J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 2549--2572

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on multi-objective optimization strategy and algorithm improvement based on genetic algorithm in large-scale computing environment

Bojun Liu[1, ✉]

[1] *Faculty of Faculty University of Sydney, Sydney, Australia*

ABSTRACT

The basic genetic algorithm suffers from problems such as precocity and low search efficiency when solving multi-objective optimization problems in large-scale computing environments. Aiming at these problems, this paper introduces various improvement strategies such as neighborhood operation, adaptive strategy, chaos optimization and cooling into the classical genetic algorithm, and designs an improved genetic algorithm process that organically combines various improvement strategies. The improved genetic algorithm and other existing large-scale multi-objective optimization algorithms are tested using LSMOP test problems, and the improved genetic algorithm has better convergence and diversity than other algorithms on both two-objective and three-objective LSMOP test problems. The PF curves of the seven algorithms are plotted separately for the two-objective on LSMOP6 and the three-objective on LSMOP5 when the decision variable is 200, and the images show that the improved genetic algorithm has the most uniform population distribution. The experimental results confirm the effectiveness of the improved genetic algorithm in solving large-scale multi-objective optimization problems.

Keywords: Genetic algorithm, Multi-objective optimization, Chaotic optimization, Large-scale computing environment

1. Introduction

With the continuous development of science and technology, many real-life problems have become more and more complex, requiring trade-offs and optimization among multiple objectives. Large-scale multi-objective optimization problem is one of the important problems. In real life, many problems often involve multiple objectives, such as resource allocation, path planning, product design and so on [1-4]. Sometimes there are conflicts and contradictions between these objectives, and it is necessary to make trade-offs and optimization between each objective, which forms a multi-objective

✉ Corresponding author.
E-mail address: xfyjrLBJ0628@163.com (B. Liu).
Received 15 April 2024; Revised 05 July 2024; Accepted 25 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-144

optimization problem. When the problem scale is huge, traditional optimization methods often cannot meet the requirements, so evolutionary algorithms are needed to solve such problems, especially genetic algorithms are widely used [5-8].

Genetic algorithm is a kind of optimization algorithm that simulates the natural genetic and evolutionary process, which simulates the process of biological evolution through gene coding, selection, crossover and mutation, so as to find the optimal solution [9-10]. In solving large-scale multi-objective optimization problems, genetic algorithms are able to transform the problem into a single-objective optimization problem by introducing fitness functions and multi-objective optimization algorithms, which are then solved using genetic algorithms. Then hybrid genetic algorithm is used, which improves the performance of the algorithm by incorporating the advantages of multiple genetic algorithms [11-14]. However, genetic algorithms also have some shortcomings. For example, the traditional genetic algorithm lacks diversity and is not sufficient in the exploration of the solution space, while it does not take into account factors such as the weighting relationships and constraints of the objectives. Therefore, it is important to study the improved genetic algorithm model for solving multi-objective optimization problems [15-18].

For multi-objective optimization in large-scale computing environment, this paper improves the classical genetic algorithm and designs an improved genetic algorithm process that organically combines various improvement strategies. One, the neighborhood operation is carried out by simulated annealing algorithm and dynamic programming to improve the local search efficiency. Secondly, the adaptive strategy is utilized to overcome the early maturity and degradation phenomenon of the classical genetic algorithm. Third, the planning mechanism to avoid the search falling into local extremes through chaotic optimization method. The improved genetic algorithm is evaluated using the LSMOP test problem and other existing large-scale multi-objective optimization algorithms, and the convergence and diversity of the algorithm is measured by the Inverse Generation Distance (IGD) and Hypervolume (HV) metrics.

2. Multi-objective optimization in large-scale computing environments

2.1. Mathematical description of multi-objective optimization problems

A general multi-objective optimization problem consists of a set of objective functions and associated constraints on some equations and inequalities, and can be described mathematically as follows:

Minimize/Maximize $f_m(x), m-1,2,LM$ such that:

$$\begin{cases} g_i(x) \geq 0, j=1,2,L,J; \\ h_k(x)=0, k=1,2,L,K; \\ x_i^{(L)} \leq x_i \leq x_i^{(U)}, i=1,2,L,V; \end{cases} \tag{1}$$

Parameter x is a V -dimensional vector with V decision variables, and the solution of a multi-objective optimization problem is mathematically formulated as a non-inferior or optimal solution (optimal in the Pareto sense).

In the presence of multiple Pareto optimal solutions, without further information about the problem, it is difficult to say which solution is preferable and all Pareto optimal solutions are considered equally important. Therefore, given the ideal method, it is important to find as many Pareto optimal solutions about the problem as possible. Thus, it can be inferred that there are two objectives in multi-objective optimization:

1) Find a set of solutions that are as close as possible to the Pareto optimal domain.

2) Find a set of solutions that are as different as possible.

The first objective is necessary in any optimization effort. Convergence to a set of solutions that are not close to the true Pareto optimal set of solutions is undesirable. Only if the solutions converge to an approximation of the true optimal solution can their near-optimal properties be guaranteed. In addition to converging to the near-optimal domain, the solutions obtained must also be sparsely distributed in the Pareto optimal domain. Only having a diverse set of solutions ensures that there is a good set of agreement solutions across multiple objectives. Since multi-objective evolutionary algorithms deal with two spaces a decision variable space and an objective space, the diversity between solutions (individuals) can be defined in both spaces [19]. For example, two individuals are said to be mutually diverse in the decision variable space if their Eulerian distance is large in the decision variable space. Similarly, if the Euler distance between two individuals is large in the goal space, then they are said to be mutually dissimilar in the goal space. Although for most problems diversity in one space often implies diversity in another, this does not hold for all problems. For such complex nonlinear problems, it is also an important task to find a set of solutions that have a good distribution of diversity in the required space.

2.2. Description of large-scale multi-objective optimization problem

A mathematical description of a multi-objective optimization problem MOPs with M objective function and D-dimensional decision variables is as follows:

$$\begin{aligned} &\min F(X) = (f_1(X), f_2(X), \ldots, f_M(X)) \\ &s.t.\ X = (x_1, x_2, \ldots, x_D) \in R^D \end{aligned} \tag{2}$$

Where: X is the position vector of the individual and $f_M(X)$ is the M rd objective value of the individual. When $D \geq 100, M \geq 2$, it is a large-scale multi-objective optimization problem.

3. Improved genetic algorithm

3.1. Classical genetic algorithms

Genetic algorithm is an adaptive global optimization search No. 1 method that simulates the process of inheritance and evolution of organisms in nature [20]. Drawing on the genetic process of individuals in populations in nature, the fitness of the offspring individuals is improved through mechanisms such as individual selection, gene crossover and gene mutation. Genetic algorithms operate with the coding of decision variables, the chromosomes of individuals correspond to the feasible solutions of the problem v, and the population constitutes the entire set of feasible solutions Ω. In this research problem, the objective function can be interpreted as the fitness function in genetic algorithms $f(v)$. The general process of genetic algorithms can be summarized in the following six steps, and the corresponding algorithmic flow is shown in Fig. 1.

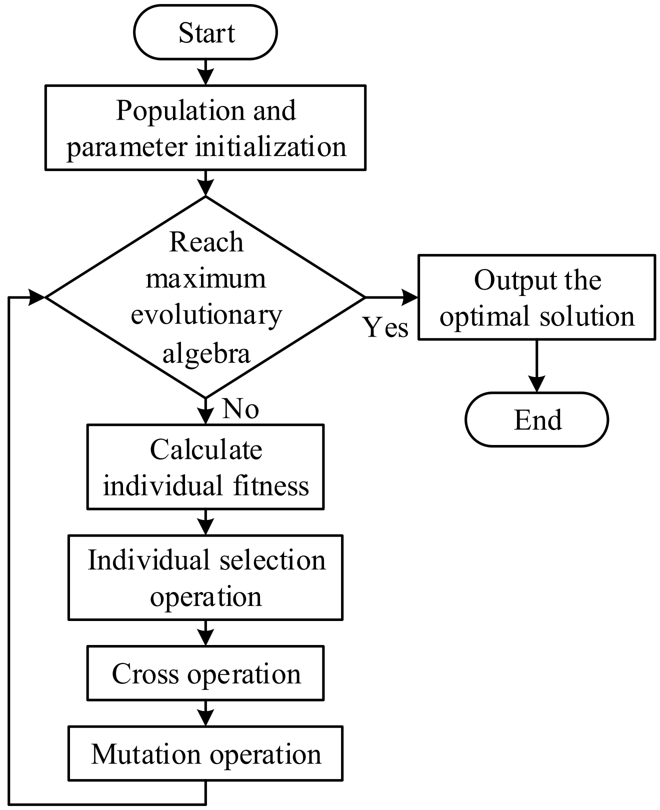


Fig. 1. Genetic algorithm process

1) Population initialization. Set the evolutionary generation counter t, maximum evolutionary generation T, and randomly generate the initial population.
2) Individual evaluation. Evaluate all individuals according to the fitness function $f(v)$.
3) Individual selection. Apply the selection operator to the individuals of the existing population, and retain the individuals carrying good genes to the next generation.
4) Crossover operation. Select individuals in pairs from the existing population based on the crossover probability P_c, and use the crossover subselection genes for crossover operation to generate new individuals.
5) Mutation operation. Based on the mutation probability P_m select individuals from the existing population and use the mutation operator to select genes of individuals for mutation operation and mutate into alleles with other coding values.
6) Terminate evolutionary judgment. Update the evolutionary generation counter t and judge whether the maximum evolutionary generation is reached T. If $t \le T$, repeat steps (2) to (5). If $t > T$, terminate the evolution. The most adapted individual in the population is the optimal solution.

3.2. Improvement strategies for genetic algorithms

3.2.1. Introducing a Neighborhood Operation Strategy. If the population size is limited, it is difficult to converge to the optimal solution through normal genetic operations. And if the whole solution space is covered once, it will increase the computation time and complexity. Therefore, local optimization of the neighborhood of each chromosome in each generation of the population can improve the local solving ability of the algorithm, increase the convergence speed and prevent premature maturity. Two types of neighborhood operations are introduced here:
1) Updating of genes in a chromosome (Neighborhood I). Arbitrarily select n (n<lchrom, lchrom is the length of the individual chromosome) genes in the chromosome, and randomly introduce new genes from the neighborhood of each gene to form a new chromosome. Such a transformation is performed

on each chromosome to constitute a new population.

Drawing on the idea of simulated annealing algorithms, starting from the neighborhood of each gene to accept the probability:

$$A_{ij}(t_k) = \min\left\{1, \exp\left(-\frac{f(j)-f(i)}{t_k}\right)\right\} \tag{3}$$

The new gene is selected. Where $f(i)$ is the value of the objective function for the current gene i and the algorithm accepts or rejects the gene j according to A_{ij}. t_k is the temperature parameter for simulated annealing. The gene neighborhood is a subset of the solution space of width m centered on the current gene. m gets progressively smaller as the convergence rate slows down, thus the search for the neighborhood is performed from coarse to fine.

2) Alignment of genes in chromosomes (Neighborhood II). The optimization of the arrangement of each chromosome is performed to obtain the optimal solution with respect to all genes, thus making the population an optimal combination of chromosomes. This type of operation can be performed using dynamic programming algorithms, which are fast and accurate.

3.2.2. Introducing Adaptive Strategies. During the evolutionary process, crossover operator p_c and mutation operator p_m are performed thousands of times. Genetic algorithms generally use fixed values of $p_c(0.5\sim0.9)$ and $p_m(0.001\sim0.2)$. Simply increasing the values of p_c and p_m enhances the global search capability of the genetic algorithm, but at the same time reduces its local optimization capability. Introducing an adaptive strategy for the crossover and mutation operators can balance the global search and local optimization, and also dynamically control the frequency of genetic operations to improve the adaptability of the search to spatial variations [21].

The adaptive strategy is based on the following two ideas:

1) The magnitude of the crossover, variance probability is based on the state of the population. When the population tends to be homogeneous, it is expected to jump out of the local optimum by increasing the values of p_c and p_m. And when the population is more dispersed, reduce the values of p_c and p_m.

2) Treat different chromosomes with different crossover and mutation probabilities, and adaptively change the sizes of p_c and p_m according to their fitness values. An operator that introduces an adaptive strategy is called an adaptive operator.

There are various kinds of adaptive operators. One is given here:

$$p_c = \begin{cases} \dfrac{k_1(f_{\max}-f')}{f_{\max}-f}, & f' \geq f \\ k_3 & f' < f \end{cases} \tag{4}$$

$$p_m = \begin{cases} \dfrac{k_2(f_{\max}-f)}{f_{\max}-f}, & f \geq f \\ k_4 & f' < f \end{cases} \tag{5}$$

Where: $f_{\max}$ denotes the fitness value of the optimal chromosome in the current population. f denotes the average fitness value of the current population. f' denotes the fitness value of the chromosome with the smaller fitness value among the two chromosomes to be crossed. General $k_1, k_2 = (0.6\sim0.95)$, $k_3, k_4 = (0.005\sim0.01)$.

3.2.3. Introducing Chaotic Optimization Strategies. Chaos has rich spatio-temporal dynamics, and the traversal characteristics of chaos can be used as an optimization mechanism to avoid falling into local extremes during the search process. Therefore, chaos has become a novel optimization technique, which has received more and more attention in the combinatorial optimization neighborhood.

In this paper, we consider the organic combination of chaotic optimization technology and genetic algorithm, the basic idea is: firstly, use the genetic algorithm to carry out iterative operations, and then carry out chaotic optimization on the optimal chromosome X^* in the generated new solution group, and generate a new chromosome X' after optimization. If the fitness value is $f(X') > f(X^*)$, replace X^* with X', and carry out the next generation of genetic iterative operations.

Among them, the chaotic optimization technique is embodied in the control operation of the chaotic sequence on the genetic operator:

1) Controlling the frequency of crossover and mutation operations with chaotic sequences. Using some kind of chaotic mapping, for any initial value x_0, an iterative sequence can be generated which is characterized by varying back and forth between [0, 1]. Obviously, x_{n+1} can be used as a random switch to perform a crossover, mutation operation when x_{n+1} is greater than a pre-selected value p (typically taken as 0.5). Conversely, no operation is performed. To wit:

$$\begin{array}{rll} IF(x_{n+1} \geq p) & TNEN & operator = ON \\ & ELSE & Operator = OFF \\ IF(operator = ON) & THEN & (operation) \end{array} \tag{6}$$

2) Control the selection of crossover and mutation points with chaotic sequences. The resulting chaotic sequence x_{n+1} is mapped to the gene position space of the chromosome, and crossover and mutation operations are performed at the corresponding positions. If, for example, if $x_{n+1} \in [0, 1.0]$, the gene position of the operation is:

$$site = \lfloor L \cdot x_{n+1} \rfloor \tag{7}$$

Where: L denotes chromosome length. $\lfloor x \rfloor$ denotes rounding to x.

After this transformation, the chaotic sequence can be fully utilized in the process of genetic evolution to traverse all the states within a certain range without repetition, so that the crossover and mutation operations have an intrinsic regularity, which overcomes the shortcomings caused by pure randomness in the genetic algorithm, and not only improves the local search capability of the genetic algorithm, but also accelerates the convergence speed of the genetic algorithm.

3.2.4. Introduction of cooling strategies. Genetic algorithms generally set the maximum number of iterations as the termination condition in advance based on experience. The disadvantage of this is that setting too many iteration generations will waste computing time, and too few will not guarantee the search for the global optimal solution, and the pre-determined maximum number of iteration generations can not be adjusted according to the actual situation of the search, which reduces the efficiency of the search.

The simulated annealing algorithm can learn from the strategy of using initial temperature and cooling in the iteration process. The initial temperature t_0 is selected to ensure that the transfer probability of each state in the smooth distribution is equal, even though:

$$A_{ij} = \exp\left(-\frac{\Delta f_{ij}}{t_0}\right) \approx 1 \tag{8}$$

So a numerical estimation method is used to estimate t_0. i.e., an initial temperature t_0 is selected, and the initial value is obtained after several iterations based on the ratio R_k of the number of accepted states to the number of iteration steps at the same temperature, with constant adjustment of t_0. A t_0

numerical iterative calculation step is given in literature [22].

The cooling strategy can be used:

$$t_{k+1} = d(t_k) = \alpha t_k \quad 0 < \alpha < 1 \tag{9}$$

Control α can change the convergence speed of the iteration. Using the initial temperature and cooling strategy as the starting and termination conditions of the iteration can overcome the strong subjectivity of the control parameters of the genetic algorithm.

4. Improve the multi-objective optimization process of genetic algorithm

The multi-objective optimization process in a large-scale computing environment is shown below:

1) Selection. In this paper, we use random traversal selection method.

(2) Cross-reorganization. Discrete recombination of selected individual genes is randomly peformed due to real number coding of optimization parameters.

3) Variation. Within the range of each optimization variable, the coded variables are randomly selected and then real-valued mutation is performed.

4) Algorithm termination conditions. Determine the maximum number of evolutionary generations and terminate the computation when the maximum number of evolutionary generations is reached.

5. Large-scale multi-objective optimization experiment based on improved genetic algorithm

5.1. *Evaluation indicators*

The experiments use the Inverse Generation Distance (IGD) and Hypervolume (HV) metrics to measure the convergence and diversity of the algorithm.The smaller the value of IGD, the closer the Paretofront (PF) composed of the final solution set obtained by the algorithm is to the true PF.The larger the value of HV, the better is the comprehensive performance of the algorithm.The metrics are calculated using the following formulas:

$$IGD(S, Pop) = \frac{\sum_{i=1, s_i \in S}^{|S|} \min d(s_i, Pop)}{|S|} \tag{10}$$

$$HV = \bigcup_{j=1}^{N} \left\{ \sum_{p_j \in Pop} (p_j, Z) \right\} \tag{11}$$

where: S is the set of solutions on the real PF. pop is the last generation of the population. p_j is a solution of population Pop. N is the size of the population Pop. δ is the Leberger measure, which measures the volume of the hypercube formed by the individual p_j and the reference point Z.

5.2. *Experimental data sets*

The experimental dataset used this time is a benchmark problem for large-scale multi-objective optimization as shown in Table 1.The LSMOP test suite is very challenging in that it has multiple solutions that should converge simultaneously to different regions of the Pareto front. So it is often used as a test problem for large-scale multi-objective optimization as a way to measure the goodness of large-scale multi-objective optimization.

Table 1. LSMOP1-LSMOP9

Benchmark problem	Frontal feature	Landscape	Separability

		characteristics	
LSMOP1	Linearity	Single peak distribution	Separable
LSMOP2	Linearity	Mixed distribution	Partially separable
LSMOP3	Linearity	Multimodal distribution	Inseparable
LSMOP4	Linearity	Mixed distribution	Inseparable
LSMOP5	Concave	Single peak distribution	Separable
LSMOP6	Concave	Mixed distribution	Partially separable
LSMOP7	Concave	Multimodal distribution	Inseparable
LSMOP8	Concave	Mixed distribution	Inseparable
LSMOP9	Discontinuous	Mixed distribution	Separable

5.3. Experimental design

Compare the methodology of this paper with six state-of-the-art MOEAs, namely MOEA/D, GDE3, CMOPSO, DGEA, S3CMAES, and FDVCSO.It should be emphasized here that we chose these comparative algorithms because they involve different types of replication methodologies or large-scale decision-variable handling strategies.

5.4. Experimental platforms

The experiments in this paper were mainly done on the PlatEMO platform, which is based on the MTALAB multi-objective evolution platform and has the following main advantages: more than 100 open-source evolutionary algorithms, more than 200 open-source multi-objective test problems, a powerful graphical user interface for parallel execution of the experiments, the generation of the results in Excel or LaTeX tabular format through one-click operations, and the ability to continue to Includes state-of-the-art algorithms. Contains 50 multi-objective optimization algorithms and 110 multi-objective optimization problems.

5.5. Experimental results and analysis

Comparison of the experimental results with MOEA/D, GDE3, CMOPSO, DGEA, S3CMAES and FDVCSO are shown in Tables 2 to 5. The datasets used for the experiments are LSMOP1-LSMOP9, the number of targets are 2 and 3, and the number of decision variables are 100, 200, and 500.The variance values are in parentheses, based on the average of the results, and the better results from the compared algorithms are emphasized in bold form.

The experimental results of IGD on two-objective LSMOP are shown in Table 2. The statistical results show that the improved genetic algorithm has better convergence on the LSMOP1, 100-dimensional LSMOP2, LSMOP4, LSMOP5, LSMOP8, and LSMOP9 test problems.The CMOPSO algorithm performs optimally on the LSMOP3 test problem.The FDVCSO algorithm performs optimally on the 500-dimensional LSMOP7 test problem. The DGEA algorithm performs optimally on the 200- and 500-dimensional LSMOP2 and LSMOP6, and the 100- and 200-dimensional LSMOP7 test problems. The total number of test problems is 27, and the improved algorithm has 16 advantages on the average. It can be inferred that the algorithm has better convergence than the other compared algorithms on the two-objective LSMOP test problems.

Table 2. IGD values on LSMOP on the second target

Prob.	Dim.	MOEA/D	GDE3	CMOPSO	FDVCSO	S3CMAES	DGEA	Ours
LSMOP1	100	6.2E-4 (6.176E-4)	5.529E+0 (1.227E+1)	2.259E-4 (2.12E-4)	7.272E-4 (3.876E-4)	1.177E-4 (7.133E-4)	7.926E-4 (7.434E-4)	4.864E-5 (4.214E-5)
	200	5.141E-4 (6.265E-4)	8.851E-4 (5.528E-4)	8.82E-5 (7.625E-4)	8.168E-4 (6.368E-4)	5.805E-4 (2.734E-4)	1.25E-5 (4.327E-4)	5.399E-6 (6.764E-6)
	500	3.29E-4 (6.656E-4)	3.485E-4 (1.164E-4)	8.9E-5 (8.542E-4)	6.077E-4 (5.74E-4)	2.899E-4 (5.713E-4)	6.258E-4 (6.782E-4)	2.36E-6 (6.795E-5)
LSMOP2	100	6.525E-4 (1.21E-5)	4.869E-4 (3.776E-4)	6.759E-4 (4.853E-4)	8.184E-4 (4.674E-4)	1.207E-4 (2.08E-4)	5.723E-4 (3.776E-4)	5.353E-6 (4.03E-5)
	200	3.771E-4 (7.204E-4)	2.395E-4 (2.927E-4)	1.791E-4 (4.503E-4)	2.42E-5 (6.113E-4)	8.079E-4 (4.04E-5)	8.76E-6 (4.37E-6)	3.559E-4 (1.904E-4)
	500	5.885E-4 (2.859E-4)	7.123E-4 (7.956E-4)	5.136E-4 (4.597E-4)	2.018E-4 (6.668E-4)	5.289E-4 (2.252E-4)	6.915E-5 (7.887E-5)	1.3E-4 (2.05E-4)
LSMOP3	100	1.567E-4 (4.057E-4)	6.121E+3 (6.773E+3)	1.421E-6 (1.855E-5)	8.504E-4 (8.033E-4)	5.31E-5 (5.458E-4)	7.294E-4 (2.081E-4)	3.43E-5 (7.558E-4)
	200	2.562E-4 (8.536E-4)	5.52E+4 (3.6E+4)	6.736E-6 (5.759E-5)	2.453E-4 (8.343E-4)	7.336E-4 (8.593E-4)	6.16E-5 (8.675E-4)	3.519E-4 (4.37E-4)
	500	6.912E-4 (5.636E-4)	5.919E+4 (1.583E+3)	5.452E-5 (7.07E-5)	2.838E-4 (8.795E-4)	3.46E+0 (5.495E+1)	7.915E-4 (2.297E-4)	4.351E-4 (6.335E-4)
LSMOP4	100	2.607E-4 (7.064E-4)	3.011E+4 (6.289E+3)	8.005E-4 (4.111E-4)	6.796E-4 (1.057E-4)	7.792E-4 (8.64E-4)	1.356E-4 (1.74E-4)	4.418E-5 (3.233E-4)
	200	2.001E-4 (3.805E-4)	5.365E-4 (2.161E-4)	3.559E-4 (4.388E-4)	7.8E-4 (3.511E-4)	2.582E-4 (2.66E-4)	5.082E-4 (4.665E-4)	7.407E-5 (5.112E-5)
	500	2.792E-4 (4.379E-4)	8.8E-4 (2.819E-4)	4.59E-5 (5.184E-4)	2.812E-4 (7.157E-4)	3.3E-4 (8.525E-4)	3.193E-4 (7.589E-4)	4.447E-5 (6.1E-6)
LSMOP5	100	7.62E+0 (1.653E+1)	2.605E+0 (2.365E+0)	3.942E-4 (7.074E-4)	2.481E-4 (5.829E-4)	4.903E-4 (6.43E-4)	4.145E-4 (6.6E-4)	4.781E-6 (3.664E-5)
	200	1.61E+0 (2.43E+1)	3.434E+0 (6.004E+0)	3.39E-5 (7.774E-4)	2.607E-4 (5.314E-4)	5.235E-4 (3.86E-4)	2.15E-5 (4.849E-4)	1.125E-6 (7.642E-5)
	500	5.121E+0 (3.603E+0)	5.716E+0 (6.88E+1)	6.27E-5 (5.135E-4)	1.428E-4 (1.22E-4)	2.465E-4 (3.44E-4)	1.987E-4 (6.446E-4)	3.697E-6 (5.73E-5)
LSMOP6	100	1.002E+2 (4.328E+1)	1.55E+3 (5.829E+1)	6.651E-2 (6.302E-1)	3.566E-1 (6.692E-1)	1.959E-1 (9.84E-1)	3.645E-4 (2.1E-4)	6.83E-1 (3.42E-1)
	200	1.643E+4 (4.557E-4)	7.708E+4 (8.916E+3)	1.449E-2 (2.455E-1)	6.038E-1 (2.596E-2)	2.697E-1 (1.905E-1)	1.084E-4 (6.75E-4)	7.58E-1 (6.15E-2)
	500	6.282E+4 (4.728E+3)	4.46E+4 (8.343E+3)	6.542E-4 (5.888E-4)	4.9E-5 (7.676E-4)	1.55E-5 (7.466E-4)	3.638E-6 (2.685E-5)	4.734E-4 (5.343E-4)
LSMOP7	100	3.888E+3 (2.796E+2)	5.878E+4 (5.938E+3)	2.835E-4 (6.479E-4)	3.64E-4 (6.185E-4)	6.38E-4 (8.701E-4)	6.434E-5 (2.859E-5)	8.553E-4 (6.468E-4)
	200	8.42E+3 (4.91E+2)	2.159E+4 (3.594E+3)	1.974E-4 (7.615E-4)	5.146E-4 (8.122E-4)	6.043E-4 (6.425E-4)	4.673E-5 (4.909E-4)	7.857E-4 (2.647E-4)
	500	7.5E+4 (5.829E+3)	2.466E+4 (5.92E+3)	8.784E-4 (8.767E-4)	1.12E-5 (7.196E-4)	3.813E-4 (8.051E-4)	7.345E-4 (2.635E-4)	1.399E-4 (3.981E-4)
LSMOP8	100	6.58E-5 (6.493E-4)	5.861E+0 (2.589E+1)	8.86E-5 (7.215E-4)	5.61E-5 (8.271E-4)	2.609E-3 (4.535E-4)	7.927E-3 (5.515E-4)	4.757E-4 (1.378E-5)
	200	2.735E-4 (3.46E-4)	6.274E+0 (4.45E+1)	3.832E-1 (5.04E-1)	5.028E-1 (5.438E-1)	1.505E-3 (8.972E-2)	8.574E-3 (8.506E-2)	5.884E-4 (1.02E-5)
	500	6.034E-4 (6.699E-4)	7.597E+0 (2.266E+1)	2.596E-1 (1.569E-1)	6.413E-2 (3.224E-3)	1.13E-3 (1.179E-2)	4.211E-3 (5.726E-3)	6.761E-4 (5.52E-5)
LSMOP9	100	8.463E-4 (8.869E-3)	1.101E+0 (8.323E+1)	4.423E-2 (3.032E-3)	4.367E-1 (1.791E-1)	3.742E-3 (2.553E-4)	8.716E-2 (4.263E-2)	4.368E-4 (6.891E-4)
	200	2.284E-4 (5.742E-4)	7.364E+0 (5.212E+1)	3.053E-1 (6.474E-2)	2.61E-1 (5.251E-2)	8.269E-3 (5.249E-4)	7.889E-1 (7.732E-2)	2.07E-4 (6.5E-5)
	500	4.064E-4 (2.311E-4)	8.822E+0 (2.398E+1)	7.058E-2 (1.081E-1)	4.191E-2 (9.54E-3)	5.688E-3 (3.502E-4)	2.79E-2 (3.811E-2)	5.337E-4 (5.33E-5)

The experimental results of HV on two-objective LSMOP are shown in Table 3. The statistical results show that the improved genetic algorithm has good diversity on the LSMOP1, 100-dimensional LSMOP2, LSMOP4, LSMOP5, 100-dimensional and 200-dimensional LSMOP8 test problems, where the

CMOPSO algorithm performs the best on the LSMOP3 test problem.The FDVCSO algorithm performs the best on the 200-dimensional, 500-dimensional LSMOP6, 500 dimensional LSMOP7, and LSMOP9 test problems.The DGEA algorithm has good diversity on 200- and 500-dimensional LSMOP2, 100-dimensional LSMOP6, 100- and 200-dimensional LSMOP7, and 500-dimensional LSMOP8 test problems. There are 27 test problems and the improved algorithm has 12 advantages in terms of mean value. It can be inferred that the algorithm shows better diversity than other comparative algorithms on bi-objective LSMOP test problems.

Table 3. HV values on LSMOP on the second target

Prob.	Dim.	MOEA/D	GDE3	CMOPSO	FDVCSO	S3CMAES	DGEA	Ours
LSMOP1	100	4.56E-3 (5.818E-2)	1.053E+0 (6.448E-1)	2.171E-2 (6.031E-2)	1.467E-2 (7.814E-3)	6.7E-2 (1.923E-2)	5.603E-4 (2.537E-4)	2.337E-5 (8.764E-6)
	200	9.92E-3 (2.741E-3)	8.701E+0 (6.718E+1)	3.578E-4 (5.245E-4)	1.375E-4 (6.254E-4)	8.035E-4 (3.9E-3)	5.212E-4 (1.487E-4)	1.684E-5 (7.307E-5)
	500	8.616E-4 (8.472E-4)	2.98E+1 (2.076E+1)	5.63E-2 (4.023E-2)	2.904E-2 (4.577E-2)	8.175E-2 (5.269E-2)	6.708E-4 (5.479E-3)	7.647E-5 (2.048E-5)
LSMOP2	100	5.734E-4 (3.637E-4)	8.696E-4 (1.45E-4)	1.198E-4 (4.458E-4)	8.15E-2 (3.29E-2)	2.304E-4 (8.188E-4)	8.511E-4 (5.775E-4)	2.784E-5 (6.044E-5)
	200	8.86E-3 (1.169E-2)	6.274E-4 (2.397E-4)	6.324E-4 (3.922E-4)	2.645E-4 (4.337E-4)	8.131E-4 (2.363E-4)	9.84E-5 (4.179E-4)	2.162E-4 (1.369E-4)
	500	5.669E-4 (5.108E-4)	4.808E-4 (6.518E-4)	2.218E-4 (8.502E-4)	4.765E-4 (4.233E-4)	6.978E-4 (2.6E-5)	5.221E-5 (5.595E-6)	3.828E-4 (3.807E-5)
LSMOP3	100	4.945E+3 (5.698E+1)	5.605E+4 (8.75E+3)	5.241E-5 (7.808E-4)	3.798E-4 (8.622E-4)	2.585E-4 (4.313E-4)	2.086E-4 (1.131E-4)	4.649E-4 (4.717E-4)
	200	5.535E-4 (8.292E-4)	1.414E+4 (7.333E+3)	7.752E-5 (2.698E-5)	4.772E-4 (4.918E-4)	7.945E-4 (2.364E-4)	3.169E-4 (2.791E-4)	6.736E-4 (5.454E-4)
	500	4.57E+3 (3.359E+2)	1.82E+4 (8.654E+3)	3.131E-6 (4.467E-5)	1.32E-5 (1.016E-4)	5.232E-4 (8.647E-4)	3.977E-4 (7.562E-4)	3.247E-4 (2.756E-4)
LSMOP4	100	4.068E-4 (8.688E-4)	2.732E-4 (2.76E-3)	6.626E-4 (3.673E-4)	8.864E-4 (4.192E-4)	6.804E-4 (2.595E-4)	1.347E-4 (5.86E-4)	3.932E-5 (8.999E-4)
	200	4.411E-4 (8.154E-4)	3.083E-4 (3.01E-4)	1.718E-4 (4.952E-4)	5.751E-4 (3.154E-4)	7.824E-4 (8.035E-4)	8.977E-4 (2.814E-4)	1.45E-5 (2.532E-5)
	500	1.855E-4 (4.963E-4)	5.437E-4 (8.5E-4)	4.299E-4 (5.41E-4)	5.305E-4 (7.164E-4)	8.241E-4 (3.241E-4)	8.943E-4 (7.008E-4)	3.168E-5 (8.584E-5)
LSMOP5	100	7.824E+0 (3.16E+1)	2.439E+0 (1.737E+1)	5.829E+0 (5.523E+1)	5.516E-4 (6.996E-4)	8.807E-4 (5.727E-4)	6.324E-4 (7.072E-4)	4.197E-5 (3.73E-5)
	200	8.413E+0 (4.556E+1)	8.259E+0 (7.462E+1)	2.247E+0 (3.243E+1)	6.285E-4 (4.603E-4)	1.552E-4 (8.769E-4)	8.29E-4 (1.195E-4)	8.3E-6 (2.638E-5)
	500	2.745E+0 (2.099E+1)	7.539E+0 (4.072E+1)	4.838E+0 (1.43E+2)	6.843E-4 (4.685E-3)	1.963E-4 (5.58E-4)	7.635E-4 (4.04E-4)	4.924E-5 (5.73E-5)
LSMOP6	100	8.665E+0 (5.915E+1)	8.537E+0 (3.38E+1)	3.92E+0 (4.331E+1)	6.66E-4 (3.592E-4)	5.489E-4 (4.352E-4)	1.542E-5 (7.56E-5)	5.76E-4 (5.636E-4)
	200	6.005E+0 (3.026E+1)	2.406E+0 (2.961E+1)	2.393E-4 (6.4E-3)	8.045E-5 (2.611E-5)	1.732E-4 (4.856E-4)	7.85E-3 (3.376E-4)	1.526E-4 (3.852E-4)
	500	6.044E+0 (6.553E+1)	2.798E+0 (2.939E+1)	3.219E+0 (6E+1)	6.72E-5 (4.487E-5)	2.028E-4 (1.33E-5)	3.507E-4 (1.11E-4)	1.558E-4 (2.911E-4)
LSMOP7	100	5.189E+2 (4.871E+1)	1.443E+4 (6.385E+2)	4.622E+0 (1.533E+1)	7.317E-4 (4.741E-4)	8.337E-4 (8.384E-4)	5.001E-5 (5.4E-5)	3.029E-4 (1.631E-4)
	200	7.991E+2 (2.035E+1)	5.023E+3 (7.4E+2)	2.402E+0 (1.805E+1)	4.228E-4 (8.378E-3)	1.76E-4 (6.966E-3)	5.85E-5 (5.317E-4)	6.363E-4 (1.03E-3)
	500	7.362E+3 (6.1E+1)	1.024E+2 (4.66E+3)	2.086E+1 (5.768E+2)	8.6E-6 (3.641E-5)	6.6E-5 (6.174E-4)	8.594E-4 (1.826E-4)	3.601E-4 (7.655E-4)
LSMOP8	100	3.386E-4 (5.093E-4)	9.7E-5 (2.226E-4)	4.296E-4 (3.178E-4)	5.256E-4 (3.046E-4)	3.499E-4 (1.217E-4)	2.091E-4 (5.1E-3)	1.439E-6 (2.89E-5)
	200	4.599E-4 (2.195E-4)	6.574E-4 (5.549E-4)	4.888E-4 (3.514E-4)	5.766E-4 (4.846E-4)	5.221E-4 (2.274E-4)	6.743E-4 (2.347E-4)	5.132E-5 (7.92E-5)
	500	5.78E-4 (7.39E-4)	8.827E-4 (5.039E-4)	8.102E-4 (7.28E-2)	7.462E-4 (3.446E-4)	6.55E-5 (5.399E-4)	3.343E-6 (5.721E-5)	2.316E-4 (3.091E-4)
LSMOP9	100	1.854E-4 (1.971E-4)	1.987E-4 (6.205E-4)	1.993E-4 (8.95E-5)	3.24E-5 (3.872E-5)	5.113E-4 (8.948E-4)	4.686E-4 (8.73E-4)	6.14E-4 (8.902E-4)
	200	4.679E-4 (1.25E-3)	4.973E-4 (7.498E-4)	2.129E-4 (8.277E-4)	7.923E-5 (6.677E-6)	1.146E-4 (9.92E-5)	8.702E-4 (8.18E-5)	7.208E-4 (5.487E-4)
	500	7.13E-5 (8.918E-4)	4.606E-4 (4.809E-4)	5.282E-4 (6.456E-4)	7.792E-6 (3.833E-5)	6.83E-5 (8.765E-4)	7.783E-4 (5.75E-4)	7.098E-4 (2.673E-4)

The experimental results of IGD on three-objective LSMOP are shown in Table 4. The statistical results show that the improved genetic algorithm has good convergence on 500-dimensional LSMOP3, LSMOP5, LSMOP6, LSMOP7, 500-dimensional LSMOP8, and LSMOP9 test problems.MOEA/D on

LSMOP1, 100-dimensional LSMOP2, LSMOP4, 100-dimensional and 200-dimensional LSMOP8 test problems has good convergence.FDVCSO performs best on the 100- and 200-dimensional LSMOP3 test problems.DGEA performs best on the 200- and 500-dimensional LSMOP2 test problems. There are a total of 27 test problems and the improved algorithm has 14 advantages in terms of mean value. It can be concluded that the improved genetic algorithm has significantly improved in terms of convergence compared to the comparative algorithms on the three-objective LSMOP test problems.

Table 4. IGD values on LSMOP on the third target

Prob.	Dim.	MOEA/D	GDE3	CMOPSO	FDVCSO	S3CMAES	DGEA	Ours
LSMOP1	100	2.411E-4 (1.525E-4)	7.001E-1 (2.66E+0)	6.191E-1 (2.389E-1)	8.911E-2 (3.452E-2)	6.81E-3 (6.61E-3)	9.1E-1 (5.6E-1)	6.081E-3 (2.269E-2)
	200	1.431E-4 (2.269E-3)	8.801E-1 (5.88E-1)	1.511E-1 (5.227E-1)	3.331E-2 (4.371E-2)	5.401E-1 (1.706E-2)	1.251E-2 (2.87E-2)	3.101E-3 (8.717E-2)
	500	1.181E-4 (1.715E-3)	2.081E-1 (1.55E-1)	6.431E-2 (7.294E-2)	2.71E-1 (8.554E-1)	6.261E-2 (6.379E-2)	5.231E-1 (5.25E-1)	6.881E-3 (3.749E-2)
LSMOP2	100	2.681E-4 (8.75E-3)	7.391E-3 (7.159E-2)	3.791E-2 (3.112E-2)	8.681E-1 (1.467E+0)	1.411E-2 (7.429E-1)	3.31E-1 (8.942E-1)	4.1E-2 (5.813E-2)
	200	5.531E-4 (4.252E-4)	4.971E-4 (1.14E-3)	2.591E-4 (7.614E-4)	7.271E-4 (7.562E-4)	5.121E-4 (3.09E-4)	6.441E-5 (7.69E-5)	8.811E-4 (8.039E-4)
	500	2.941E-4 (8.432E-4)	7.051E-4 (1.087E-4)	4.81E-3 (3.21E-4)	7.091E-4 (8.193E-4)	3.111E-4 (5.475E-4)	2.961E-5 (7.486E-5)	1.871E-4 (3.953E-3)
LSMOP3	100	8.201E+3 (5.454E+2)	1.191E+2 (3.291E+1)	4.911E+3 (2.51E+2)	5.71E-5 (4.369E-4)	7.841E-4 (2.996E-3)	5.011E-4 (8.386E-3)	6.551E-4 (5.425E-3)
	200	1.901E+1 (4.932E+1)	1.321E+2 (3.678E+2)	7.251E+3 (4.509E+1)	8.051E-3 (4.509E-2)	8.631E-1 (8.768E+0)	2.641E-2 (2.235E-1)	1.471E-2 (6.145E-1)
	500	7.331E-4 (2.202E-4)	7.401E-4 (8.799E-4)	1.691E-4 (2.496E-4)	6.351E-4 (8.825E-4)	4.511E-4 (6.757E-4)	5.981E-4 (2.292E-4)	2.101E-5 (6.229E-5)
LSMOP4	100	7.91E-2 (3.05E-2)	6.481E+1 (2.922E+1)	7.91E+2 (7.829E+2)	3.151E+3 (2.846E+2)	2.081E+1 (1.215E+2)	8.621E+3 (4.323E+2)	1.961E+2 (6.768E+1)
	200	6.501E-2 (2.153E-2)	7.751E+1 (5.067E+2)	7.31E+2 (6.486E+1)	1.201E-1 (5.92E-1)	4.551E-1 (8.23E+0)	1.451E-1 (6.794E-1)	5.741E-1 (7.763E-1)
	500	4.201E-4 (5.156E-3)	8.841E-3 (9.8E-2)	1.681E-3 (7.749E-2)	2.591E-3 (5.179E-2)	6.921E-3 (4.706E-2)	3.641E-3 (2.07E-2)	7.791E-2 (7.62E-1)
LSMOP5	100	2.121E-1 (4.441E-1)	1.991E-1 (2.144E-1)	7.371E-1 (5.317E-1)	3.371E-2 (7.493E-2)	4.931E-2 (4.469E-2)	3.611E-3 (5.897E-2)	5.791E-4 (3.509E-3)
	200	6.801E-1 (8.865E-2)	1.841E-1 (6.792E-2)	1.591E-1 (6.912E-2)	3.831E-1 (6.653E-2)	4.691E-3 (1.715E-2)	2.301E-3 (6.794E-2)	4.551E-4 (5.174E-3)
	500	6.701E-2 (6.02E-1)	4.611E-2 (6.269E-1)	2.451E-3 (7.248E-2)	4.211E-1 (3.153E-2)	7.831E-2 (7.947E-2)	7.721E-1 (2.884E-2)	2.471E-4 (4.019E-3)
LSMOP6	100	6.991E-3 (1.959E-2)	6.461E-2 (7.383E-2)	2.671E-2 (5.163E-2)	1.361E-2 (2.328E-2)	5.271E-1 (2.163E-2)	4.321E-3 (2.27E-2)	1.871E-4 (3.722E-3)
	200	2.841E-1 (7.134E-1)	1.411E-1 (1.21E-1)	8.51E-2 (7.245E-2)	5.141E-1 (3.022E-1)	6.911E-3 (1.89E-2)	2.31E-2 (5.499E-2)	3.591E-4 (6.348E-3)
	500	5.531E-1 (7.671E-1)	6.741E+2 (1.371E+1)	3.901E+2 (2.482E+1)	1.041E+1 (7.945E+2)	9.71E+2 (8.55E+2)	6.981E+1 (6.387E+1)	7.321E-2 (1.629E-2)
LSMOP7	100	6.561E-1 (1.607E-1)	4.801E+3 (7.131E+2)	3.471E+3 (4.071E+2)	5.391E+2 (5.86E+1)	6.901E+1 (8.346E+2)	1.691E+3 (8.03E+2)	2.501E-2 (5.919E-2)
	200	7.521E+2 (4.948E+1)	3.11E+1 (7.083E+2)	8.511E+2 (2.433E+1)	5.131E+3 (7.479E+2)	6.341E+1 (3.167E+2)	6.631E+2 (8.544E+2)	4.701E-3 (8.986E-2)
	500	2.171E+1 (8.131E+2)	1.571E+2 (1.948E+1)	2.331E+3 (4.226E+1)	1.1E+3 (1.91E+2)	2.411E+1 (4.613E+2)	7.151E-1 (5.225E+0)	5.471E-2 (6.528E-2)
LSMOP8	100	8.231E-4 (6.538E-4)	5.641E-3 (3.937E-2)	1.381E-3 (4.999E-2)	7.611E-3 (4.798E-2)	6.681E-2 (4.409E-2)	4.781E-1 (5.28E-1)	4.501E-1 (6.076E-1)
	200	3.311E-5 (6.094E-4)	2.001E-4 (7.324E-3)	6.61E-5 (3.177E-3)	9.11E-5 (3.413E-3)	3.331E-4 (1.869E-3)	3.071E-4 (1.91E-3)	2.711E-4 (5.115E-3)
	500	1.881E-4 (1.615E-4)	7.591E-4 (6.65E-4)	7.561E-4 (5.323E-4)	3.201E-4 (3.188E-4)	1.071E-4 (6.332E-4)	4.31E-1 (4.01E-1)	2.061E-5 (8.605E-5)
LSMOP9	100	2.241E-4 (6.91E-3)	4.001E-4 (7.642E-3)	4.71E-2 (6.441E-1)	3.261E-4 (6.552E-4)	1.201E-4 (3.274E-4)	5.11E-2 (7.783E-2)	6.241E-5 (3.982E-5)
	200	5.551E-4 (1.92E-4)	8.091E-4 (6.794E-4)	4.031E-4 (8.64E-4)	3.271E-4 (8.946E-4)	1.951E-4 (6.238E-4)	8.781E-4 (3.326E-4)	1.31E-5 (2.606E-5)
	500	7.351E+2 (6.275E+1)	6.161E-2 (4.302E-2)	2.491E-1 (2.158E-1)	1.951E-2 (2.333E-2)	7.301E+1 (2.865E+1)	8.531E+2 (5.544E+2)	7.031E-3 (5.38E-3)

The experimental results of HV on three-objective LSMOP are shown in Table 5. The statistical results show that the improved genetic algorithm has better diversity on 100-, 200-, and 500-dimensional LSMOP1, LSMOP5, LSMOP6, and LSMOP7 test problems.MOEA/D algorithm has better diversity on 500-dimensional LSMOP3, 100-dimensional LSMOP8, and LSMOP9 test problems.CMOPSO algorithm performs best on the 200- and 500-dimensional LSMOP2, 200- and 500-dimensional LSMOP4, and 500-dimensional LSMOP5 test problems.FDVCSO algorithm performs optimally on 100- and 200-dimensional LSMOP3 test problems. The DGEA algorithm performed optimally on the 100- and 100-dimensional LSMOP2, 100-dimensional LSMOP4, 200- and 500-dimensional LSMOP8 test problems. In total, there are 27 test problems and the improved algorithm has 12 advantages on the average. The algorithm performs better in terms of diversity on three-objective LSMOP test problems.

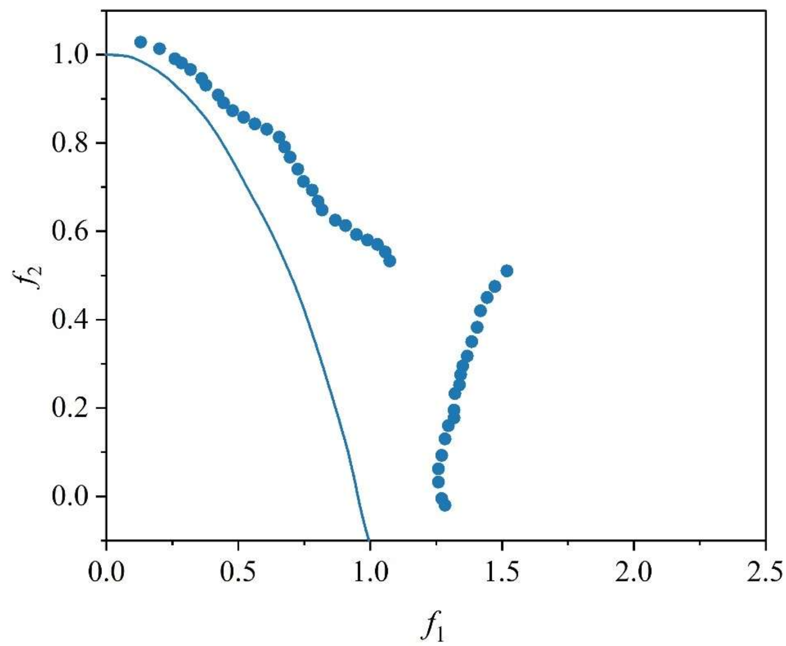
Table 5. HV values on LSMOP on the third target

Prob.	Dim.	MOEA/D	GDE3	CMOPSO	FDVCSO	S3CMAES	DGEA	Ours
LSMOP1	100	3.371E-2 (4.6E-2)	5.11E-1 (5.284E-1)	6.601E-2 (1.435E-2)	4.21E-1 (2.742E-1)	3.281E-1 (8.561E-1)	1.061E-2 (5.64E-2)	1.511E-3 (3.66E-4)
	200	7.921E-1 (3.592E-1)	6.81E-1 (2.877E-1)	7.181E-2 (8.22E-2)	1.711E-2 (4.553E-2)	5.781E-1 (8.224E-2)	4.741E-2 (8.476E-2)	4.161E-3 (2.96E-3)
	500	4.361E-2 (8.44E-2)	1.91E-3 (2.11E-3)	7.521E-2 (7.976E-2)	8.151E-1 (7.295E-1)	7.951E-2 (3.609E-2)	7.481E-3 (6.473E-2)	3.981E-4 (1.082E-4)
LSMOP2	100	8.971E-2 (5.24E-2)	8.741E-2 (8.208E-2)	6.531E-2 (4.48E-2)	7.271E-2 (1.688E-2)	2.821E-2 (5.659E-2)	6.141E-4 (6.701E-3)	1.751E-3 (6.72E-2)
	200	5.981E-3 (2.261E-2)	1.151E-3 (2.281E-2)	4.821E-4 (6.005E-3)	8.041E-3 (4.65E-2)	2.021E-2 (4.347E-3)	1.951E-2 (7.678E-2)	6.281E-3 (8.075E-2)
	500	4.131E-3 (5.957E-1)	4.491E-3 (1.941E-2)	5.311E-4 (8.124E-3)	8.121E-2 (2.846E-2)	4.101E-2 (5.323E-1)	3.11E-2 (1.562E-1)	2.651E-3 (3.879E-2)
LSMOP3	100	4.251E-3 (5.614E-2)	5.91E-2 (5.37E-1)	6.671E-3 (4.714E-2)	4.261E-4 (3.337E-3)	1.071E-2 (8.684E-2)	4.401E-2 (2.301E-2)	4.491E-3 (8.025E-3)
	200	5.931E-4 (3.286E-2)	1.811E-4 (2.517E-3)	7.571E-4 (6.46E-2)	1E-5 (6.43E-4)	4.531E-4 (4.069E-3)	8.561E-4 (7.22E-3)	5.211E-4 (8.652E-2)
	500	6.781E-5 (5.076E-5)	7.941E-4 (3.56E-4)	2.011E-4 (5.87E-4)	7.361E-4 (5.671E-4)	3.641E-4 (1.393E-4)	6.481E-4 (9.89E-3)	6.341E-4 (5.628E-4)
LSMOP4	100	4.671E-4 (3.999E-4)	1.701E-4 (7.624E-4)	5.641E-4 (4.484E-4)	4.371E-4 (4.408E-4)	1.141E-4 (2.019E-4)	2.41E-5 (3.497E-5)	7.821E-4 (3.05E-3)
	200	3.511E-4 (5.784E-4)	1.721E-4 (5.236E-4)	7.381E-5 (2.061E-5)	1.041E-4 (2.99E-3)	3.351E-4 (1.01E-3)	9.51E-5 (4.78E-4)	8.101E-4 (8.906E-4)
	500	2.051E-4 (8.754E-4)	4.471E-4 (7.068E-4)	4.061E-5 (2.398E-5)	2.891E-4 (3.307E-4)	7.601E-4 (8.409E-4)	8.431E-4 (8.403E-4)	7.81E-3 (2.72E-4)
LSMOP5	100	3.951E-2 (4.077E-1)	5.131E-2 (8.102E-2)	6.561E-2 (5.06E-1)	4.151E-2 (7.376E-2)	3.411E-2 (7.774E-2)	7.991E-2 (2.684E-1)	3.101E-3 (2.332E-3)
	200	2.801E-2 (3.833E-2)	5.801E-1 (4.171E-1)	4.031E-1 (4.291E-1)	2.831E-1 (4.363E-1)	8.61E-1 (4.254E-1)	2.21E-2 (7.071E-2)	7.431E-3 (5.191E-3)
	500	6.131E-4 (1.77E-3)	4.671E-4 (5.712E-4)	3.081E-4 (1.361E-4)	3.351E-4 (8.39E-3)	6.171E-4 (3.48E-4)	7.071E-4 (5.03E-4)	1.301E-5 (6.043E-5)
LSMOP6	100	5.361E-4 (4.006E-4)	7.281E-4 (3.8E-3)	3.631E-4 (3.986E-4)	6.211E-4 (1.11E-3)	6.321E-4 (3.94E-4)	1.901E-4 (4.167E-4)	6.381E-5 (7.716E-5)
	200	1.331E-4 (3.056E-4)	6.621E-4 (4.69E-4)	3.851E-4 (7.307E-4)	3.821E-4 (1.885E-4)	6.101E-4 (1.514E-4)	5.641E-4 (5.743E-4)	5.81E-5 (2.626E-5)
	500	8.361E-4 (1.257E-4)	7.111E-4 (5.121E-4)	5.601E-4 (7.199E-4)	2.691E-4 (8.966E-4)	6.191E-4 (5.514E-4)	4.181E-4 (7.871E-4)	8.961E-5 (2.897E-5)
LSMOP7	100	6.131E-4 (4.932E-4)	6.371E-4 (7.304E-4)	8.871E-4 (8.103E-4)	7.291E-4 (4.501E-4)	5.711E-4 (2.406E-4)	2.541E-4 (2.38E-4)	2.541E-5 (1.927E-5)
	200	5.631E-3 (6.967E-2)	8.81E-3 (1.936E-2)	2.161E-3 (5.784E-2)	1.251E-2 (3.85E-2)	2.761E-2 (5.047E-2)	4.621E-2 (7.482E-2)	7.021E-4 (4.578E-3)
	500	7.811E-2 (6.589E-1)	6.181E-1 (4.22E-1)	6.211E-1 (8.133E-1)	8.711E-1 (3.5E-1)	2.381E-2 (7.834E-1)	7.921E-2 (6.021E-1)	4.661E-3 (1.103E-2)
LSMOP8	100	2.211E-5 (5.407E-5)	5.481E-4 (8.418E-4)	8.671E-4 (3.321E-4)	1.551E-4 (5.7E-3)	5.1E-3 (1.244E-2)	1.901E-4 (8.994E-4)	7.741E-4 (5.93E-3)
	200	4.731E-4 (7.637E-4)	6.791E-4 (5.263E-4)	7.201E-4 (8.64E-4)	5.591E-4 (2.753E-4)	7.251E-4 (4.18E-3)	2.791E-5 (7.47E-5)	1.361E-4 (7.063E-4)
	500	2.991E-4 (2.259E-4)	5.691E-4 (2.201E-4)	3.171E-4 (4.855E-4)	2.41E-3 (2.335E-4)	6.831E-4 (5.081E-4)	7.111E-5 (3.278E-5)	7.261E-4 (6.84E-4)
LSMOP9	100	8.081E-2 (2.961E-2)	8.031E-1 (7.902E-1)	5.721E-1 (5.061E-1)	9.11E-1 (6.14E-1)	2.221E-1 (5.198E-1)	1.661E+1 (1.601E+0)	3.051E+2 (6.547E+1)
	200	2.581E-2 (7.913E-2)	7.21E+2 (6.778E+1)	6.411E+2 (4.555E+1)	4.421E+1 (7.456E+1)	3.691E+1 (2.565E+1)	3.111E-1 (4.871E-1)	1.401E-1 (8.132E-1)
	500	4.261E-2 (5.302E-2)	3.1E+1 (3.595E+1)	7.01E+1 (1.286E+1)	6.241E+2 (3.307E+2)	7.231E+2 (8.984E+2)	1.081E-1 (2.444E+0)	8.921E-1 (3.883E-1)

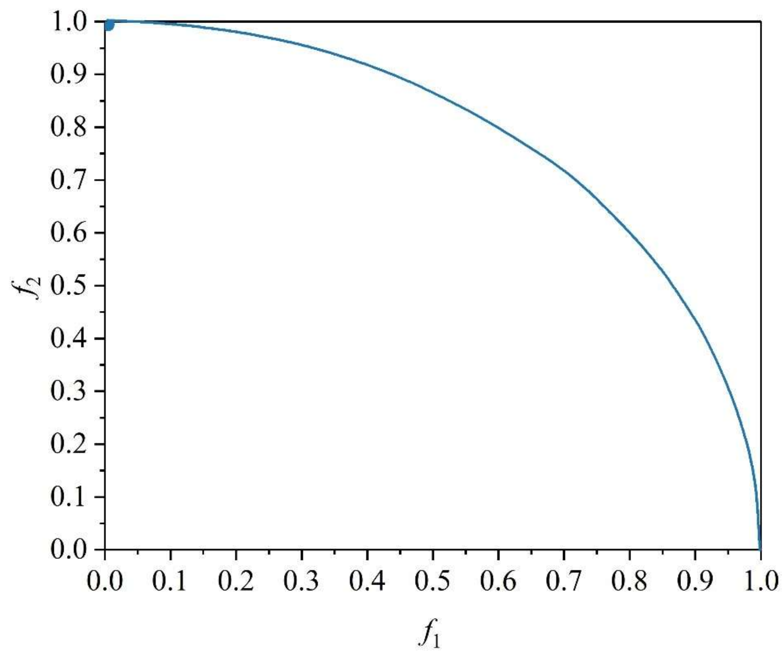
The PF results of GDE3, CMOPSO, MOEA/D, DGEA, S3CMAES, FDVCSO, and the method of this paper on LSMOP6 with two objectives and a decision variable of 200 are shown in Fig. 2. The PF results of the seven methods on LSMOP5 with three objectives and a decision variable of 200 are shown in Fig.

3. The curved solid line in the figure is the real PF curve, and the blue solid meta-dots are the populations obtained by the algorithm. (a)~(g) denote the PF results of GDE3, CMOPSO, MOEADVA, DGEA, S3CMAES, FDVCSO and improved genetic algorithm, respectively.

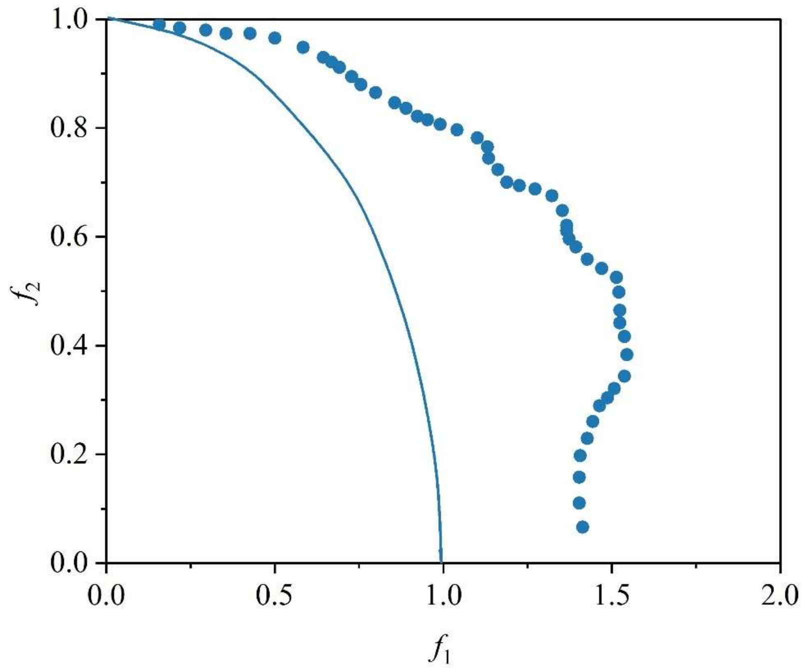
From Fig. 2, it can be seen that the improved genetic algorithm has the best results.GDE3 cannot obtain a complete population.CMOPSO's PF is incomplete.MOEA/D's population converges poorly.S3CMAES cannot obtain the corresponding candidate solutions in a limited number of iterations.FDVCSO has not been completely fitted to the real PF, at this time, the FDVCSO may be in a process of fuzzy evolution, the population has approximated convergence and is ready to enter into the process of exact evolution as a way to improve the diversity of the algorithm.



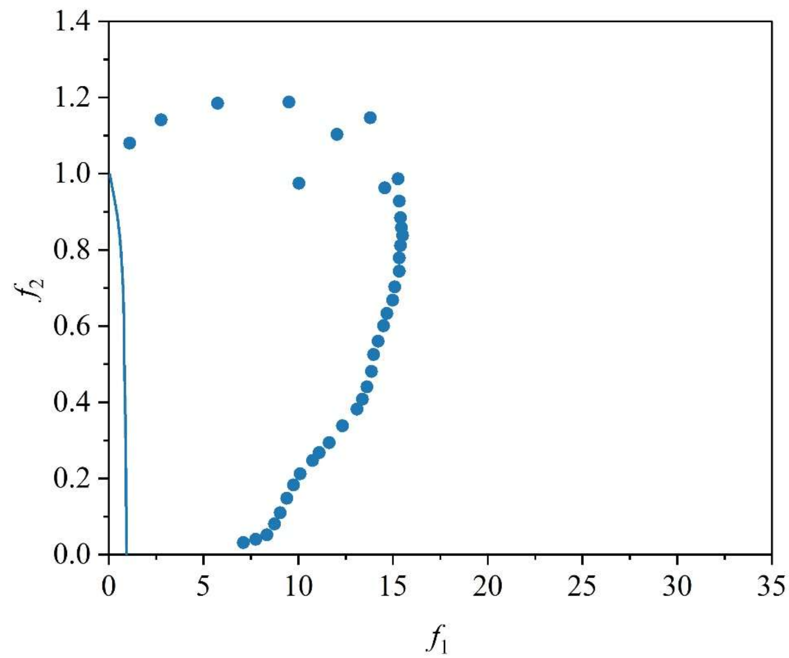
(a) GDE3



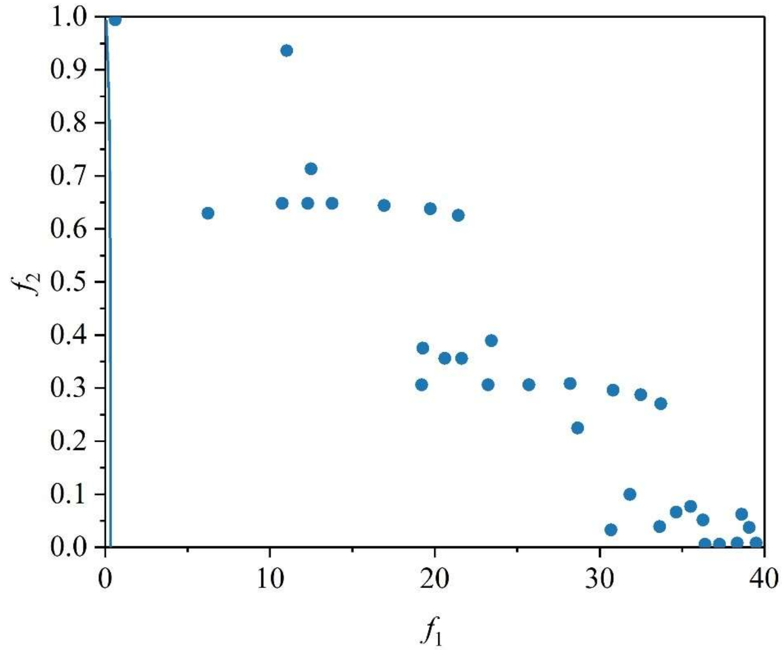
(b) CMOPSO



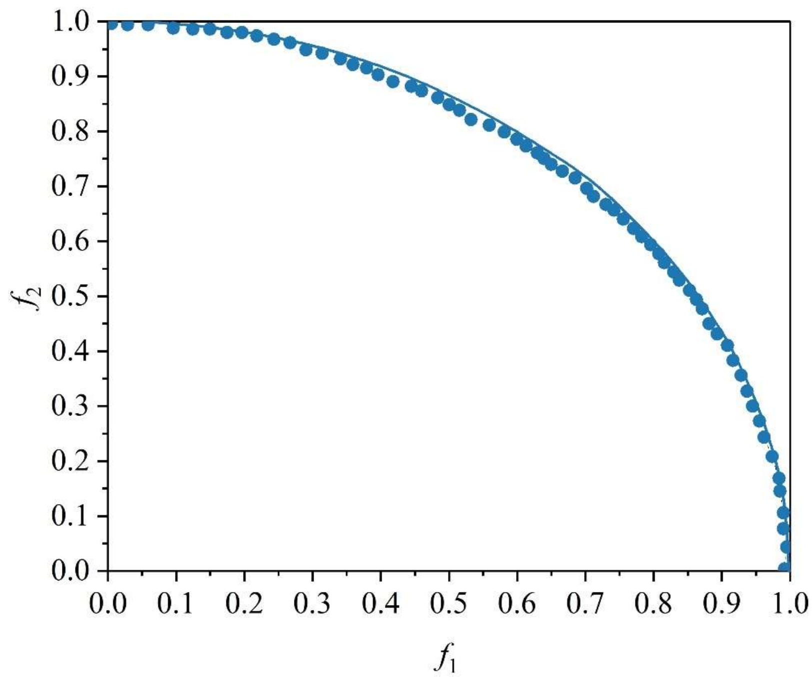
(c) MOEA/D



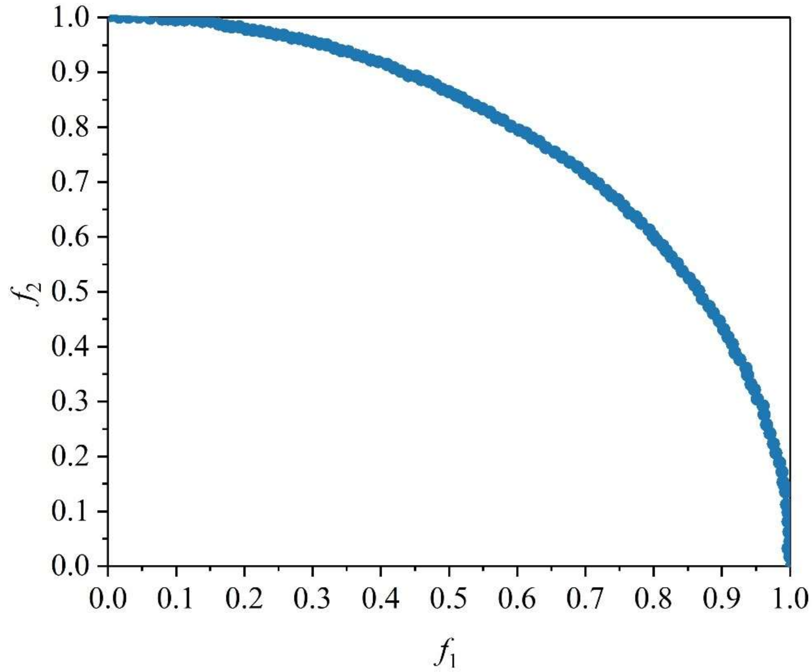
(d) DGEA



(e) S3CMAES



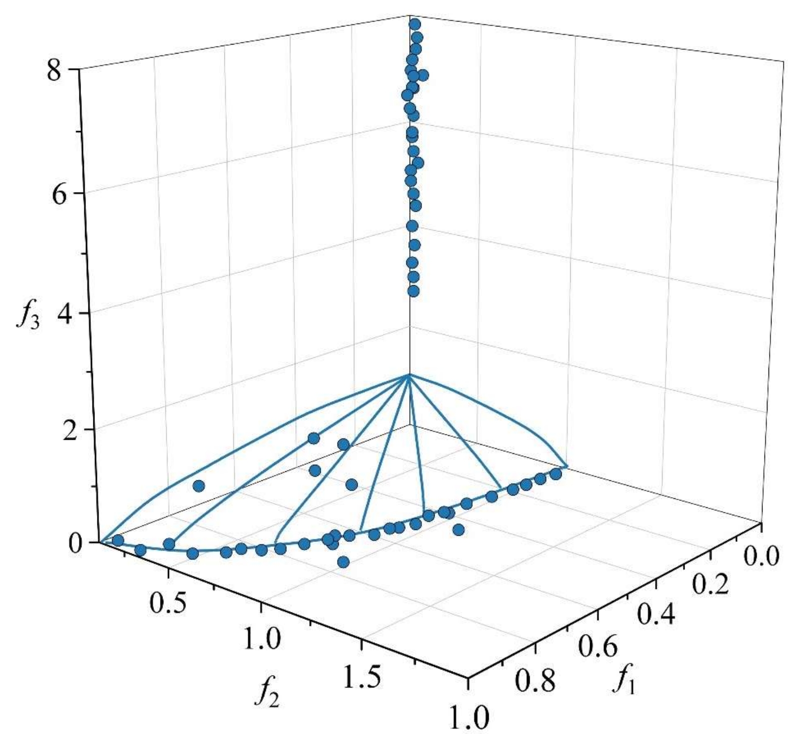
(f) FDVCSO



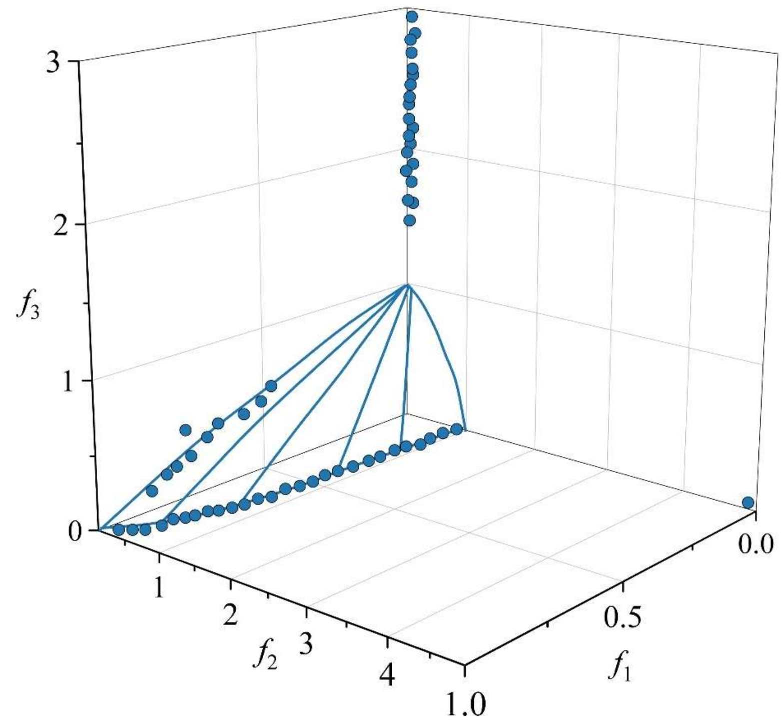
(g) Improved GA

Fig. 2. The f results of the LSMOP6 on the second target

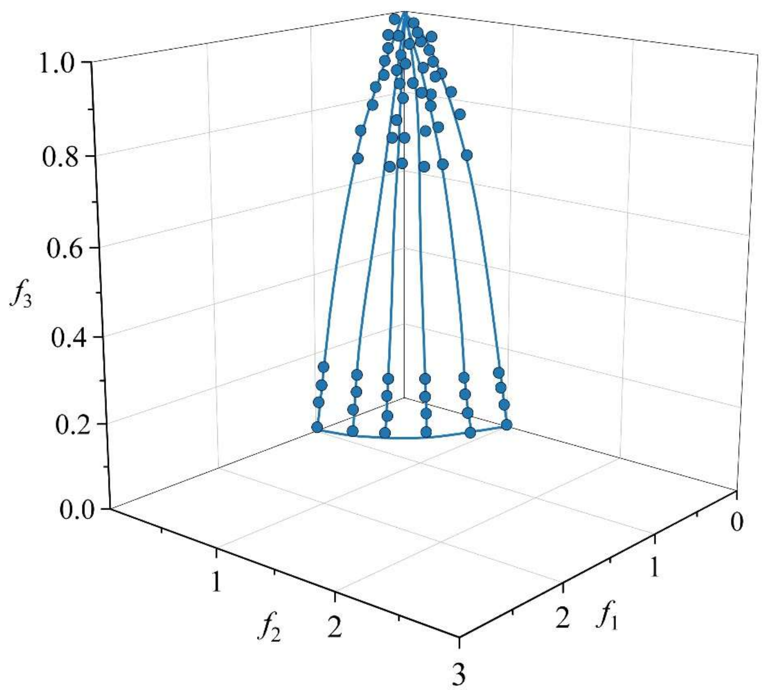
As can be seen from Fig. 3, the population distribution of the improved genetic algorithm is very uniform. The PFs obtained by GDE3, CMOPSO, and DGEA were less than ideal, with GDE3 and DGEA being the least effective. MOEA/D and FDVCSO populations are more evenly distributed. S3CMAES failed to obtain populations in a limited number of iterations.



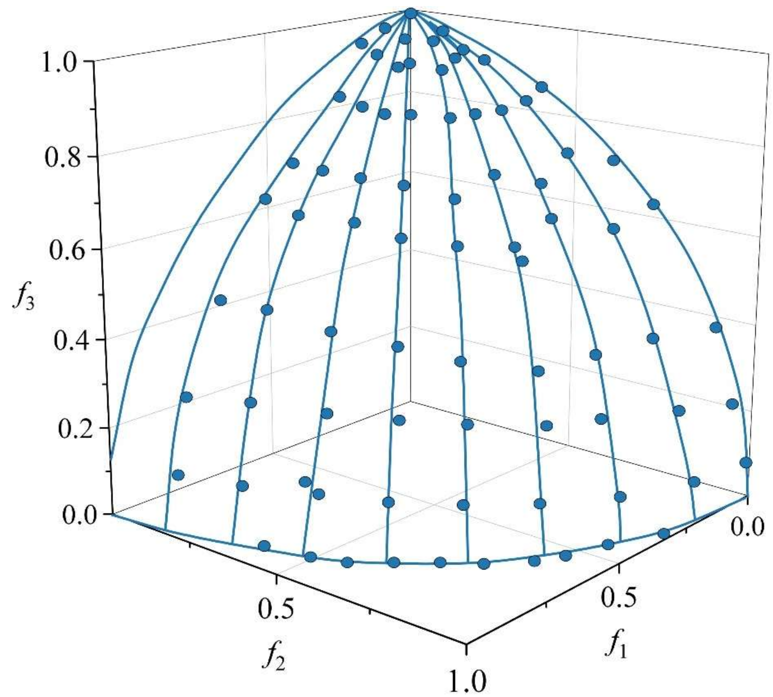
(a) GDE3



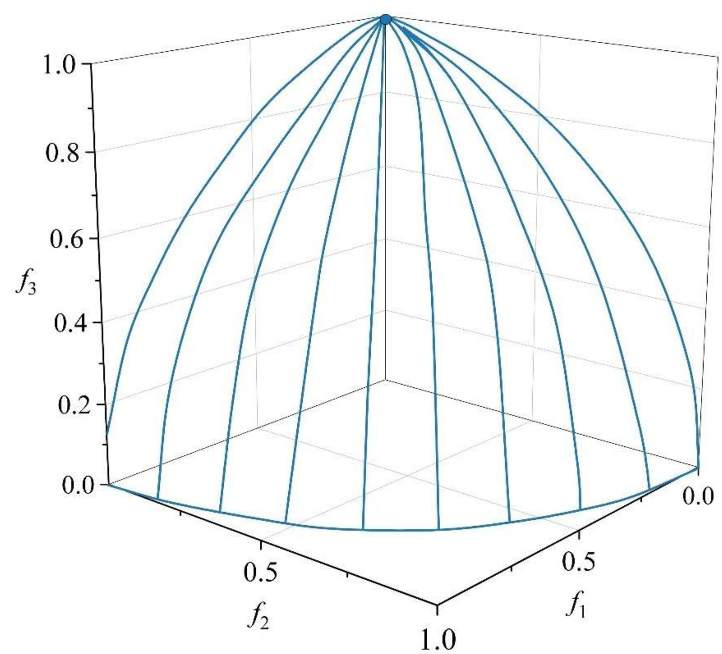
(b) CMOPSO



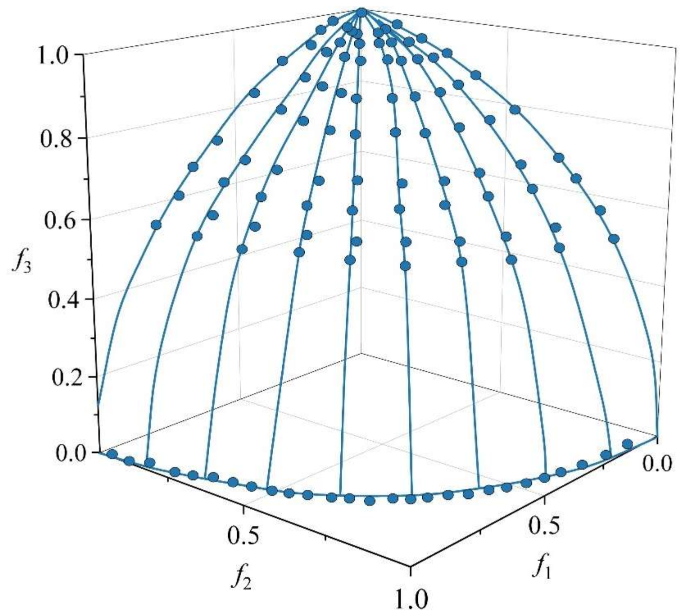
(c) MOEA/D



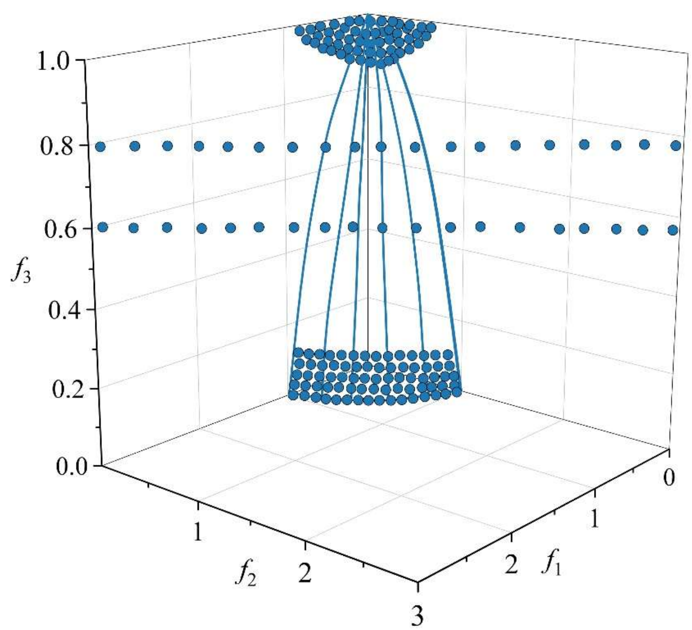
(d) DGEA



(e) S3CMAES



(f) FDVCSO



(g) Improved GA

Fig. 3. The f results of the LSMOP5 on the third target

6. Conclusion

In this paper, improved genetic algorithms are designed to address multi-objective optimization strategies in large-scale computing environments. The improved genetic algorithm designed in this paper is compared with six state-of-the-art MOEAs, and the seven algorithms are evaluated for convergence and diversity via the inverse generation distance (IGD) and hypervolume (HV) metrics using the LSMOP test problem.

1) The experimental results for IGD and HV on the two-objective LSMOP show that the improved genetic algorithms have 16 and 12 advantages on average, respectively. The experimental results of IGD and HV on three-objective LSMOP show that the improved genetic algorithm has 14 and 12

advantages on the average value, respectively. It shows that the improved genetic algorithm has better convergence and diversity than other compared algorithms on two- and three-objective LSMOP test problems.

2) The PF curves for both the two-objective on LSMOP6 and the three-objective on LSMOP5 show the most uniform population distribution for the improved genetic algorithm when the decision variable is 200. It shows that the improved genetic algorithm can ensure its better convergence to the optimal solution when solving large-scale multi-objective optimization problems.

References

[1] Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K. C., & Jin, Y. (2021). Evolutionary large-scale multi-objective optimization: A survey. ACM Computing Surveys (CSUR), 54(8), 1-34.

[2] Hong, W. J., Yang, P., & Tang, K. (2021). Evolutionary computation for large-scale multi-objective optimization: A decade of progresses. International Journal of Automation and Computing, 18(2), 155-169.

[3] Zhang, Y., Tian, Y., & Zhang, X. (2023). Improved SparseEA for sparse large-scale multi-objective optimization problems. Complex & Intelligent Systems, 9(2), 1127-1142.

[4] Qi, S., Zou, J., Yang, S., & Zheng, J. (2022). A level-based multi-strategy learning swarm optimizer for large-scale multi-objective optimization. Swarm and Evolutionary Computation, 73, 101100.

[5] Antonio, L. M., Coello, C. A. C., Brambila, S. G., González, J. F., & Tapia, G. C. (2019, July). Operational decomposition for large scale multi-objective optimization problems. In Proceedings of the genetic and evolutionary computation conference companion (pp. 225-226).

[6] Liu, R., Liu, J., Li, Y., & Liu, J. (2020). A random dynamic grouping based weight optimization framework for large-scale multi-objective optimization problems. Swarm and Evolutionary Computation, 55, 100684.

[7] Xu, Y., Xu, C., Zhang, H., Huang, L., Liu, Y., Nojima, Y., & Zeng, X. (2022). A multi-population multi-objective evolutionary algorithm based on the contribution of decision variables to objectives for large-scale multi/many-objective optimization. IEEE Transactions on Cybernetics, 53(11), 6998-7007.

[8] Seifollahi-Aghmiuni, S., & Bozorg Haddad, O. (2018). Multi objective optimization with a new evolutionary algorithm. Water resources management, 32, 4013-4030.

[9] Mirjalili, S., & Mirjalili, S. (2019). Genetic algorithm. Evolutionary algorithms and neural networks: theory and applications, 43-55.

[10] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia tools and applications, 80, 8091-8126.

[11] Mayer, M. J., Szilágyi, A., & Gróf, G. (2020). Environmental and economic multi-objective optimization of a household level hybrid renewable energy system by genetic algorithm. Applied Energy, 269, 115058.

[12] Starke, A. R., Cardemil, J. M., Escobar, R., & Colle, S. (2018). Multi-objective optimization of hybrid CSP+ PV system using genetic algorithm. Energy, 147, 490-503.

[13] Costa-Carrapiço, I., Raslan, R., & González, J. N. (2020). A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. Energy and Buildings, 210, 109690.

[14] Sahoo, L., Bhunia, A. K., & Kapur, P. K. (2012). Genetic algorithm based multi-objective reliability optimization in interval environment. Computers & Industrial Engineering, 62(1), 152-160.

[15] Akopov, A. S., & Hevencev, M. A. (2013, October). A multi-agent genetic algorithm for multi-objective optimization. In 2013 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1391-1395).

IEEE.

[16] Azzouz, R., Bechikh, S., & Ben Said, L. (2017). Dynamic multi-objective optimization using evolutionary algorithms: a survey. Recent advances in evolutionary multi-objective optimization, 31-70.

[17] Bouzid, S. E., Seresstou, Y., Raoof, K., Omri, M. N., Mbarki, M., & Dridi, C. (2020). MOONGA: multi-objective optimization of wireless network approach based on genetic algorithm. IEEE Access, 8, 105793-105814.

[18] Guerrero, C., Lera, I., & Juiz, C. (2018). Genetic algorithm for multi-objective optimization of container allocation in cloud architecture. Journal of Grid Computing, 16, 113-135.

[19] Bo Liu,Dan Wang & Jian Gao. (2024). A Multi-objective Community Detection Algorithm with a Learning-Based Strategy. International Journal of Computational Intelligence Systems(1),311-311.

[20] Xiaoda Liu,Jing Yang,Liya Yi,Donghu Li,Qian Wang,Huayun Du... & Yinghui Wei. (2024). Multi-objective optimization of fracturing ball strength and corrosion rate with genetic algorithms and interpretable machine learning. Advanced Composites and Hybrid Materials(1),100-100.

[21] Xiaohan Sun,Yuan Ren & Linghui Yu. (2024). Multi-Adaptive Strategies-Based Higher-Order Quantum Genetic Algorithm for Agile Remote Sensing Satellite Scheduling Problem. Sensors (Basel, Switzerland)(15),4938-4938.

[22] Mauricio Torres Arellano,Aaron Burgos Vergara,Saul Piedra Gonzalez,Erik César Herrera Hernández & Edgar A. Franco Urquiza. (2024). Optimal design of laminates orientation in a composite material by genetics algorithm and simulated annealing. Mechanics of Advanced Materials and Structures(28),10592-10600.