. Generally speaking, the number of decision variables D takes the value of $M \times 100$, and the number of subcomponents $n_k = 5$. In order to measure the performance of the algorithm on different problems, LSMOP1-9 is designed by combining the link function ($L(x^s)$), the correlation matrix (C), and the shape matrix ($H\left(x^f\right)$), which correspond to the separability of the test functions, the correlation of the variables, and the shape of the Pareto front, respectively. As for the PFs, LSMOP1-4 has linear PFs, LSMOP5-8 has convex PFs, and LSMOP9 has disconnected PFs. According to the above description, the properties and characteristics of each test problem are shown in Table 1. Among them, $L_1(x^s)$ is a linear variable connection and $L_2(x^s)$ is a nonlinear variable connection.

Table 1. The properties and characteristics of LSMOP1-9.

Problems	PF	PS	Modality	Separability
LSMOP1	linear	$L_1(x^s)$	unimodal	fully separable
LSMOP2	linear	$L_1(x^s)$	mixed	partially separable
LSMOP3	linear	$L_1(x^s)$	multimodal	mixed
LSMOP4	linear	$L_1(x^s)$	mixed	mixed
LSMOP5	convex	$L_2(x^s)$	unimodal	fully separable
LSMOP6	convex	$L_2(x^s)$	mixed	partially separable
LSMOP7	convex	$L_2(x^s)$	multimodal	mixed
LSMOP8	convex	$L_2(x^s)$	mixed	mixed
LSMOP9	disconnected	$L_2(x^s)$	mixed	fully separable

5.1.2. Performance Metrics

The metrics used to measure the performance of algorithms in multi-objective optimization are usually classified into four types [43]: capacity, convergence, diversity, and convergence-diversity. Capacity mainly measures the ability of the algorithm to obtain non-dominated solutions in terms of the number or proportion of non-dominated solutions in the optimal solution set that satisfy the predefined conditions. The convergence assesses how well the optimal solution set obtained by the algorithm fits the true PF. The diversity measures the distribution and spread of solutions in the optimal solution set. The convergence-diversity metrics measure both the convergence and diversity of solutions.

Considering the convergence and diversity of the algorithms, we choose the inter-generational distance (GD) [71], the inverted generational distance (IGD) [72,73], and the metric for diversity (DM) [74] as the performance metrics for the experiments in this paper.

GD is a classical convergence metric. This metric calculates the squared sum of the Euclidean distances from the optimal solution set S to the nearest reference point on the true PF. IGD is a comprehensive metric proposed by Coello et al. in 2005. This metric calculates the average distance from each reference point on the Pareto approximate frontier to the closest solution in the optimal solution set S. The mathematical representation of the two is as follows:

$$\mathrm{GD}(S,P) = \frac{\left(\sum_{i=1}^{|S|} d_i^q\right)^{\frac{1}{q}}}{|S|} \quad (16)$$

$$\mathrm{IGD}(P,S) = \frac{\left(\sum_{i=1}^{|P|} d_i^q\right)^{\frac{1}{q}}}{|P|} \quad (17)$$

where $q = 2$ and $d_i = \min_{\vec{s}\in S}\|F(\vec{p}_i) - F(\vec{s})\|, \vec{p}_i \in P$ computes the shortest Euclidean distance from the i-th solution s_i in the optimal solution set S to the nearest P point on the Pareto approximation front. However, the calculation of d_i in IGD is just the opposite, which is the shortest Euclidean distance from a point in the set of reference points P to a point in the optimal solution set S. The smaller the value of GD, the better the convergence of the optimal solution set S. The smaller the value of IGD, the better the convergence or diversity of solution set S.

DM was proposed by Kalyanmoy and Sachin in 2002, and the basic idea is that the non-dominated points obtained at each generation are projected on a suitable hyperplane, thus losing points of one dimension. The plane is divided into many small grids, and the diversity is judged according to whether or not each grid contains an obtained non-dominated point. If all grids are represented by at least one point, then the best diversity is achieved. The mathematical expression is as follows:

$$D\left(P^{(t)}\right) = \frac{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(h(i,j,\ldots))}{\sum_{\substack{i,j,\ldots,\ldots \\ H(i,\ldots,\ldots\neq 0}} m(H(i,j,\ldots))}, \quad (18)$$

where

$$H(i,j,\ldots) = \begin{cases} 1, \text{ if the grid has a representative point in } P^* \\ 0, \text{ otherwise} \end{cases}, \quad (19)$$

$$h(i,j,\ldots) = \begin{cases} 1, \text{ if } H(i,j,\ldots) = 1 \\ \qquad \text{and the grid has a representative point in } F^{(t)}, \\ 0, \text{ otherwise} \end{cases} \quad (20)$$

where $F^{(t)}$ is the non-dominated set to P^* which is determined from $P^{(t)}$.

5.2. Experimental Settings

NSGA-II/SDR-OLS is compared with other MaOEAs in LSMOP1-9. The three parameters to be set in the LSMOP1-9 objective function are as follows: the number of objectives M ranges from 3 to 15 (3, 5, 8, 10, 12, and 15), the dimension of the decision variables $D = M * 100$, and the number of subcomponents of each variable group $n_k = 5$. For the fairness of the results, the same parameters in all algorithms are set to be consistent. Specifically, the control parameters of DEA-GNG are chosen such that aph = 0.1 and eps = 0.314. In RVEA, the parameters α and fr are set to 0.9 and 2, respectively. In the experiment of NSGA-II-conflict, the number of subspaces and cycles were set to 2 and 10, which achieved the best performance of the algorithm, so the same parameter settings are used in this paper. These parameter settings were set empirically in the same manner as in the original studies [17,18,22,24–28].

In order to improve the credibility of the experimental results, in this section, the population size N of LSMOP is set to 100, which allows the algorithm to find the global solution as much as possible, while ensuring operational efficiency, and the maximum fitness function evaluation value FE is set to 10^4. Finally, to ensure the validity of the experimental results, each comparison algorithm independently run 20 times on each problem.

5.3. Comparison

5.3.1. Comparative Algorithms

In order to verify the performance of our proposed algorithm in dealing with large-scale MaOPs, NSGA-II/SDR-OLS and six existing MaOEAs are compared and tested on LSMOP. The performance of NSGA-II/SDR-OLS was comprehensively measured by IGD, GD, and DM indicators. The following is a brief introduction to the comparative algorithm. Among them, our chosen algorithm covers almost all the classification of MaOEAs, involving the latest algorithms and classical algorithms.

The promising region-based multi-objective evolutionary algorithm (PREA) [24] is a MaOEA based on ratio indicator. Scalable small subpopulation-based covariance matrix adaptation evolution strategy (S3-CMA-ES) [25] is used to solve MOPs with large-scale decision variables. The decomposition-based multi-objective evolutionary algorithm guided by growing neural gas (DEA-GNG) [26] is a novel decomposition-based MaOP, which can optimize the performance degradation of decomposition-based MOEAs in solving MOPs with irregular PFs. Reference vector-guided evolutionary algorithm (RVEA) [27] is a reference vector-based MaOEA. NSGA-II with a conflict-based partitioning strategy (NSGA-II-conflict) [28] is a MaOEA based on conflict partition strategy. NSGA-III [17,18] is a MaOEA based on reference points, following the NSGA-II framework.

5.3.2. Comparing NSGA-II/SDR-OLS with Other MaOEAs

In this section, the performance of the improved NSGA-II/SDR done in this paper will be investigated through two-stage experiments. The first stage is to verify whether the addition of the two policies improves the performance of the original NSGA-II/SDR for processing large-scale MaOPs. At this stage, the NSGA-II/SDR-OLS was compared with the original NSGA-II/SDR on nine benchmark functions (LSMOP1-9), with the objective number M ranging from 3 to 15 (3, 5, 8, 10, 12, and 15). Statistical results (mean and standard deviation) of the IGD and DM values were recorded. The reason for recording the mean and standard deviation is that experience from a series of previous studies has shown that the mean and standard deviation can more accurately describe the results. The second stage is to test whether our proposed new algorithm is more competitive than other state of the art algorithms in solving large-scale MaOPs. In Section 5.3.1, six existing algorithms were selected to compare NSGA-II/SDR-OLS with them. Experiments were also carried out on LSMOP1-9 with the number of objectives M ranging from 3 to 15 (3, 5, 8, 10, 12, and 15). Statistical results (mean and standard deviation) of the IGD and GD values were recorded. The above experiments were run under Windows 10 and carried out on MATLAB R2021b. Their performance was compared according to the recorded experimental results. The performance of NSGA-II/SDR-OLS is verified according to the experimental design in Sections 5.1 and 5.2. Detailed experimental results are shown in Tables 2–17. The best results are highlighted.

Tables 2–5, respectively, show the IGD and DM results of NSGA-II/SDR-OLS and the original NSGA-II/SDR on LSMOPs with the objective number M ranging from 3 to 15 (3, 5, 8, 10, 12, and 15). According to Tables 2 and 3, the results of IGD show that the performance of our improved algorithm does improve significantly, achieving better results than the original algorithm in seven of the nine test problems with objective numbers ranging from 3 to 15. According to Tables 4 and 5, the GD values also reflect the better results achieved by our improved algorithm. Our algorithm achieves better results on five out of nine instances of the 3/5/10-objective test problem, and better on six out of nine instances of the 12-objective test problem. The NSGA-II/SDR-OLS is as good as that of the original NSGA-II/SDR on 8/15-objective test problems.

Table 2. IGD values compared with original NSGA-II/SDR with M = 3, 5, and 8.

	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS
LSMOP1	3	3.7090×10^{0} (3.29×10^{-1}) -	8.6033×10^{-1} (3.39×10^{-4})	5	2.8559×10^{0} (4.73×10^{-1}) -	9.4329×10^{-1} (1.68×10^{-3})	8	3.6633×10^{0} (9.42×10^{-1}) -	9.6768×10^{-1} (8.05×10^{-3})
LSMOP2	3	9.3298×10^{-2} (8.96×10^{-3}) +	1.6047×10^{-1} (1.28×10^{-2})	5	1.8932×10^{-1} (1.34×10^{-2}) +	2.8667×10^{-1} (2.03×10^{-2})	8	3.3750×10^{-1} (4.43×10^{2}) +	3.7228×10^{-1} (1.75×10^{-2})
LSMOP3	3	1.1110×10^{1} (8.65×10^{-1}) -	8.6072×10^{-1} (1.14×10^{-16})	5	1.1574×10^{1} (2.43×10^{0}) -	9.5883×10^{-1} (0.00×10^{0})	8	2.4557×10^{1} (5.32×10^{0}) -	1.8394×10^{0} (1.16×10^{-3})
LSMOP3	3	1.1110×10^{1} (8.65×10^{-1}) -	8.6072×10^{-1} (1.14×10^{-16})	5	1.1574×10^{1} (2.43×10^{0}) -	9.5883×10^{-1} (0.00×10^{0})	8	2.4557×10^{1} (5.32×10^{0}) -	1.8394×10^{0} (1.16×10^{-3})
LSMOP4	3	2.2664×10^{-1} (3.10×10^{-3}) +	3.7947×10^{-1} (1.70×10^{-2})	5	3.4663×10^{-1} (1.14×10^{-1}) +	4.2448×10^{-1} (2.93×10^{-2})	8	3.7218×10^{-1} (3.30×10^{-2}) +	4.4210×10^{-1} (1.68×10^{-2})
LSMOP5	3	1.1456×10^{1} (9.34×10^{-1}) -	5.8970×10^{-1} (1.59×10^{-2})	5	5.8429×10^{0} (7.95×10^{-1}) -	5.2580×10^{-1} (1.41×10^{-2})	8	5.5801×10^{0} (1.72×10^{0}) -	6.0238×10^{-1} (1.29×10^{-2})
LSMOP6	3	1.1647×10^{3} (4.86×10^{2}) -	1.2947×10^{0} (1.23×10^{-2})	5	6.7717×10^{1} (4.37×10^{1}) -	1.2176×10^{0} (1.84×10^{-2})	8	1.5859×10^{0} (1.40×10^{-1}) -	1.1487×10^{0} (1.67×10^{-2})
LSMOP7	3	1.2360×10^{0} (9.53×10^{-2}) -	1.0098×10^{0} (1.80×10^{-2})	5	1.8709×10^{0} (5.37×10^{-1}) -	1.2208×10^{0} (3.72×10^{-2})	8	1.4606×10^{2} (5.10×10^{1}) -	1.2954×10^{0} (1.39×10^{-2})
LSMOP8	3	8.9147×10^{-1} (8.29×10^{-2}) -	4.0702×10^{-1} (1.19×10^{-2})	5	1.1757×10^{0} (2.94×10^{-2}) -	4.7892×10^{-1} (1.98×10^{-2})	8	1.8231×10^{0} (2.73×10^{-1}) -	6.0531×10^{-1} (1.57×10^{-2})
LSMOP9	3	1.2551×10^{1} (1.33×10^{0}) -	1.1138×10^{0} (1.18×10^{-1})	5	3.3787×10^{1} (4.37×10^{0}) -	2.1552×10^{0} (2.24×10^{-1})	8	1.1562×10^{2} (9.66×10^{0}) -	4.1550×10^{0} (3.29×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation of IGD values, and the number out of the brackets represents the mean of IGD values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm in terms of IGD values.

Table 3. IGD values compared with original NSGA-II/SDR with M = 10, 12, and 15.

	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS
LSMOP1	10	4.1673×10^{0} (7.21×10^{-1})	9.7484×10^{-1} (1.25×10^{-2})	12	4.8455×10^{0} (1.25×10^{0})	9.6213×10^{-1} (2.50×10^{-2})	15	4.5034×10^{0} (6.08×10^{-1})	9.9075×10^{-1} (2.43×10^{-2})
LSMOP2	10	4.1109×10^{-1} (5.19×10^{-2})	3.8972×10^{-1} (1.71×10^{-2})	12	4.2597×10^{-1} (8.03×10^{-2})	4.0385×10^{-1} (1.11×10^{-2})	15	3.8220×10^{-1} (3.96×10^{-2}) +	4.2081×10^{-1} (9.61×10^{-3})
LSMOP3	10	2.8745×10^{1} (1.55×10^{1}) -	1.9183×10^{0} (6.67×10^{-4})	12	2.9680×10^{1} (1.71×10^{1}) -	1.9136×10^{0} (5.28×10^{-4})	15	2.7104×10^{1} (5.34×10^{0}) -	1.0440×10^{0} (2.28×10^{-16})
LSMOP4	10	4.3062×10^{-1} (6.00×10^{-2}) +	4.5483×10^{-1} (1.69×10^{-2})	12	4.2320×10^{-1} (5.08×10^{-2}) +	4.5818×10^{-1} (1.46×10^{-2})	15	3.9997×10^{-1} (3.12×10^{-2}) +	4.6769×10^{-1} (1.53×10^{-2})
LSMOP5	10	5.8210×10^{0} (9.85×10^{-1}) -	6.4375×10^{-1} (1.83×10^{-2})	12	4.5858×10^{0} (6.25×10^{-1}) -	6.7580×10^{-1} (1.20×10^{-2})	15	4.2185×10^{0} (5.91×10^{-1}) -	7.1099×10^{-1} (1.00×10^{-2})
LSMOP6	10	2.0789×10^{0} (2.21×10^{0}) -	1.1948×10^{0} (1.34×10^{-2})	12	3.1374×10^{1} (1.33×10^{2}) -	1.2374×10^{0} (1.15×10^{-2})	15	6.2836×10^{2} (1.69×10^{2}) -	1.3672×10^{0} (8.65×10^{-3})
LSMOP7	10	1.2489×10^{3} (5.08×10^{2}) -	1.3225×10^{0} (1.38×10^{-2})	12	7.0748×10^{2} (2.94×10^{2}) -	1.3444×10^{0} (9.64×10^{-3})	15	1.6070×10^{0} (9.35×10^{-2}) -	1.3393×10^{0} (1.99×10^{-2})
LSMOP8	10	3.4899×10^{0} (7.42×10^{-1}) -	6.4259×10^{-1} (1.60×10^{-2})	12	2.6803×10^{0} (4.21×10^{-1}) -	6.7286×10^{-1} (1.18×10^{-2})	15	1.1750×10^{0} (7.71×10^{-2}) -	6.9831×10^{-1} (5.73×10^{-3})
LSMOP9	10	2.2373×10^{2} (2.03×10^{1}) -	4.9644×10^{0} (3.96×10^{-1})	12	3.8086×10^{2} (2.06×10^{1}) -	6.3764×10^{0} (3.18×10^{-1})	15	7.7762×10^{2} (2.82×10^{1}) -	9.8092×10^{0} (6.73×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation of IGD values, and the number out of the brackets represents the mean of IGD values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm in terms of IGD values.

Table 4. DM values compared with original NSGA-II/SDR with M = 3, 5, and 8.

	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS
LSMOP1	3	4.3688×10^{-1} (6.75×10^{-2}) +	2.3190×10^{-1} (5.33×10^{-2})	5	3.0156×10^{-1} (8.47×10^{-2}) +	2.3317×10^{-1} (3.88×10^{-2})	8	3.8057×10^{-1} (5.12×10^{-2}) +	2.5704×10^{-1} (3.80×10^{-2})
LSMOP2	3	6.6415×10^{-1} (3.09×10^{-2}) +	5.7650×10^{-1} (4.07×10^{-2})	5	4.1607×10^{-1} (3.86×10^{-2}) =	4.0589×10^{-1} (5.23×10^{-2})	8	3.8271×10^{-1} (5.15×10^{-2}) =	3.9909×10^{-1} (3.88×10^{-2})
LSMOP3	3	1.0831×10^{-1} (3.14×10^{-2}) -	1.7626×10^{-1} (3.63×10^{-2})	5	1.8195×10^{-1} (4.28×10^{-2}) +	1.1092×10^{-1} (1.77×10^{-2})	8	2.5991×10^{-1} (6.67×10^{-2}) +	1.4907×10^{-1} (3.20×10^{-2})
LSMOP4	3	6.5914×10^{-1} (2.83×10^{-2}) +	5.0649×10^{-1} (4.17×10^{-2})	5	4.5996×10^{-1} (7.20×10^{-2}) =	4.3840×10^{-1} (5.22×10^{-2})	8	3.7126×10^{-1} (4.33×10^{-2}) =	3.9097×10^{-1} (4.98×10^{-2})
LSMOP5	3	1.0128×10^{-1} (2.24×10^{-2}) -	5.0011×10^{-1} (3.92×10^{-2})	5	1.7735×10^{-1} (2.62×10^{-2}) -	4.2999×10^{-1} (2.80×10^{-2})	8	2.0054×10^{-1} (1.96×10^{-2}) -	3.6625×10^{-1} (3.48×10^{-2})
LSMOP6	3	7.2973×10^{-2} (2.38×10^{-2}) -	2.5882×10^{-1} (4.54×10^{-2})	5	1.3241×10^{-1} (5.10×10^{-2}) -	2.5065×10^{-1} (3.35×10^{-2})	8	8.2730×10^{-2} (7.37×10^{-2}) -	2.5153×10^{-1} (5.27×10^{-2})
LSMOP7	3	2.7434×10^{-1} (1.44×10^{-1}) +	1.7412×10^{-1} (4.19×10^{-2})	5	1.3049×10^{-1} (7.73×10^{-2}) -	2.3920×10^{-1} (4.20×10^{-2})	8	1.2310×10^{-1} (3.86×10^{-2}) -	2.4803×10^{-1} (4.05×10^{-2})
LSMOP8	3	1.6534×10^{-1} (3.49×10^{-2}) -	4.9234×10^{-1} (5.43×10^{-2})	5	1.7849×10^{-1} (2.97×10^{-2}) -	4.1702×10^{-1} (3.32×10^{-2})	8	2.0365×10^{-1} (4.05×10^{-2}) -	3.8176×10^{-1} (3.70×10^{-2})
LSMOP9	3	3.9367×10^{-1} (4.14×10^{-2}) -	6.9026×10^{-1} (1.26×10^{-1})	5	3.8226×10^{-1} (4.86×10^{-2}) -	5.8524×10^{-1} (6.11×10^{-2})	8	5.6552×10^{-1} (5.27×10^{-2}) +	5.2402×10^{-1} (5.14×10^{-2})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation of DM values, and the number out of the brackets represents the mean of DM values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm in terms of DM values.

Table 5. DM values compared with original NSGA-II/SDR with M = 10, 12, and 15.

	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS	M	NSGA-II/SDR	NSGA-II/SDR-OLS
LSMOP1	10	3.5985×10^{-1} (6.36×10^{-2}) +	2.3715×10^{-1} (3.79×10^{-2})	12	3.5080×10^{-1} (5.22×10^{-2}) +	2.4499×10^{-1} (2.40×10^{-2})	15	3.9656×10^{-1} (4.14×10^{-2}) +	2.6189×10^{-1} (2.38×10^{-2})
LSMOP2	10	3.6948×10^{-1} (5.81×10^{-2}) -	4.2031×10^{-1} (3.63×10^{-2})	12	3.4826×10^{-1} (4.56×10^{-2}) =	3.7356×10^{-1} (3.47×10^{-2})	15	3.5880×10^{-1} (4.82×10^{-2}) =	3.7683×10^{-1} (3.98×10^{-2})
LSMOP3	10	2.6713×10^{-1} (1.05×10^{-1}) +	1.2479×10^{-1} (2.99×10^{-2})	12	2.8237×10^{-1} (1.69×10^{-1}) +	1.5130×10^{-1} (1.90×10^{-2})	15	6.0595×10^{-1} (3.74×10^{-1}) +	2.0025×10^{-1} (1.98×10^{-2})
LSMOP4	10	3.5042×10^{-1} (3.46×10^{-2}) -	4.0313×10^{-1} (3.98×10^{-2})	12	3.3504×10^{-1} (3.33×10^{-2}) -	3.7049×10^{-1} (2.99×10^{-2})	15	3.9695×10^{-1} (3.01×10^{-2}) =	3.8980×10^{-1} (2.86×10^{-2})
LSMOP5	10	2.2188×10^{-1} (3.80×10^{-2}) -	3.0851×10^{-1} (3.86×10^{-2})	12	2.5446×10^{-1} (3.10×10^{-2}) -	3.2003×10^{-1} (3.64×10^{-2})	15	2.5721×10^{-1} (3.38×10^{-2}) =	2.7269×10^{-1} (2.96×10^{-2})
LSMOP6	10	1.2635×10^{-1} (1.17×10^{-1}) =	1.9221×10^{-1} (3.84×10^{-2})	12	1.1194×10^{-1} (1.17×10^{-1}) -	2.2167×10^{-1} (2.63×10^{-2})	15	2.0575×10^{-1} (3.50×10^{-2}) -	2.5950×10^{-1} (3.72×10^{-2})
LSMOP7	10	9.3230×10^{-2} (1.85×10^{-2}) -	2.3057×10^{-1} (3.57×10^{-2})	12	1.4920×10^{-1} (3.73×10^{-2}) -	2.3232×10^{-1} (3.01×10^{-2})	15	1.3554×10^{-1} (9.96×10^{-2}) -	1.4875×10^{-1} (2.36×10^{-2})
LSMOP8	10	2.2456×10^{-1} (3.59×10^{-2}) -	3.1097×10^{-1} (3.22×10^{-2})	12	2.7443×10^{-1} (3.16×10^{-2}) -	3.2955×10^{-1} (3.18×10^{-2})	15	2.2614×10^{-1} (8.18×10^{-2}) =	2.7508×10^{-1} (2.67×10^{-2})
LSMOP9	10	6.3809×10^{-1} (5.37×10^{-2}) =	6.7788×10^{-1} (5.83×10^{-2})	12	6.1838×10^{-1} (4.05×10^{-2}) -	6.6460×10^{-1} (4.95×10^{-2})	15	8.5501×10^{-1} (2.87×10^{-2}) -	8.7993×10^{-1} (2.83×10^{-2})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation of DM values, and the number out of the brackets represents the mean of DM values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm in terms of DM values.

Table 6. IGD values for 9 three-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	3	3.0767×10^{0} (3.07×10^{-1}) -	4.3517×10^{1} (2.62×10^{1}) -	3.7222×10^{0} (4.98×10^{-1}) -	5.3556×10^{0} (6.04×10^{-1}) -	1.1407×10^{1} (1.70×10^{0}) -	3.7779×10^{0} (3.32×10^{-1}) -	8.6035×10^{-1} (4.13×10^{-4})
LSMOP2	3	9.3421×10^{-2} (7.71×10^{-4}) +	5.8003×10^{-1} (1.33×10^{-1}) -	9.0760×10^{-2} (9.29×10^{-4}) +	9.4270×10^{-2} (5.12×10^{-4}) +	5.1654×10^{-1} (7.22×10^{-2}) -	9.3575×10^{-2} (4.63×10^{-4}) +	1.6243×10^{-1} (1.70×10^{-2})
LSMOP3	3	1.0161×10^{1} (8.02×10^{-1}) -	3.1281×10^{4} (2.12×10^{4}) -	1.4772×10^{1} (3.31×10^{0}) -	1.4262×10^{1} (1.33×10^{0}) -	2.5019×10^{1} (1.20×10^{1}) -	1.1950×10^{1} (7.87×10^{-1}) -	8.6072×10^{-1} ($3.39 \times 10^{-1}6$)
LSMOP4	3	2.3656×10^{-1} (3.12×10^{-3}) +	1.0368×10^{0} (4.28×10^{-1}) -	2.4505×10^{-1} (3.82×10^{-3}) +	2.7426×10^{-1} (4.75×10^{-3}) +	7.6900×10^{-1} (6.86×10^{-2}) -	2.7648×10^{-1} (2.79×10^{-3}) +	3.8314×10^{-1} (2.01×10^{-2})
LSMOP5	3	5.9359×10^{0} (7.32×10^{-1}) -	8.1858×10^{1} (6.62×10^{1}) -	1.0077×10^{1} (1.16×10^{0}) -	1.0541×10^{1} (4.18×10^{0}) -	9.3210×10^{0} (1.58×10^{0}) -	1.0345×10^{1} (1.57×10^{0}) -	5.8922×10^{-1} (1.58×10^{-2})
LSMOP6	3	1.6612×10^{3} (5.15×10^{2}) -	4.2039×10^{5} (4.52×10^{5}) -	1.1939×10^{3} (3.46×10^{2}) -	2.3750×10^{3} (8.82×10^{2}) -	2.4290×10^{4} (6.67×10^{3}) -	1.2083×10^{3} (4.33×10^{2}) -	1.2965×10^{0} (1.34×10^{-2})
LSMOP7	3	8.5039×10^{1} (4.57×10^{2}) -	4.3709×10^{5} (5.25×10^{5}) -	4.4980×10^{3} (3.87×10^{3}) -	1.2590×10^{0} (9.05×10^{-2}) -	1.5529×10^{0} (3.21×10^{-2}) -	1.5607×10^{0} (1.90×10^{-2}) -	1.0071×10^{0} (1.73×10^{-2})
LSMOP8	3	9.4869×10^{-1} (5.82×10^{-2}) -	2.8983×10^{1} (2.29×10^{1}) -	8.8753×10^{-1} (8.54×10^{-2}) -	7.8391×10^{-1} (1.22×10^{-1}) -	9.8088×10^{-1} (4.27×10^{-4}) -	9.7098×10^{-1} (1.49×10^{-2}) -	4.0971×10^{-1} (1.39×10^{-2})
LSMOP9	3	2.4436×10^{1} (3.16×10^{0}) -	3.1941×10^{2} (1.38×10^{2}) -	2.0080×10^{1} (2.48×10^{0}) -	5.5539×10^{1} (1.13×10^{1}) -	2.9830×10^{1} (4.11×10^{0}) -	1.7330×10^{1} (1.94×10^{0}) -	1.1125×10^{0} (1.41×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 7. GD values for 9 three-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	3	1.1963×10^{0} (1.62×10^{-1}) +	1.1377×10^{1} (7.15×10^{0}) -	1.6237×10^{0} (1.86×10^{-1}) =	2.4642×10^{0} (4.31×10^{-1}) -	1.1419×10^{1} (5.70×10^{0}) -	1.5878×10^{0} (1.87×10^{-1}) =	1.4978×10^{0} (2.83×10^{-1})
LSMOP2	3	9.5680×10^{-3} (1.73×10^{-4}) +	1.8721×10^{-2} (3.94×10^{-3}) -	9.1128×10^{-3} (2.69×10^{-4}) +	1.0818×10^{-2} (1.27×10^{-4}) =	2.9460×10^{-2} (1.39×10^{-2}) -	1.0627×10^{-2} (1.16×10^{-4}) =	1.0625×10^{-2} (8.89×10^{-4})
LSMOP3	3	7.5458×10^{2} (1.47×10^{2}) +	8.7258×10^{3} (5.40×10^{3}) -	1.2121×10^{3} (2.79×10^{2}) +	1.1078×10^{3} (6.19×10^{2}) +	1.2737×10^{4} (5.03×10^{3}) -	1.2431×10^{3} (3.17×10^{2}) +	3.4454×10^{3} (6.60×10^{2})
LSMOP4	3	3.8612×10^{-2} (1.11×10^{-3}) +	1.4360×10^{-1} (7.04×10^{-2}) -	4.1272×10^{-2} (1.22×10^{-3}) +	5.0926×10^{-2} (1.00×10^{-3}) +	1.3967×10^{-1} (9.01×10^{-2}) -	5.0508×10^{-2} (6.36×10^{-4}) +	7.4564×10^{-2} (1.41×10^{-2})
LSMOP5	3	4.0713×10^{0} (4.49×10^{-1}) -	1.9971×10^{1} (1.57×10^{1}) -	4.4563×10^{0} (4.81×10^{-1}) -	8.0563×10^{0} (4.56×10^{0}) -	1.0063×10^{1} (2.33×10^{0}) -	4.6429×10^{0} (3.84×10^{-1}) -	5.7865×10^{-1} (9.09×10^{-2})
LSMOP6	3	7.8265×10^{3} (3.74×10^{3}) -	7.4233×10^{4} (5.50×10^{4}) -	7.7020×10^{3} (1.96×10^{3}) -	4.5317×10^{4} (3.14×10^{4}) -	3.9467×10^{4} (1.33×10^{4}) -	8.0897×10^{3} (2.63×10^{3}) -	3.5122×10^{2} (7.88×10^{1})
LSMOP7	3	4.9453×10^{3} (7.65×10^{2}) +	1.2754×10^{5} (1.43×10^{5}) =	3.6885×10^{3} (5.36×10^{2}) +	2.0162×10^{3} (2.58×10^{3}) +	4.9498×10^{4} (1.26×10^{4}) +	4.0954×10^{3} (7.54×10^{2}) +	1.2618×10^{5} (5.05×10^{4})
LSMOP8	3	9.7197×10^{-1} (8.57×10^{-2}) +	8.1480×10^{0} (6.02×10^{0}) =	9.0624×10^{-1} (1.05×10^{-1}) +	8.4953×10^{-1} (7.66×10^{-1}) +	4.5659×10^{0} (1.65×10^{0}) =	1.0962×10^{0} (1.18×10^{-1}) +	4.7494×10^{0} (1.45×10^{0})
LSMOP9	3	5.8018×10^{0} (1.07×10^{0}) -	7.8507×10^{1} (3.44×10^{1}) -	4.0675×10^{0} (5.28×10^{-1}) -	1.5183×10^{1} (1.46×10^{1}) -	2.3250×10^{1} (7.79×10^{0}) -	3.5068×10^{0} (3.70×10^{-1}) -	2.1065×10^{-2} (3.27×10^{-2})

Table 8. IGD values for 9 five-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	5	4.9095×10^{0} (5.52×10^{-1}) -	4.6128×10^{1} (4.04×10^{1}) -	3.4118×10^{0} (3.94×10^{-1}) -	3.5133×10^{0} (4.74×10^{-1}) -	1.3418×10^{1} (2.33×10^{0}) -	6.4850×10^{0} (7.90×10^{-1}) -	9.4323×10^{-1} (2.02×10^{-3})
LSMOP2	5	1.9800×10^{-1} (4.67×10^{-3}) +	7.2196×10^{-1} (1.73×10^{-1}) -	1.7677×10^{-1} (1.60×10^{-2}) +	1.7606×10^{-1} (1.77×10^{-3}) +	6.7702×10^{-1} (1.66×10^{-1}) -	1.7901×10^{-1} (4.03×10^{-4}) +	2.8843×10^{-1} (1.96×10^{-2})
LSMOP3	5	1.2179×10^{1} (8.23×10^{-1}) -	8.1564×10^{4} (5.14×10^{4}) -	1.1149×10^{3} (8.20×10^{2}) -	2.3483×10^{1} (9.67×10^{0}) -	2.5708×10^{1} (8.92×10^{0}) -	2.0077×10^{1} (5.47×10^{0}) -	9.5883×10^{-1} $(4.52 \times 10^{-1}6)$
LSMOP4	5	3.4103×10^{-1} (1.38×10^{-2}) +	1.8979×10^{0} (1.04×10^{0}) -	3.3577×10^{-1} (5.70×10^{-2}) +	3.0844×10^{-1} (6.19×10^{-3}) +	8.1678×10^{-1} (2.17×10^{-1}) -	3.3691×10^{-1} (2.06×10^{-3}) +	4.1999×10^{-1} (2.84×10^{-2})
LSMOP5	5	1.1422×10^{1} (2.08×10^{0}) -	5.7216×10^{1} (3.62×10^{1}) -	1.0483×10^{1} (1.65×10^{0}) -	2.5149×10^{0} (1.09×10^{0}) -	1.8062×10^{1} (4.91×10^{0}) -	1.2514×10^{1} (2.14×10^{0}) -	5.2796×10^{-1} (1.97×10^{-2})
LSMOP6	5	8.6350×10^{2} (5.31×10^{2}) -	3.4687×10^{5} (4.55×10^{5}) -	4.6473×10^{2} (3.23×10^{2}) -	1.0232×10^{2} (7.80×10^{1}) -	7.1809×10^{4} (4.92×10^{4}) -	2.5899×10^{3} (4.18×10^{3}) -	1.2174×10^{0} (1.61×10^{-2})
LSMOP7	5	4.0660×10^{2} (2.21×10^{3}) -	4.4562×10^{5} (4.75×10^{5}) -	5.7024×10^{3} (8.74×10^{3}) -	1.8118×10^{0} (2.60×10^{-1}) -	2.9687×10^{0} (1.52×10^{-1}) -	2.7954×10^{0} (1.12×10^{-1}) -	1.2177×10^{0} (3.74×10^{-2})
LSMOP8	5	1.6360×10^{0} (1.32×10^{0}) -	2.4307×10^{1} (2.30×10^{1}) -	3.0791×10^{0} (2.49×10^{0}) -	9.4960×10^{-1} (8.72×10^{-2}) -	1.3702×10^{0} (1.04×10^{0}) -	1.1863×10^{0} (1.61×10^{-2}) -	4.7855×10^{-1} (1.80×10^{-2})
LSMOP9	5	7.4682×10^{1} (7.65×10^{0}) -	5.8820×10^{2} (2.51×10^{2}) -	7.7794×10^{1} (9.12×10^{0}) -	2.1686×10^{2} (3.25×10^{1}) -	9.3178×10^{1} (1.21×10^{1}) -	9.3151×10^{1} (1.05×10^{1}) -	2.1596×10^{0} (2.43×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 9. GD values for 9 five-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	5	1.2629×10^{0} (1.31×10^{-1}) +	1.0029×10^{1} (9.18×10^{0}) -	1.1907×10^{0} (2.03×10^{-1}) +	1.0561×10^{0} (3.51×10^{-1}) +	7.3281×10^{0} (4.34×10^{0}) -	1.7659×10^{0} (1.07×10^{-1}) -	1.6214×10^{0} (1.87×10^{-1})
LSMOP2	5	1.1963×10^{-2} (1.99×10^{-4}) -	2.8321×10^{-2} (5.16×10^{-3}) -	9.7645×10^{-3} (4.05×10^{-4}) +	1.2604×10^{-2} (5.94×10^{-4}) -	3.8167×10^{-2} (3.77×10^{-2}) -	1.3559×10^{-2} (7.55×10^{-5}) -	1.0635×10^{-2} (6.92×10^{-4})
LSMOP3	5	2.7623×10^{3} (3.00×10^{2}) +	1.9383×10^{4} (1.21×10^{4}) -	2.7358×10^{3} (6.72×10^{2}) +	1.5596×10^{3} (1.87×10^{3}) +	1.5293×10^{4} (6.90×10^{3}) -	5.1390×10^{3} (9.08×10^{2}) +	7.1726×10^{3} (1.40×10^{3})
LSMOP4	5	6.4989×10^{-2} (2.40×10^{-2}) -	2.9986×10^{-1} (2.06×10^{-1}) -	5.3963×10^{-2} (8.41×10^{-3}) -	5.3808×10^{-2} (5.53×10^{-3}) -	4.3879×10^{-1} (4.28×10^{-1}) -	8.0706×10^{-2} (2.20×10^{-3}) -	3.7189×10^{-2} (3.21×10^{-3})
LSMOP5	5	9.3364×10^{0} (5.45×10^{-1}) -	1.5085×10^{1} (9.55×10^{0}) -	6.9220×10^{0} (9.38×10^{-1}) -	2.8038×10^{0} (3.80×10^{0}) -	1.1902×10^{1} (2.06×10^{0}) -	1.0975×10^{1} (1.95×10^{0}) -	3.8445×10^{-2} (4.46×10^{-3})
LSMOP6	5	1.4927×10^{4} (5.35×10^{3}) -	9.0072×10^{4} (1.21×10^{5}) -	3.2039×10^{4} (1.43×10^{4}) -	7.7020×10^{3} (9.92×10^{3}) -	7.3474×10^{4} (1.48×10^{4}) -	2.4241×10^{4} (9.27×10^{3}) -	1.5447×10^{-1} (9.69×10^{-3})
LSMOP7	5	3.3870×10^{4} (5.98×10^{3}) -	1.3023×10^{5} (1.13×10^{5}) -	2.9125×10^{4} (6.16×10^{3}) -	2.3477×10^{3} (3.16×10^{3}) +	6.1440×10^{4} (7.54×10^{3}) -	7.2063×10^{4} (1.07×10^{4}) -	6.3879×10^{3} (1.36×10^{4})
LSMOP8	5	3.6270×10^{0} (1.83×10^{-1}) -	5.1737×10^{0} (5.57×10^{0}) =	2.7691×10^{0} (2.50×10^{-1}) -	1.9039×10^{0} (1.64×10^{0}) =	5.4621×10^{0} (6.43×10^{-1}) -	5.8008×10^{0} (3.57×10^{-1}) -	1.4251×10^{0} (5.12×10^{-1})
LSMOP9	5	1.8822×10^{1} (4.67×10^{0}) -	1.4001×10^{2} (6.21×10^{1}) -	1.7280×10^{1} (1.88×10^{0}) -	1.5350×10^{2} (1.06×10^{2}) -	4.3751×10^{1} (1.32×10^{1}) -	2.3092×10^{1} (2.22×10^{0}) -	5.1549×10^{-2} (2.51×10^{-2})

Table 10. IGD values for 9 eight-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	8	7.1110×10^{0} (1.25×10^{0}) -	3.7558×10^{1} (2.60×10^{1}) -	4.9285×10^{0} (8.52×10^{-1}) -	4.0295×10^{0} (6.85×10^{-1}) -	1.1896×10^{1} (3.15×10^{0}) -	7.2860×10^{0} (8.74×10^{-1}) -	9.6768×10^{-1} (8.05×10^{-3})
LSMOP2	8	3.3358×10^{-1} (6.93×10^{-3}) +	1.6115×10^{0} (6.78×10^{-1}) -	3.9133×10^{-1} (8.08×10^{-2}) =	3.0174×10^{-1} (3.03×10^{-2}) +	6.9579×10^{-1} (1.78×10^{-1}) -	2.6291×10^{-1} (3.75×10^{-3}) +	3.7228×10^{-1} (1.75×10^{-2})
LSMOP3	8	1.6657×10^{1} (1.63×10^{0}) -	6.1460×10^{5} (7.39×10^{5}) -	1.3203×10^{4} (4.06×10^{3}) -	2.6963×10^{1} (3.99×10^{0}) -	2.4306×10^{1} (4.52×10^{0}) -	1.1786×10^{3} (1.79×10^{3}) -	1.8394×10^{0} (1.16×10^{-3})
LSMOP4	8	3.8695×10^{-1} (5.73×10^{-3}) +	8.7746×10^{-1} (1.55×10^{-1}) -	3.8617×10^{-1} (4.60×10^{-2}) +	3.0904×10^{-1} (3.23×10^{-2}) +	7.4061×10^{-1} (1.55×10^{-1}) -	3.1513×10^{-1} (4.46×10^{-3}) +	4.4210×10^{-1} (1.68×10^{-2})
LSMOP5	8	1.8409×10^{1} (2.59×10^{0}) -	3.8467×10^{1} (3.34×10^{1}) -	1.2795×10^{1} (7.11×10^{0}) -	3.3177×10^{0} (6.39×10^{-1}) -	2.4220×10^{1} (5.89×10^{0}) -	1.3842×10^{1} (1.94×10^{0}) -	6.0238×10^{-1} (1.29×10^{-2})
LSMOP6	8	1.8153×10^{0} (6.46×10^{-2}) -	1.3184×10^{5} (1.95×10^{5}) -	1.1924×10^{4} (5.73×10^{3}) -	1.5629×10^{0} (7.45×10^{-2}) -	1.8967×10^{0} (6.11×10^{-2}) -	1.8018×10^{0} (3.35×10^{-2}) -	1.1487×10^{0} (1.67×10^{-2})
LSMOP7	8	1.9612×10^{4} (1.71×10^{4}) -	2.3192×10^{5} (2.85×10^{5}) -	6.6669×10^{3} (9.27×10^{3}) -	1.6613×10^{2} (7.58×10^{1}) -	1.1901×10^{5} (4.19×10^{4}) -	3.0521×10^{3} (7.18×10^{3}) -	1.2954×10^{0} (1.39×10^{-2})
LSMOP8	8	6.9849×10^{0} (4.06×10^{0}) -	3.1277×10^{1} (1.88×10^{1}) -	3.0763×10^{0} (7.01×10^{-1}) -	1.7168×10^{0} (2.61×10^{-1}) -	2.3617×10^{1} (5.82×10^{0}) -	5.7218×10^{0} (1.75×10^{0}) -	6.0531×10^{-1} (1.57×10^{-2})
LSMOP9	8	3.7140×10^{2} (3.15×10^{1}) -	1.7437×10^{3} (9.18×10^{2}) -	4.4525×10^{2} (5.23×10^{1}) -	5.9298×10^{2} (6.41×10^{1}) -	3.2087×10^{2} (3.17×10^{1}) -	5.6127×10^{2} (5.75×10^{1}) -	4.1550×10^{0} (3.29×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 11. GD values for 9 eight-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	8	1.7119×10^{0} (1.88×10^{-1}) -	9.2071×10^{0} (6.48×10^{0}) -	1.4178×10^{0} (4.66×10^{-1}) =	1.1380×10^{0} (1.88×10^{-1}) +	5.3823×10^{0} (2.83×10^{0}) -	2.0047×10^{0} (2.31×10^{-1}) -	1.4869×10^{0} (1.46×10^{-1})
LSMOP2	8	2.4807×10^{-2} (1.55×10^{-3}) -	2.0204×10^{-1} (1.36×10^{-1}) -	2.3595×10^{-2} (4.74×10^{-3}) -	2.3700×10^{-2} (2.99×10^{-3}) -	7.4241×10^{-2} (6.90×10^{-2}) -	3.3813×10^{-2} (1.29×10^{-3}) -	1.7061×10^{-2} (2.12×10^{-3})
LSMOP3	8	1.0739×10^{4} (3.36×10^{3}) -	1.5365×10^{5} (1.85×10^{5}) -	1.2041×10^{4} (4.19×10^{3}) -	5.5895×10^{2} (5.33×10^{2}) +	3.7963×10^{4} (2.36×10^{4}) -	1.0446×10^{4} (2.56×10^{3}) -	2.3370×10^{3} (3.59×10^{2})
LSMOP4	8	5.0203×10^{-2} (2.36×10^{-3}) -	4.1471×10^{-2} (1.64×10^{-2}) +	2.7628×10^{-2} (6.82×10^{-3}) +	3.1235×10^{-2} (5.21×10^{-3}) +	9.8200×10^{-2} (6.54×10^{-2}) -	5.7510×10^{-2} (3.01×10^{-3}) -	4.3584×10^{-2} (4.08×10^{-3})
LSMOP5	8	1.5380×10^{1} (6.35×10^{-1}) -	9.4776×10^{0} (8.36×10^{0}) -	7.6432×10^{0} (3.41×10^{0}) -	8.9827×10^{-1} (9.17×10^{-1}) -	1.1600×10^{1} (3.37×10^{0}) -	1.2330×10^{1} (3.04×10^{0}) -	4.1447×10^{-2} (3.33×10^{-3})
LSMOP6	8	1.0570×10^{5} (1.90×10^{4}) -	3.2960×10^{4} (4.87×10^{4}) =	8.2954×10^{4} (1.81×10^{4}) -	5.4382×10^{2} (1.12×10^{3}) +	7.4940×10^{4} (3.88×10^{4}) -	1.1012×10^{5} (1.41×10^{4}) -	1.8739×10^{4} (1.08×10^{4})
LSMOP7	8	1.1385×10^{5} (3.52×10^{4}) -	5.7979×10^{4} (7.13×10^{4}) -	7.1486×10^{4} (4.26×10^{4}) -	3.0804×10^{3} (3.28×10^{3}) -	1.0481×10^{5} (3.12×10^{4}) -	6.0418×10^{4} (3.56×10^{4}) -	1.4507×10^{-1} (8.80×10^{-3})
LSMOP8	8	7.6531×10^{0} (4.22×10^{-1}) -	7.6830×10^{0} (4.70×10^{0}) -	3.4645×10^{0} (9.08×10^{-1}) -	1.8465×10^{0} (2.13×10^{0}) -	6.7090×10^{0} (1.50×10^{0}) -	5.8351×10^{0} (1.43×10^{0}) -	4.0622×10^{-2} (5.47×10^{-3})
LSMOP9	8	9.8530×10^{1} (1.33×10^{1}) -	4.3465×10^{2} (2.30×10^{2}) -	9.7022×10^{1} (1.09×10^{1}) -	4.7201×10^{2} (2.32×10^{2}) -	1.2373×10^{2} (3.32×10^{1}) -	1.4639×10^{2} (1.48×10^{1}) -	9.2175×10^{-2} (1.29×10^{-2})

Tables 6 and 7, respectively, show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine three-objective LSMOPs. As can be seen from Table 6, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by PREA and DEA-GNG, which respectively achieved the optimal values in the other one test instance. It can be seen from Table 7 that the GD of NSGA-II/SDR-OLS has a general performance, with three optimal values obtained in nine test instances, which is the same as PREA.

Tables 8 and 9 respectively show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine five-objective LSMOPs. As can be seen from Table 8, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by RVEA, which achieved the optimal values in the other two test instances. It can be seen from Table 9 that the GD of NSGA-II/SDR-OLS has a best performance, with five optimal values obtained in nine test instances, followed by RVEA, which achieved the optimal values in three test instances.

Tables 10 and 11, respectively, show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine eight-objective LSMOPs. As can be seen from Table 10, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by RVEA and NSGA-III, which, respectively, achieved the optimal values in the other one test instance. It can be seen from Table 11 that the GD of NSGA-II/SDR-OLS has the best performance, with five optimal values obtained in nine test instances, followed by RVEA, which achieved the optimal values in three test instances.

Tables 12 and 13 respectively show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine eight-objective LSMOPs. As can be seen from Table 12, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by RVEA, which achieved the optimal values in the other two test instances. It can be seen from Table 13 that the GD of NSGA-II/SDR-OLS has the best performance, with five optimal values obtained in nine test instances, followed by RVEA, which achieved the optimal values in three test instances.

Tables 14 and 15, respectively, show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine eight-objective LSMOPs. As can be seen from Table 14, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by RVEA, which achieved the optimal values in the other two test instances. It can be seen from Table 15 that the GD of NSGA-II/SDR-OLS has the best performance, with five optimal values obtained in nine test instances, followed by RVEA, which achieved the optimal values in three test instances.

Tables 16 and 17, respectively, show the IGD and GD results of NSGA-II/SDR-OLS and another six MaOEAs on nine fifteen-objective LSMOPs. As can be seen from Table 16, the IGD of NSGA-II/SDR-OLS has the best performance, achieving seven optimal values in nine test instances, followed by NSGA-II-conflict, which achieved the optimal values in the other two test instances. It can be seen from Table 17 that the GD of NSGA-II/SDR-OLS has the best performance, with six optimal values obtained in nine test instances, followed by RVEA, which achieved the optimal values in three test instances.

Table 12. IGD values for 9 ten-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	10	9.1611×10^{0} (9.34×10^{-1}) -	4.0702×10^{1} (3.68×10^{1}) -	7.6280×10^{0} (1.37×10^{0}) -	4.4396×10^{0} (5.60×10^{-1}) -	1.3562×10^{1} (2.60×10^{0}) -	8.4103×10^{0} (7.07×10^{-1}) -	9.7484×10^{-1} (1.25×10^{-2})
LSMOP2	10	3.8055×10^{-1} (8.30×10^{-3}) =	1.0184×10^{0} (3.06×10^{-1}) -	4.7975×10^{-1} (7.49×10^{-2}) -	3.1423×10^{-1} (3.30×10^{-2}) +	7.1517×10^{-1} (1.60×10^{-1}) -	3.9616×10^{-1} (1.63×10^{-2}) =	3.8972×10^{-1} (1.71×10^{-2})
LSMOP3	10	1.9897×10^{1} (3.73×10^{0}) -	4.6358×10^{5} (6.03×10^{5}) -	3.1674×10^{4} (1.57×10^{4}) -	3.1857×10^{1} (1.15×10^{1}) -	2.8527×10^{1} (2.96×10^{0}) -	8.3392×10^{2} (2.12×10^{3}) -	1.9183×10^{0} (6.67×10^{-4})
LSMOP4	10	4.1529×10^{-1} (9.83×10^{-3}) +	8.2777×10^{-1} (1.34×10^{-1}) -	4.3654×10^{-1} (5.95×10^{-2}) =	3.7070×10^{-1} (5.84×10^{-2}) +	6.9616×10^{-1} (1.35×10^{-1}) -	4.3034×10^{-1} (1.02×10^{-2}) +	4.5483×10^{-1} (1.69×10^{-2})
LSMOP5	10	2.1696×10^{1} (2.75×10^{0}) -	5.5017×10^{1} (3.52×10^{1}) -	1.8103×10^{1} (1.02×10^{1}) -	5.2392×10^{0} (1.02×10^{0}) -	2.1103×10^{1} (3.66×10^{0}) -	1.9892×10^{1} (5.62×10^{0}) -	6.4375×10^{-1} (1.83×10^{-2})
LSMOP6	10	5.1772×10^{3} (2.07×10^{4}) -	9.5585×10^{4} (1.53×10^{5}) -	3.5781×10^{4} (2.30×10^{4}) -	1.4441×10^{0} (4.19×10^{-2}) -	1.5413×10^{0} (9.38×10^{-3}) -	1.5159×10^{0} (7.32×10^{-3}) -	1.1948×10^{0} (1.34×10^{-2})
LSMOP7	10	4.8588×10^{4} (3.80×10^{4}) -	2.3075×10^{5} (2.21×10^{5}) -	2.2062×10^{4} (2.97×10^{4}) -	1.1820×10^{3} (3.89×10^{2}) -	1.1480×10^{5} (5.69×10^{4}) -	5.7196×10^{3} (2.61×10^{3}) -	1.3225×10^{0} (1.38×10^{-2})
LSMOP8	10	1.1766×10^{1} (6.05×10^{0}) -	2.8291×10^{1} (1.45×10^{1}) -	7.5096×10^{0} (4.03×10^{0}) -	2.8017×10^{0} (3.91×10^{-1}) -	1.8570×10^{1} (5.64×10^{0}) -	9.2507×10^{0} (2.25×10^{0}) -	6.4259×10^{-1} (1.60×10^{-2})
LSMOP9	10	7.6628×10^{2} (3.65×10^{1}) -	1.7607×10^{3} (5.08×10^{2}) -	9.4821×10^{2} (1.18×10^{2}) -	9.9598×10^{2} (9.80×10^{1}) -	7.9528×10^{2} (6.19×10^{1}) -	7.5698×10^{2} (3.79×10^{1}) -	4.9644×10^{0} (3.96×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 13. GD values for 9 ten-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	10	2.0481×10^{0} (1.97×10^{-1}) -	9.9986×10^{0} (9.16×10^{0}) -	2.2090×10^{0} (6.60×10^{-1}) -	1.3587×10^{0} (2.25×10^{-1}) +	7.1926×10^{0} (4.02×10^{0}) -	2.9154×10^{0} (3.00×10^{-1}) -	1.4531×10^{0} (8.70×10^{-2})
LSMOP2	10	2.2880×10^{-2} (8.67×10^{-4}) -	7.6257×10^{-2} (3.96×10^{-2}) -	2.6425×10^{-2} (3.59×10^{-3}) -	2.3759×10^{-2} (4.62×10^{-3}) -	6.0869×10^{-2} (2.94×10^{-2}) -	3.3843×10^{-2} (1.00×10^{-3}) -	1.8524×10^{-2} (1.28×10^{-3})
LSMOP3	10	1.8663×10^{4} (3.54×10^{3}) -	1.1589×10^{5} (1.51×10^{5}) -	3.5758×10^{4} (1.81×10^{4}) -	9.4464×10^{2} (8.66×10^{2}) +	9.3773×10^{4} (8.10×10^{4}) -	2.6933×10^{4} (5.64×10^{3}) -	2.7920×10^{3} (4.41×10^{2})
LSMOP4	10	5.1812×10^{-2} (3.35×10^{-3}) -	3.4311×10^{-2} (1.14×10^{-2}) =	2.1847×10^{-2} (2.29×10^{-3}) +	2.5421×10^{-2} (5.59×10^{-3}) +	4.2724×10^{-2} (2.43×10^{-2}) =	7.0390×10^{-2} (2.10×10^{-3}) -	3.5861×10^{-2} (3.86×10^{-3})
LSMOP5	10	1.7632×10^{1} (6.83×10^{-1}) -	1.3609×10^{1} (8.80×10^{0}) -	1.0959×10^{1} (4.01×10^{0}) -	1.4193×10^{0} (1.01×10^{0}) -	9.5817×10^{0} (2.59×10^{0}) -	1.9252×10^{1} (2.20×10^{0}) -	6.3491×10^{-2} (4.45×10^{-3})
LSMOP6	10	1.4557×10^{5} (1.18×10^{4}) -	2.3896×10^{4} (3.82×10^{4}) -	6.4101×10^{4} (2.06×10^{4}) -	1.1389×10^{2} (2.49×10^{2}) +	7.6239×10^{4} (3.75×10^{4}) -	1.3287×10^{5} (1.01×10^{4}) -	1.7193×10^{4} (6.99×10^{3})
LSMOP7	10	1.5503×10^{5} (3.35×10^{4}) -	5.7688×10^{4} (5.54×10^{4}) -	8.3025×10^{4} (4.85×10^{4}) -	1.6441×10^{3} (2.13×10^{3}) -	7.8358×10^{4} (3.70×10^{4}) -	7.9420×10^{4} (2.88×10^{4}) -	1.2505×10^{0} (2.24×10^{-1})
LSMOP8	10	8.6690×10^{0} (4.31×10^{-1}) -	6.9236×10^{0} (3.63×10^{0}) -	4.4941×10^{0} (2.46×10^{0}) -	1.0520×10^{0} (7.24×10^{-1}) -	5.8513×10^{0} (1.51×10^{0}) -	9.0945×10^{0} (1.01×10^{0}) -	6.1613×10^{-2} (6.31×10^{-3})
LSMOP9	10	2.1676×10^{2} (2.78×10^{1}) -	4.3858×10^{2} (1.27×10^{2}) -	2.0987×10^{2} (2.56×10^{1}) -	8.7893×10^{2} (3.06×10^{2}) -	2.9160×10^{2} (4.58×10^{1}) -	2.0653×10^{2} (1.45×10^{1}) -	9.3000×10^{-2} (1.43×10^{-2})

Table 14. IGD values for 9 twelve-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	12	1.0018×10^{1} (8.23×10^{-1}) -	5.2629×10^{1} (3.83×10^{1}) -	7.6130×10^{0} (8.85×10^{-1}) -	4.5943×10^{0} (4.94×10^{-1}) -	1.4711×10^{1} (2.72×10^{0}) -	7.7126×10^{0} (6.19×10^{-1}) -	9.6213×10^{-1} (2.50×10^{-2})
LSMOP2	12	4.0437×10^{-1} (1.62×10^{-2}) =	1.1529×10^{0} (3.85×10^{-1}) -	4.1742×10^{-1} (5.62×10^{-2}) =	3.0819×10^{-1} (2.94×10^{-2}) +	6.8779×10^{-1} (1.81×10^{-1}) -	3.8339×10^{-1} (4.28×10^{-2}) =	4.0385×10^{-1} (1.11×10^{-2})
LSMOP3	12	2.1513×10^{1} (2.69×10^{0}) -	5.6524×10^{5} (7.81×10^{5}) -	2.7587×10^{4} (1.13×10^{4}) -	7.1167×10^{1} (7.82×10^{1}) -	2.9781×10^{2} (1.20×10^{3}) -	1.8718×10^{2} (6.90×10^{2}) -	1.9136×10^{0} (5.28×10^{-4})
LSMOP4	12	4.3832×10^{-1} (1.98×10^{-2}) +	8.3568×10^{-1} (1.17×10^{-1}) -	4.0061×10^{-1} (3.62×10^{-2}) +	3.3585×10^{-1} (3.48×10^{-2}) +	7.2743×10^{-1} (1.48×10^{-1}) -	4.2190×10^{-1} (2.50×10^{-2}) +	4.5818×10^{-1} (1.46×10^{-2})
LSMOP5	12	2.0907×10^{1} (3.38×10^{0}) -	4.1103×10^{1} (3.58×10^{1}) -	2.0702×10^{1} (1.07×10^{1}) -	4.8794×10^{0} (8.78×10^{-1}) -	1.8620×10^{1} (2.87×10^{0}) -	1.6587×10^{1} (2.23×10^{0}) -	6.7580×10^{-1} (1.20×10^{-2})
LSMOP6	12	1.9445×10^{4} (4.73×10^{4}) -	1.1786×10^{5} (1.47×10^{5}) -	5.7236×10^{4} (5.29×10^{4}) -	1.4707×10^{0} (5.78×10^{-2}) -	1.6429×10^{0} (1.46×10^{-2}) -	1.6112×10^{0} (1.40×10^{-2}) -	1.2374×10^{0} (1.15×10^{-2})
LSMOP7	12	5.4971×10^{4} (3.81×10^{4}) -	1.4258×10^{5} (1.25×10^{5}) -	1.6984×10^{4} (2.34×10^{4}) -	7.8930×10^{2} (3.91×10^{2}) -	7.3450×10^{4} (2.06×10^{4}) -	6.3644×10^{3} (6.18×10^{3}) -	1.3444×10^{0} (9.64×10^{-3})
LSMOP8	12	1.7657×10^{1} (5.67×10^{0}) -	2.2444×10^{1} (1.29×10^{1}) -	7.9232×10^{0} (5.60×10^{0}) -	2.4409×10^{0} (4.56×10^{-1}) -	1.6166×10^{1} (2.46×10^{0}) -	8.3617×10^{0} (2.73×10^{0}) -	6.7286×10^{-1} (1.18×10^{-2})
LSMOP9	12	1.2788×10^{3} (7.23×10^{1}) -	3.1846×10^{3} (1.43×10^{3}) -	1.6947×10^{3} (1.78×10^{2}) -	1.3847×10^{3} (1.29×10^{2}) -	1.4649×10^{3} (4.71×10^{1}) -	1.3642×10^{3} (1.23×10^{2}) -	6.3764×10^{0} (3.18×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 15. GD values for 9 twelve-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	12	1.8878×10^{0} (1.22×10^{-1}) -	1.2963×10^{1} (9.56×10^{0}) -	1.8933×10^{0} (4.10×10^{-1}) -	1.2862×10^{0} (3.14×10^{-1}) +	4.9266×10^{0} (1.89×10^{0}) -	2.1918×10^{0} (1.13×10^{-1}) -	1.5142×10^{0} (1.13×10^{-1})
LSMOP2	12	2.5086×10^{-2} (1.47×10^{-3}) -	9.7305×10^{-2} (5.83×10^{-2}) -	2.2495×10^{-2} (2.87×10^{-3}) -	2.2197×10^{-2} (6.57×10^{-3}) =	6.6262×10^{-2} (5.83×10^{-2}) -	2.8516×10^{-2} (1.28×10^{-3}) -	1.9342×10^{-2} (1.13×10^{-3})
LSMOP3	12	1.7296×10^{4} (3.38×10^{3}) -	1.4131×10^{5} (1.95×10^{5}) -	2.5025×10^{4} (1.02×10^{4}) -	7.1193×10^{2} (1.01×10^{3}) +	6.4390×10^{4} (4.62×10^{4}) -	2.1277×10^{4} (4.55×10^{3}) -	3.4474×10^{3} (4.19×10^{2})
LSMOP4	12	4.8747×10^{-2} (4.23×10^{-3}) -	3.9808×10^{-2} (1.28×10^{-2}) -	1.9421×10^{-2} (1.46×10^{-3}) +	4.5609×10^{-2} (2.02×10^{-2}) -	4.5609×10^{-2} (2.02×10^{-2}) -	4.8970×10^{-2} (4.77×10^{-3}) -	3.0675×10^{-2} (1.90×10^{-3})
LSMOP5	12	1.8421×10^{1} (7.68×10^{-1}) -	1.0125×10^{1} (8.95×10^{0}) -	1.0140×10^{1} (4.49×10^{0}) -	1.1658×10^{0} (8.74×10^{-1}) -	1.0389×10^{1} (2.51×10^{0}) -	1.5316×10^{1} (2.10×10^{0}) -	5.4845×10^{-2} (3.65×10^{-3})
LSMOP6	12	1.6254×10^{5} (1.19×10^{4}) -	2.9465×10^{4} (3.68×10^{4}) =	6.5509×10^{4} (1.84×10^{4}) -	9.2101×10^{2} (2.77×10^{3}) +	6.1254×10^{4} (1.81×10^{4}) -	1.2567×10^{5} (1.33×10^{4}) -	1.3654×10^{4} (4.35×10^{3})
LSMOP7	12	1.6165×10^{5} (3.41×10^{4}) -	3.5645×10^{4} (3.14×10^{4}) -	5.7448×10^{4} (3.95×10^{4}) -	1.1899×10^{3} (2.94×10^{3}) -	7.0558×10^{4} (2.68×10^{4}) -	6.9028×10^{4} (2.83×10^{4}) -	2.9176×10^{-1} (6.30×10^{-2})
LSMOP8	12	9.1311×10^{0} (3.05×10^{-1}) -	5.4579×10^{0} (3.23×10^{0}) -	4.3283×10^{0} (1.92×10^{0}) -	8.5704×10^{-1} (7.90×10^{-1}) -	5.6593×10^{0} (5.93×10^{-1}) -	7.2745×10^{0} (1.61×10^{0}) -	5.1678×10^{-2} (4.92×10^{-3})
LSMOP9	12	3.6672×10^{2} (4.30×10^{1}) -	7.9421×10^{2} (3.57×10^{2}) -	3.2447×10^{2} (4.47×10^{1}) -	8.3872×10^{2} (5.59×10^{2}) -	4.6681×10^{2} (4.80×10^{1}) -	2.9837×10^{2} (1.80×10^{1}) -	1.0028×10^{-1} (9.78×10^{-3})

Table 16. IGD values for 9 fifteen-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	15	9.9732×10^{0} (1.05×10^{0}) -	4.1929×10^{1} (2.77×10^{1}) -	1.2033×10^{1} (8.99×10^{0}) -	7.0254×10^{0} (1.31×10^{0}) -	1.3113×10^{1} (3.71×10^{0}) -	7.5635×10^{0} (1.09×10^{0}) -	9.9075×10^{-1} (2.43×10^{-2})
LSMOP2	15	4.1963×10^{-1} (2.13×10^{-2}) =	1.0024×10^{0} (2.44×10^{-1}) -	7.7469×10^{-1} (1.25×10^{-1}) -	6.3630×10^{-1} (1.55×10^{-1}) -	7.2566×10^{-1} (1.62×10^{-1}) -	4.0412×10^{-1} (3.68×10^{-2}) +	4.2081×10^{-1} (9.61×10^{-3})
LSMOP3	15	2.4822×10^{1} (2.74×10^{0}) -	1.0943×10^{5} (5.35×10^{4}) -	6.3063×10^{2} (1.24×10^{3}) -	1.1128×10^{2} (1.12×10^{2}) -	3.2185×10^{1} (8.72×10^{0}) -	4.8734×10^{1} (2.52×10^{1}) -	1.0440×10^{0} $(2.28 \times 10^{-1}6)$
LSMOP4	15	4.6559×10^{-1} (2.52×10^{-2}) =	1.1002×10^{0} (3.36×10^{-1}) -	9.7195×10^{-1} (2.90×10^{-1}) -	6.9424×10^{-1} (2.27×10^{-1}) -	7.0874×10^{-1} (1.85×10^{-1}) -	4.2075×10^{-1} (3.54×10^{-2}) +	4.6769×10^{-1} (1.53×10^{-2})
LSMOP5	15	2.2980×10^{1} (6.66×10^{0}) -	3.1725×10^{1} (2.51×10^{1}) -	1.0521×10^{1} (2.27×10^{0}) -	6.7168×10^{0} (1.04×10^{0}) -	2.0583×10^{1} (3.34×10^{0}) -	1.0772×10^{1} (3.31×10^{0}) -	7.1099×10^{-1} (1.00×10^{-2})
LSMOP6	15	5.0787×10^{4} (4.47×10^{4}) -	1.8963×10^{5} (1.28×10^{5}) -	7.9605×10^{3} (6.71×10^{3}) -	2.5290×10^{3} (1.16×10^{3}) -	8.5084×10^{4} (3.02×10^{4}) -	5.5259×10^{3} (4.06×10^{3}) -	1.3672×10^{0} (8.65×10^{-3})
LSMOP7	15	3.0068×10^{3} (1.04×10^{4}) -	1.1473×10^{5} (1.35×10^{5}) -	8.3748×10^{3} (9.85×10^{3}) -	2.2724×10^{1} (6.06×10^{1}) -	1.8327×10^{0} (1.74×10^{-2}) -	1.8135×10^{3} (4.01×10^{3}) -	1.3393×10^{0} (1.99×10^{-2})
LSMOP8	15	2.3444×10^{0} (2.50×10^{0}) -	2.0002×10^{1} (1.61×10^{1}) -	4.6646×10^{0} (4.72×10^{0}) -	1.3125×10^{0} (2.70×10^{-2}) -	1.3188×10^{0} (1.08×10^{-2}) -	1.3239×10^{0} (3.10×10^{-4}) -	6.9831×10^{-1} (5.73×10^{-3})
LSMOP9	15	2.3615×10^{3} (8.29×10^{1}) -	5.4298×10^{3} (2.01×10^{3}) -	4.6285×10^{3} (2.13×10^{3}) -	2.6975×10^{3} (1.10×10^{2}) -	2.5878×10^{3} (7.84×10^{1}) -	3.0633×10^{3} (1.26×10^{3}) -	9.8092×10^{0} (6.73×10^{-1})

The gray background represents that this algorithm has the best performance on this problem. The number in the brackets represents the standard deviation, and the number out of the brackets represents the mean values. "+/-/=" means that the relevant algorithm performs better than/worse than/as well as the NSGA-II/SDR-OLS algorithm. Same below table.

Table 17. GD values for 9 fifteen-objective benchmark problems.

	M	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-conflict	NSGA-III	NSGA-II/SDR-OLS
LSMOP1	15	1.9288×10^{0} (1.30×10^{-1}) -	1.0283×10^{1} (6.89×10^{0}) -	5.3807×10^{0} (1.96×10^{0}) -	6.1332×10^{0} (3.60×10^{0}) -	6.8459×10^{0} (4.39×10^{0}) -	4.1265×10^{0} (7.12×10^{-1}) -	1.7214×10^{0} (1.13×10^{-1})
LSMOP2	15	2.6408×10^{-2} (1.85×10^{-3}) -	7.5055×10^{-2} (3.47×10^{-2}) -	4.4323×10^{-2} (9.91×10^{-3}) -	1.1681×10^{-1} (5.12×10^{-2}) -	5.6459×10^{-2} (2.87×10^{-2}) -	5.1511×10^{-2} (7.24×10^{-3}) -	2.4715×10^{-2} (1.67×10^{-3})
LSMOP3	15	6.0677×10^{3} (6.33×10^{2}) +	2.7358×10^{4} (1.34×10^{4}) -	3.2451×10^{3} (1.43×10^{3}) +	1.7921×10^{2} (2.62×10^{2}) +	7.2088×10^{3} (4.48×10^{3}) =	4.8678×10^{3} (1.70×10^{3}) +	6.5649×10^{3} (6.32×10^{2})
LSMOP4	15	4.5910×10^{-2} (3.65×10^{-3}) -	8.5655×10^{-2} (4.73×10^{-2}) -	7.5265×10^{-2} (2.20×10^{-2}) -	1.1174×10^{-1} (5.08×10^{-2}) -	5.2622×10^{-2} (2.68×10^{-2}) -	5.5012×10^{-2} (4.37×10^{-3}) -	2.9969×10^{-2} (2.10×10^{-3})
LSMOP5	15	1.8984×10^{1} (8.51×10^{-1}) -	7.7651×10^{0} (6.27×10^{0}) -	6.7842×10^{0} (3.76×10^{0}) -	3.3211×10^{0} (1.24×10^{0}) -	1.5367×10^{1} (2.89×10^{0}) -	1.4577×10^{1} (6.60×10^{0}) -	5.4065×10^{-2} (4.49×10^{-3})
LSMOP6	15	1.5097×10^{5} (3.16×10^{4}) -	4.7408×10^{4} (3.21×10^{4}) -	2.4633×10^{4} (3.43×10^{4}) -	1.4922×10^{3} (9.28×10^{2}) -	1.0303×10^{5} (4.51×10^{4}) -	6.9810×10^{4} (6.09×10^{4}) -	2.6383×10^{-1} (4.00×10^{-2})
LSMOP7	15	1.7603×10^{5} (8.38×10^{3}) -	2.8682×10^{4} (3.37×10^{4}) -	4.7758×10^{4} (2.80×10^{4}) -	3.5873×10^{2} (1.56×10^{3}) +	1.2662×10^{5} (4.04×10^{4}) -	1.3996×10^{5} (3.37×10^{4}) -	4.4953×10^{3} (1.54×10^{3})
LSMOP8	15	9.4726×10^{0} (3.94×10^{-1}) -	4.8316×10^{0} (4.04×10^{0}) -	4.9044×10^{0} (1.80×10^{0}) -	3.9321×10^{-2} (3.39×10^{-2}) +	7.5267×10^{0} (1.27×10^{0}) -	1.0948×10^{1} (2.01×10^{0}) -	5.0486×10^{-1} (1.59×10^{-1})
LSMOP9	15	6.3391×10^{2} (6.14×10^{1}) -	1.3545×10^{3} (5.03×10^{2}) -	1.0780×10^{3} (3.77×10^{2}) -	2.6131×10^{3} (3.67×10^{2}) -	6.8321×10^{2} (3.96×10^{1}) -	7.9693×10^{2} (2.35×10^{2}) -	1.7417×10^{-1} (1.16×10^{-2})

For further intuitive understanding, Figures 4–6, respectively, show the distribution of the optimal solution set of LSMOP1/5/9 with the objective number of 15 for each algorithm. For PFs of LSMOPs, LSMOP1-4 have linear PFs, LSAMOP5-8 have convex PFs, and LSMOP9 has discontinuous PFs. Therefore, the results of the optimal solution set obtained on LSMOP1/5/9 are selected to evaluate the performance of the proposed algorithm on different PFs. LSMOP1 has linear PFs. Figure 4 shows the optimal solution set obtained by each algorithm running on the 15-objective LSMOP1, according to the same function evaluation values (FEs). Among them, NSGA-II/SDR has the best performance, converges to PF on each objective, and maintains good diversity. NSGA-II/SDR-OLS takes second place, with good diversity, but poor convergence to PF, which is also reflected in the data above. The diversity of other algorithms is poor, and the effect of convergence to PF is not good.

Figure 5 shows the optimal solution set obtained by each algorithm running on 15-objective LSMOP5, according to the same FEs. LSMOP5 has convex PFs. Among them, NSGA-II/SDR-OLS has the best performance, converging to the PF on each target and maintaining good diversity. The diversity of PREA, NSGA-II-conflict and NSGA-III is good, but the effect of convergence to the PF is not good. The diversity of other algorithms is poor, and the effect of convergence to the PF is also poor, and some even do not converge to the PF. This reflects the advantages of NSGA-II/SDR-OLS in solving convex PF.

Figure 6 shows the optimal solution set obtained by each algorithm running on 15-objective LSMOP9, according to the same FEs. LSMOP5 has discontinuous PFs. It can be seen from Figure 6 that all algorithms can successfully converge to the PF, but their distribution in the objective space is very different. Among them, the diversity of PREA, NSGA-II-conflict and NSGA-II/SDR-OLS is the best, and each of its objectives has well-maintained diversity. Other algorithms perform poorly in terms of diversity, and maintain poor diversity. In the above comparison, NSGA-II/SDR-OLS has achieved good performance, indicating that the added OBL and LS strategies are competitive in maintaining convergence and diversity.

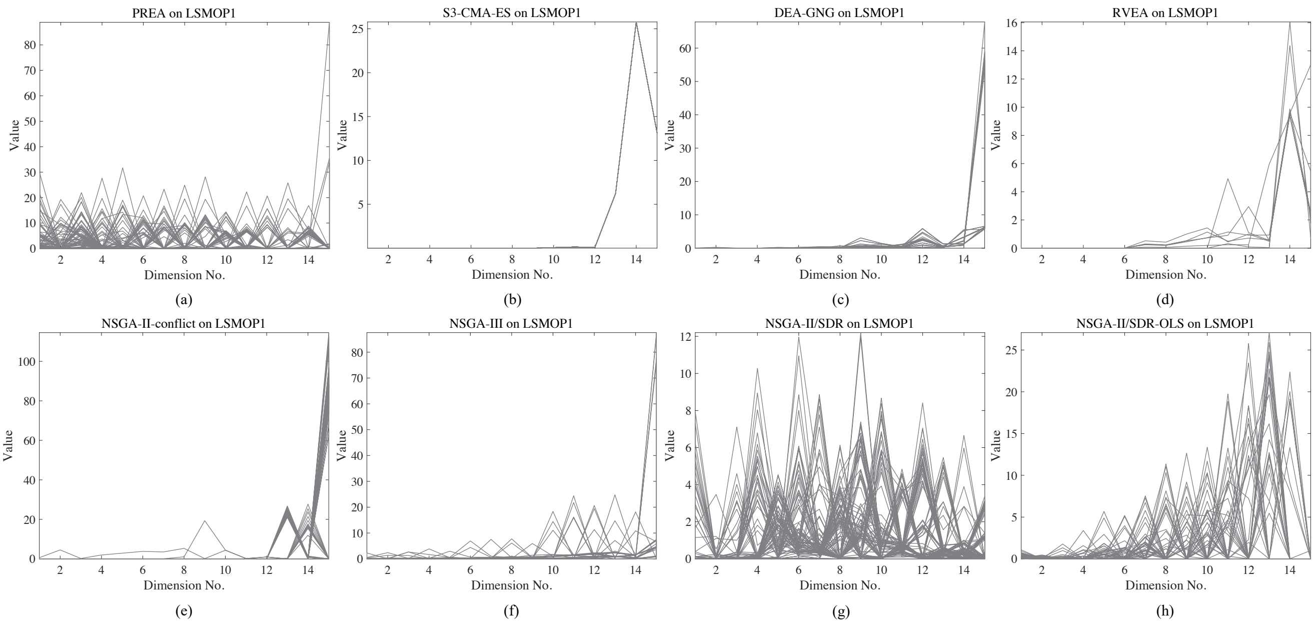

Figure 4. The final solution set obtained by NSGA-II/SDR-OLS and other seven algorithms on 15-objective LSMOP1. (**a**) The final solution set of PREA; (**b**) The final solution set of S3-CMA-ES; (**c**) The final solution set of DEA-GNG; (**d**) The final solution set of RVEA; (**e**) The final solution set of NSGA-II-conflict; (**f**) The final solution set of NSGA-III; (**g**) The final solution set of NSGA-II/SDR; (**h**) The final solution set of NSGA-II/SDR-OLS.

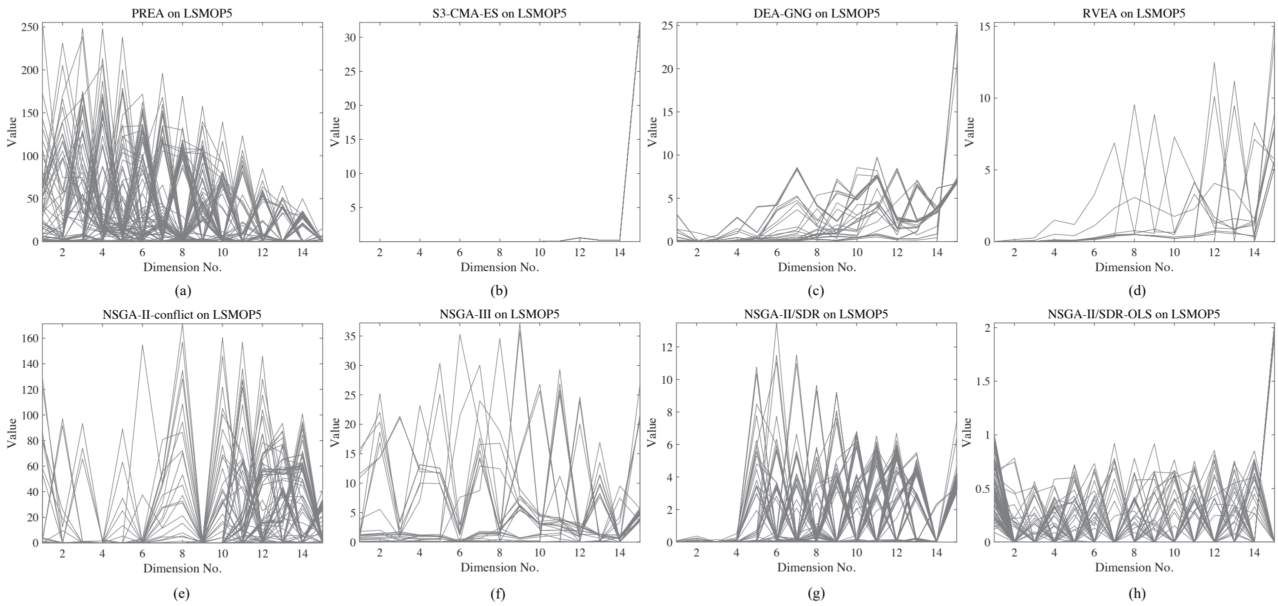

Figure 5. The final solution set obtained by NSGA-II/SDR-OLS and other seven algorithms on 15-objective LSMOP5. (**a**) The final solution set of PREA; (**b**) The final solution set of S3-CMA-ES; (**c**) The final solution set of DEA-GNG; (**d**) The final solution set of RVEA; (**e**) The final solution set of NSGA-II-conflict; (**f**) The final solution set of NSGA-III; (**g**) The final solution set of NSGA-II/SDR; (**h**) The final solution set of NSGA-II/SDR-OLS.

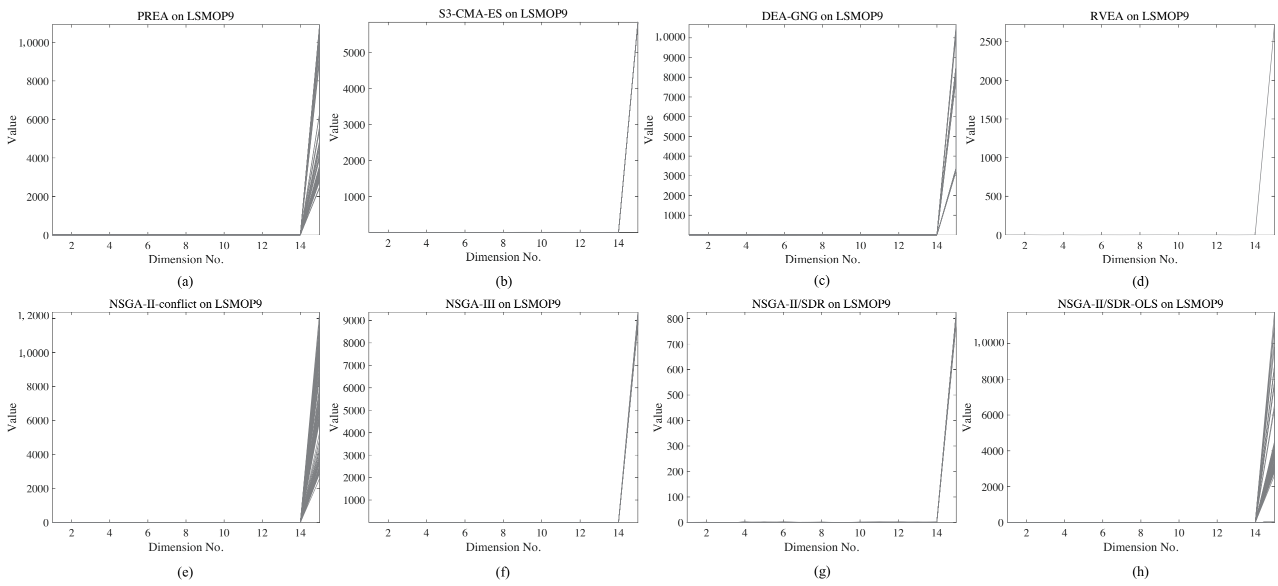

Figure 6. The final solution set obtained by NSGA-II/SDR-OLS and other seven algorithms on 15-objective LSMOP9. (**a**) The final solution set of PREA; (**b**) The final solution set of S3-CMA-ES; (**c**) The final solution set of DEA-GNG; (**d**) The final solution set of RVEA; (**e**) The final solution set of NSGA-II-conflict; (**f**) The final solution set of NSGA-III; (**g**) The final solution set of NSGA-II/SDR; (**h**) The final solution set of NSGA-II/SDR-OLS.

5.3.3. Discussion and Statistical Analysis

According to the above results, it can be easily seen that the performance of our algorithm is significantly better than that of the original algorithm and the other six comparison algorithms. The following is a detailed analysis. The IGD value reflects the convergence and diversity of the algorithm at the same time. In the comparison of the other seven algorithms, our algorithm achieved the best results in most instances of LSMOPs when the objective number changed from 3 to 15. It can be found that, except for LSMOP2 and LSMOP4, NSGA-II/SDR-OLS fully covers the lowest IGD values of LSMOP1/3/5/6/7/8/9 in test instances of 3, 5, 8, 10, 12, and 15 objectives, including linear, convex and disconnected PFs. Therefore, the added strategy further balances the convergence and diversity of the solution set, so that the algorithm can obtain better comprehensive performance when solving LSMOPs. In contrast, for LSMOP2 and LSMOP4, the original algorithm NSGA-II/SDR achieves better results than our improved algorithm, which may be because our strategy ignores the characteristics of such functional landscapes. However, RVEA achieved the best results for a large proportion of instances of LSMOP2 and LSMOP4. The reason may be the effectiveness of preference expression methods, based on reference vectors, in solving such problems.

In the comparative experiment with the original algorithm, another performance metric we use is DM. It can be found that on LSMOP1-LSMOP4 with linear PFs, the diversity of NSGA-II/SDR is slightly better than that of NSGA-II/SDR-OLS, but NSGA-II/SDR-OLS achieves better results on most test instances of LSMOP5-LSMOP9. To some extent, it can be considered that NSGA-II/SDR is more suitable for solving linear PF problems than NSGA-II/SDR-OLS in terms of diversity, while NSGA-II/SDR-OLS can deal with a wider and more complex PF range, which can be explained by the hybrid characteristics of PF. The sampling points of ideal PFs in the proposed algorithm are mostly Pareto optimal solutions.

In the comparison experiment with the other six algorithms, another performance metric we use is the convergence metric GD. According to the experimental results, our NSGA-II/SDR-OLS achieved poor performance on the three-objective test problems, achieving three optimal values in nine test instances, which was the same as PREA. However, with the increase of the number of objectives, the GD values became significantly better, until achieving three optimal values in nine test instances, which significantly outperformed other comparison algorithms on 15-objective test problems. This indicates that the convergence of our algorithm is enhanced as the number of objectives increases. Thus, our algorithm can accommodate most LSMOPs in high-dimensional space.

To further demonstrate the excellent overall performance of NSGA-II/SDR-OLS while preventing unnecessary errors, the Friedman ranking test was used to analyze the metric datasets. In this test, the mean and standard deviation (Std) values are considered separately to check the differences between all comparison algorithms, and the statistical results are presented. The purpose of statistical testing is to verify whether there are statistically significant differences between the proposed algorithm and other comparison algorithms. All non-parametric tests were conducted on SPSS 26.

Table 18 shows the ranking of the Friedman test on IGD values for 15 objectives, which reflects the overall performance of NSGA-II/SDR-OLS. The reason for choosing this dataset is that, firstly, the IGD values can comprehensively reflect the overall performance of the algorithm, and, secondly, the performance under 15 objectives better reflects the algorithm's performance in terms of solving large-scale problems. Firstly, from the perspective of the average value, the algorithm is arranged in ascending order of rank, as NSGA-II/SDR-OLS, NSGA-II/SDR, RVEA, NSGA-III, PREA, NSGA-II-conflict, DEA-GNG, and S3-CMA-ES. Secondly, from the perspective of variance, the ascending order of rank is NSGA-II/SDR-OLS, NSGA-II/SDR, NSGA-III, PREA, NSGA-II-conflict, RVEA, DEA-GNG, and S3-CMA-ES. It can be noted that from any perspective, NSGA-II/SDR-OLS always ranks first.

Table 18. The ranking of the Friedman test.

	PREA	S3-CMA-ES	DEA-GNG	RVEA	NSGA-II-Conflict	NSGA-III	NSGA-II/SDR	NSGA-II/SDR-OLS
Friedman rank (Mean)	4.67	8.00	6.33	3.78	5.33	4.22	2.00	1.67
Final rank (Mean)	5	8	7	3	6	4	2	1
Friedman rank (Std)	4.44	7.89	6.22	4.67	4.56	4.11	2.89	1.22
Final rank (Std)	4	8	7	6	5	3	2	1

In terms of non-parametric statistics significance, because the confidence level is 95%, all Friedman rank test results are subject to χ^2 distribution with seven degrees-of-freedom, and the p-values of both rank tests are lower than the given confidence level of 0.05. This indicates a significant difference between the samples participating in the test, which further confirms the significant difference between NSGA-II/SDR-OLS and other comparison algorithms. The above results all indicate that our improvement is meaningful.

6. Conclusions

In order to further improve the performance of the algorithm in solving large-scale MaOPs, this paper proposed the NSGA-II/SDR-OLS based on NSGA-II/SDR, combining the opposition-based learning strategy and the local search strategy. Firstly, an opposition-based learning strategy was utilized to update the initial population and enhance its quality. Secondly, a local search strategy was incorporated during the population-updating process to prevent the current optimal solution from being trapped in a local optimum and to allow it to explore the objective space further. The combination of the two strategies effectively balanced the convergence and diversity of the population.

To verify the performance, NSGA-II/SDR-OLS was compared with the original NSGA-II/SDR model and six other existing algorithms (PREA, S3-CMA-ES, DEA-GNG, RVEA, NSGA-II-conflict, and NSGA-III). The experimental results showed that the two strategies added in this paper did improve the performance of the original NSGA-II/SDR in solving large-scale MaOPs, and also had strong competitiveness in other comparative algorithms. While ensuring operational efficiency and time, it effectively balances the convergence and diversity of the solution set. In addition, statistical analysis shows that NSGA-II/SDR-OLS has significant differences compared to other algorithms.

In the future, research will be carried out from the following aspects:

1. From the experimental results, we can see that NSGA-II/SDR-OLS does not perform well in solving some problems of linear PFs. In the future, we will conduct more in-depth research on it and try to introduce new effectiveness strategies to further enhance the performance of the algorithm;
2. In this paper, the verification was conducted on a problem set with a maximum number of objectives of 15 and a maximum number of decision variables of 1500. In the future, we can improve the applicability of the algorithm by evaluating the test problem set with more objective numbers and more decision variables;
3. With the rapid development of the Internet and big data, machine learning and deep learning technologies are developing day by day. Now, many researchers have been committed to finding effective strategies in the field of machine learning and deep learning, as well as to introducing them into multi-objective evolutionary algorithms to improve their performance. We can also work in this direction in the future. In addition, by applying the algorithm to solve large-scale problems in the real world, such as hyperparameter optimization of the model, which meets the characteristics of MaOEAs due to its numerous parameters, such applications can further prove the effectiveness of the algorithm, as well as demonstrate its practical significance.

Author Contributions: Conceptualization, Y.Z.; Methodology, G.W.; Software, G.W.; Validation, Y.Z.; Investigation, Y.Z.; Resources, G.W.; Data curation, Y.Z.; Writing—original draft, Y.Z.; Writing—review & editing, G.W. and H.W.; Supervision, G.W. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fundamental Research Funds for the Central Universities.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krishna, R.; Tang, C.; Sullivan, K.; Ray, B. Conex: Efficient exploration of big-data system configurations for better performance. *IEEE Trans. Softw. Eng.* **2022**, *48*, 893–909. [CrossRef]
2. Cheng, S.; Ma, L.B.; Lu, H.; Lei, X.J.; Shi, Y.H. Evolutionary computation for solving search-based data analytics problems. *Artif. Intell. Rev.* **2021**, *54*, 1321–1348. [CrossRef]
3. Guo, G.H.; Wu, N.J.; Liu, L.Y. Low power application specific soc chip for uncooled infrared image processing. *J. Infrared Millim. Waves* **2023**, *42*, 122–131.
4. Sreedhara, S.; Sorensen, T.J.; Poursaee, A.; Ross, B.E. Practical application of digital image processing in measuring concrete crack widths in field studies. *Pract. Period. Struct. Des. Constr.* **2023**, *28*, 05022006. [CrossRef]
5. Hu, Y.; Zhang, Y.; Gong, D.W. Multiobjective particle swarm optimization for feature selection with fuzzy cost. *IEEE Trans. Cybern.* **2021**, *51*, 874–888. [CrossRef] [PubMed]
6. Song, X.F.; Zhang, Y.; Gong, D.W.; Gao, X.Z. A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Trans. Cybern.* **2022**, *52*, 9573–9586. [CrossRef] [PubMed]
7. Tahmasebi, S.; Moradi, P.; Ghodsi, S.; Abdollahpouri, A. An ideal point based many-objective optimization for community detection of complex networks. *Inf. Sci.* **2019**, *502*, 125–145. [CrossRef]
8. Cui, Z.H.; Zhang, J.J.; Wu, D.; Cai, X.J.; Wang, H.; Zhang, W.S.; Chen, J.J. Practical application of digital image processing in measuring concrete crack widths in field studies. *Inf. Sci.* **2020**, *518*, 256–271. [CrossRef]
9. El-Shorbagy, M.A.; El-Refaey, A.M. A hybrid genetic–firefly algorithm for engineering design problems. *J. Comput. Des. Eng.* **2022**, *9*, 706–730.
10. Xi, B.J.; Lei, D.M. Q-learning-based teaching-learning optimization for distributed two-stage hybrid flow shop scheduling with fuzzy processing time. *Complex Syst. Model. Simul.* **2022**, *2*, 113–129. [CrossRef]
11. Fu, Y.P.; Hou, Y.S.; Wang, Z.F.; Wu, X.W.; Gao, K.Z.; Wang, L. Distributed scheduling problems in intelligent manufacturing systems. *Tsinghua Sci. Technol.* **2021**, *26*, 625–645. [CrossRef]
12. Zheng, Y.; Wang, Y.Q.; Liu, J.X. Research on structure optimization and motion characteristics of wearable medical robotics based on improved particle swarm optimization algorithm. *Future Gener. Comput. Syst.* **2022**, *129*, 187–198. [CrossRef]
13. Shang, K.; Ishibuchi, H.; Chen, W.Y.; Nan, Y.; Liao, W.D. Hypervolume-optimal μ-distributions on line/plane-based Pareto fronts in three dimensions. *IEEE Trans. Evol. Comput.* **2022**, *26*, 349–363. [CrossRef]
14. Tan, H.Q.; Du, M.Q.; Chen, A. Accelerating the gradient projection algorithm for solving the non-additive traffic equilibrium problem with the barzilai-borwein step size. *Comput. Oper. Res.* **2022**, *141*, 105723. [CrossRef]
15. Zhou, A.M.; Qu, B.Y.; Li, H.; Zhao, S.Z.; Suganthan, P.N.; Zhang, Q.F. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm Evol. Comput.* **2011**, *1*, 32–49. [CrossRef]
16. Srinivas, N.; Deb, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
17. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
18. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [CrossRef]
19. Jain, H.; Deb, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **2014**, *18*, 602–622. [CrossRef]
20. Zhang, Q.F.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
21. Li, K.; Deb, K.; Zhang, Q.; Kwong, S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **2015**, *19*, 694–716. [CrossRef]
22. Liu, R.C.; Wang, R.N.; Bian, R.Y.; Liu, J.; Jiao, L.C. A decomposition-based evolutionary algorithm with correlative selection mechanism for many-objective optimization. *Evol. Comput.* **2021**, *29*, 269–304. [CrossRef] [PubMed]
23. Tian, Y.; Cheng, R.; Zhang, X.Y.; Su, Y.S.; Jin, Y.C. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 331–345. [CrossRef]
24. Cheng, R.; Jin, Y.C.; Olhofer, M.; Sendhoff, B. Test problems for large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2017**, *47*, 4108–4121. [CrossRef]
25. Yuan, J.W.; Liu, H.L.; Gu, F.Q.; Zhang, Q.F.; He, Z.S. Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions. *IEEE Trans. Evol. Comput.* **2021**, *25*, 75–86. [CrossRef]
26. Chen, H.; Cheng, R.; Wen, J.; Li, H.; Weng, J. Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. *Inf. Sci.* **2020**, *509*, 457–469. [CrossRef]
27. Liu, Y.P.; Ishibuchi, H.; Masuyama, N.; Nojima, Y. Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts. *IEEE Trans. Evol. Comput.* **2020**, *24*, 439–453. [CrossRef]
28. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 773–791. [CrossRef]

29. Jaimes, A.L.; Coello, C.A.C.; Aguirre, H.; Tanaka, K. Objective space partitioning using conflict information for solving many-objective problems. *Inf. Sci.* **2014**, *268*, 305–327. [CrossRef]
30. Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine learning. *Addison-Wesley* **1989**, *1989*, 36.
31. Yuan, Y.; Xu, H.; Wang, B.; Yao, X. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 16–37. [CrossRef]
32. Fabre, M.G.; Pulido, G.T.; Coello, C.A.C. Alternative fitness assignment methods for many-objective optimization problems. In Proceedings of the 9th International Conference on Evolution Artificial, Strasbourg, France, 26–28 October 2009; pp. 146–157.
33. Zhang, X.Y.; Tian, Y.; Jin, Y. C. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 761–776. [CrossRef]
34. Ishibuchi, H.; Tsukamoto, N.; Nojima, Y. Evolutionary many-objective optimization: A short review. In Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 2419–2426.
35. Laumanns, M.; Thiele, L.; Deb, K.; Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.* **2002**, *10*, 263–282. [CrossRef]
36. McClymont, K.; Keedwell, E. Deductive sort and climbing sort: New methods for non-dominated sorting. *Evol. Comput.* **2012**, *20*, 1–26. [CrossRef]
37. Li, M.; Yang, S.; Liu, X. Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *18*, 348–365. [CrossRef]
38. Elarbi, M.; Bechikh, S.; Gupta, A.; Ben Said, L.; Ong, Y.S. A new decomposition-based NSGA-II for many-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1191–1210. [CrossRef]
39. Pan, L.Q.; Xu, W.T.; Li, L.H.; He, C.; Cheng, R. Adaptive simulated binary crossover for rotated multi-objective optimization. *Swarm Evol. Comput.* **2021**, *60*, 100759. [CrossRef]
40. Zhao, L.D.; Wang, B.; Jiang, X.P.; Lu, Y.C.; Hu, Y.H. DIP-MOEA: A double-grid interactive preference based multi-objective evolutionary algorithm for formalizing preferences of decision makers. *Front. Inf. Technol. Electron. Eng.* **2022**, *23*, 1714–1732. [CrossRef]
41. Zheng, J.H.; Du, Z.F.; Zou, J.; Yang, S.X. A weight vector generation method based on normal distribution for preference-based multi-objective optimization. *Swarm Evol. Comput.* **2023**, *77*, 101250. [CrossRef]
42. Ben Said, L.; Bechikh, S.; Ghedira, K. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Trans. Evol. Comput.* **2010**, *14*, 801–818. [CrossRef]
43. Wang, F.; Li, Y.X.; Zhang, H.; Hu, T.; Shen, X.L. An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm Evol. Comput.* **2019**, *49*, 220–233. [CrossRef]
44. He, Z.; Yen, G.G. Many-objective evolutionary algorithms based on coordinated selection strategy. *IEEE Trans. Evol. Comput.* **2017**, *21*, 220–233. [CrossRef]
45. Gong, D.; Sun, F.; Sun, J.; Sun, X. Set-based many-objective optimization guided by a preferred region. *Neurocomputing* **2017**, *228*, 241–255. [CrossRef]
46. Hou, Z.; He, C.; Cheng, R. Reformulating preferences into constraints for evolutionary multi- and many-objective optimization. *Inf. Sci.* **2020**, *514*, 1–15. [CrossRef]
47. Ke, L.J.; Zhang, Q.F.; Battiti, R. MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony. *IEEE Trans. Cybern.* **2013**, *43*, 1845–1859. [CrossRef] [PubMed]
48. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern.* **1996**, *26*, 29–41. [CrossRef]
49. Jiao, R.W.; Zeng, S.Y.; Li, C.H.; Ong, Y.S. Two-type weight adjustments in MOEA/D for highly constrained many-objective optimization. *Inf. Sci.* **2021**, *578*, 592–614. [CrossRef]
50. Peng, H.; Wang, C.; Han, Y.P.; Xiao, W.H.; Zhou, X.Y.; Wu, Z.J. Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization. *Future Gener. Comput. Syst.* **2022**, *131*, 59–74. [CrossRef]
51. Xu, H.; Zeng, W.H.; Zhang, D.F.; Zeng, X.X. MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition. *IEEE Trans. Cybern.* **2019**, *49*, 517–526. [CrossRef] [PubMed]
52. Zhang, Y.; Wang, G.G.; Li, K.; Yeh, W.C.; Jian, M.; Dong, J. Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. *Inf. Sci.* **2020**, *522*, 1–16. [CrossRef]
53. Jiang, S.; Ong, Y.S.; Zhang, J.; Feng, L. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Trans. Cybern.* **2014**, *44*, 2391–2404. [CrossRef]
54. Zitzler, E.; Kunzli, S. Indicator-based selection in multiobjective search. In Proceedings of the Parallel Problem Solving from Nature, Birmingham, UK, 18–22 September 2004; pp. 832–842.
55. Bader, J.; Zitzler, E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **2011**, *19*, 45–76. [CrossRef]
56. Liu, Y.; Gong, D.; Sun, J.; Jin, Y. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **2017**, *47*, 2689–2702. [CrossRef]
57. Cai, X.; Sun, H.; Fan, Z. A diversity indicator based on reference vectors for many-objective optimization. *Inf. Sci.* **2018**, *430–431*, 467–486. [CrossRef]

58. Sun, Y.; Yen, G.G.; Yi, Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2018**, *23*, 173–187. [CrossRef]
59. Liang, Z.P.; Hu, K.F.; Ma, X.L.; Zhu, Z.X. A many-objective evolutionary algorithm based on a two-round selection strategy. *IEEE Trans. Cybern.* **2021**, *51*, 1417–1429. [CrossRef]
60. Ma, X.L.; Liu, F.; Qi, Y.T.; Wang, X.D.; Li, L.L.; Jiao, L.C.; Yin, M.L.; Gong, M.G. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Trans. Evol. Comput.* **2016**, *20*, 275–298. [CrossRef]
61. Mandavi, S.; Shiri, M.E.; Rahnamayan, S. Metaheuristics in large-scale global continues optimization: A survey. *Inf. Sci.* **2015**, *295*, 407–428. [CrossRef]
62. Ma, L.B.; Huang, M.; Yang, S.X.; Wang, R.; Wang, X. W. An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2022**, *52*, 6684–6696. [CrossRef] [PubMed]
63. Wang, F.; Wang, X.J.; Sun, S.L. A Reinforcement Learning Level-based Particle Swarm Optimization Algorithm for Large-scale Optimization. *Inf. Sci.* **2022**, *602*, 298–312. [CrossRef]
64. Gu, Z.M.; Wang, G.G. Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization. *Future Gener. Comput. Syst.* **2020**, *107*, 49–69. [CrossRef]
65. Xiang, Y.; Zhou, Y.; Li, M.; Chen, Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Trans. Evol. Comput.* **2017**, *21*, 131–152. [CrossRef]
66. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, Vienna, Austria, 28–30 November 2005; pp. 695–701.
67. Chen, B.L.; Zeng, W.H.; Lin, Y.B.; Zhang, D.F. A new local search-based multiobjective optimization algorithm. *IEEE Trans. Evol. Comput.* **2015**, *19*, 50–73. [CrossRef]
68. Gao, S.C.; Yu, Y.; Wang, Y.R.; Wang, J.H.; Cheng, J.J.; Zhou, M. C. Chaotic local search-based differential evolution algorithms for optimization. *IEEE Trans. Syst. Man Cybern.* **2021**, *51*, 3954–3967. [CrossRef]
69. Wang, Y.; Cai, S.; Chen, J.; Yin, M. SCCWalk: An efficient local search algorithm and its improvements for maximum weight clique problem. *Artif. Intell.* **2020**, *280*, 103230. [CrossRef]
70. Deb, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evol. Comput.* **1999**, *7*, 205–230. [CrossRef] [PubMed]
71. Menchaca-Mendez, A.; Coello, C.A.C. GDE-MOEA: A new MOEA based on the generational distance indicator and e-dominance. In Proceedings of the IEEE Congress on Evolutionary Computation, Sendai, Japan, 25–28 May 2015; pp. 947–955.
72. Tian, Y.; Zhang, X.Y.; Cheng, R.; Jin, Y.C. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016; pp. 5222–5229.
73. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Reference point specification in inverted generational distance for triangular linear pareto front. *IEEE Trans. Evol. Comput.* **2018**, *22*, 961–975. [CrossRef]
74. Deb, K.; Jain, S. Running performance metrics for evolutionary multi-objective optimization. In Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 18–22 November 2002; pp. 13–20.