

## **AN IMPROVED BARE-BONES PARTICLE SWARM ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION WITH APPLICATION TO THE ENGINEERING STRUCTURES**

**Zhaohua Wang<sup>1</sup>, Guobiao Yang<sup>1</sup>, Yinxu Sun<sup>2</sup>, Yongxin Li<sup>2</sup>,  
Fenghe Wu<sup>2</sup>**

<sup>1</sup>School of Mechanical Engineering,  
Taiyuan University of Science and Technology, Taiyuan, China  
<sup>2</sup>School of Mechanical Engineering, Yanshan University, Qinhuangdao, China

**Abstract.** *In this paper, an improved bare-bones multi-objective particle swarm algorithm is proposed to solve the multi-objective size optimization problems with non-linearity and constraints in structural design and optimization. Firstly, the development of particle individual guide and the randomness of gravity factor are increased by modifying the updated form of particle position. Then, the combination of spatial grid density and congestion distance ranking is used to maintain the external archive, which is divided into two parts: feasible solution set and infeasible solution set. Next, the global best positions are determined by increasing the probability allocation strategy which varies with time. The algorithmic complexity is given and the performance of solution ability, convergence and constraint processing are analyzed through standard test functions and compared with other algorithms. Next, as a case study, a support frame of triangle track wheel is optimized by the BB-MOPSO and improved BB-MOPSO. The results show that the improved algorithm improves the cross-region exploration, optimal solution distribution and convergence of the bare-bones particle swarm optimization algorithm, which can effectively solve the multi-objective size optimization problem with non-linearity and constraints.*

**Key words:** *Optimization, Multi-objective, Particle swarm, BB-MOPSO*

### 1. INTRODUCTION

At present, engineering structures are developing towards high reliability and lightweight design. In the design process, the stiffness, strength and vibration characteristics

---

Received: August 29, 2022 / Accepted January 16, 2023

**Corresponding author:** Fenghe Wu  
School of Mechanical Engineering, Yanshan University, No 438, West of Hebei Avenue, Qinhuangdao, China, 066004.  
E-mail: risingwu@ysu.edu.cn

of the structure must be taken into account at the same time [1, 2]. Therefore, the multi-objective optimization problem (MOP) in the process of structural design is becoming more and more important [3]. In the conceptual design stage, the topology configuration of structure is determined by the topology optimization methods [4]. Then, the local size parameters are further optimized to enhance the mechanical performance in the detailed design stage. Size optimization needs to consider many mechanical performances of the structure at the same time, including stiffness, strength, fatigue life, etc. Local extremum and non-convergence often occur in the calculation process, especially the stress-based optimization, which belongs to a typical nonlinear MOP with constraints.

With the increasing complexity of multi-objective optimization problems, more and more intelligent algorithms are applied to size optimization. Bekdas et al. [5] and Assimi et al. [6] improved different intelligent algorithms to optimize the truss size with good results obtained. A new collision box is designed and optimized by using archive-based micro-genetic algorithm and improved, non-dominated sorting genetic algorithm, which improve the energy absorption characteristics and comprehensive crashworthiness [7]. Although many achievements have been made in the application of intelligent algorithms to size optimization, there are still problems of local extremum and non-convergence for the solution with multi-constraints. Evolutionary algorithm or genetic algorithm [8-9] has a special evaluation mechanism, and has gradually matured after several generations of development, but its local search ability is poor, and the selection of more parameters affects the quality of the solution. Simulated annealing method [10-11] draws lessons from the phenomenon of sudden jump of solid properties during annealing heat treatment and becomes a good global optimization method, but it has low efficiency and slow convergence speed. Ant colony algorithm [12-13] is a parallel algorithm with fast convergence speed and obvious advantages in dealing with complex combinatorial optimization problems, but it is prone to premature termination. Inspired by the predation behavior of birds, Particle Swarm Optimization (PSO) [14, 15] is a random iterative parallel algorithm, which is simple and easy to implement, has strong global search ability for nonlinear multi-peak problems, and has obvious advantages in solving multi-objective size optimization problems. However, the algorithm is limited to unconstrained optimization problems, and it is ignored to solve general multi-objective optimization problems with constraints.

Since the PSO does not have the ability to deal with constraint problems, it is often necessary to transform the constraint problems with unconstrained problems. An improved multi-objective particle swarm optimization algorithm (MOPSO) based on the concept of constraint domination is proposed in [16], which perturbs particles with small probability to improve the diversity of the algorithm. Mohamad et al. [17] proposed a MOPSO with good convergence and constraint processing ability for high-dimensional problems to solve complex engineering problems with many optimization variables. Li et al. [18] proposed an adaptive particle swarm optimization algorithm with different learning strategies to solve the path planning problem of mobile robot under different types of constraints in complex environment. Xu et al. [19] proposed an improved adaptive weighted PSO and applied it to multi-objective optimization design of planetary gears, which solved the problem that PSO is not easy to converge or fall into local optimum under complex constraints. It can be seen that many scholars have done extensive research on PSO in solving MOP and constrained

problems [20], but the above algorithms need to set parameters such as learning factor and inertia weight, which affects the computational efficiency of PSO.

A bare-bones particle swarm optimization algorithm (BB-PSO), using a Gaussian distribution of personal best positions and global best positions to update particle positions, was first proposed by Kennedy [21], which has the advantage of not setting inertia weights, learning factors and other control parameter. Zhang et al. [22] first extended the BB-PSO to multi-objective optimization problem, and the multi-objective economic/environmental scheduling problems with constraints are solved. However, the bare-bones multi-objective particle swarm optimization algorithm (BB-MOPSO) focuses more on the globality in particle update and global best positions selection, which makes the algorithm's optimization ability stronger, but the local development ability, boundary search ability and particle diversity are correspondingly reduced. Therefore, it needs to be further improved for the problems that there are many local extremum and the optimal solution may be located at the boundary when solving nonlinear multi-objective optimization with constraints.

Based on the above analysis, this paper proposes an improved BB-MOPSO to solve the multi-objective size optimization problems with non-linearity and constraints. The MOP and BB-MOPSO are introduced in Section 2 and Section 3. An improved BB-MOPSO is proposed in Section 4. The algorithmic complexity is given and the performance of solution ability, convergence and constraint processing are analyzed in Section 5. As a case study, a support frame of triangle track wheel is optimized in Section 6.

## 2. MULTI-OBJECTIVE OPTIMIZATION PROBLEM

A classical multi-objective optimization problem can be defined as finding a decision variable  $x^* = [x_1^*, x_2^*, \dots, x_n^*]$  that satisfies the following conditions:

Decision space:

$$x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, 2, \dots, n \quad (x_i \subseteq x) . \quad (1)$$

Equality constraint:

$$h_j(x) = 0, \quad j = 1, 2, \dots, J . \quad (2)$$

Inequality constraint:

$$g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K . \quad (3)$$

Minimized objective function:

$$f(x) = [f_1(x), f_2(x), \dots, f_m(x)] . \quad (4)$$

Function  $f(x)$  is called the objective function. Each component value of decision variable  $x$  is constrained by two boundary values  $x_i^{\min}$  and  $x_i^{\max}$ . The boundaries of all components of the decision variable constitute the decision space of multi-objective optimization. The output values of  $m$  objective functions constitute the objective space of multi-objective optimization. The decision variables that conform to all constraints

constitute the feasible region of multi-objective optimization, and the solution that conforms to Eq. (4) in the feasible region is called the optimal solution.

Different from the single objective optimization problem, the MOP contains many conflicting objective functions, and it is difficult for each objective to achieve the optimal at the same time. Therefore, its solution becomes a solution set containing infinite elements. Usually, we find the effective solution of this solution set, also known as Pareto solution, which is defined as: if there is not  $x \in X$ , make  $f(x) \leq f(x^*)$ , and then  $x^*$  is defined as an effective solution to the MOP.

### 3. BB-MOPSO

In the PSO, each particle represents a solution of the optimized problem. The position of the  $i$ -th particle in the  $n$ -dimensional space is represented as  $x_i=[x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ , and the velocity is represented as  $v_i=[v_{i,1}, v_{i,2}, \dots, v_{i,n}]$ . Each particle has an adaptation value determined by the optimization function, and it is known that the optimal value of the current particle individual is  $p_i=[p_{i,1}, p_{i,2}, \dots, p_{i,n}]$  and the global optimal value of the particle is  $g_i=[g_1, g_2, \dots, g_n]$  at the  $i$ -th iteration. At the next iteration, the particle determines the next motion state according to its own experience and the experience of particles in the same neighborhood until the end of the iteration. The velocity update equation is,

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1r_1(p_{i,j}(t) - x_{i,j}(t)) + c_2r_2(g_j(t) - x_{i,j}(t)). \quad (5)$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are learning factors.  $r_1$  and  $r_2$  are random obeying uniformly distributed  $U(0,1)$ ;  $j=1, 2, \dots, n$ , and  $i=1, 2, \dots, N$ ,  $N$  is the number of particles.

The particle position is,

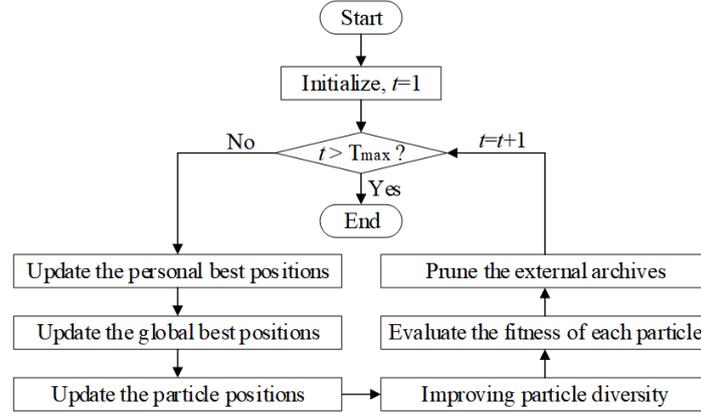
$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1). \quad (6)$$

BB-MOPSO has the same optimization idea as PSO, but it is different from traditional PSO. On the one hand, the control parameters such as inertia coefficient and learning factor are deleted to overcome the disadvantage that traditional PSO relies too much on parameters. The mathematical model to determine the updating position of particles is realized by a Gaussian sampling (normal distribution) of personal best positions and global best positions. The mathematical model of BB-MOPSO [18] is as follows:

$$x_{i,j}(t+1) = \begin{cases} N\left([r_1 \cdot p_{i,j}(t) + (1-r_1) \cdot g_{i,j}(t)]/2, |p_{i,j}(t) - g_{i,j}(t)|\right) & U(0,1) < 0.5 \\ p_{i,j}(t) & \text{others} \end{cases}. \quad (7)$$

On the other hand, PSO belongs to the single-objective optimization method, which has a definite optimal solution. BB-MOPSO belongs to the multi-objective optimization method. Due to the addition of a non-dominated relationship between each particle, the algorithm may get more than one set of optimal solutions (non-inferior solutions) after iteration. Therefore, for each particle, there will be more than one candidate point when updating the global best positions, and these candidate points do not dominate each other. All candidate points are stored in a set different from particle swarm, which is the external archive

mentioned above. When the particle position needs to be updated, the global best position of the particle is selected from the external archive, and the external archive element is also submitted to the decision maker as the final result of the algorithm. The flow of the algorithm is shown in Fig. 1, and the  $T_{\max}$  is the maximum number of iterations.



**Fig. 1** Flow chart of BB-MOPSO

#### 4. IMPROVED BB-MOPSO

Multi-objective size optimization is a nonlinear multi-objective optimization problem with constraints. There are many local extremum, and the Pareto solution may be at the boundary, and the constraint relationship is complex. Although BB-MOPSO has some advantages for solving such problems, but the algorithm focuses more on globality in particle update and global best positions selection, which reduces the algorithm's boundary and cross-region search ability and particle diversity. Therefore, the algorithm is difficult to find the global optimal solution when dealing with multi-objective size optimization problems.

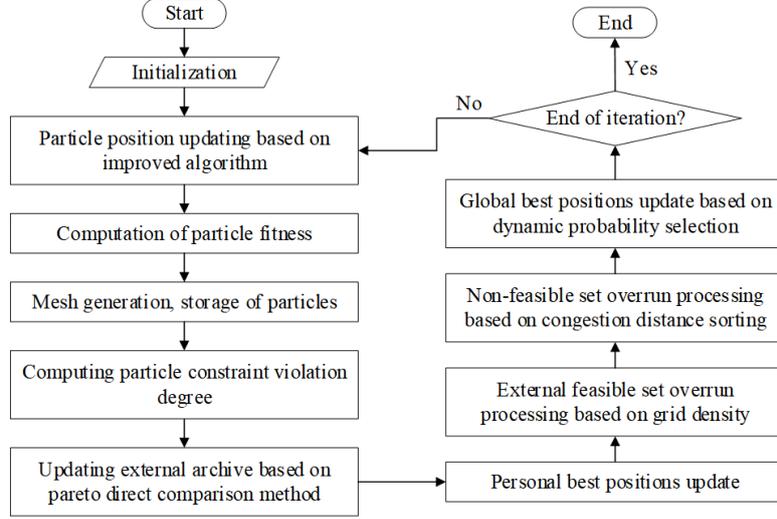
In this paper, the BB-MOPSO is improved from three aspects, including particle update mode, maintenance strategy of external archive and global best positions selection, to increase the algorithm's boundary and cross-region search ability and particle diversity. The calculation process of new algorithm is shown in Fig. 2, and the improvement of each part of the algorithm will be introduced in detail in sections 4.1~4.3.

##### 4.1 Particle Position Updates Approach

The particle position update of BB-MOPSO (Eq. 7) is more inclined to the global best positions. But for the size optimization problems, the optimal solution is likely to appear near the constrained boundary, especially the stress-based optimization. Therefore, in the process of particle updating, it is necessary to increase the local development of particles. In this paper, the development of individual particle guides and the randomness of gravitational factors are increased in this paper. The new update mode of particle position is shown in Eq. (8).

$$x_{i,j}(t+1) = \begin{cases} N\left([r_1 \cdot p_{i,j}(t) + r_2 \cdot g_{i,j}(t)]/2, |p_{i,j}(t) - g_{i,j}(t)|\right) & U(0,1) < 0.5 \\ p_{i,j}(t) & \text{others} \end{cases}, \quad (8)$$

where  $r_1$  and  $r_2$  are random numbers in the range of 0~1.



**Fig. 2** Flow chart of improved BB-MOPSO

The new update mode can make the particle have a 50% probability to select the relevant component of the current local optimal particle in each iteration calculation. At the same time, the random numbers  $r_1$  and  $r_2$  can improve the possibility of particles entering another feasible solution region, which expands the particle search range and theoretically increases the boundary and cross region search ability of the algorithm.

The updated particle  $x_i(t+1)$  needs to be compared with the historical optimal particle  $p_i(t)$  to select the personal best positions. The principle of selection is as follows: when  $x_i(t+1)$  dominates  $p_i(t)$ , then  $p_i(t+1)=x_i(t+1)$ ; when  $x_i(t+1)$  and  $p_i(t)$  do not dominate each other, anyone can be selected as  $p_i(t+1)$ , which can increase the probability of "excellent particles" entering the next generation; Otherwise,  $p_i(t+1)=p_i(t)$ .

#### 4.2 Maintenance Strategy of External Archive

The constraint dominance relation is constructed by direct comparison method in BB-MOPSO, which is easy to fall into local optimum in multi-island problem. In order to reduce the possibility of local optimum, the external archive is divided into two parts: feasible solution set and infeasible solution set. In this way, the development of isolated regions and the global capability of the algorithm can be enhanced [23].

After each update, the feasible solution set is updated first, that is, the particle set with constraint violation degree of 0. A new feasible solution set is composed of the newly obtained particles and the particles in the original feasible solution set. Pareto domination

relationship is analysed, and the particles that do not dominate each other are left. In order to ensure the diversity of learning particles, the original BB-MOPSO method uses crowded distances sorting [24] as the external archive maintenance strategy. It needs to calculate the Euclidean distance of each particle relative to other particles, which has a large computational complexity.

In this paper, the combination of spatial grid and crowded distance sorting are used. The objective space is first divided into  $M_1 \times M_2 \times \dots \times M_k$  grid,  $k$  is the objective dimension,  $M_k$  is the grid number of each dimension. The grid position of each particle is calculated to determine the grid density, which can reduce the computational complexity and maintain the optimal solution distribution in the global range.

The grid density can be calculated by the following method. The upper bounds  $F_i^{\max}$  and lower bounds  $F_i^{\min}$  of the objective space are given by analysing the values range of each dimension. The objective width of  $i$ -th dimension of each grid is

$$d_i = (F_i^{\max} - F_i^{\min}) / M_i. \quad (9)$$

Suppose the coordinate of a particle corresponding to the objective space is  $s=[s_1, s_2, \dots, s_m]$ , then the particle position in each dimension of the objective space is

$$L_i = \text{fix}(s_i - F_i^{\min}, d_i) + 1, \quad (10)$$

where  $\text{fix}(a, b)$  is a rounding function.

Each grid is numbered and placed into different grids according to the position values of each dimension. The number of particles put into the grids is recorded as  $N(j, t)$ , which indicates the number of particles in the region where the  $j$ -th particle is located at time  $t$ . Suppose the maximum theoretical capacity of the grid is  $N_{pi}$ , the density is

$$\rho_i^j = N(j, t) / N_{pi}. \quad (11)$$

When the total number of feasible solutions in a certain grid is larger than the grid capacity, the crowding distance of each particle is calculated and  $N_{pi}$  with larger distance as the new external archive particles are selected, to ensure the distribution of the solution. The crowding distance can be calculated by the following method. The objective values of each dimension are sorted from small to large, and composition of objective value sequence  $[f_{i,1}, f_{i,2}, \dots, f_{i,p}, \dots, f_{i,sn}]$ , then the crowding distance of a objective value is

$$dy_p = \sum_{i=1}^k dy_{i,p} = \frac{1}{2} \sum_{i=1}^k (f_{i,p+1} - f_{i,p-1}) / (f_i^{\max} - f_i^{\min}), \quad (12)$$

where  $f_i^{\max}$  and  $f_i^{\min}$  are represent the maximum and minimum values of the  $i$ -th objective in the current archive.  $dy_{i,p}$  is the crowding distance of the  $p$ -th particle in the objective value of  $i$ -dimension.  $sn$  is the size of particle swarm.

For the update of the infeasible solution set (the particle set with constraint violation degree greater than 0), the Pareto dominance relationship analysis is first performed to obtain the mutually un-dominated particles. And then, the adaptive grid technique is used to divide the particles. For the grid exceeding the capacity, instead of congestion distance sorting,  $N_{pi}$  particles are randomly selected and re-placed into the grid. Although this

approach may reduce the diversity of infeasible solution set, it can still ensure the ability of the algorithm to explore the unknown feasible region and reduce the computational complexity.

#### 4.3 Global Best Positions Selection

Whether the selection of global best positions is reasonable or not is related to the ability of the algorithm to accurately converge to the Pareto front. In the BB-MOPSO, the global best positions for each particle are selected by the tournament selection method with scale of 2 based on the crowding distance value. The larger the congestion distance value, the more likely it is selected as the global best positions. In this paper, the dynamic probability selection method is used to give the global best positions, that is, the global best positions are selected from the infeasible solution set and the feasible solution set with the probability of  $p_{sl}$  and  $1-p_{sl}$ .

$$P_{sl} = P_{sl1} - P_{sl2} \cdot t / T, \quad (13)$$

where  $p_{sl1}$  and  $p_{sl2}$  are dynamic probability constants,  $p_{sl1}=0.7$  and  $p_{sl2}=0.6$ .  $t$  is the current number of iterations.  $T$  is the total number of iterations.

In this way, the algorithm has sufficient global search ability. The diversity of learning particles in the early stage is improved, and the convergence in the later stage is accelerated. After determining the global best positions, a probability selection equation (Eq. 14) is constructed based on the density values of each grid, so that the probability of selecting a value with small density is high and that of selecting a value with large density is low. The global best positions selected in this way will have good distribution and convergence.

$$P_i^t = 1 - \rho_i^t / \left( \sum_{i=1}^{ks} \rho_i^t \right), \quad (14)$$

where  $p_i^t$  is the final choice probability.  $ks$  is total number of grids.

#### 4.4 Analysis of Algorithmic Complexity

The computational cost of the improved BB-MOPSO mainly concentrates on the update of the reserve set. Suppose an optimization problem with  $M$  objectives and  $K$  constraints, the number of feasible and infeasible solutions in the new particle swarm optimization is  $N_p$  and  $N_q$ , the feasible reserve capacity is  $N_a$ , and the infeasible reserve capacity is  $N_b$ .

Firstly, the renewal process of feasible reserve set is analysed. The size of the new population is  $N_p+N_a$ , and the number of comparisons is  $M \times (N_p+N_a)$  to judge whether a particle is advantages and disadvantages. Considering the worst case (the particles in the population are not dominated by each other), it takes  $M \times (N_p+N_a)^2$  comparisons to select all non-inferior solutions from  $N_p+N_a$  particles. On the other hand, the computational complexity of using crowded distance to maintain the reserve set is  $\mathcal{O}(M \times (N_p+N_a) \times \log(N_p+N_a))$  and  $\log(N_p+N_a) < N_p+N_a$  [25]. As a result, the computational complexity of updating the feasible reserve set is  $\mathcal{O}(M \times (N_p+N_a)^2)$ .

Next, the update process of infeasible reserve set is analyzed. To judge whether a particle in the new population (new infeasible solution and infeasible reserve set) is dominated by the feasible solution, the number of comparisons is  $M \times (N_q+N_b)^2$ . Because

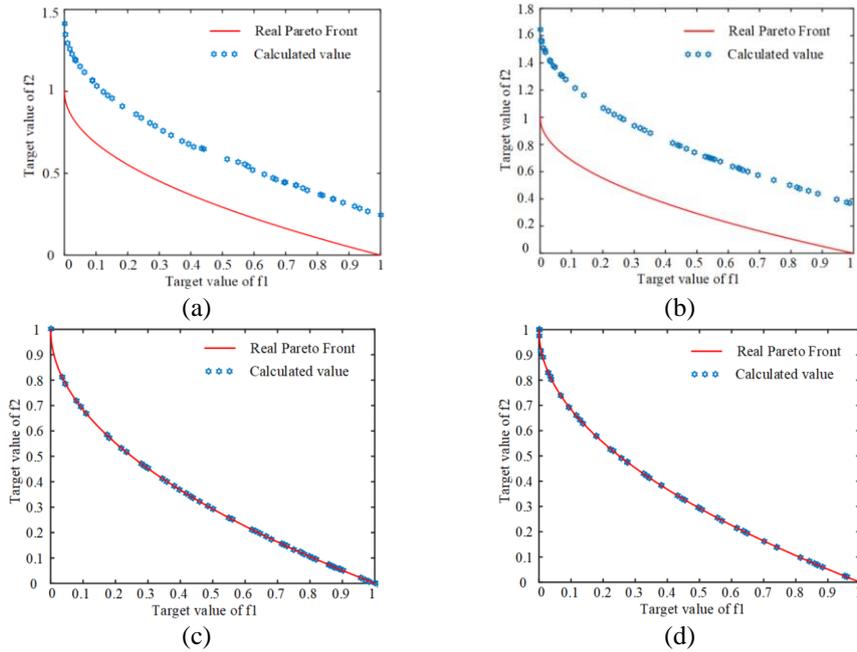
there is no need for crowded distance, the computational complexity of updating infeasible reserve set is  $\mathcal{O}(M \times (N_q + N_b)^2)$ .

In summary, the computational complexity of the improved BB-MOPSO is  $\mathcal{O}(M \times (N_p + N_a)^2) + \mathcal{O}(M \times (N_q + N_b)^2)$ .

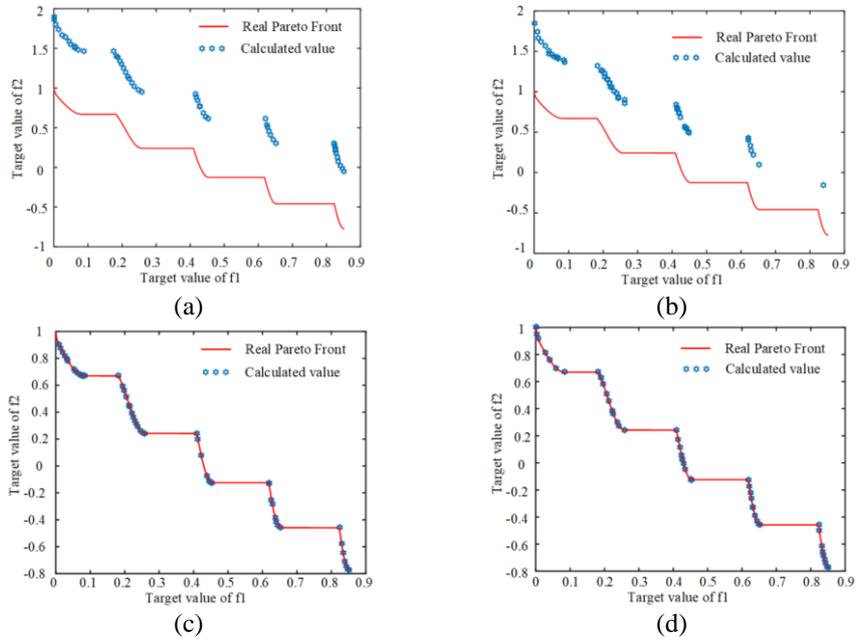
## 5. PERFORMANCE ANALYSIS OF IMPROVED BB-MOPSO

The performance of BB-MOPSO have been tested by a large number of functions and compared with other intelligent algorithms [22]. Therefore, the performance of improved BB-MOPSO was compared with original BB-MOPSO, NSGA2 and SPEA2 in this paper. Although the improved BB-MOPSO is proposed for the problem with constraints, it does not change the constraint processing method. So, the classical functions of ZDT1, ZDT3 and ZDT4 [26] are used to test the ability of the algorithm to solve convex functions, the ability to solve discontinuous and multi-connected domain functions, and the ability to solve global optimization. In addition, the function of DTLZ3 [27] is used to test the ability to converge to global frontier.

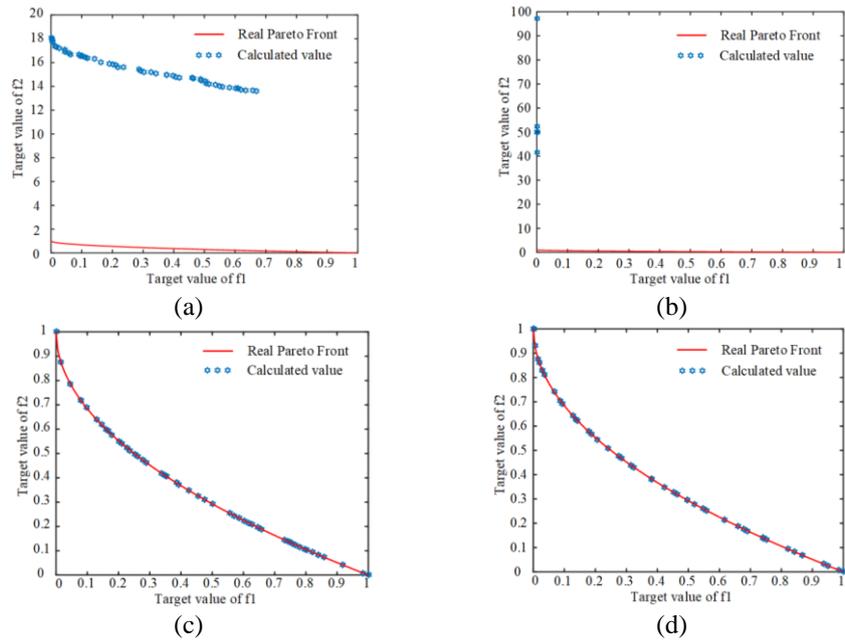
In the test, the algorithm runs 30 times independently for each function. ZDT1 and ZDT3 are taken as 100 dimensions, ZDT4 is 30 dimensions, particle number is 100, and iteration number is 300. DTLZ3 is taken as 10 dimensions particle number is 500, iteration number is 1000, external archive capacity is 50, and grid capacity is 10. The Pareto fronts by each algorithm are shown in Figs. 3~6.



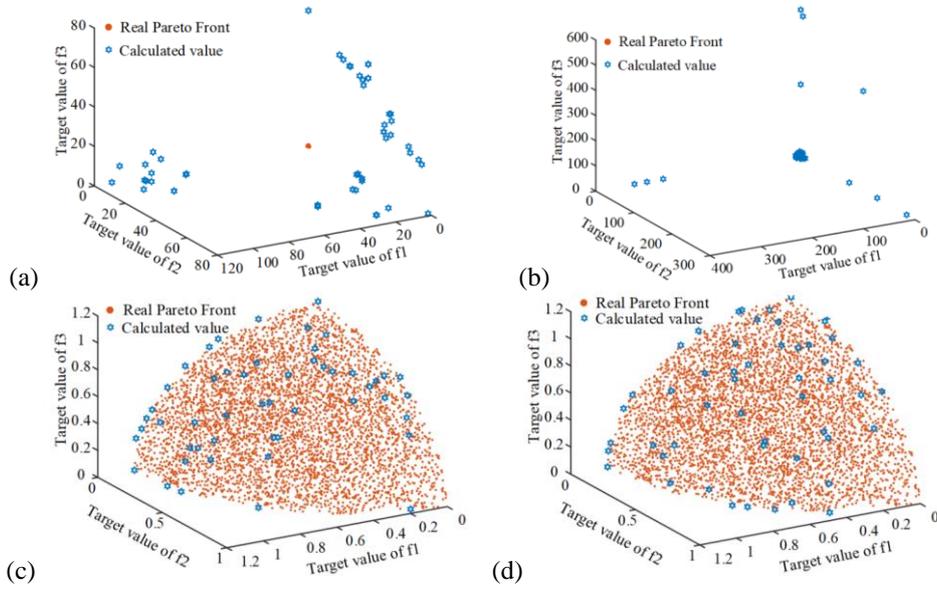
**Fig. 3** Pareto front of test function ZDT1 (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO



**Fig. 4** Pareto front of test function ZDT3 (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO



**Fig. 5** Pareto front of test function ZDT4 (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO



**Fig. 6** Pareto front of test function DTLZ3 (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO

As can be seen from Figs. 3-6, neither NSGA2 nor SPEA2 can find the real frontier in the above test functions. In addition, the solution falls into the local optimum. The improved algorithm is significantly different from the original algorithm in the distribution of the solution. The solution of the improved algorithm converges more uniformly to the frontier. The performance of the four algorithms is quantified by calculating the values of Spacing Metric (SP) and Generational Distance Metric (GD) [26, 28]. The SP reflects the distribution and diversity of feasible solution set in the objective space, while the GD reflects the distance between feasible solution and Pareto frontier, as shown in Tables 1-4, where the best results are given in bold.

**Table 1** Performance statistics of ZDT1

	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
GDAV	1.53E-4	<b>1.49E-4</b>	4.99E-2	5.09E-2
GDVAR	5.35E-5	<b>5.31E-5</b>	5.21E-2	5.10E-3
GDBEST	1.53E-4	<b>1.09E-4</b>	1.55E-2	3.97E-2
GDWORST	4.38E-4	<b>4.05E-4</b>	2.64E-1	6.04E-2
SPAV	<b>1.67E-2</b>	2.42E-2	1.64E-2	2.25E-2
SPVAR	<b>1.70E-3</b>	6.60E-3	3.00E-3	1.08E-2
SPBEST	1.35E-2	<b>1.30E-2</b>	1.15E-2	1.22E-2
SPWORST	<b>1.97E-2</b>	4.19E-2	2.46E-2	5.91E-2

**Table 2** Performance statistics of ZDT3

	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
GDAV	<b>6.53E-4</b>	7.74E-4	7.87E-2	5.31E-2
GDVAR	<b>4.57E-5</b>	5.60E-5	4.64E-2	6.50E-3
GDBEST	<b>5.87E-4</b>	6.78E-4	2.75E-2	4.38E-2
GDWORST	<b>7.10E-4</b>	8.60E-4	2.15E-1	7.50E-2
SPAV	<b>2.05E-2</b>	2.91E-2	2.07E-2	3.21E-2
SPVAR	<b>2.50E-3</b>	4.70E-3	4.60E-3	1.78E-2
SPBEST	<b>1.54E-2</b>	2.13E-2	1.02E-2	1.59E-2
SPWORST	<b>2.78E-2</b>	4.33E-2	2.97E-2	7.62E-2

**Table 3** Performance statistics of ZDT4

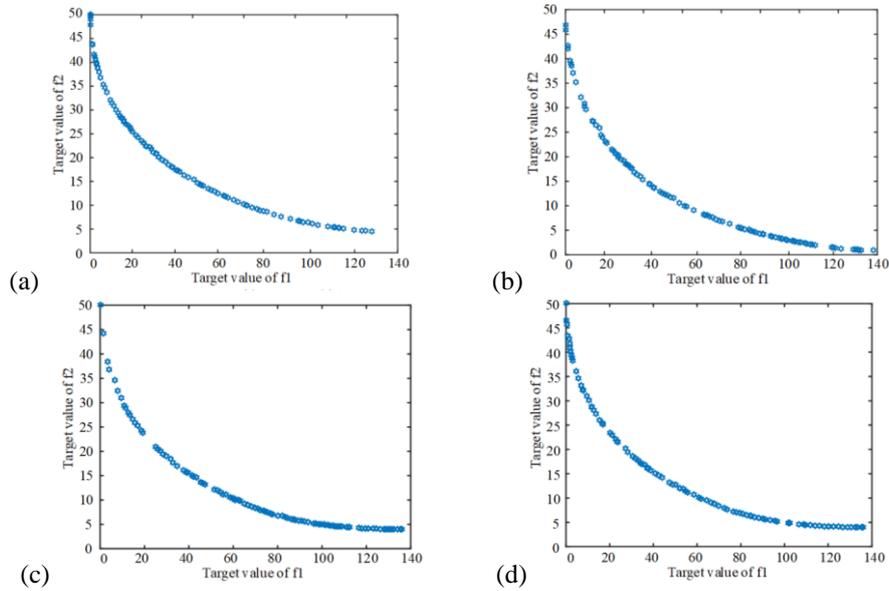
	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
GDAV	<b>2.91E-4</b>	3.38E-4	2.73E+0	6.81E+0
GDVAR	<b>4.46E-4</b>	5.70E+4	1.51E+0	3.96E+0
GDBEST	<b>1.80E-4</b>	2.80E-4	6.55E-1	2.50E+0
GDWORST	<b>4.34E-4</b>	5.80E-4	5.58E+0	1.66E+1
SPAV	<b>1.80E-2</b>	2.58E-2	1.50E-2	1.41E+0
SPVAR	<b>1.30E-3</b>	7.80E-3	1.61E-2	5.37E+0
SPBEST	<b>1.39E-2</b>	1.42E-2	0.00E+0	0.00E+0
SPWORST	<b>2.04E-2</b>	4.36E-2	4.98E-2	2.29E+1

**Table 4** Performance statistics of DTLZ3

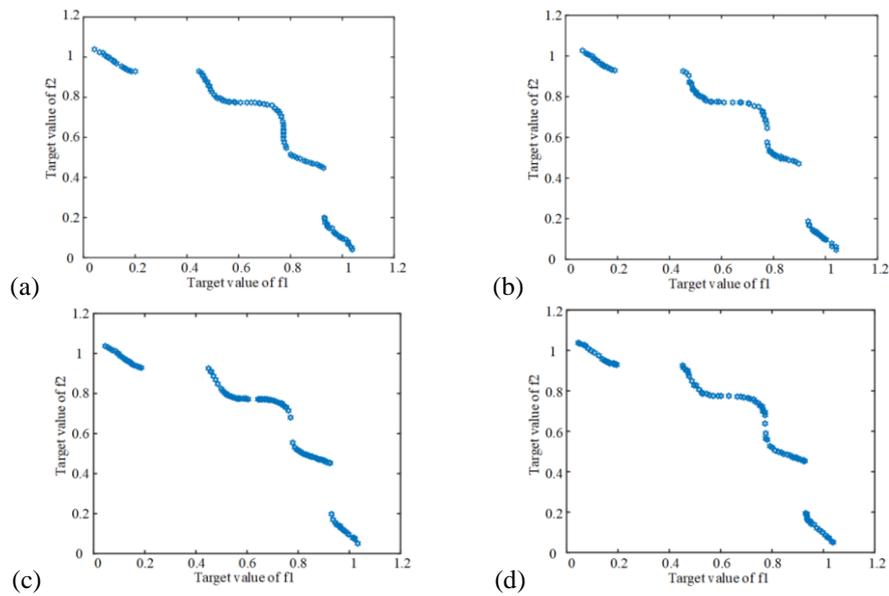
	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
GDAV	<b>3.70E-3</b>	4.50E-3	7.45E+0	1.92E+1
GDVAR	<b>5.10E-4</b>	7.95E-4	2.42E+0	6.12E+0
GDBEST	<b>2.70E-3</b>	3.50E-3	4.01E+0	7.34E+0
GDWORST	<b>3.16E-3</b>	5.50E-3	1.42E+1	2.91E+1
SPAV	<b>7.73E-2</b>	8.61E-2	4.78E+0	8.50E+1
SPVAR	<b>4.90E-3</b>	6.40E-3	5.01E+0	3.59E+1
SPBEST	<b>6.53E-2</b>	7.82E-2	6.17E-1	2.00E+1
SPWORST	<b>8.94E-2</b>	9.20E-2	1.87E+1	1.73E+2

Tables show that the distribution and diversity of feasible solutions of the improved algorithm are much stronger than that of the original algorithm when dealing with ZDT1, ZTD3, ZTD4 and DTLZ3. Then, the functions of BNH, TNK and DTLZ8 [27, 29] are used to test the ability of constraints processing for the improved BB-MOPSO. BNH and TNK are set as follows: particle number is 100, iteration number is 500, maximum capacity of reserve set is 100, and grid capacity is 10. DTLZ8 is set as follows: particle number is 500,

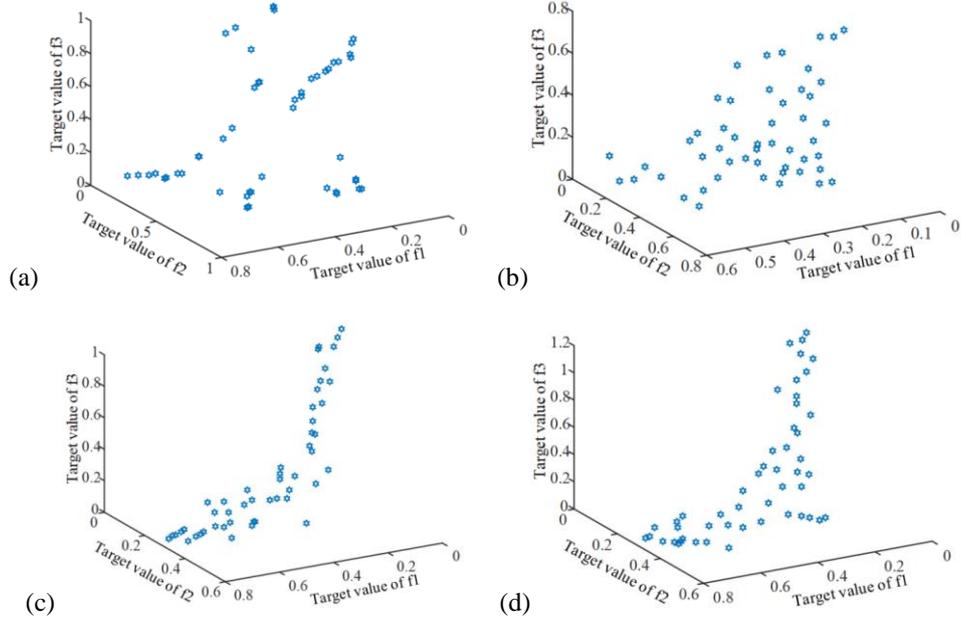
iteration number is 1000. Similarly, the algorithm runs 30 times independently for each function. The Pareto front by each algorithm is shown in Figs. 7-9.



**Fig. 7** Pareto front of test function BNH (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO



**Fig. 8** Pareto front of test function TNK (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO



**Fig. 9** Pareto front of test function DTLZ8 (a) NSGA2 (b) SPEA2 (c) BB-MOPSO (d) Improved BB-MOPSO

In order to express the proportion of the elements that are dominant to each other in the solution set of the improved algorithm and other algorithms, the C measure [30] is used for the test functions of BNH, TNK and DTLZ8. The results are shown in Tables 5-7 and the best results are marked in bold.

**Table 5** Performance statistics of TNK

	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
C(IM-BB)	-	3.50E-1	2.32E-1	1.63E-1
C(BB-D)	<b>3.72E-1</b>	-	-	-
C(NSGA2)	<b>3.25E-1</b>	-	-	-
C(SPEA2)	<b>2.36E-1</b>	-	-	-
SPAV	6.70E-3	7.00E-3	<b>5.60E-3</b>	7.20E-3
SPVAR	<b>8.56E-4</b>	1.00E-3	9.12E-4	1.90E-3
SPBEST	4.50E-3	5.10E-3	<b>2.40E-3</b>	4.60E-3
SPWORST	8.40E-3	9.00E-3	<b>7.10E-3</b>	1.47E-2

As can be seen from Tables 5-7, the distribution of solutions and the dominant number of solutions of the improved algorithm are better than those of other algorithms when dealing with BNH, TNK and DTLZ8. Although the distribution of NAGA2 solutions is better when dealing with TNK, the distribution of solutions of the improved algorithm is

more stable. This benefits from external archive maintenance based on grid density and congestion distance sorting, which maintains a certain number of solutions in each region.

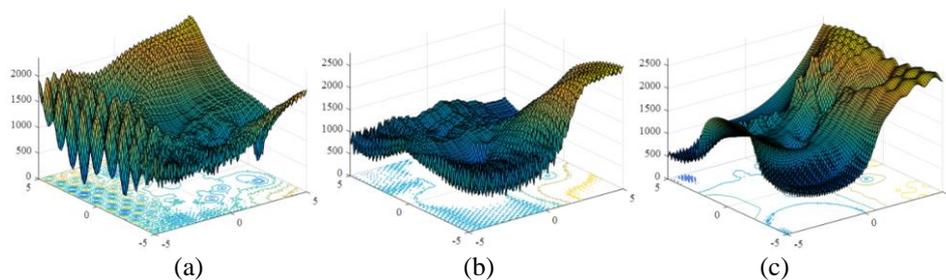
**Table 6** Performance statistics of BNH

	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
C(IM-BB)	-	7.40E-2	9.50E-2	2.47E-1
C(BB-D)	<b>9.97E-2</b>	-	-	-
C(NSGA2)	<b>2.03E-1</b>	-	-	-
C(SPEA2)	<b>3.70E-1</b>	-	-	-
SPAV	<b>8.42E-1</b>	8.77E-1	8.65E-1	9.40E-1
SPVAR	4.34E-2	1.66E-1	6.75E-2	<b>2.51E-1</b>
SPBEST	<b>5.07E-1</b>	6.20E-1	5.15E-1	6.13E-1
SPWORST	<b>6.97E-1</b>	1.39E+0	7.74E-1	1.57E+0

**Table 7** Performance statistics of DTLZ8

	<b>IM-BB-MOPSO</b>	<b>BB-MOPSO</b>	<b>NSGA2</b>	<b>SPEA2</b>
C(IM-BB)	-	1.50E-1	3.55E-2	3.35E-2
C(BB-D)	<b>1.70E-1</b>	-	-	-
C(NSGA2)	<b>1.30E-1</b>	-	-	-
C(SPEA2)	<b>9.40E-2</b>	-	-	-
SPAV	<b>2.99E-2</b>	3.11E-2	3.60E-2	3.79E-2
SPVAR	<b>4.44E-3</b>	5.00E-3	9.40E-3	6.90E-3
SPBEST	<b>1.91E-2</b>	2.16E-2	2.34E-2	2.37E-2
SPWORST	<b>4.13E-2</b>	4.17E-2	5.80E-2	5.24E-2

In order to further verify the optimization ability of the algorithm, the hybrid composition functions of CF4, CF5 and CF6 provided by Liang et al. [31] are used to calculate. The shape of three functions is shown in Fig. 10.



**Fig. 10** The shape of three functions (a) CF4 (b) CF5 (c) CF6

For each test function, each algorithm is run 20 times and the maximum fitness evaluations are set at 50,000 for all algorithms. The results are compared as shown in Table 8. The first six groups of data about functions of PSO, CPSO, CLPSO, CMA-ES, G3-PCX, DE are from reference [31]. The results show that the improved algorithm has good performance in dealing with complex functions. However, the stability of the solution is low when dealing with CF6 function.

**Table 8** Performance statistics of CF4, CF5 and CF6

	PSO	CPSO	CLPSO	CMA-ES	G3-PCX	DE	IM-BBMOPSO
CF4(AV)	3.14E+2	5.22E+2	3.22E+2	6.16E+2	4.93E+2	3.25E+2	<b>3.10E+2</b>
CF4(STD)	2.01E+1	1.22E+2	2.75E+1	6.72E+2	1.42E+2	1.48E+1	<b>7.55E+0</b>
CF5(AV)	8.35E+1	2.56E+2	5.37 E+0	3.59E+2	2.60E+1	1.08E+1	<b>3.22E+0</b>
CF5(STD)	1.01E+0	1.76E+2	2.61E+0	1.68E+2	4.16E+1	2.60E+1	<b>7.42E-1</b>
CF6(AV)	8.61E+2	8.53E+2	5.01E+2	9.00E+2	7.72E+2	4.91E+2	<b>4.26E+2</b>
CF6(STD)	1.26E+2	1.28E+2	<b>7.78E-1</b>	8.32E-2	1.89E+2	3.95E+1	1.07E+1

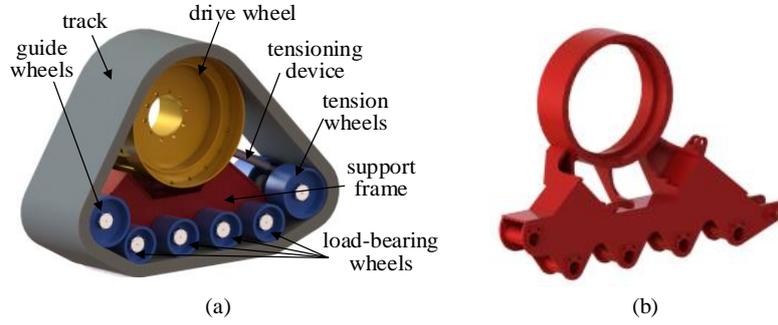
As can be seen from the above data, the convergence of the improved algorithm is slightly weaker when dealing with ZDT1. But the convergence of the improved algorithm is slightly stronger for dealing with ZTD3, ZTD4 and DTLZ3. When dealing with BNH, TNK and DTLZ8, the improved algorithm has a higher proportion of dominance. When dealing with CF4, CF5 and CF6 functions, the search results of the improved algorithm are best, this shows that the improved algorithm has a higher accuracy. This is due to the updating of particle location for boundary search and the strategy of selecting local particles, which enhances the ability of developing local particles.

Based on the above analysis, the new algorithm proposed in this paper has better applicability and accuracy for multi-connected and multi-extreme complex problems, and can be used as a good tool for solving multi-objective size optimization problems with constraints.

## 6. CASE ANALYSIS

As a case study, a support frame of triangle track wheel is optimized by the new algorithm proposed in this paper to verify the applicability and accuracy. The structure is shown in Fig. 11, which can be found in reference [32]. In the process of travelling, the triangle track wheel has various working conditions such as climbing, crossing obstacle, starting, braking, turning, crossing the soft road, snow, muddy land, marsh, sand and so on. The support frame has different load-bearing mode under different working-conditions. The three wheel supported ground is the worst working condition, which is considered in this paper. The loads borne by the support frame include: self-gravity 4.5 kN, hydraulic cylinder preload 36 kN, and chassis pressure 78.4 kN. The weight is 0.366 tons. The stiffness and strength of the support frame are more sensitive to the change of mass, and the restriction relationship is more significant. Local extremum and non-convergence is easy to occur in the calculation process when the stress is taken as performance index, which is a

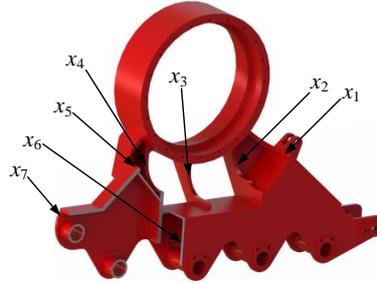
nonlinear MOP with constraints. In this section, multi-objective size optimization of support frame is carried out.



**Fig. 11** The structure of support frame (a) Triangle track wheel (b) Support frame

### 6.1 Mathematical Model of Multi-objective Size Optimization

The parametric model of the support frame is established as shown in Fig. 12. Seven important dimensions (plate thickness of local area) are marked including  $x_1 \sim x_7$ , which are design variables.



**Fig. 12** Parametric model of the support frame

According to the design and manufacturing process [33], each size parameter of the support frame is constrained, and the first-order natural frequency should exceed the excitation frequency transmitted by the engine. The constraints can be expressed as

$$s.t. \begin{cases} x_{i\min} \leq x_i \leq x_{i\max}, i = 1, 2, \dots, 7 \\ f(x_i) \geq f_0 \end{cases}, \quad (15)$$

where  $x_{i\min}$  and  $x_{i\max}$  are the minimum and maximum value of  $x_i$ .  $f(x_i)$  is the first-order natural frequency of the support frame.  $f_0$  is the excitation frequency of engine, the value is 50 Hz.

For the support frame, the optimization objective is how to reduce the weight, displacement and stress. So, the objective function can be defined as shown in Eq. (16):

$$\min [\sigma_{\max}(x_i), d_{\max}(x_i), V(x_i)], \quad (16)$$

where  $\sigma_{\max}(x_i)$  and  $d_{\max}(x_i)$  are maximum stress and displacement of support frame.  $V(x_i)$  is the weight.  $x_i$  is the design parameter.

So, the mathematical model of size optimization is established, as shown in Eq. (17), which belongs to a typical nonlinear MOP with constraints.

$$\begin{cases} \text{Find } x_i (i = 1, 2, 3, 4, 5, 6, 7) \\ \min \sigma_{\max}(x_i), d_{\max}(x_i), V(x_i) \\ \text{s.t. } x_{i\min} \leq x_i \leq x_{i\max} \\ f(x_i) \geq f_0 \end{cases} \quad (17)$$

In the optimization model, the stress, displacement, volume and natural frequency of the support frame lack a functional relationship with the design variables. Therefore, the approximate model of structural performance and parameters is established in the next section.

## 6.2 Setting an Approximate Model

The response surface methodology is a method to establish an approximate model. It synthesizes experimental design and mathematical model, and obtains the relationship between design objectives and design variables through limited experimental design of the set of sample points in the designated design space. It is also possible to smooth the response function and reduce the "numerical noise", which is conducive to faster convergence to the global optimum in optimization process.

In this paper, a second-order polynomial is used to construct the response surface methodology. The basis function is as follows:

$$Y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{\substack{i=1 \\ i < j}}^k \beta_{ij} x_i x_j, \quad (18)$$

where  $\beta$  is an unknown coefficient;  $k$  is the number of design variables,  $k=7$ ;  $Y$  is the predicted response value;  $\beta_0$ ,  $\beta_i$  and  $\beta_{ii}$  are deviation term coefficient, linear deviation term coefficient and second-order deviation term coefficient, respectively;  $\beta_{ij}$  is the interaction coefficient.

Four approximate models of response are obtained through experimental design as follows:

$$\begin{aligned} f_1 &= X^T \cdot C_1 \\ f_2 &= X^T \cdot C_2 \\ f_3 &= X^T \cdot C_3 \\ f_4 &= X^T \cdot C_4 \end{aligned}, \quad (19)$$

where  $f_1, f_2, f_3, f_4$  represent the approximate model of stress, natural frequency, displacement and volume, respectively. The parameters of approximate model are given in Table 9.

**Table 9** Parameters of approximate model

$X$	$C_1$	$C_2$	$C_3$	$C_4$
Constant	2.60E+2	3.36E+1	3.30E-1	2.80E+7
$x_1$	9.84E-1	3.73E-1	-1.10E-2	3.81E+5
$x_2$	-5.42E+0	2.44E-1	-5.00E-3	3.82E+5
$x_3$	2.57E+0	1.19E+0	-4.00E-3	7.96E+4
$x_4$	-1.04E+1	1.79E-1	3.00E-3	4.59E+4
$x_5$	-7.52E+0	3.81E-1	1.00E-2	5.39E+4
$x_6$	-2.11E+0	7.36E-1	-5.00E-3	6.87E+4
$x_7$	-1.82E+0	2.19E-1	2.00E-3	5.16E+4
$x_1^2$	-1.03E-1	-3.00E-3	0	4.29E+2
$x_2^2$	1.10E-1	-2.00E-3	0	4.53E+2
$x_3^2$	-7.80E-2	-2.10E-2	0	6.26E+2
$x_4^2$	2.34E-1	2.00E-3	0	4.89E+2
$x_5^2$	1.42E-1	-5.00E-3	0	5.82E+2
$x_6^2$	4.70E-2	-1.30E-2	0	9.01E+2
$x_7^2$	1.00E-2	1.00E-3	0	5.20E+2
$x_1 \cdot x_2$	8.40E-2	-6.00E-3	0	-8.96E+1
$x_1 \cdot x_3$	-3.90E-2	-1.00E-3	0	1.69E+1
$x_1 \cdot x_4$	2.80E-2	1.00E-3	0	-1.70E+2
$x_1 \cdot x_5$	2.10E-2	-1.00E-3	0	-1.41E+2
$x_1 \cdot x_6$	1.00E-3	-1.00E-3	0	-3.64E+2
$x_1 \cdot x_7$	-2.20E-2	0	0	-1.45E+2
$x_2 \cdot x_3$	1.40E-2	3.00E-3	0	9.87E+0
$x_2 \cdot x_4$	2.10E-2	1.00E-3	0	-1.74E+2
$x_2 \cdot x_5$	4.60E-2	0	0	-1.63E+2
$x_2 \cdot x_6$	7.00E-3	2.00E-3	0	-3.91E+2
$x_2 \cdot x_7$	-5.00E-3	-1.00E-3	0	-1.38E+2
$x_3 \cdot x_4$	-2.70E-2	-2.00E-3	0	-1.19E+1
$x_3 \cdot x_5$	4.50E-2	-3.00E-3	0	1.44E+0
$x_3 \cdot x_6$	-3.70E-2	0	0	1.01E+1
$x_3 \cdot x_7$	-1.70E-2	-1.00E-3	0	3.03E-1
$x_4 \cdot x_5$	4.40E-2	-1.00E-3	0	-2.86E+1
$x_4 \cdot x_6$	6.00E-3	-1.00E-3	0	-3.97E+1
$x_4 \cdot x_7$	5.00E-2	-2.00E-3	0	-8.88E+0
$x_5 \cdot x_6$	3.00E-3	0	0	-6.02E+0
$x_5 \cdot x_7$	2.00E-3	0	0	6.68E+0
$x_6 \cdot x_7$	4.80E-2	3.00E-3	0	1.47E+1

After establishing the approximate model, it is necessary to evaluate its predictive power. Commonly used evaluation index are mean error (AE), maximum error (ME), root mean square error (RMSE), and correlation coefficient ( $R^2$ ). AE is the average of all errors and can be expressed as

$$AE = \frac{\sum_{i=1}^N (f_i - f'_i)}{N}, \quad (20)$$

where  $i$  is the  $i$ -th sample point;  $f_i$  is the finite element analysis result of the  $i$ -th sample point;  $f'_i$  is the calculated value of the approximate model of the  $i$ -th sample point;  $N$  is the number of sample points. The parameters of the approximate model are given in Table 4.

ME is the maximum value of all errors, that is  $\max (f_i - f'_i)$ . RMSE also known as standard error and can be expressed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (f_i - f'_i)^2}{N - p - 1}}, \quad (21)$$

where  $p$  is the number of terms of the polynomial.

$R^2$  is an index for evaluating the fitting accuracy of the approximate model to the experimental data. The closer  $R^2$  is to 1, the smaller the error.  $R^2$  can be described as:

$$R^2 = \frac{\sum_{i=1}^N (f_i - \bar{f})^2 - \sum_{i=1}^N (f_i - \bar{f}'_i)^2}{\sum_{i=1}^N (f_i - \bar{f})^2}, \quad (22)$$

where  $\bar{f}$  is the average of the finite element analysis of all sample points.

Table 10 shows the error analysis results of approximate model for different performance functions. It shows that the AE of the four responses is less than 0.1, ME is less than 0.3, RSME is less than 0.2,  $R^2$  is greater than 0.9, which shows that the approximate model has high prediction accuracy. Therefore, the approximate model can be used for subsequent optimization design. It is not difficult to see that the accuracy of stress prediction is little poor, because stress is easy to produce numerical errors and has high non-linearity. While the volume prediction accuracy is very high, because volume function of structure is an explicit function of design variables in size optimization.

**Table 10** Error analysis

Variables	Mass	Stress	Displacement	Mode
AE	1.76E-3	5.04E-2	2.04E-2	2.85E-2
ME	5.12E-3	2.05E-1	6.58E-2	6.52E-2
RMSE	2.17E-3	6.42E-2	2.70E-2	3.39E-2
$R^2$	9.99E-1	9.23E-1	9.86E-1	9.83E-1

### 6.3 Multi-Objective Size Optimization

In this section, the mathematical model in Eq. (17) are optimized by using the BB-MOPSO and the improved BB-MOPSO. The optimization results of design variables are shown in Table 11.

**Table 11** The results of size optimization

Method	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
BB-MOPSO	20	13.96	10	10	14.68	10	19.69
Improved BB-MOPSO	20	10	11.15	16.49	10	10	20

The model is modified according to the optimization results in Table 11. A new model is established and the finite element analysis of the worst working conditions is carried out. The results using the BB-MOPSO and the improved BB-MOPSO are listed in Table 12 to compare the applicability and accuracy for multi-connected and multi-extreme complex problem.

**Table 12** The comparison of mechanical performance

Method	Mass (kg)	Displacement (mm)	Stress (MPa)
Not optimized	376.65	0.217	132.90
BB-MOPSO	337.28	<b>0.196</b>	73.111
Improved BB-MOPSO	<b>335.70</b>	0.217	<b>68.167</b>

From Table 12, the mechanical performance of the support frame is improved by optimization using BB-MOPSO and improved BB-MOPSO. However, the optimization model using BB-MOPSO obtains better results in displacement, but it is worse in mass and stress than the improved BB-MOPSO. Generally, displacement reflects the overall stiffness of the structure, while stress reflects the local strength. The results of Table 12 show that the improved BB-MOPSO has better boundary and cross-region search ability. So, the improved BB-MOPSO is more suitable for structures with requirements of light weight and high strength.

## 7. CONCLUSION

In this paper, an improved BB-MOPSO is proposed to solve the multi-objective size optimization problems with non-linearity and constraints in structural design and optimization. The ability of particle searching for boundary and cross-region is enhanced by modifying the updating form of particle location. The optimal solution distribution in the global scope is maintained by combining the spatial grid density with the ranking of crowding distance. The global best positions are determined by increasing the probability allocation strategy changing with time, and the exploration of unknown region and the

convergence of the algorithm are increased. The performance of the improved algorithm is verified by the test functions. As a case study, a support frame of triangle track wheel is optimized by the BB-MOPSO and improved BB-MOPSO. The results show that the improved algorithm can enhance the search ability of boundary and cross-region and diversity, which can effectively solve the multi-objective size optimization problem with non-linearity and constraints.

**Acknowledgement:** This research is supported by Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (2021L291), Fundamental Research Program of Shanxi Province (202103021223290), Taiyuan University of Science and Technology Scientific Research Initial Funding (20212040) and National Natural Science Foundation of China (52205278).

#### REFERENCES

1. Wu, F.H., Wang, Z.H., Song, D.Z., Lian, H., 2022, *Lightweight design of control arm combining load path analysis and biological characteristics*, Reports in Mechanical Engineering, 3(1), pp. 71-82.
2. Safaei, B., Onyibo, E.C., Hurdoganoglu, D., 2022, *Effect of static and harmonic loading on the honeycomb sandwich beam by using finite element method*, Facta Universitatis-Series Mechanical Engineering, 20(2), pp. 279-306.
3. Lu, Z.Y., Wang, N., Yang, C.G., 2021, *A novel iterative identification based on the optimised topology for common state monitoring in wireless sensor networks*, International Journal of Systems Science, 53(1), pp. 25-39.
4. Gao, K., Do, D.M., Chu, S., Wu, G., Kim, H.A., Featherston, C.A., 2022, *Robust topology optimization of structures under uncertain propagation of imprecise stochastic-based uncertain field*, Thin-Walled Structures, 175, 109238.
5. Bekda, G., Nigdeli, S.M., Yang, X.S., 2015, *Sizing optimization of truss structures using flower pollination algorithm*, Applied Soft Computing, 37, pp. 322-331.
6. Assimi, H., Jamali, A., Nariman-Zadeh, N., 2017, *Sizing and topology optimization of truss structures using genetic programming*, Swarm and Evolutionary Computation, 37, pp. 90-103.
7. Wang, C.Y., Li, Y., Zhao, W.Z., Zou, S.C., Zhou, G., Wang, Y.L., 2018, *Structure design and multi-objective optimization of a novel crash box based on biomimetic structure*, International Journal of Mechanical Sciences, 138-139, pp. 489-501.
8. Woolway, M., Majoz, T., 2018, *A novel metaheuristic framework for the scheduling of multipurpose batch plants*, Chemical Engineering Science, 192, pp. 678-687.
9. Tsai, C.C., Huang, H.C., Chan, C.K., 2011, *Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation*. IEEE Transactions on Industrial Electronics, 58(10), pp. 4813-4821.
10. Alvarez, A., Munari, P., Morabito, R., 2018, *Iterated local search and simulated annealing algorithms for the inventory routing problem*. International Transactions in Operational Research, 25(4), pp. 1785-1809.
11. Hui, M., Tian, Y.C., 2013, *Dynamic Robot Path Planning using An Enhanced Simulated Annealing Approach*, Applied Mathematics and Computation, 222, pp. 420-437.
12. Kumazawa, T., Takada, K., Takimoto, M., Kambayashi, Y., 2019, *Ant colony optimization based model checking extended by smell-like pheromone with hop counts*, Swarm and Evolutionary Computation, 44, pp. 511-521.
13. Xiong, C., Kong, Y.Y., Fang, X., Wu, Q.D., 2013, *A fast two-stage ACO algorithm for robotic path planning*, Neural Computing & Applications, 22(2), pp. 313-319.
14. Cao, L.L., Xu, L.H., Goodman, E.D., 2019, *A collaboration-based particle swarm optimizer with history-guided estimation for optimization in dynamic environments*, Expert Systems with Application, 120, pp. 1-13.
15. Sadoughi, M., Pourdadashnia, A., Farhadi-Kangarlu, M., Galvani, S., 2022, *PSO-optimized SHE-PWM technique in a cascaded H-bridge multilevel inverter for variable output voltage applications*, IEEE Transactions on Power Electronics, 37(7), pp. 8065-8075.
16. Li, L.D., Li, X.D., Yu, X.H., 2008, *A multi-objective constraint-handling method with PSO algorithm for constrained engineering optimization problems*, 2018 IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, pp. 1528-1535.

17. Zain, M.Z.B., Kanesan, J., Chuah, J.H., Dhanapal, S., Kendall, G., 2018, *A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization*, Applied Soft Computing, 70, pp. 680-700.
18. Li, G.S., Chou, W.S., 2018, *Path planning for mobile robot using self-adaptive learning particle swarm optimization*, Science China Information Sciences, 61(5), pp. 263-280.
19. Xu, X.Y., Han, X., Ai, Xing., Song, Fu., 2018, *Research on multi-objective optimization of planetary gear system with improved particle swarm optimization*, Mechanical Science and Technology for Aerospace Engineering, 37(9), pp. 1352-1358.
20. Hasanoglu, M.S., Dolen, M., 2018, *Multi-objective feasibility enhanced particle swarm optimization*, Engineering Optimization, 50(12), pp. 2013-2037.
21. Kennedy, J., 2008, *Bare bones particle swarms*, Swarm Intelligence Symposium, pp. 80-87.
22. Zhang, Y., Gong, D.W., Ding, Z.H., 2012, *A Bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch*, Information Sciences, 192, pp. 213-227.
23. Leong, W.F., Yen, G.G., 2008, *PSO-based multi-objective optimization with dynamic population size and adaptive local archives*, IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society, 38(5), pp. 1270-1293.
24. Raquel, C.R., Jr, P.C.N., 2005, *An effective use of crowding distance in multi-objective particle swarm optimization*, Genetic and Evolutionary Computation Conference, Washington DC, USA, June 25-29, pp. 257-264.
25. Leong, W.F., 2008, *Multi-objective particle swarm optimization: integration of dynamic population and multiple-swarm concepts and constraint handling*, PhD Thesis, Oklahoma State University, America, 213 p.
26. Zitzler, K., Deb, K., Thiele, L., 2000, *Comparison of multi-objective evolutionary algorithms: empirical results*, Evolutionary Computation, 8(2), pp. 173-195.
27. Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2001, *Scalable test problems for evolutionary multi-objective optimization*, Evolutionary Computation, 8(2), pp. 105-145.
28. David, A.V., Gary, B.L., 1998, *Multi-objective evolutionary algorithm research: a history and analysis*, Evolutionary Computation, 8(2), pp. 125-147.
29. Tanaka, M., Watanabe, H., Furukawa, Y., Tanino, T., 1995, *GA-based decision support system for multicriteria optimization*, IEEE International Conference on Systems, Vancouver, October 22-25, 2, pp. 1556-1561.
30. Xia, S., Zhou, M., Li, G.I., 2011, *Multi-objective optimization algorithm for distributed generation locating and sizing*, Power System Technology, 35(9), pp. 115-121. (in Chinese)
31. Liang, J.J., Suganthan, P.N., Deb, K., 2005, *Novel composition test functions for numerical global optimization*, Swarm Intelligence Symposium, Pasadena, CA, USA, pp. 68-75.
32. Wu, F.H., Wang, Z.H., Sun, Y.X., Yang, Y.L., Li, Y.X., Guo, B.S., Peng, Q.J., 2018, *Multi-objective topological optimization of support frame for high-speed and heavy-load triangle track wheel based on analytic hierarchy process*, Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Canada, <https://doi.org/10.1115/DETC2018-85124>
33. Liu, Q., Liao, Z.R., Axinte, D., 2020, *Temperature effect on the material removal mechanism of soft-brittle crystals at nano/micron scale*, International Journal of Machine Tools and Manufacture, 159, 103620.