



OPEN

An improved multi-objective particle swarm optimization algorithm for the design of foundation pit of rail transit upper cover project

Jinyan Shao^{1,4}, Yuan Lu^{2✉}, Yi Sun³ & Lei Zhao⁴

In this study, a multi-objective particle swarm optimization (MOIPSO) algorithm is proposed to address complex optimization problems, including real-world engineering challenges. The algorithm retains the basic convergence mechanism of particle swarm optimization (PSO) as its core, while innovatively combining the fast non-dominated sorting technique to effectively evaluate and approximate the Pareto optimal solution set. To enhance the diversity and generalization of the solution set, the crowding distance mechanism is introduced, ensuring a good balance between multiple optimization objectives and a wider coverage of the solution space. Additionally, an acceleration factor based on trigonometric functions and an adaptive Gaussian mutation strategy are incorporated, improving the exploration ability of the particles in the search space and facilitating their movement towards the global optimal solution more effectively. The performance of the algorithm is verified using the multi-modal multi-objective benchmark function set provided by CEC2020, and comparisons are made with five advanced multi-objective metaheuristics. The MOIPSO algorithm is also applied to solve the design problem of rail transit upper cover foundation pit, further demonstrating the practical effectiveness of the proposed algorithm. The results show that MOIPSO not only performs well in multi-objective function testing but also proves highly competitive in solving real-world engineering problems. Note that the source codes of MOGWO are publicly available at <https://au.mathworks.com/matlabcentral/fileexchange/177404-moipso-optimization-engineering-problem>.

Keywords Multi-objective optimization problems, Particle swarm optimization, Multi-objective particle swarm optimization (MOIPSO), Rail transit upper cover foundation pit

With the continuous development of rail transit infrastructure, the number of complex TOD (public transport-oriented development) projects close to existing rail transit lines has been increasing. Under the premise of maintaining the normal operation of rail transit, the construction demand for new foundation pit projects has been growing¹. This necessitates effectively reducing the cost of foundation pit projects while ensuring their safety². In recent years, metaheuristic techniques have garnered significant attention from researchers due to their ability to solve complex problems efficiently. Such techniques have found widespread application in multi-domain optimization problems owing to their relatively low computational cost³. In particular, multi-objective optimization problems pose significant technical challenges due to the non-existence of a unique optimal solution, the nonlinearity of the objective function, the high dimensionality of the problems, and the complexity of the search spaces. To this end, specially designed multi-objective optimization algorithms (MOAs) have emerged and become the mainstream method for solving such problems over the past decade. Many efficient algorithms have been developed to overcome these technical challenges⁴. It is worth noting that a significant number of multi-objective problems are NP-hard in nature, which means that there is no universally

¹TOD Institute, Beijing Jiaotong University, Beijing 100044, China. ²School of Architecture and Design, Beijing Jiao Tong University, Beijing 100044, China. ³China Architecture Design and Research Institute Co., Ltd. Shanghai Branch, Shanghai, China. ⁴Beijing Urban Construction Design and Development Group Co., Ltd, Xicheng District, Beijing 100037, China. ✉email: luyuan@bjtu.edu.cn

accepted exact solution method. It is in this context that metaheuristic algorithms have been widely recognized by researchers for their outstanding performance in multi-objective optimization⁵.

The purpose of multi-objective algorithms is to effectively deal with multiple conflicting objective functions, and this is a crucial metric for evaluating their ability to solve complex optimization problems. For multi-objective optimization problems, common processing methods include prior methods, posterior methods, and interactive methods, and they primarily differ in the extent to which decision makers express preferences during the decision-making process⁶. The prior method requires the decision maker to preset preferences by assigning weights to the sub-objectives of the multi-objective optimization problem and then solving it using a single-objective method. Although this method is simple and feasible, it is challenging to address non-convex objective spaces. The posteriori method involves making decisions after the optimization process is completed. Despite the large computational effort required, multiple solutions can be generated in a single run, forming the Pareto front (PF). Thus, it has become the preferred method for researchers addressing multi-objective optimization problems⁷. Interactive methods focus on making decisions in real-time during the optimization process, an approach known as the man-machine in the loop technique⁸. However, the main challenge is that it is difficult to clearly define decision preferences in the optimization process, which results in relatively low efficiency.

David Schaffer initially utilized stochastic technology to address multi-objective optimization problems⁹. This approach combines evolutionary and heuristic stochastic techniques to address multi-objective optimization challenges. A principal advantage of stochastic techniques is their ability to operate without derivations and circumvent local optima. Consequently, this technique is highly favored by researchers, fostering the development of diverse multi-objective algorithms extensively employed in fields such as scheduling¹⁰, economics¹¹, image processing^{12,13} and engineering. Among these, He et al.¹⁴ introduced a novel multi-objective optimization algorithm to address the complex scheduling problems of microgrids, extensively analyzing the impact of new energy proportions and energy storage systems on optimal microgrid scheduling. Upadhyay et al.¹⁵ developed a multi-objective Gravitational Search algorithm (MOGSA) to enhance image segmentation efficiency. To address real-world engineering optimization challenges, Jameel et al.⁸ introduced a novel multi-objective Mantis search algorithm (MOMSA). These studies demonstrate that the multi-objective optimization algorithms offer researchers superior decision-making frameworks, clearly highlighting their significant benefits in addressing realistic multi-objective optimization challenges. However, according to the no free lunch (NFL) theorem proposed by Wolpert and Macready¹⁶, the theorem demonstrates that when specific assumptions are made about the data, an optimization algorithm only exhibits performance advantages for specific problems. There is no universal optimization technique that can be applied to all optimization problems. It has been logically proven that the performance of different optimization algorithms depends on how well the data assumptions align with the characteristics of the problem, and no single optimization technique performs well across all scenarios. This theorem not only lays a theoretical foundation for numerous studies in the literature but also inspires researchers to apply existing techniques to new optimization challenges or to develop novel optimization algorithms¹⁷.

As the most popular swarm intelligence algorithm, PSO¹⁸ has been widely used in solving various problems. It not only effectively addresses global optimization challenges but also penetrates deeply into domains such as text analysis¹⁹, image segmentation²⁰, feature selection²¹, and data clustering²², demonstrating broad applicability in industrial and engineering practices²³. However, the "no free lunch" (NFL) theorem asserts that no single optimization algorithm can address all optimization challenges, and currently, there lacks a multi-objective PSO literature addressing the design challenges of foundation pit construction in rail transit. These insights prompt the development of a multi-objective particle swarm optimization algorithm (MOIPSO) based on PSO, aimed at tackling the engineering challenges of foundation pit design for rail transit.

The main contributions of this research are as follows:

- (1) An archiving strategy based on crowding degree is introduced to store the non-dominated Pareto optimal solutions found so far.
- (2) A fast non-dominated sort is proposed to assign rank to each individual in particle swarm.
- (3) Acceleration factor based on trigonometric function and adaptive Gaussian mutation strategy are proposed to enhance the search ability of particles in the search space.
- (4) The performance of MOIPSO is verified by using the CEC2020 multi-objective test set and the rail transit upper cover foundation pit engineering design problem.

Literature review

In this section, we will outline the basic concepts related to multi-objective optimization, such as Pareto optimality, Pareto dominance, Pareto optimal set, and Pareto optimal front. Next, we will discuss and review various types of multi-objective optimization methods, covering both classic and recent research results.

Basic concepts of multi-objective optimization

Multi-objective optimization is dedicated to addressing the challenges of conflicts among multiple objectives. Unlike traditional optimization, which seeks the optimal solution for a single objective, multi-objective optimization addresses the complexities of scenarios where multiple objectives coexist. This approach emphasizes discovering a set of balanced solutions to mediate between conflicting objectives. In mathematical terms, multi-objective optimization is typically formulated as a problem of minimizing a multi-objective function, as specified below²⁴:

$$\begin{aligned}
 & \text{Minimize: } F(x) = (f_1(x), f_2(x), \dots, f_M(x))^T \\
 & g_i(x) = 0, i = 1, 2, \dots, P \\
 & h_i(x) \leq 0, i = 1, 2, \dots, K \\
 & x = (x_1, \dots, x_N)^T \\
 & L_i \leq x_i \leq U_i, i = 1, 2, \dots, N
 \end{aligned} \tag{1}$$

where, M represents the number of objective functions, K is the number of inequality constraints, and P denotes the number of equality constraints. For specific constraints, we use $h_i(x)$ to represent the i th inequality constraint and $g_i(x)$ to present the i th equality constraint. In addition, N refers to the total number of design variables, and the value range of each design variable x is defined by its lower limit L_i and upper limit U_i .

Traditional methods often falter when evaluating multiple solutions to multi-objective problems. Consequently, we propose the adoption of a new operator, Pareto optimality, to evaluate and optimize these issues more effectively. The fundamental definition of Pareto optimality is outlined as follows:

Definition 1 Pareto dominance relation.

It refers to a set of solutions where one solution is considered to Pareto dominate another if it is not inferior to the other across all evaluation metrics and superior on at least one metric. Assuming that solution x is superior to solution y (denoted by $x \leq y$), the mathematical model is formulated as follows²⁵:

$$\forall i \in \{1, 2, \dots, r\} : f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1, 2, \dots, r\} : f_i(\vec{x}) < f_i(\vec{y}) \tag{2}$$

Definition 2 Pareto optimal set.

If no alternative y is superior to it, then x is considered a non-dominated alternative (also termed the Pareto optimal alternative). Clearly, all sets of non-dominated solutions form the Pareto solution set (PS), defined as follows²⁶:

$$PS = \{x \in X \mid \nexists y \in X : y > x\} \tag{3}$$

Definition 3 The Pareto optimal front.

The PS is substituted into the objective function to calculate the result called Pareto optimal front (PF), and its mathematical expression is as follows²⁷:

$$PF = \{F(x) \mid x \in PS\} \tag{4}$$

The blue circle in Fig. 1 highlights the Pareto optimal vector. In the decision variable space, these vectors constitute the PS. However, in the objective space, this set is referred to as PF. PF consists entirely of non-dominated vectors, and its shape varies. It may be either continuous or discrete, and may display either singular or composite shapes, such as concave or convex.

Related work

Multi-objective optimization centers on two primary objectives. Firstly, it aims to identify as many solutions as possible on the Pareto front to assess the convergence of the solution set. Secondly, the goal is to secure

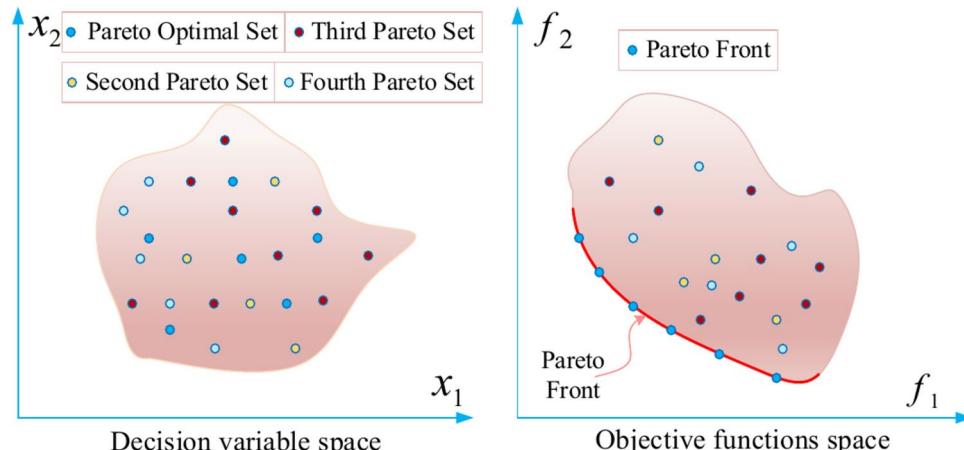


Fig. 1.. Decision space and objective space in MOO.

diversified solutions on the optimal frontier, thereby gauging the distribution breadth of the solution set. In recent decades, the field of multi-objective optimization has experienced significant growth, with numerous efficient algorithms emerging²⁸.

One of the foremost multi-objective optimization algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II), proposed by Deb et al.²⁹ in 2002. Compared to its predecessor, NSGA, NSGA-II offers significant enhancements across three dimensions: First, it integrates parents and offspring through non-dominated sorting, significantly lowering computational costs and reducing complexity from $O(MN^3)$ to $O(MN)$; Second, the introduction of an elite selection strategy ensures the preservation of high-quality individuals and enhances the algorithm's accuracy. Third, the use of crowding distance and its comparison mechanism automatically maintains population diversity, ensures the uniform distribution of Pareto fronts, and bolsters the robustness and operational efficiency of the algorithm. In summary, NSGA-II demonstrates significant improvements in performance, speed, and stability.

The multi-objective evolutionary algorithm based on decomposition (MOEA/D), proposed by Zhang et al.³⁰, has garnered significant attention in the field of multi-objective optimization. The essence of the algorithm involves decomposing complex multi-objective problems into multiple single-objective or multi-objective sub-problems, aiming to cover the entire Pareto front through neighborhood cooperation and parallel optimization among these sub-problems. Each sub-problem is assigned a weight vector, with the neighborhood relationship defined by the Euclidean distance between these vectors. This approach allows the optimal solution to quickly propagate to neighboring individuals, thereby accelerating population convergence. Compared to NSGA-II, MOEA/D exhibits lower computational complexity.

In addition to the aforementioned algorithms, other notable algorithms include: multi-objective particle swarm optimization (MOPSO)³¹, multi-objective artificial bee colony (MOABC)³², multi-objective grey wolf optimizer (MOGWO)²⁴, multi-objective ant lion optimization algorithm (MOALO)³³, multi-objective pathfinder algorithm (MOPFA)³⁴, non-dominated sorting genetic algorithm third edition (NSGA-III)³⁵, multi-objective Slime Mould algorithm (MOSMA)³⁶, an enhanced version of NSGA-II utilizing a special congestion strategy and adaptive crossover strategy (ASDNGA-II)³⁷, and the multi-objective symbiotic organism search algorithm based on decomposition (MOSOS/D)³⁸. Recent proposals in 2024 for multi-objective optimization algorithms include: multi-objective war strategy optimization (MOWSO)³⁹, multi-objective siege and conquest algorithm (MOBCA)⁴⁰, Multi-objective Variant of Moth-flame optimization algorithm (MnMOMFO)⁴¹, multi-objective geometric mean optimizer (MOGMO)⁴², and multi-objective quantum evolutionary algorithm based on decomposition mechanism (MOQEA/D)⁴³, among others. Table 1 summarizes these popular multi-objective optimization algorithms.

Author	Variants	Improvement strategy	Field of application	Year
Deb et al. ²⁹	NSGA-II	1. A fast non-dominated sorting approach 2. Propose a selection operator	Solve constrained multi-objective problems	2002
Coello et al. ³¹	MOPSO	1. Propose pareto dominance strategy 2. Propose special mutation operator	Multi-objective optimization problems	2004
Zhang et al. ³⁰	MOEA/D	1. Decomposition strategy Weight vector method	Multi-objective 0–1 knapsack problem and optimization problem	2007
Akbari et al. ³²	MOABC	1. Propose a grid-based approach 2. Propose a fixed size archive strategy	Multi-objective optimization problems	2012
Mirjalili et al. ²⁴	MOGWO	1. A fixed-sized external archive 2. A leader selection mechanism	Challenging multi-objective test problems	2016
Mirjalili et al. ³³	MOALO	1. Employ a repository method 2. Using a roulette wheel mechanism	Multi-objective engineering design problems	2017
Yapici et al. ³⁴	MOPFA	1. Leader mechanism 2. New location update rule	Real engineering applications	2019
Yi et al. ³⁵	NSGA-III	1. Introducing the concept of Stud 2. Design several improved crossover operators of SBX, UC, and SI	Large-scale optimization problems	2020
Premkumar et al. ³⁶	MOSMA	1. Using the positive-negative feedback system 2. Elitist non-dominated sorting approach	real-world engineering design problems	2021
Deng et al. ³⁷	ASDNGA-II	1. Propose a special congestion strategy 2. Introducing adaptive crossover strategy	Multi-objective optimization problems	2022
Ganesh et al. ³⁸	MOSOS/D	1. Introducing a decomposition framework	large complex multi-objective problems	2023
He et al. ³⁹	MOWSO	1. Propose a random selection strategy 2. Active update of soldier position	Microgrid and Power dispatch	2024
Jiang et al. ⁴⁰	MOBCA	1. Propose the grid mechanism, archiving mechanism, and leader selection mechanism	Multi-objective optimization	2024
Sahoo et al. ⁴¹	MnMOMFO	1. Introducing arithmetic and geometric mean strategies 2. Using NDS and CD strategies	Real-world engineering problems	2024
Pandya et al. ⁴²	MOGMO	1. Propose an elite non-dominated sorting approach 2. A Crowding Distance coupled with an Information Feedback Mechanism selection strategy	Multi-objective engineering design challenges	2024
Deng et al. ⁴³	MOQEA/D	1. Design a new decomposition mechanism 2. Propose a new optimal crossover strategy	Multi-objective knapsack problem and the multi-objective GAP	2024

Table 1. Some popular multi-objective algorithms.

In summary, the field of multi-objective optimization algorithms has witnessed significant advancements, with several algorithms demonstrating exemplary performance in practical applications. However, according to the no free lunch (NFL) theorem, there remains a need for the development of new algorithms in this field to address multi-objective optimization problems that currently lack effective solutions.

Multi-objective PSO algorithm

The PSO algorithm

The PSO algorithm, devised by Eberhart and Kennedy¹⁸, draws inspiration from the foraging behavior of birds and is conceptualized as the stochastic exploration of particles within a search space, aiming to identify the optimal solution to the objective function. In every iteration, each particle adjusts its trajectory based on its historical optimum and the global optimum shared by the swarm. Ultimately, all particles converge to the region with the most abundant resources, thereby facilitating the discovery of the global optimal solution.

Basic principle of PSO

In the PSO algorithm, the positions or food sources of the bird flock represent potential solutions to the optimization problem. Through the exchange of information and interactions between individuals, the algorithm guides the population to gradually converge towards the optimal solution while maintaining diversity. In this process, the flock individuals are simplified to "particles," focusing on their position information while ignoring mass and volume. The particle update is influenced by the topology structure, and during each update, the particle considers both its own historical best position and the historical best position of the swarm. The update mechanism of the particle combines the individual and population historical optimal solutions, as illustrated in Fig. 2.

In Fig. 2, the next velocity $v_{i,t+1}$ of particle i is determined by its current velocity v_i , individual optimal position $p_{best,i}$ and global optimal position g_{best} . The particle then moves from the current position $x_i(t)$ to the new position $x_{i,t+1}$ at an updated speed. As the iterative process progresses, the entire particle swarm incrementally completes the search for the optimal solution in the decision space, guided by the "leader".

The PSO optimization process begins with the random initialization of the particle swarm, including the size, initial position, and velocity of each particle. Each particle is then evaluated using the fitness function, and the globally and individually optimal particles are selected. Based on the PSO update rules, the velocity and position of each particle are adjusted, with simultaneous updates to the individual and group optimal solutions. If the new solution surpasses the historical optimum, it replaces it. This process combines both individual and group intelligence to efficiently guide the swarm toward the optimal solution. The flowchart is depicted in Fig. 3.

Update formula of PSO

In the decision space, the particle swarm size is N , the current number of iterations is t , the position of the r th ($r = 1, 2, \dots, N$) particle in the population is: $x_r(t) = [x_{r,1}(t), x_{r,2}(t), \dots, x_{r,d}(t)]$, and the velocity of the i th particle is: $v_r(t) = [v_{r,1}(t), v_{r,2}(t), \dots, v_{r,d}(t)]$. The historical optimal position of all particles in the whole population in the d dimension is: $g_{best}(t) = [g_1(t), g_2(t), \dots, g_d(t)]$, and the historical optimal position of particle r is: $p_{best,r}(t) = [p_{r,1}(t), p_{r,2}(t), \dots, p_{r,d}(t)]$. After the next iteration of particle r , the velocity and position update formula as follows:

$$\nu_{r,d}(t+1) = \omega\nu_{r,d}(t) + c_1 r_1(p_{best,r,d}(t) - x_{r,d}(t)) + c_2 r_2(g_{best,d}(t) - x_{r,d}(t)) \quad (5)$$

$$x_{r,d}(t+1) = x_{r,d}(t) + v_{r,d}(t+1) \quad (6)$$

where, r_1 and r_2 are random numbers in the interval $[0,1]$. c_1 and c_2 are acceleration constants in the range of $[0,2]$. ω is the inertia weight.

The proposed MOIPSO algorithm

This section introduces the proposed MOIPSO algorithm, designed to enhance the efficiency of solving multi-objective optimization problems. The essence of MOIPSO lies in integrating two key technologies: fast non-dominated sorting and the crowding distance mechanism, which together effectively search for and extract

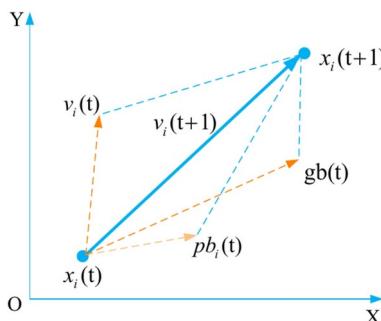
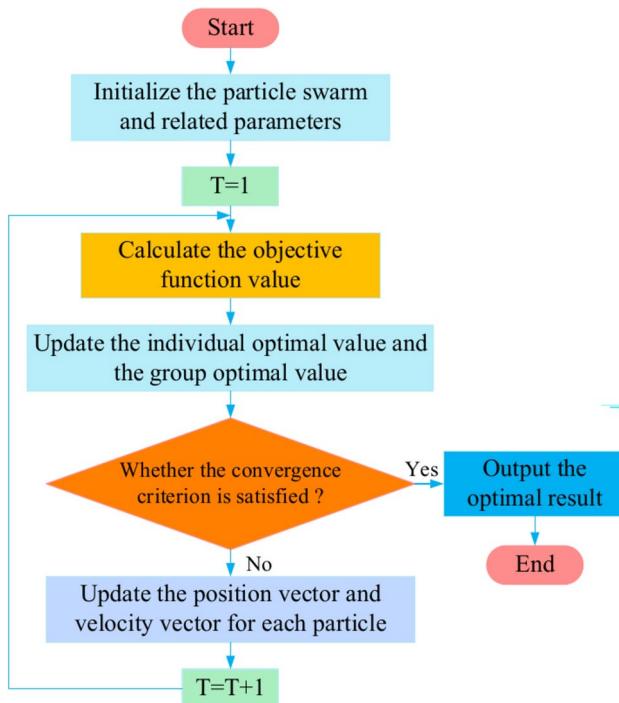
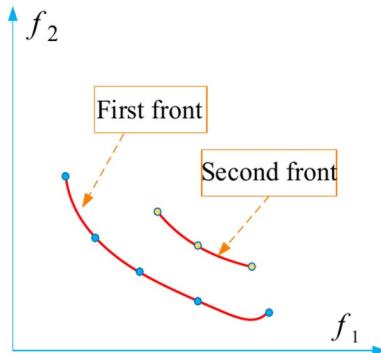


Fig. 2. Illustration of particle movement in decision space.

**Fig. 3..** The flow chart of PSO.**Fig. 4..** The illustration of non-dominated sorting.

the non-dominated solution set across all particles, while preserving the diversity of this set. This integration facilitates a more efficient optimization search.

Fast non-dominated sorting

The fast non-dominated sorting method categorizes individuals in the population into different layers based on their dominance relationships across multiple objectives, thereby creating multiple non-dominated fronts. The individuals within each front exhibit the same level of Pareto superiority, with a clear ordered superiority relationship between different fronts. The optimization process of this method primarily focuses on the efficient calculation of dominance relationships and the optimization of the sorting process. By minimizing comparisons, optimizing data structures, and utilizing parallel processing, computational efficiency can be significantly enhanced, particularly in large-scale populations and complex multi-objective problems. This method enables the effective selection and retention of the most promising individuals in multi-objective optimization, facilitating the algorithm's ability to find a more uniform and accurate solution set along the Pareto front²⁹. In multi-objective optimization problems, the dominance relationship directly assesses the merits and shortcomings of each solution, thereby effectively distinguishing between the advantages of different solutions. The non-dominated sorting process is depicted in Fig. 4. The leading solution (highlighted in blue) excels, outperforming all other solutions. Conversely, solutions in the second echelon (marked in yellow) are outperformed by at least one solution in the first echelon. The higher a solution's ranking, the more significant its position of non-dominance.

Crowding distance

Utilizing the fast non-dominated sorting technique described above, we can precisely partition the particles into multiple quality tiers, where the first tier represents the optimal Pareto solution. As the tier ascends, the mass of the particles correspondingly diminishes. Although the sorting mechanism effectively distinguishes particle masses, the presence of multiple particles with similar masses at the same tier often presents challenges in role assignment for the MOIPSO algorithm, complicating smooth operation. To address this issue, we introduce the crowding degree strategy, as depicted in Fig. 5. This strategy establishes a distance threshold for each tier of particles, computes the relative distance between each particle and its neighbors within this range, and normalizes these distances to derive the crowding degree index. This measure allows particles within the same tier to be further differentiated based on their crowding degree, effectively addressing the challenge of sorting particles at equivalent tiers. The specific calculation method for crowding degree is presented in Eq. (7).

$$Cd_j^i = \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}} \quad (7)$$

where, Cd_j^i denotes the crowding degree between particles, f_j^{i+1} and f_j^{i-1} are the fitness values of the $(i+1)$ th particle and the $(i-1)$ th particle in the j th objective function, respectively, and f_j^{\max} and f_j^{\min} present the maximum and minimum values of the objective function, respectively.

Acceleration factor based on trigonometric function

Changing the acceleration factor to a trigonometric function-based acceleration factor can improve the adaptability and search efficiency of the PSO algorithm. The search behavior of particles is dynamically adjusted by the periodic changes of trigonometric functions, promoting global search in the early stage to explore more possibilities and favoring local search in the later stage to improve convergence accuracy and solution quality. This adaptive adjustment not only balances the ability of global and local search but also increases diversity and flexibility in the search process, helping the particle swarm find the optimal solution more effectively in a complex problem space. The mathematical model of this strategy is given in Eq. (8).

$$\begin{aligned} c_1 &= 2.05 \times |\cos(2 \times \pi \times \text{rand})| \\ c_2 &= 2.05 \times |\sin(2 \times \pi \times \text{rand})| \end{aligned} \quad (8)$$

where, rand is a random number between 0 and 1.

Adaptive Gaussian mutation strategy

In the PSO algorithm, the adaptive Gaussian mutation strategy is used to dynamically adjust the mutation step size according to the search process. This approach not only enhances the global search ability of the algorithm in the early stage to explore a wider solution space but also improves the local search accuracy in the later stage to approach the optimal solution while maintaining population diversity to avoid premature convergence. Thus, the robustness, stability, and solution quality of the algorithm are comprehensively improved. The mathematical expression for updating the particle position is as follows:

$$\begin{aligned} dx &= \text{randn}(\mu, \sigma) \times (ub - lb) \\ X(idx) &= X(idx) + dx \end{aligned} \quad (9)$$

where, randn denotes the Gaussian distribution, the expectation $\mu = 0$, the variance $\sigma = 0.1 \times (1 - \frac{t}{T})$, ub and lb are the upper and lower bounds of the problem, respectively, and idx is the index of half of the population randomly selected from the population.

To sum up, the flow chart of the MOIPSO algorithm is shown in Fig. 6.

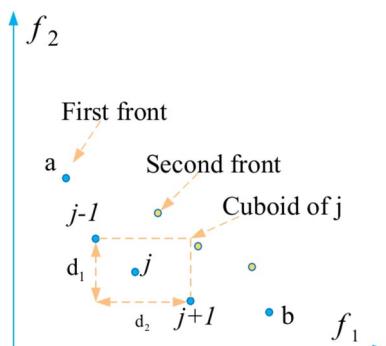


Fig. 5. Diagram of the crowding distance of particles.

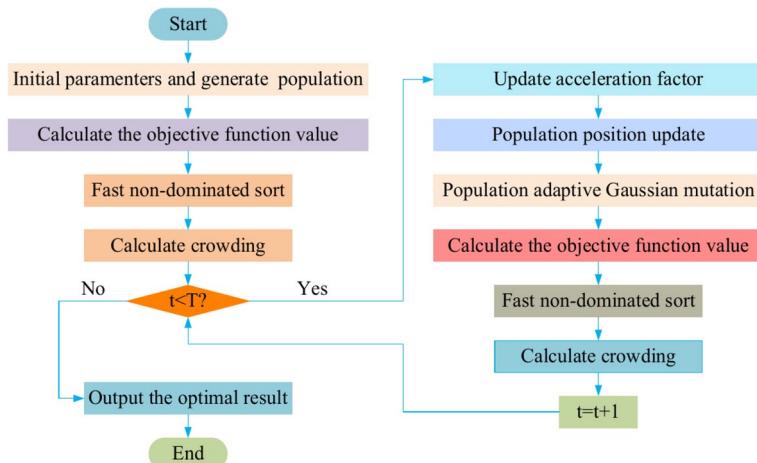


Fig. 6. The flowchart of the MOIPSO algorithm.

Experimental results and analysis

In this section, the performance of the MOIPSO algorithm is evaluated against five other algorithms using the CEC2020 benchmark functions. Using four key performance indicators as the evaluation basis, we analyze the test results in detail to verify the advantages and characteristics of the MOIPSO algorithm in the competition.

Experimental settings

The CEC2020 benchmark is a set of multimodal multi-objective optimization functions proposed by the Chinese Computer Society to evaluate and compare the performance of multi-objective optimization algorithms. In the field of optimization algorithms, it is essential to have a unified set of test functions to compare the strengths and weaknesses of different algorithms, and CEC2020 serves as such a tool. It helps researchers assess the effectiveness of algorithms in solving multimodal multi-objective problems in a fair and objective manner. This section verifies the efficiency and excellence of the MOIPSO algorithm on the CEC2020⁴⁴ multimodal multi-objective optimization benchmark functions. This benchmark covers²⁴ test functions of various types, including linear, nonlinear, concave, and convex functions. To fairly evaluate the performance of different algorithms, scientific evaluation criteria and reference data were designed. In the decision space evaluation, we used two key indicators, the reciprocal of Proximity (rPSP)⁴⁵ and Backward distance (IGDX)⁴⁶, to measure the quality of Pareto solutions. In the target space, the inverse Hypervolume (rHV)⁴⁷ and IGD in the target space (IGDF)⁴⁸ are used for performance evaluation. The specific information on the CEC2020 benchmark functions is shown in Table 2.

This experiment was conducted in a computing environment consisting of Intel Core i7-12700F, 2.10 GHz CPU, and 16 GB RAM. The experiment was carried out using Windows 11 operating system and Matlab R2023a as running software.

We compared the proposed MOIPSO algorithm with five widely recognized algorithms, including MOGWO²⁴, MOMVO⁴⁹, MOSMA³⁶, MOSSA⁵⁰, and MOPSO³¹. The parameter settings of the competitor algorithms are shown in Table 3. Each algorithm was run 21 times independently, with 100 iterations performed in each run to ensure a fair evaluation environment. We recorded the mean and standard deviation data obtained by each algorithm in each function test in detail, which is an important basis for evaluating the quality of the algorithms. To visually show the relative performance of our algorithm, we carried out the Wilson sign test⁵¹. When the performance of the MOIPSO algorithm is worse than that of the other comparison algorithms, we used "-". If the MOIPSO algorithm is better than other algorithms, it is marked as "+". When the performance is comparable, with no significant advantage, it is marked with "=".

CEC2020 test function results

Tables 4, 5, 6, and 7 detail the mean and standard deviation results of the MOIPSO algorithm and other comparison algorithms when calculating rPSP, rHV, IGDX, and IGDF indicators. The rPSP index is used to evaluate the convergence degree and coverage between the solution set obtained by the algorithm and the true solution set. The rHV reflects the convergence performance between the Pareto front generated by the algorithm and the true PF, and the diversity level of the solution. IGDX and IGDF focus on quantifying the convergence accuracy between the PS and PF obtained by the algorithm and the true PS and PF, respectively.

The rPSP values obtained by the six algorithms, including MOIPSO, are recorded in detail in Table 4, where the data highlighted in black font indicate that the algorithm has achieved the best performance in the corresponding test functions. It can be clearly seen from the data in the table that the mean value of MOIPSO on 11 test functions reaches the lowest value, while the mean value of MOSMA on 12 test functions also reaches the lowest value. This result places first among all compared algorithms. Although MOIPSO does not achieve the largest number of minima, its number of minima differs by only one from MOSMA. In contrast, although some

Problem number	Function name	Geometry of PS	Geometry of PF	number of global and local optima
TP1	MMF1	Nonlinear	Convex	2
TP2	MMF2	Nonlinear	Convex	2
TP3	MMF4	Nonlinear	Concave	2
TP4	MMF5	Nonlinear	Convex	2
TP5	MMF7	Nonlinear	Convex	2
TP6	MMF8	Nonlinear	Concave	2
TP7	MMF10	Linear	Convex	1
TP8	MMF11	Linear	Convex	1
TP9	MMF12	Linear	Convex	1
TP10	MMF13	Nonlinear	Convex	1
TP11	MMF14	Linear	Concave	2
TP12	MMF15	Linear	Concave	1
TP13	MMF1_e	Nonlinear	Convex	2
TP14	MMF14_a	Nonlinear	Concave	2
TP15	MMF15_a	Nonlinear	Concave	1
TP16	MMF10_l	Linear	Convex	2
TP17	MMF11_l	Linear	Convex	2
TP18	MMF12_l	Linear	Convex	2
TP19	MMF13_l	Nonlinear	Convex	2
TP20	MMF15_l	Linear	Concave	2
TP21	MMF15_a_l	Nonlinear	Concave	2
TP22	MMF16_l1	Linear	Concave	3
TP23	MMF16_l2	Linear	Concave	3
TP24	MMF16_l3	Linear	Concave	4

Table 2. Details of the CEC2020 test suite.

Algorithm	Parameter values
MOGWO	Grid Inflation Parameter, alpha = 0.1; Number of Grids per each Dimension, nGrid = 10; Leader Selection Pressure Parameter, beta = 4; Extra (to be deleted) Repository Member Selection Pressure, gamma = 2
MOMVO	Maximum of Wormhole, WEP_Max = 1; Minimum of Wormhole, WEP_Min = 0.2; Number of Grids per each Dimension, nGrid = 30
MOSSA	Number of Grids per each Dimension, nGrid = 30
MOSMA	a = arctan h (1 - (t/T)) b = 1 - (t/T); vb = [-a,a], vc = [-b,b]
MOPSO	Inertia weight, w = 0.5; Inertia Weight Damping Rate, wdamp = 0.99; c1 = 1; c2 = 2; Number of Grids per Dimension, nGrid = 7; Inflation Rate, alpha = 0.1; Leader Selection Pressure, beta = 2; Deletion Selection Pressure, gamma = 2; Mutation Rate, mu = 0.1
MOIPSO	Inertia weight, w = 0.4; Number of Grids per Dimension, nGrid = 30; maxvel = 5; u_mut = 0.5

Table 3. Parameter Settings for competitor algorithms.

algorithms, such as MOGWO, MOMVO, MOSSA, and MOPSO, achieve the lowest value in some test functions, MOIPSO demonstrates superior performance in most test functions. Through comparative analysis of rPSP values, it is evident that the Pareto solution set generated by MOIPSO closely covers the real Pareto solution set, demonstrating the powerful global optimization ability of MOIPSO in solving multimodal optimization problems.

Table 5 presents the mean and standard deviation of the rHV obtained by the competing algorithms on the test functions. The data clearly show that in most test functions, MOIPSO achieves the minimum value of rHV, which is significantly better than that of other competing algorithms. By analyzing the search results for the rHV index, it is evident that the MOIPSO algorithm demonstrates strong competitiveness in search efficiency and effect, and excels among all competing algorithms. This indicates that MOIPSO can explore the solution space more comprehensively and discover more high-quality solutions when solving complex optimization problems.

Tables 6 and 7 show the mean and standard deviation of IGDX and IGDF of MOIPSO and other algorithms on 24 test functions, respectively. In Table 6, the MOIPSO algorithm outperforms the other algorithms on IGDX, having the most bold entries. Although it fails to reach the minimum value in some functions, the results are still noteworthy, which highlights the strong competitiveness of MOIPSO in searching the decision space. Combined with the rPSP results in Table 4, it is further confirmed that the global search ability of MOIPSO is optimal among the competing algorithms. Table 7, on the other hand, shows the IGDF metrics in the objective space. In this respect, MOIPSO also performs well, often reaching or coming close to the minimum value on most test functions, which fully demonstrates the excellent optimization ability of MOIPSO. This coincides with the trend exhibited by rHV in Table 5, which jointly verifies the superior search performance of MOIPSO in finding the

ID		MOGWO	MOMVO	MOSSA	MOSMA	MOPSO	MOIPSO
TP1	Mean	1.1012E-01	9.2683E-02	2.5109E-01	9.1889E-02	8.3672E-02	5.4385E-02
	Std	2.9629E-02	1.4531E-02	3.4825E-02	1.0003E-02	1.2197E-02	7.7707E-03
TP2	Mean	1.1324E-01	1.4080E-01	1.1918E-01	9.4114E-02	2.0691E-01	8.5138E-02
	Std	7.0104E-02	2.0535E-01	5.2085E-02	5.0751E-02	1.4925E-01	7.8347E-02
TP3	Mean	4.3001E-01	3.5372E-01	2.8917E-01	1.0208E-01	8.9527E-02	5.9398E-02
	Std	1.6193E-01	1.6064E-01	1.0642E-01	2.7109E-02	3.7621E-02	3.1215E-02
TP4	Mean	3.4088E-01	2.3561E-01	5.1424E-01	1.5380E-01	1.3969E-01	1.0766E-01
	Std	1.0898E-01	1.0427E-01	1.4318E-01	2.1718E-02	2.4896E-02	1.1482E-02
TP5	Mean	1.8195E-01	1.9855E-01	2.5042E-01	6.7849E-02	6.4618E-02	3.5242E-02
	Std	1.2527E-01	7.8746E-02	7.8351E-02	1.3136E-02	3.0263E-02	1.2993E-02
TP6	Mean	8.8831E+00	1.2399E+00	1.4811E+00	2.5865E-01	4.7782E-01	1.2213E+00
	Std	8.6534E+00	1.0190E+00	6.5144E-01	1.2543E-01	4.1867E-01	1.6875E+00
TP7	Mean	8.7154E-03	1.8662E-01	1.1529E-01	3.6695E-02	1.4793E-01	4.4906E-02
	Std	2.0725E-02	1.2765E-01	5.1716E-02	2.9370E-02	1.6744E-01	1.0281E-01
TP8	Mean	5.3293E-03	9.9323E-03	1.3220E-02	1.3003E-02	3.8478E-02	4.0496E-03
	Std	7.7506E-04	1.2932E-03	2.4852E-03	3.7185E-03	9.1681E-02	5.8026E-04
TP9	Mean	2.5277E-03	5.8519E-03	7.7857E-03	8.7170E-03	3.9416E-03	1.6934E-03
	Std	7.6627E-04	9.6231E-03	1.6378E-03	2.3313E-03	1.3932E-03	1.0589E-04
TP10	Mean	5.5681E-01	9.1396E-02	9.9005E-02	6.9005E-02	1.9875E-01	1.0341E-01
	Std	3.4374E-01	2.9785E-02	1.5248E-02	1.3889E-02	4.6297E-01	8.8311E-02
TP11	Mean	7.7747E-01	6.1085E-01	1.3270E-01	8.5277E-02	4.3314E-01	4.0837E-01
	Std	2.9286E-01	1.7823E-01	2.3403E-02	8.9896E-03	2.6958E-01	2.3973E-01
TP12	Mean	1.1493E-01	6.0599E-02	9.2958E-02	6.9899E-02	2.0243E-01	4.6701E-02
	Std	2.1119E-02	8.5733E-03	1.2552E-02	6.6266E-03	1.0685E-01	2.3205E-03
TP13	Mean	6.4272E+00	7.0965E+00	7.8923E+00	3.4514E-01	6.6372E+00	3.3479E+00
	Std	5.4311E+00	3.4115E+00	6.3353E+00	3.4341E-02	6.1112E+00	1.9111E+00
TP14	Mean	2.4121E-01	3.1796E-01	1.8611E-01	1.9112E-01	2.4020E-01	1.1666E-01
	Std	2.7024E-02	1.4895E-01	2.0036E-02	2.7831E-02	1.9138E-01	4.3297E-02
TP15	Mean	8.2247E-02	8.7223E-02	1.1285E-01	9.7511E-02	1.0797E-01	6.4591E-02
	Std	8.2660E-03	1.2498E-02	1.3708E-02	1.7586E-02	3.5348E-02	1.4292E-02
TP16	Mean	5.4762E+00	5.7169E-01	1.7194E-01	3.5362E-02	1.0377E+01	4.9638E+00
	Std	6.0079E-01	1.2604E+00	1.3001E-01	1.9541E-02	1.2605E+01	3.7132E-01
TP17	Mean	2.5338E+00	1.5650E+00	1.9440E+00	3.0097E-01	4.1590E+00	1.8379E+00
	Std	3.7133E-01	4.6129E-01	5.4560E-01	3.0698E-01	5.4662E+00	1.7876E-01
TP18	Mean	2.3705E+00	2.0637E+00	2.2497E+00	1.1967E-01	1.6413E+00	1.9767E+00
	Std	2.9015E-01	4.9850E-01	5.4768E-01	3.4073E-03	2.7261E-01	1.4305E-01
TP19	Mean	2.3640E+00	6.3835E-01	7.0925E-01	4.7902E-01	7.1592E-01	8.7085E-01
	Std	7.8488E-01	9.9415E-02	1.1076E-01	1.5283E-01	3.3169E-01	2.8270E-01
TP20	Mean	8.8048E-01	6.8048E-01	6.8718E-01	3.2605E-01	8.6536E-01	6.3559E-01
	Std	2.2536E-01	5.7326E-02	1.0778E-01	1.6922E-01	1.8636E-01	9.0462E-02
TP21	Mean	3.1593E-01	2.9842E-01	3.2916E-01	3.0805E-01	2.9242E-01	2.5334E-01
	Std	1.4840E-02	1.1302E-02	1.1425E-02	1.4357E-02	2.5641E-02	2.1459E-02
TP22	Mean	8.6789E-01	5.2815E-01	2.7605E-01	1.9069E-01	4.3289E-01	3.5634E-01
	Std	5.7184E-01	2.7167E-01	1.2208E-02	4.3683E-02	2.5412E-01	1.5597E-01
TP23	Mean	1.1402E+00	8.6724E-01	8.7520E-01	4.0243E-01	1.3147E+00	7.9159E-01
	Std	2.4453E-01	7.3099E-02	6.7151E-02	1.8837E-01	6.2159E-01	8.6013E-02
TP24	Mean	9.6331E-01	7.5929E-01	3.5502E-01	3.0075E-01	6.9314E-01	6.2975E-01
	Std	5.9561E-01	4.4299E-01	1.8899E-02	3.9300E-02	4.7115E-01	3.3886E-01

Table 4. The rPSP values obtained by the competitive algorithms. Significant values are in bold

Pareto-optimal frontier. In summary, whether in the measure of decision space or objective space, the MOIPSO algorithm shows significant advantages in finding multiple Pareto-optimal solutions and their corresponding frontiers, and its search performance is highly competitive compared to other leading algorithms.

Figures 7, 8, 9, 10, 11 and 12 show the coverage of Pareto obtained by different algorithms in decision space and objective space for test problems TP1, TP11, and TP12. In particular, in Figs. 7, 9 and 11, the Pareto solution sets generated by the MOIPSO algorithm are distributed in the decision space of TP1, TP11, and

ID		MOGWO	MOMVO	MOSSA	MOSMA	MOPSO	MOIPSO
TP1	Mean	1.1570E+00	1.1528E+00	1.1852E+00	1.1699E+00	1.1527E+00	1.1456E+00
	Std	4.2081E-03	1.4500E-03	5.9764E-03	1.3939E-02	1.8185E-03	1.8757E-04
TP2	Mean	1.1602E+00	1.1787E+00	1.1959E+00	1.1685E+00	1.2803E+00	1.1663E+00
	Std	8.3879E-03	4.4029E-02	2.3095E-02	8.2137E-03	3.5125E-02	8.4971E-03
TP3	Mean	1.8680E+00	1.9003E+00	1.9881E+00	1.8588E+00	1.8851E+00	1.8529E+00
	Std	7.1853E-03	8.5067E-03	6.4983E-02	8.3678E-04	4.4464E-02	5.6354E-04
TP4	Mean	1.1545E+00	1.1518E+00	1.1846E+00	1.1897E+00	1.1525E+00	1.1454E+00
	Std	3.8022E-03	1.8115E-03	6.6763E-03	2.2729E-02	2.4537E-03	1.5926E-04
TP5	Mean	1.1578E+00	1.1612E+00	1.1757E+00	1.1551E+00	1.1508E+00	1.1454E+00
	Std	5.8143E-03	1.9751E-03	4.0478E-03	7.4899E-03	4.2216E-03	1.0910E-04
TP6	Mean	2.3868E+00	2.4601E+00	2.9926E+00	2.4230E+00	2.4344E+00	2.3739E+00
	Std	7.3031E-03	2.4587E-01	1.8237E+00	6.8788E-02	2.6480E-02	1.5502E-03
TP7	Mean	7.7826E-02	8.2614E-02	8.3504E-02	8.3998E-02	9.5610E-02	7.9041E-02
	Std	1.4243E-04	2.6569E-03	1.3806E-03	7.1764E-04	2.4528E-02	2.9242E-03
TP8	Mean	6.8968E-02	6.9152E-02	6.9330E-02	6.9861E-02	7.7308E-02	6.8879E-02
	Std	3.9694E-05	4.9654E-05	8.2215E-05	5.2233E-04	2.1392E-02	1.7268E-05
TP9	Mean	6.3888E-01	6.5197E-01	6.4724E-01	8.7813E-01	6.3846E-01	6.3558E-01
	Std	3.3641E-03	5.4462E-02	7.6354E-03	1.4489E-01	2.8035E-03	1.2665E-04
TP10	Mean	5.4336E-02	5.4495E-02	5.4640E-02	5.4549E-02	5.5037E-02	5.4247E-02
	Std	3.6199E-05	4.0563E-05	6.9886E-05	4.1626E-04	1.5270E-03	7.2067E-06
TP11	Mean	4.1292E-01	3.6776E-01	3.6411E-01	3.2889E-01	3.9861E-01	3.5143E-01
	Std	3.0093E-02	3.5211E-02	3.6077E-02	3.3862E-02	4.9353E-02	9.8215E-03
TP12	Mean	2.7049E-01	2.5046E-01	2.5118E-01	2.2766E-01	2.7181E-01	2.3924E-01
	Std	1.4374E-02	1.9206E-02	2.1430E-02	2.7284E-02	3.9512E-02	4.2977E-03
TP13	Mean	1.1655E+00	1.1501E+00	1.2183E+00	-7.2895E-02	4.4915E-01	1.1519E+00
	Std	2.7001E-02	7.8964E-04	1.0608E-01	4.0785E-02	4.5621E+00	1.4503E-02
TP14	Mean	3.7261E-01	3.5471E-01	3.5378E-01	3.3909E-01	3.8481E-01	3.4739E-01
	Std	2.8360E-02	3.2546E-02	3.0049E-02	2.6671E-02	8.0526E-02	1.2268E-02
TP15	Mean	2.5106E-01	2.4897E-01	2.4229E-01	2.2181E-01	2.4829E-01	2.3403E-01
	Std	9.1150E-03	1.4895E-02	1.9800E-02	2.5702E-02	2.3067E-02	4.6928E-03
TP16	Mean	7.7788E-02	8.3715E-02	8.3578E-02	8.3793E-02	1.2745E-01	7.7721E-02
	Std	2.0296E-05	2.7118E-03	2.2878E-03	6.3767E-04	1.2644E-01	1.3618E-05
TP17	Mean	6.8965E-02	6.9178E-02	6.9369E-02	6.9787E-02	8.5111E-02	6.8883E-02
	Std	3.3408E-05	6.8952E-05	1.0279E-04	3.1088E-04	3.5404E-02	1.4803E-05
TP18	Mean	6.3956E-01	6.3957E-01	6.5233E-01	9.2448E-01	6.3918E-01	6.3559E-01
	Std	3.0895E-03	3.2479E-03	8.0959E-03	1.4951E-01	5.1054E-03	1.0329E-04
TP19	Mean	5.4328E-02	5.4508E-02	5.4649E-02	5.4578E-02	5.4584E-02	5.4248E-02
	Std	2.0207E-05	4.1924E-05	8.5727E-05	5.1561E-04	4.7495E-04	1.2092E-05
TP20	Mean	2.6596E-01	2.4536E-01	2.4589E-01	2.2288E-01	2.6805E-01	2.3916E-01
	Std	1.1087E-02	1.9432E-02	1.6988E-02	2.1727E-02	3.0021E-02	4.3317E-03
TP21	Mean	2.4994E-01	2.4721E-01	2.3935E-01	2.3285E-01	2.5103E-01	2.3549E-01
	Std	1.0471E-02	2.1624E-02	1.4942E-02	2.4451E-02	1.5094E-02	5.1365E-03
TP22	Mean	2.6269E-01	2.4000E-01	2.4497E-01	2.2303E-01	2.5204E-01	2.3496E-01
	Std	1.4045E-02	1.6589E-02	1.4528E-02	2.8151E-02	1.5695E-02	3.3715E-03
TP23	Mean	2.6644E-01	2.5416E-01	2.4113E-01	2.2017E-01	2.7758E-01	2.3498E-01
	Std	1.3370E-02	1.5939E-02	1.0315E-02	2.0365E-02	3.4474E-02	4.1594E-03
TP24	Mean	2.5805E-01	2.4337E-01	2.3938E-01	2.2101E-01	2.5665E-01	2.3590E-01
	Std	1.1050E-02	1.2546E-02	1.5220E-02	2.5725E-02	3.1715E-02	3.9984E-03

Table 5. The rHV values obtained by the competitive algorithms. Significant values are in bold

TP12, providing better coverage than other algorithms. This phenomenon demonstrates that MOIPSO exhibits excellent global search ability when dealing with multimodal optimization tasks, and effectively explores and identifies a large number of global non-dominated solutions. This advantage confirms the importance of the improvement strategy within the framework of the MOIPSO algorithm. When solving multimodal test problems, the MOIPSO algorithm can efficiently and in real-time save global non-dominated solutions, and

ID		MOGWO	MOMVO	MOSSA	MOSMA	MOPSO	MOIPSO
TP1	Mean	1.0815E-01	9.1835E-02	2.3745E-01	9.0922E-02	8.3006E-02	5.4111E-02
	Std	2.7625E-02	1.3765E-02	3.0662E-02	9.6854E-03	1.2151E-02	7.7540E-03
TP2	Mean	1.1324E-01	1.0359E-01	1.0981E-01	9.4114E-02	1.7959E-01	8.1614E-02
	Std	7.0104E-02	9.8603E-02	4.8048E-02	5.0751E-02	1.0088E-01	6.9110E-02
TP3	Mean	3.3583E-01	2.8392E-01	2.4877E-01	1.0129E-01	8.8396E-02	5.9038E-02
	Std	8.5673E-02	7.6395E-02	6.2205E-02	2.6583E-02	3.6306E-02	3.0555E-02
TP4	Mean	3.1353E-01	2.1797E-01	4.3012E-01	1.5234E-01	1.3807E-01	1.0714E-01
	Std	8.4977E-02	7.6813E-02	7.0775E-02	2.1247E-02	2.4130E-02	1.1472E-02
TP5	Mean	1.1655E-01	1.2847E-01	1.6488E-01	6.5056E-02	5.5947E-02	3.4615E-02
	Std	5.2408E-02	3.1421E-02	2.9516E-02	1.0111E-02	1.9717E-02	1.2477E-02
TP6	Mean	1.9186E+00	8.8082E-01	9.7200E-01	2.4443E-01	4.1773E-01	6.8073E-01
	Std	1.0757E+00	4.8293E-01	2.4219E-01	1.1027E-01	3.4783E-01	7.0307E-01
TP7	Mean	7.6841E-03	1.6461E-01	1.0798E-01	3.6353E-02	1.2910E-01	4.4719E-02
	Std	1.6011E-02	1.1269E-01	4.1692E-02	2.9335E-02	1.5299E-01	1.0281E-01
TP8	Mean	5.3054E-03	9.8815E-03	1.3059E-02	1.3003E-02	2.7600E-02	4.0496E-03
	Std	7.7870E-04	1.2743E-03	2.3697E-03	3.7185E-03	5.2440E-02	5.8026E-04
TP9	Mean	2.5114E-03	5.4662E-03	7.6724E-03	8.7170E-03	3.9416E-03	1.6934E-03
	Std	7.4540E-04	7.8681E-03	1.5399E-03	2.3313E-03	1.3932E-03	1.0589E-04
TP10	Mean	1.5576E-01	8.4436E-02	9.6038E-02	6.9005E-02	8.8842E-02	7.6895E-02
	Std	3.8448E-02	2.2601E-02	1.3951E-02	1.3889E-02	5.9098E-02	4.1058E-02
TP11	Mean	2.9958E-01	2.6187E-01	1.3157E-01	8.5277E-02	2.4279E-01	2.1056E-01
	Std	2.7528E-02	5.4606E-02	2.1656E-02	8.9896E-03	6.9887E-02	7.9779E-02
TP12	Mean	1.0338E-01	6.0186E-02	9.0737E-02	6.9899E-02	1.5373E-01	4.6701E-02
	Std	1.5793E-02	8.0147E-03	1.0511E-02	6.6266E-03	5.0901E-02	2.3205E-03
TP13	Mean	2.3406E+00	2.6669E+00	2.4170E+00	3.4161E-01	2.3142E+00	1.7148E+00
	Std	9.5532E-01	4.1436E-01	8.5325E-01	3.3346E-02	7.8158E-01	6.3244E-01
TP14	Mean	2.2094E-01	2.5729E-01	1.8353E-01	1.9112E-01	1.9347E-01	1.1666E-01
	Std	1.4114E-02	7.7693E-02	1.8726E-02	2.7831E-02	8.1081E-02	4.3297E-02
TP15	Mean	8.1285E-02	8.6885E-02	1.1169E-01	9.7511E-02	1.0395E-01	6.4591E-02
	Std	7.7381E-03	1.2182E-02	1.3757E-02	1.7586E-02	2.9431E-02	1.4292E-02
TP16	Mean	2.0210E-01	1.5399E-01	1.3048E-01	3.5279E-02	2.4175E-01	2.0115E-01
	Std	6.3018E-04	4.8953E-02	3.8524E-02	1.9597E-02	9.6086E-02	2.2792E-04
TP17	Mean	2.5174E-01	2.4997E-01	2.5581E-01	1.4929E-01	2.7787E-01	2.4952E-01
	Std	5.5317E-04	9.7728E-03	2.8372E-03	7.8061E-02	5.9382E-02	6.0500E-04
TP18	Mean	2.4631E-01	2.4161E-01	2.4876E-01	1.1966E-01	2.4476E-01	2.4491E-01
	Std	8.1588E-04	2.0458E-02	9.3149E-04	3.4062E-03	9.2298E-04	3.0598E-04
TP19	Mean	3.6860E-01	2.9231E-01	3.1362E-01	2.4603E-01	2.9487E-01	3.1087E-01
	Std	1.9479E-02	1.3831E-02	1.9620E-02	5.7147E-02	3.3292E-02	3.1796E-02
TP20	Mean	3.0677E-01	2.8150E-01	2.9593E-01	2.2435E-01	3.2383E-01	2.7004E-01
	Std	9.4500E-03	4.3968E-03	8.7269E-03	4.0812E-02	2.7420E-02	4.0231E-03
TP21	Mean	2.4929E-01	2.4020E-01	2.6371E-01	2.5044E-01	2.3982E-01	2.1478E-01
	Std	8.6490E-03	8.0713E-03	8.6781E-03	1.1715E-02	1.5412E-02	6.1825E-03
TP22	Mean	3.0512E-01	2.4608E-01	2.0770E-01	1.6475E-01	2.3455E-01	2.2376E-01
	Std	4.2573E-02	4.2194E-02	7.9771E-03	1.8105E-02	4.0121E-02	3.3156E-02
TP23	Mean	3.7773E-01	3.5421E-01	3.6080E-01	2.7556E-01	3.9902E-01	3.4009E-01
	Std	1.0553E-02	7.8161E-03	4.5015E-03	4.8059E-02	3.5818E-02	4.0820E-03
TP24	Mean	3.6300E-01	3.1665E-01	2.5324E-01	2.2613E-01	2.9904E-01	2.8134E-01
	Std	5.6254E-02	6.1161E-02	5.8695E-03	1.2214E-02	6.8440E-02	5.7832E-02

Table 6. The IGDX values obtained by the competitive algorithms.

provide high-quality candidate solutions for subsequent iterations, thus promoting the continuous optimization and improvement of the overall performance of the algorithm.

Figures 8, 10, and 12 show the coverage of Pareto-optimal fronts obtained by all competing algorithms on the objective space of TP1, TP11, and TP12 test functions, respectively. From the figures, it is evident that the Pareto optimal front generated by MOIPSO is highly consistent with the true Pareto optimal front, far outperforming the other algorithms. For the real Pareto front test functions with hypersurface shapes, such as

ID		MOGWO	MOMVO	MOSSA	MOSMA	MOPSO	MOIPSO
TP1	Mean	8.5468E-03	6.5198E-03	2.2995E-02	8.1294E-03	5.9222E-03	2.5257E-03
	Std	2.1846E-03	8.7371E-04	1.9206E-03	1.6857E-03	7.1728E-04	1.0467E-04
TP2	Mean	9.5515E-03	2.7285E-02	2.6095E-02	1.5083E-02	6.3341E-02	1.4024E-02
	Std	5.3821E-03	4.0687E-02	6.4527E-03	4.7220E-03	2.1218E-02	6.1126E-03
TP3	Mean	5.2788E-03	1.2015E-02	2.3158E-02	3.8380E-03	6.7240E-03	2.3493E-03
	Std	1.0048E-03	1.7366E-03	3.7457E-03	3.4025E-04	3.8293E-03	1.0988E-04
TP4	Mean	6.8661E-03	5.9506E-03	1.9459E-02	8.1946E-03	5.5405E-03	2.4257E-03
	Std	1.6371E-03	1.0731E-03	1.7409E-03	1.7700E-03	9.9526E-04	9.1353E-05
TP5	Mean	9.6554E-03	1.1415E-02	1.9608E-02	5.7720E-03	5.2897E-03	2.4703E-03
	Std	3.6857E-03	1.0814E-03	2.2468E-03	3.2748E-03	2.1804E-03	6.5773E-05
TP6	Mean	3.9180E-03	7.1973E-03	1.8353E-02	5.6569E-03	1.0443E-02	2.6470E-03
	Std	2.9480E-04	9.1862E-04	3.6170E-03	2.8656E-03	4.0691E-03	1.2958E-04
TP7	Mean	2.0317E-02	2.6278E-01	1.8574E-01	1.2595E-01	3.0559E-01	5.8823E-02
	Std	2.2433E-02	9.8250E-02	4.0910E-02	4.9416E-02	2.5900E-01	1.0787E-01
TP8	Mean	2.1159E-02	3.7910E-02	5.4262E-02	3.0324E-02	2.2897E-01	1.1606E-02
	Std	2.9810E-03	4.4885E-03	7.3646E-03	7.9532E-03	3.3599E-01	1.6913E-03
TP9	Mean	5.6436E-03	1.0715E-02	1.8578E-02	1.5423E-02	7.3809E-03	2.1252E-03
	Std	2.1447E-03	1.4405E-02	3.2678E-03	8.2066E-03	4.9186E-03	1.3988E-04
TP10	Mean	2.7850E-02	5.2242E-02	6.8782E-02	2.5594E-02	9.8962E-02	1.3106E-02
	Std	4.5880E-03	5.9846E-03	1.0115E-02	1.2280E-02	1.5347E-01	1.1319E-03
TP11	Mean	2.1591E-01	1.3991E-01	1.5700E-01	1.2130E-01	2.1808E-01	9.1685E-02
	Std	6.6061E-02	1.9874E-02	1.6267E-02	1.9979E-02	9.6049E-02	3.6316E-03
TP12	Mean	2.2511E-01	1.3157E-01	1.8893E-01	1.5000E-01	4.1425E-01	9.2759E-02
	Std	5.6969E-02	2.1701E-02	2.9966E-02	1.8690E-02	1.6347E-01	3.5739E-03
TP13	Mean	8.1370E-03	4.7153E-03	1.8535E-02	5.6284E-02	3.5475E-02	3.1957E-03
	Std	3.1754E-03	2.8659E-04	3.0599E-03	1.6803E-02	2.8777E-02	2.6221E-04
TP14	Mean	1.3649E-01	1.4035E-01	1.7659E-01	1.6332E-01	2.3852E-01	9.8903E-02
	Std	1.7283E-02	1.3265E-02	1.5188E-02	1.6619E-02	1.7749E-01	7.4769E-03
TP15	Mean	1.2513E-01	1.4636E-01	1.8341E-01	1.4710E-01	1.9297E-01	1.0291E-01
	Std	9.8741E-03	1.5911E-02	1.6693E-02	1.6311E-02	8.4177E-02	1.2630E-02
TP16	Mean	1.9877E-01	2.4343E-01	2.6583E-01	1.4885E-01	6.1912E-01	1.9415E-01
	Std	1.1636E-03	4.8728E-02	3.9451E-02	4.4732E-02	4.2749E-01	4.2011E-04
TP17	Mean	1.0328E-01	1.1700E-01	1.3641E-01	7.6933E-02	4.6011E-01	9.5413E-02
	Std	2.6676E-03	5.6220E-03	8.7409E-03	7.4139E-03	5.1952E-01	1.7723E-03
TP18	Mean	9.1022E-02	8.6010E-02	9.6766E-02	6.8602E-02	8.6905E-02	8.3046E-02
	Std	4.4171E-03	1.5945E-03	4.6806E-03	4.5128E-03	2.5132E-03	1.7015E-04
TP19	Mean	1.6210E-01	1.7599E-01	2.0472E-01	1.4492E-01	2.0303E-01	1.4756E-01
	Std	6.4253E-03	1.2161E-02	1.6993E-02	9.0120E-03	9.1653E-02	1.5241E-03
TP20	Mean	3.0629E-01	2.4575E-01	2.7787E-01	2.1402E-01	4.0929E-01	2.0256E-01
	Std	3.8097E-02	1.7685E-02	2.8653E-02	1.2033E-02	1.3565E-01	5.7944E-03
TP21	Mean	2.0525E-01	2.3107E-01	2.6502E-01	2.3167E-01	2.5890E-01	1.9284E-01
	Std	1.7583E-02	1.2726E-02	1.1410E-02	1.0414E-02	7.1277E-02	5.4464E-03
TP22	Mean	2.5635E-01	2.0709E-01	2.1965E-01	1.9569E-01	2.4508E-01	1.6687E-01
	Std	5.7817E-02	1.5214E-02	9.0920E-03	1.4996E-02	5.3930E-02	2.9362E-03
TP23	Mean	3.5603E-01	3.0741E-01	3.1226E-01	2.5393E-01	5.1035E-01	2.5171E-01
	Std	5.1100E-02	3.7120E-02	1.3942E-02	1.0111E-02	1.7163E-01	5.0916E-03
TP24	Mean	2.7442E-01	2.4468E-01	2.5727E-01	2.2052E-01	3.1200E-01	2.0833E-01
	Std	5.0324E-02	1.1122E-02	1.2080E-02	1.5772E-02	9.8288E-02	4.8848E-03

Table 7. The IGDF values obtained by the competitive algorithms.

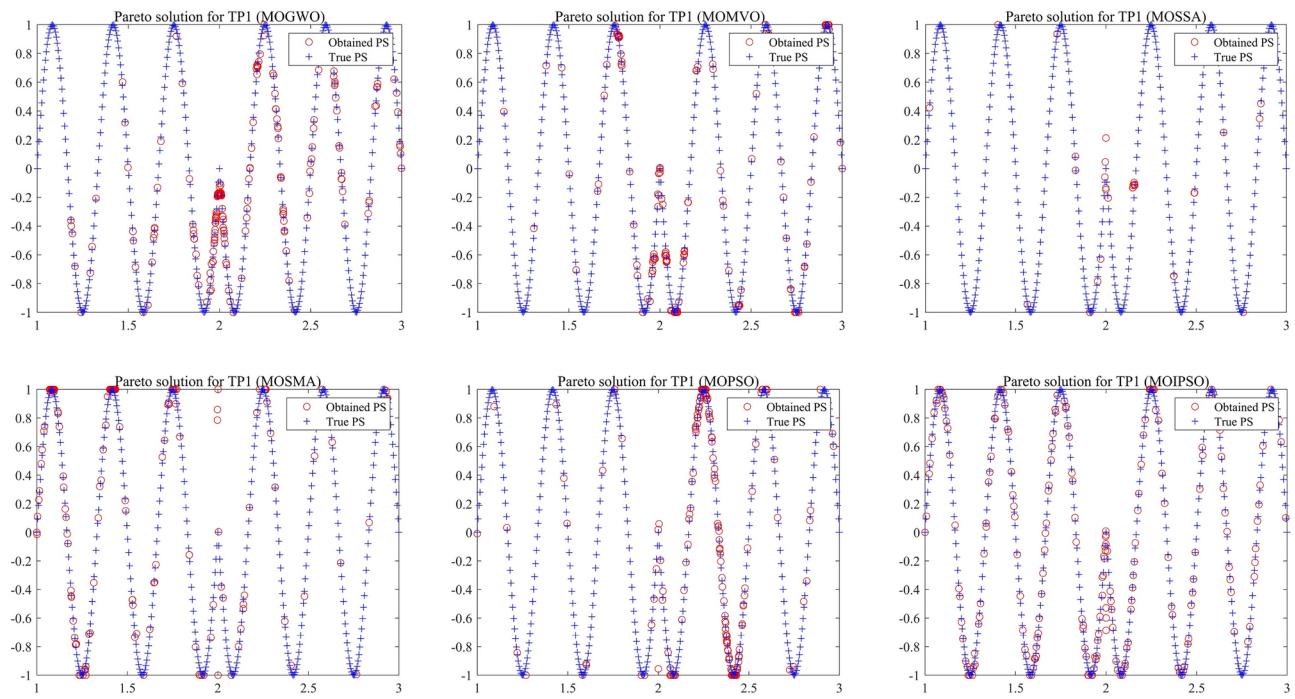


Fig. 7.. The PS obtained by competitive algorithms on TP1.

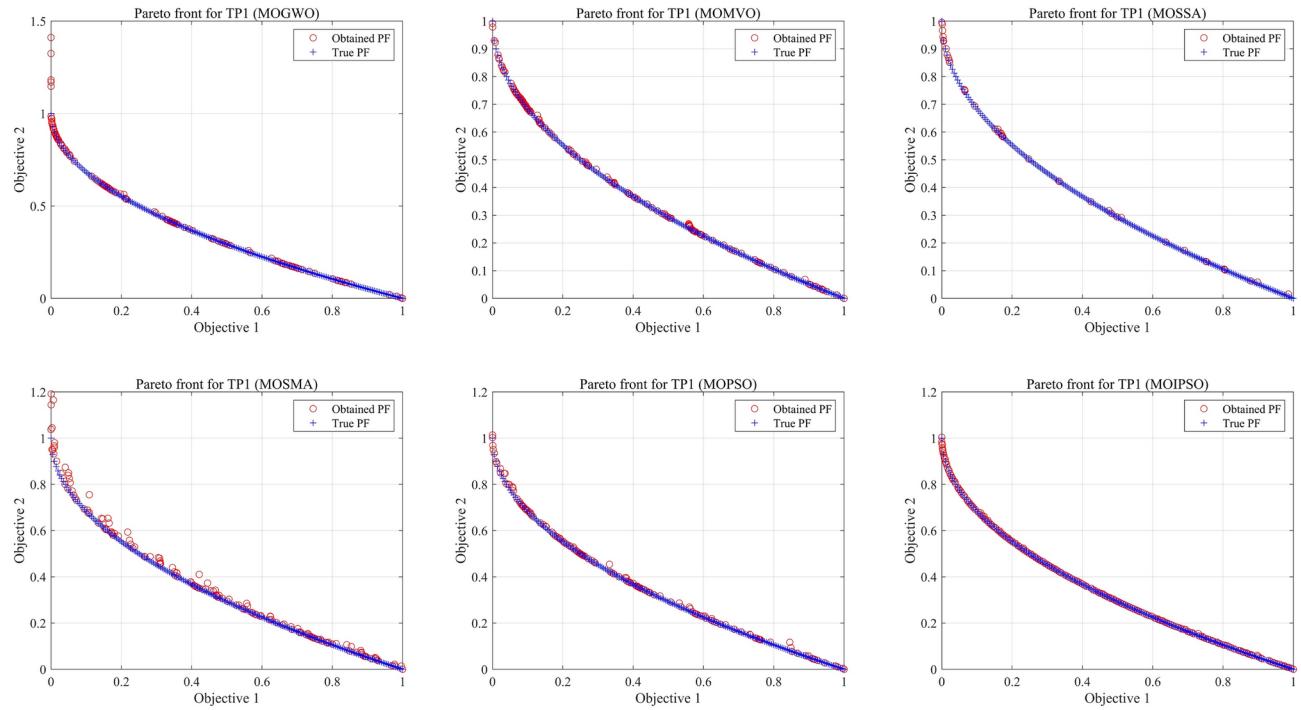
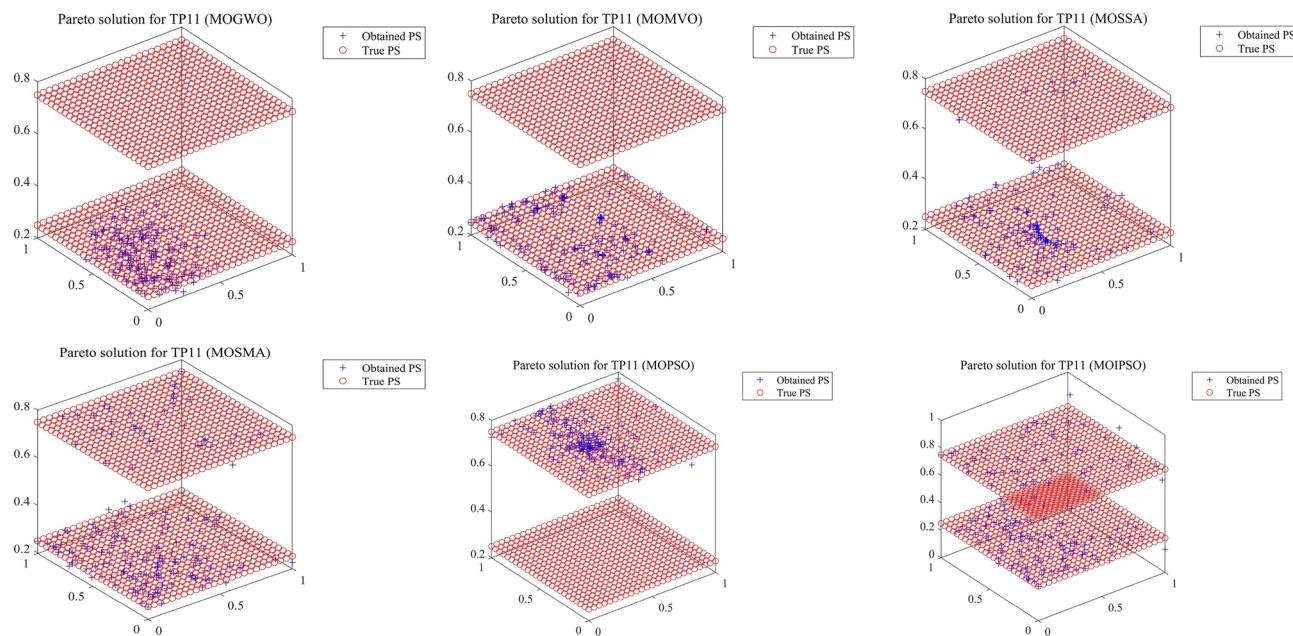
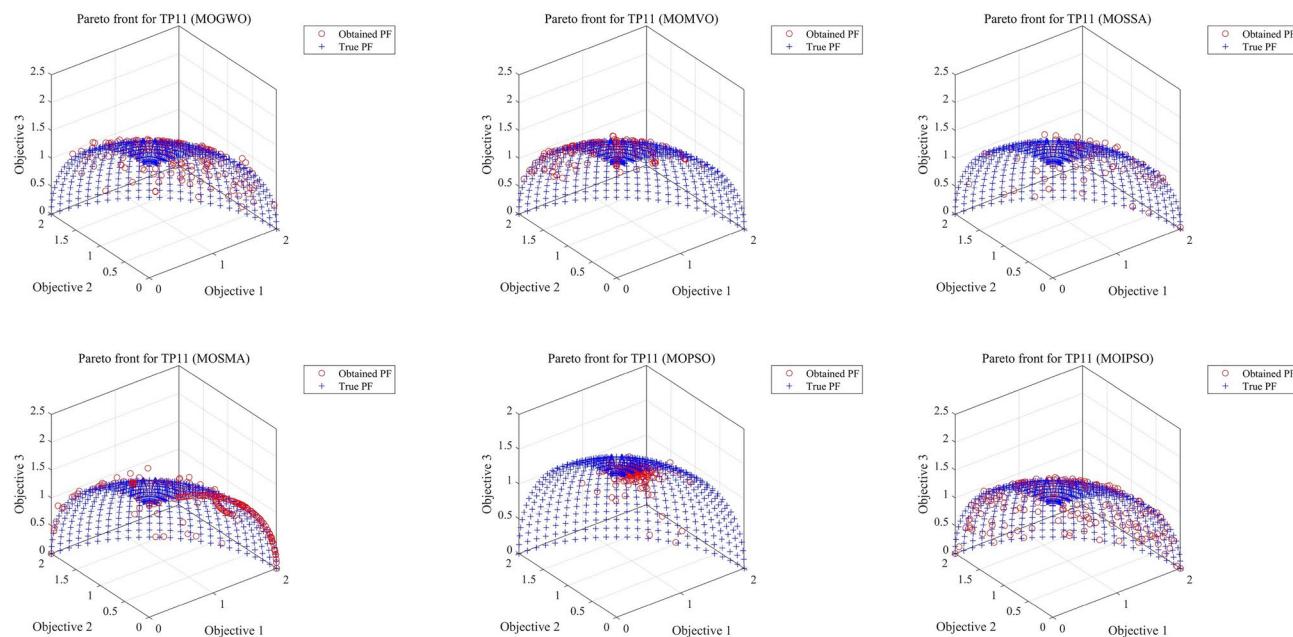


Fig. 8.. The PF obtained by competitive algorithms on TP1.

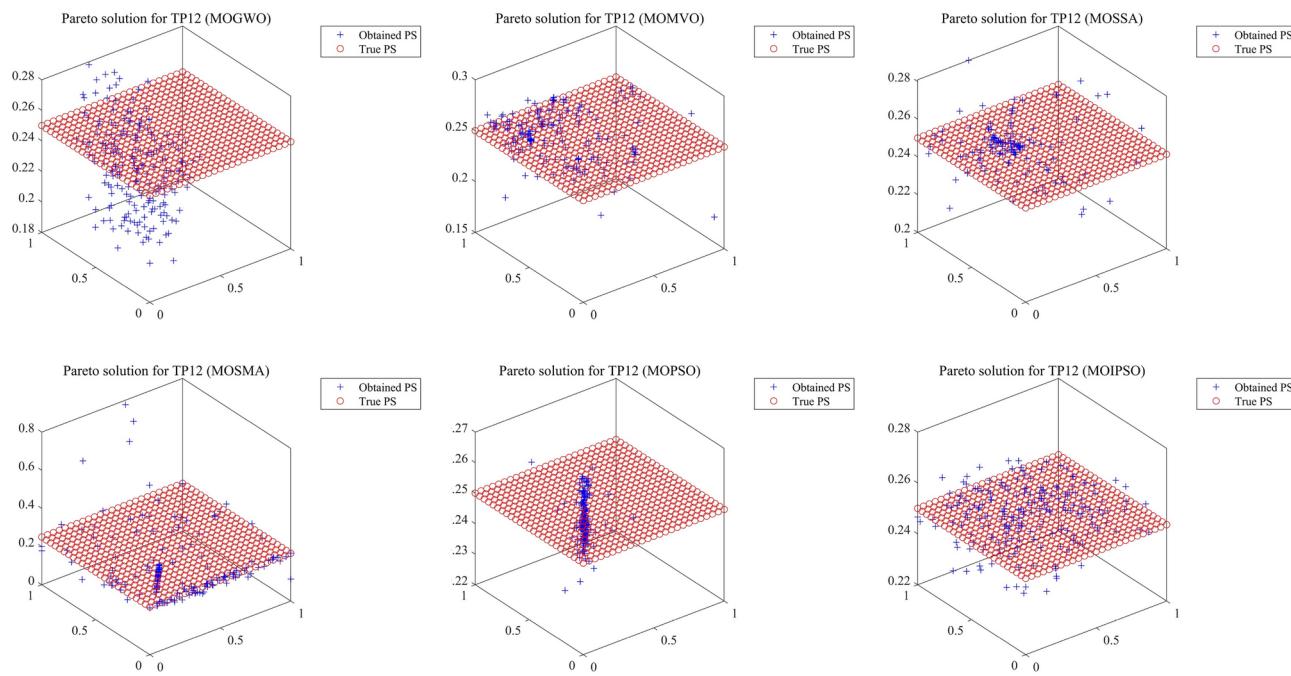
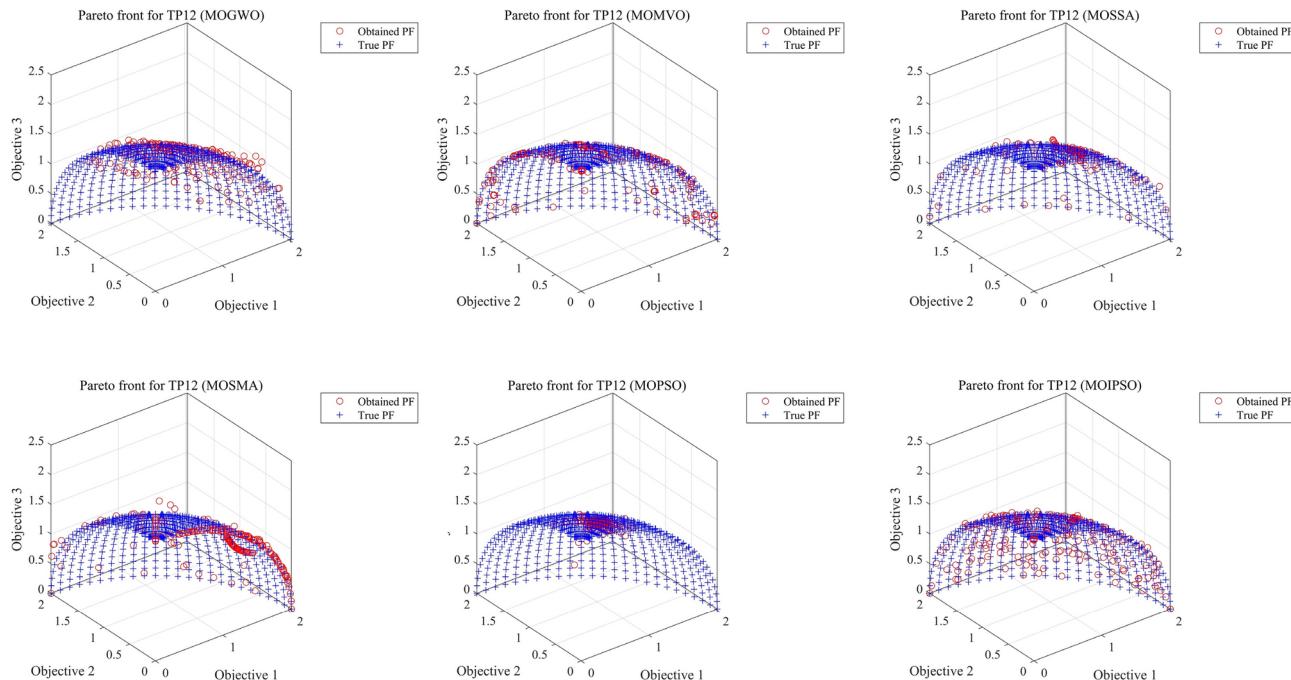
**Fig. 9.** The PS obtained by competitive algorithms on TP11.**Fig. 10.** The PF obtained by competitive algorithms on TP11.

TP11 and TP12, although MOIPSO does not completely cover the real Pareto optimal front, it still demonstrates the best performance in terms of uniform distribution and convergence of Pareto solutions. In summary, the performance of MOIPSO demonstrates significant superiority in multimodal test problems.

Statistical analysis

Friedman rank test

In order to ensure the stability and reproducibility of the experimental results, the Friedman statistical test method was used to conduct an in-depth analysis of the experimental results, and the test results were summarized in Table 8. The table detailedly presents the statistical test results of all algorithms on various evaluation indicators, including the average and specific rankings of each algorithm. Although MOIPSO did not secure first place in the rPSP index, it ranked second, surpassing other algorithms. It is worth noting that the MOIPSO algorithm achieves the lowest comprehensive average ranking score on the four key evaluation indicators, securing a firm

**Fig. 11.** The PS obtained by competitive algorithms on TP12.**Fig. 12.** The PF obtained by competitive algorithms on TP12.

first place. This result underscores the leading position of MOIPSO among all the algorithms involved in the comparison. In summary, the proposed MOIPSO algorithm has been conclusively demonstrated as a superior performing algorithm, and its effectiveness and advantages are well-supported by robust data.

Wilcoxon rank sum test

The Wilcoxon signed-rank test is a nonparametric method. Nonparametric tests differ from parametric tests in that they do not rely on specific assumptions about the population distribution (e.g., normal distribution). These tests are particularly useful when the population distribution is unknown or does not meet the assumptions

Algorithms	rPSP score (rank)	rHV score (rank)	IGDX score (rank)	IGDF score (rank)	Synthesis score (rank)
MOGWO	4.46 (6)	3.08 (2)	4.06 (5)	3.48 (4)	3.77 (4)
MOMVO	3.79 (4)	3.67 (3)	3.88 (4)	3.46 (3)	3.70 (3)
MOSSA	3.69 (3)	4.23 (5)	3.75 (3)	4.42 (5)	4.02 (5)
MOSMA	2.25 (1)	3.79 (4)	2.67 (2)	3.25 (2)	2.99 (2)
MOPSO	4.40 (5)	4.79 (6)	4.38 (6)	5.13 (6)	4.67 (6)
MOIPSO	2.42 (2)	1.44 (1)	2.27 (1)	1.27 (1)	1.85 (1)

Table 8. Friedman test results of competing algorithms on different performance metrics. Significant values are in bold

MOIPSO vs	MOGWO	MOMVO	MOSSA	MOSMA	MOPSO
TP1	+	+	+	+	+
TP2	=	=	=	=	+
TP3	+	+	+	+	+
TP4	+	+	+	+	+
TP5	+	+	+	+	+
TP6	+	=	=	-	=
TP7	=	+	+	=	+
TP8	+	+	+	+	+
TP9	+	+	+	+	+
TP10	+	=	=	=	=
TP11	+	+	-	-	=
TP12	+	+	+	+	+
TP13	=	+	+	-	+
TP14	+	+	+	+	+
TP15	+	+	+	+	+
TP16	+	-	-	-	=
TP17	+	-	=	-	+
TP18	+	=	+	-	-
TP19	+	-	=	-	-
TP20	+	+	=	-	+
TP21	+	+	+	+	+
TP22	+	+	-	-	=
TP23	+	+	+	-	+
TP24	+	=	-	-	=
Overall (+/-/=)	21/3/0	16/5/3	14/6/4	10/3/11	16/6/2

Table 9. The rPSP Wilcoxon signed rank test results.

required for parametric tests. This method is primarily used to assess whether the difference between two related samples or paired observations is statistically significant 53. We used the nonparametric Wilcoxon signed-rank test method to conduct a comparative analysis of the MOIPSO algorithm with other competitive algorithms, and the detailed results are presented in Tables 9, 10, 11, and 12. When the significance level $\alpha=0.05$, we consider that the comparison algorithm and MOIPSO optimization perform similarly. When $\alpha<0.05$, it means that there is a significant difference between the competitive algorithm and MOIPSO, otherwise there is no significant difference. To visually represent the performance of MOIPSO, we use the symbols "+/-/=—" to highlight the advantages and disadvantages of MOIPSO compared to other algorithms: "+" indicates superiority (a significant difference), "=" means similar performance (similar performance), and "-" indicates slight inferiority (no significant difference). In the last row of the table, we summarize the statistical results, clearly illustrating the differences between the MOIPSO algorithm and other algorithms across various indicators. This further confirms the significant advantage of the MOIPSO algorithm in comprehensive performance, distinguishing it from its competitors and demonstrating outstanding overall performance.

Optimization design of foundation pit engineering for rail transit upper cover project Background

With the construction of transportation infrastructure, including high-speed, intercity, suburban, and urban rail transit, and the gradual deepening of concepts such as station-city integration and transit-oriented development (TOD), an urban development and construction mode based on public transportation, is becoming increasingly

MOIPSO vs	MOGWO	MOMVO	MOSSA	MOSMA	MOPSO
TP1	+	+	+	+	+
TP2	-	=	+	=	+
TP3	+	+	+	+	+
TP4	+	+	+	+	+
TP5	+	+	+	+	+
TP6	+	+	+	+	+
TP7	=	+	+	+	+
TP8	+	+	+	+	+
TP9	+	+	+	+	+
TP10	+	+	+	+	+
TP11	+	=	=	-	+
TP12	+	+	+	=	+
TP13	+	=	+	-	+
TP14	+	=	=	=	=
TP15	+	+	=	=	+
TP16	+	+	+	+	+
TP17	+	+	+	+	+
TP18	+	+	+	+	+
TP19	+	+	+	+	+
TP20	+	=	=	-	+
TP21	+	+	=	=	+
TP22	+	=	+	=	+
TP23	+	+	+	-	+
TP24	+	+	=	-	+
Overall (+/-/=)	22/1/1	18/6/0	18/6/0	13/6/5	23/1/0

Table 10. The rHV Wilcoxon signed rank test results.

popular. This approach also demonstrates significant value in optimizing the urban spatial pattern and promoting high-quality urban development. As urban development enters the era of stock renewal and new planned lines are constructed, the comprehensive development and utilization of TOD mode—transformation and renewal at existing large-scale rail transit hubs—has emerged as an inevitable trend. As rail transit infrastructure expands, the number of complex TOD projects near existing lines continues to grow. With ongoing rail transit operations, the demand for new foundation pit construction is rising. Therefore, reducing the construction cost of foundation pit projects is crucial for effectively controlling their safety. In this paper, the optimization design of foundation pit engineering is illustrated using the example of Shaoxing North Station on Shaoxing Metro Line 1, as shown in Fig. 13 According to investigations, the primary layers within the foundation pit include complex fill soil and sandy silt at the top, silt soil and silty clay in the middle, and silt alongside dense and strong tufa sandstone in the lower section. The main formation parameters are shown in Table 13.

Mathematical model

In this study, we selected the surrounding environment within the scope of the Shaoxing North Station project to design the foundation pit engineering, and used the finite element software Midas GTS NX V2023 to numerically simulate the main construction process of the foundation pit. The software can be accessed at the following URL: <https://product.midasit.cn/index/>, as shown in Figs. 14 and 15.

According to reference⁵⁴, only the vertical displacement of the left foundation pit was selected for analysis, as the changes in vertical displacement were found to be synergistic. The unoptimized initial scheme was derived from production practice experience. Using the maximum vertical displacement of the tunnel under the initial scheme of the foundation pit excavation—with¹³longitudinal blocks, 1 m anti-floating plate thickness, 15 m solid pile length, 1 m diameter, and²⁷unilateral pullout piles (where the vertical spacing of the pullout piles is 3 m)—as the reference, the influence of foundation pit parameters was analyzed through numerical simulation. The maximum vertical displacement of the tunnel, under conditions varying by the number of vertical blocks, thickness of the anti-floating plate, length of solid piles, diameter of anti-pulling piles, and number of anti-pulling piles, was obtained for each variable. Matlab programming was employed to derive fitting polynomials that describe the relationship between the variables and the multiples of maximum vertical displacement under the initial scheme. For each fitting polynomial, the factors include: excavation partition number μ_1 , anti-floating plate thickness μ_2 , solid pile length μ_3 , anti-pulling pile diameter μ_4 , and the number of unilateral anti-pulling piles μ_5 , with the respective maximum vertical displacement expressions of the tunnel.

MOIPSO vs	MOGWO	MOMVO	MOSSA	MOSMA	MOPSO
TP1	+	+	+	+	+
TP2	=	=	=	=	+
TP3	+	+	+	+	+
TP4	+	+	+	+	+
TP5	+	+	+	+	+
TP6	+	=	=	-	=
TP7	=	+	+	=	+
TP8	+	+	+	+	+
TP9	+	+	+	+	+
TP10	+	=	=	=	=
TP11	+	+	-	-	=
TP12	+	+	+	+	+
TP13	=	+	+	-	+
TP14	+	+	+	+	+
TP15	+	+	+	+	+
TP16	+	-	-	-	+
TP17	+	+	+	-	+
TP18	+	+	+	-	=
TP19	+	=	=	-	-
TP20	+	+	+	-	+
TP21	+	+	+	+	+
TP22	+	=	=	-	=
TP23	+	+	+	-	+
TP24	+	+	=	-	=
Overall (+/-/-)	21/3/0	18/5/1	16/6/2	10/3/11	17/6/1

Table 11. The IGDX Wilcoxon signed rank test results.

$$\begin{aligned}
 & \delta(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \\
 = 17.01 \times & \left(-0.0000131849\mu_1^5 + 0.000469463\mu_1^4 - 0.00604901\mu_1^3 + 0.0343602\mu_1^2 - 0.0992232\mu_1 + 1.26013 \right) \\
 & \times \left(-0.01459\mu_3^5 + 0.0959586\mu_3^4 - 0.212419\mu_3^3 + 0.199501\mu_3^2 - 0.231241\mu_3 + 1.1189 \right) \\
 & \times \left(0.00000216142\mu_3^5 - 0.000113656\mu_3^4 + 0.00217459\mu_3^3 - 0.018897\mu_3^2 + 0.0466485\mu_3 + 1.32445 \right) \\
 & \times (-0.03351\mu_4 + 1.03334) (-0.0141426\mu_5 + 1.39165) \#
 \end{aligned} \quad (10)$$

According to the provisional specification: $\delta(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) < 20$ mm.

The empirical data collected indicate that the corresponding unit price of the small shaft project is approximately 400 yuan/m³, the corresponding unit price of the lock ring beam project of the small shaft is approximately 450 yuan/m³, the corresponding unit price of the anchor project required for the small shaft is approximately 75 yuan/m, the corresponding unit price of the anti-floating plate is approximately 550 yuan/m³, the corresponding unit price of the uplift pile is approximately 900 yuan/m³, the corresponding unit price of the uplift pile empty pile (hole) project is 200 yuan/m³, and the total price of the excavation and the overall large foundation pit support is fixed. Substituting the number of longitudinal blocks of foundation pit excavation (μ_1), the thickness of the anti-floating plate (μ_2), the length of the uplift pile (μ_3), the diameter of the uplift pile (μ_4), and the number of unilateral uplift piles (μ_5), the variable value c of the engineering cost in foundation pit engineering of the rail transit upper cover project can be obtained as follows:

$$c(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) = (119850 + 4500 \frac{80.5}{\mu_1}) \mu_1 + 1638175\mu_2 + 450\pi\mu_4^2\mu_3\mu_5 + 500\pi\mu_4^2\mu_5 \quad (11)$$

In this paper, the goal of optimizing the design of the foundation pit engineering of the rail transit upper cover project is to determine the reasonable values of the five design variables: the number of longitudinal blocks (μ_1), the thickness of the anti-floating plate (μ_2), the length of the uplift pile (μ_3), the diameter of the uplift pile (μ_4), and the number of unilateral uplift piles (μ_5) for the foundation pit excavation. The aim is to minimize the maximum vertical displacement of the tunnel and the variable part of the project cost, specifically:

$$\min \left\{ \begin{array}{l} \delta(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \\ c(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \end{array} \right\} \quad (12)$$

The following constraints exist:

$$s.t. \left\{ \begin{array}{l} 1 \leq \mu_1 \leq 13, \mu_1 \in N, 0 \leq \mu_2 \leq 2.8, \mu_2 \in 0.1 \times N, 3 \leq \mu_3 \leq 21, \mu_3 \in 0.5 \times N, 0.6 \leq \mu_4 \leq 1.6, \\ \mu_4 \in 0.05 \times N, 6 \leq \mu_5 \leq 41, \mu_5 \in N, \delta < 20 \end{array} \right\} \quad (13)$$

MOIPSO vs	MOGWO	MOMVO	MOSSA	MOSMA	MOPSO
TP1	+	+	+	+	+
TP2	-	=	+	=	+
TP3	+	+	+	+	+
TP4	+	+	+	+	+
TP5	+	+	+	+	+
TP6	+	+	+	+	+
TP7	=	+	+	+	+
TP8	+	+	+	+	+
TP9	+	+	+	+	+
TP10	+	+	+	+	+
TP11	+	+	+	+	+
TP12	+	+	+	+	+
TP13	+	+	+	+	+
TP14	+	+	+	+	+
TP15	+	+	+	+	+
TP16	+	+	+	-	+
TP17	+	+	+	-	+
TP18	+	+	+	-	+
TP19	+	+	+	=	+
TP20	+	+	+	+	+
TP21	+	+	+	+	+
TP22	+	+	+	+	+
TP23	+	+	+	=	+
TP24	+	+	+	+	+
Overall (+/-/=)	22/1/1	23/1/0	24/0/0	18/3/3	24/0/0

Table 12. The IGDF Wilcoxon signed rank test results.**Fig. 13..** Shaoxing Metro Line 1 Shaoxing North Station and the surrounding environment.

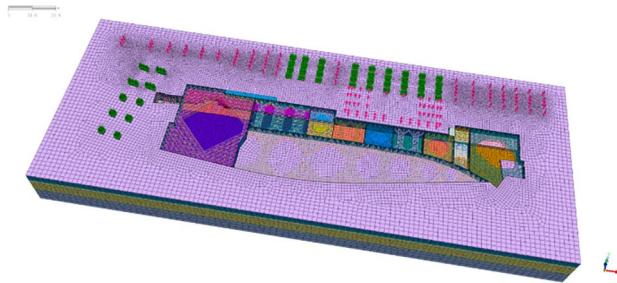
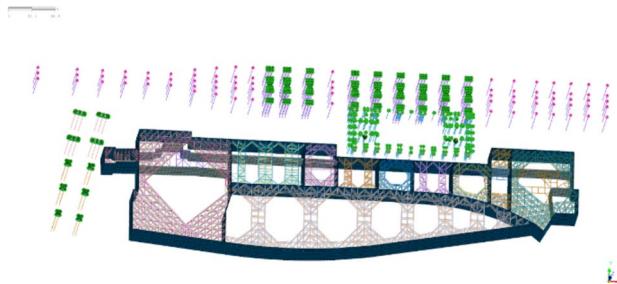
In the MOIPSO algorithm for optimizing the foundation pit design, the design variables ($\mu_1, \mu_2, \mu_3, \mu_4, \mu_5$) are first encoded as the particle position information, with each particle representing a potential foundation pit design. The fitness value of each particle is then calculated based on the objective functions (the maximum vertical displacement of the tunnel and the project cost). During the iteration process, particles update their positions based on their own historical best position, the global best position, and velocity information. The velocity update formula is as follows:

$$\nu_{r,d}(t+1) = \omega\nu_{r,d}(t) + c_1 r_1 (pbest_{r,d}(t) - x_{r,d}(t)) + c_2 r_2 (gbest_d(t) - x_{r,d}(t)) . \quad (13)$$

$$x_{r,d}(t+1) = x_{r,d}(t) + v_{r,d}(t+1) \quad (14)$$

$$\begin{aligned} c_1 &= 2.05 \times |\cos(2 \times \pi \times rand)| \\ c_2 &= 2.05 \times |\sin(2 \times \pi \times rand)| \end{aligned} \quad (15)$$

Soil Layer	Natural weight γ (kN/m ³)	Consolidation quick shear		Unconfined compressive strength qu (kPa)	modulus of compression Es1-2 (MPa)	M value M (MN/m ⁴)
		C (kPa)	φ (°)			
(1)-1 Miscellaneous fill	18.3	20.1	27.0	/	8.30	/
(2) Sandy silt	18.5	12.8	29.4	85.8	12.31	5.0*
(3) Mucky clay	17.0	12.0*	9.3	18.8	2.54	1.2*
(4)-1 Silty clay	18.8	30.9*	14.0*	66.4	5.72	5.6*
(4)-2 Clayey silt	18.7	13.8	28.5	37.2	9.55	8.0*
(5)-1 Sandy silt	18.8	13.6	28.8	43.1	10.55	10.0*
(5)-2 Muddy silty soil	17.6	18.0	12.5*	34.7	3.31	1.5*
(6)-1 Clay	18.8	47*	15.4	80.5	6.57	8.0*
(6)-2 Silty clay	18.5	35.4	15.6	50.0*	5.42	6.3*
(7)-1 Clay		34.5	14.0		4.81	5.8*
(7)-2 Clay		54.1	16.1		7.44	8.8*
(8)-1 Silty clay		39.3	20.8		7.53	9.5*
(8)-2 Silt		3.0*	30.0*		18	10*
(10)-1 Fully weathered tuffaceous rock		43.6	13.7		5.21	6.7*
(10)-2 Strongly weathered tuffaceous rock	22.0*		35			

Table 13. Formation parameter table.**Fig. 14.** Overall model diagram.**Fig. 15.** Peripheral structure and foundation pit supporting structure diagram.

$$\begin{aligned} dx &= \text{randn}(\mu, \sigma) \times (ub - lb) \\ X(idx) &= X(idx) + dx \end{aligned} \quad (16)$$

where, r_1 and r_2 represent random numbers within the interval $[0, 1]$. c_1 and c_2 are trigonometric function-based acceleration factors. ω is the inertia weight. $v_{r,d}(t)$ is the current velocity of the particle r in the d th dimension, $x_{r,d}(t)$ represents the current position of the particle r in the d th dimension, $pbest_{r,d}(t)$ presents the historical optimal position of the particle r in the d th dimension, $gbest_d(t)$ denotes the historical optimal position of the group in the d th dimension. randn denotes the Gaussian distribution, the expectation $\mu = 0$, the variance $\sigma = 0.1 \times (1 - \frac{t}{T})$, ub and lb are the upper and lower bounds of the problem, respectively, and idx is the index of half of the population randomly selected from the population.

Through continuous iterations, the MOIPSO algorithm gradually approaches the optimal foundation pit design, minimizing the project cost while satisfying the tunnel vertical displacement constraint. In this process, the algorithm adjusts the particles' velocity and position to explore different combinations of design variables, evaluating their corresponding objective function values, and ultimately finding the optimal or near-optimal solution set. The pseudocode for MOIPSO to optimize this engineering problem is shown in **Algorithm 1**. It is worth noting that, for the convenience of future researchers, we have made the code used to optimize this application publicly available at <https://au.mathworks.com/matlabcentral/fileexchange/177404-moipso-optimization-engineering-problem>.

Simulation analysis

In this section, we use MOGWO, MOMVO, MOSSA, MOSMA, MOPSO, and MOIPSO to solve the optimization design problem of foundation pit engineering, assuming that the maximum vertical displacement is Objective 1 and the project cost is Objective 2. The algorithm parameters are consistent with those in previous chapters, and Fig. 16 shows the Pareto front of six multi-objective optimization algorithms for this problem. Each subfigure shows the optimization results for the two objectives, with the set of Pareto optimal solutions of each algorithm represented by a scatter plot. The Pareto front of MOIPSO shows broad coverage for both objectives, especially in the low-value range, indicating the algorithm's effectiveness in finding near-optimal solutions. Additionally, the distribution of points on the Pareto front of MOIPSO is dense, especially when the value of Objective 1 is low. This indicates that the algorithm maintains high solution density and diversity during the exploration and development process, aiding in finding more balanced solutions. Furthermore, the figure shows that MOIPSO can find high-quality solutions on the boundary of Objective 1 and Objective 2. This indicates that MOIPSO performs better when balancing the two objectives. Finally, compared to MOGWO, MOMVO, MOSSA, and MOPSO, MOIPSO appears smoother on the Pareto front, indicating better continuity and consistency of solutions. Figure 17 shows the box plots of the evaluation metric Spacing for the six multi-objective algorithms, which is typically used to measure the distribution uniformity of Pareto front solutions. The box plot shows that MOIPSO has the lowest median spacing in the spacing index, indicating that the solutions are more closely distributed. The smallest interquartile range indicates the most uniform distribution of solutions. There are no outliers, further proving the consistency of the solution distribution. Overall, MOIPSO has a significant advantage in the distribution uniformity of Pareto front solutions.

In summary, MOIPSO demonstrates clear superiority in the performance of the Pareto front, including wider coverage, higher solution density, excellent boundary performance, and better solution continuity. These characteristics enable MOIPSO to perform better in multi-objective optimization problems, finding a more balanced and high-quality Pareto optimal solution set, and providing a superior solution for the optimization design of foundation pit engineering in rail transit upper cover projects.

Discussion

In the entire fifth section, the MOIPSO algorithm demonstrates superior optimization results compared to other algorithms in addressing complex geotechnical engineering problems, such as foundation pit engineering. In

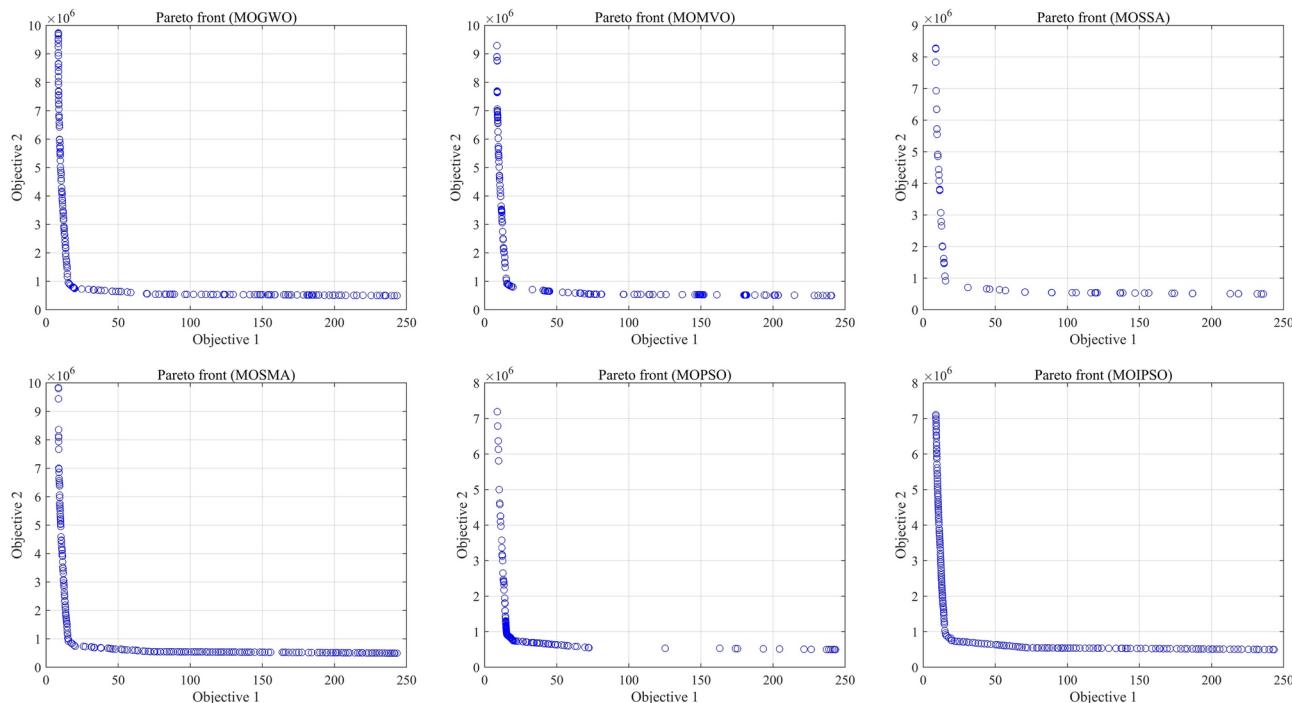


Fig. 16. Pareto fronts of different competitor algorithms.

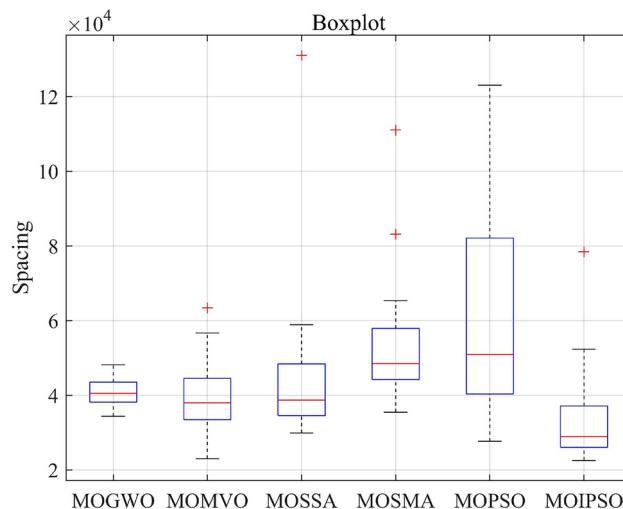


Fig. 17. Boxplots of the different competitor algorithms.

future work, the MOIPSO algorithm framework will be applied to foundation pit projects with varying geological conditions, scales, and surrounding environmental requirements. It will only be necessary to adjust the design variables and constraints according to the specific engineering context. This means that geotechnical engineers can use this method to quickly establish the corresponding optimization model for a variety of complex and variable real-world projects, providing a general and effective approach to solving multi-objective optimization problems in engineering. For example, when faced with large variations in the mechanical properties of different strata, geotechnical engineers can easily apply the MOIPSO algorithm to the project's optimization design by adjusting parameter ranges and constraint conditions, without the need to develop a complex new optimization algorithm. This approach not only improves work efficiency but also reduces the engineering design cycle.

Summary and outlook

In this paper, a multi-objective particle swarm optimization algorithm (MOIPSO) is proposed, combining the fast non-dominated sorting technique to efficiently evaluate and approximate the Pareto optimal solution set while maintaining the primary search mechanism of PSO. To enhance the diversity and generalization of the solution set, the crowding distance mechanism is introduced, ensuring that the algorithm achieves a good balance between multiple optimization objectives and covers a wider solution space. Additionally, the MOIPSO algorithm integrates a trigonometric acceleration factor and an adaptive Gaussian mutation strategy, enhancing the exploration ability of particles in the search space and enabling them to move towards the global optimal solution more effectively. MOIPSO is tested on the multi-objective benchmark set CEC2020, with its performance indicators such as IGDX, rPSP, IGDF, and rHV analyzed. The results show that MOIPSO performs better than MOPSO, MOGWO, MOSSA, MOMVO, and MOSMA. Finally, the practical effectiveness of the proposed algorithm is further verified by solving the design problem of foundation pit engineering for rail transit upper cover projects. Compared to other algorithms, MOIPSO demonstrates strong competitiveness. Based on extensive comparative analysis of various optimization problems, MOIPSO significantly outperforms existing multi-objective optimization methods in terms of performance and robustness.

In future work, we will further verify the performance of MOIPSO through more challenging engineering applications and continuously improve its optimization capabilities. We plan to apply MOIPSO to more complex real-world problems, such as microgrid allocation, wind prediction, WSN node coverage, and mobile cell frequency assignment. In addition, multi-objective versions of MOIPSO are also a promising research topic.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 14 August 2024; Accepted: 17 January 2025

Published online: 26 March 2025

References

- Zuo, F. M. & Li, T. Z. In *19th COTA International Conference of Transportation Professionals (CICTP) - Transportation in China 2025*. 4841–4852 (2019).
- Zhou, Y. E., Li, L. B. & Zhang, Y. H. Location of transit-oriented development stations based on multimodal network equilibrium: Bi-level programming and paradoxes. *Transp. Res. Part a-Policy Pract.* <https://doi.org/10.1016/j.tra.2023.103729> (2023).
- Li, K. et al. A multi-strategy enhanced northern goshawk optimization algorithm for global optimization and engineering design problems. *Comput. Methods Appl. Mech. Eng.* **415**, 116199. <https://doi.org/10.1016/j.cma.2023.116199> (2023).
- Huang, H., Zheng, B., Wei, X., Zhou, Y. & Zhang, Y. NSCSO: A novel multi-objective non-dominated sorting chicken swarm optimization algorithm. *Sci. Rep.* **14**, 4310. <https://doi.org/10.1038/s41598-024-54991-0> (2024).

5. Agarwal, P., Agrawal, R. K. & Kaur, B. Multi-objective particle swarm optimization with guided exploration for multimodal problems. *Appl. Soft Comput.* **120**, 108684. <https://doi.org/10.1016/j.asoc.2022.108684> (2022).
6. Zhong, K., Zhou, G., Deng, W., Zhou, Y. & Luo, Q. MOMPA: Multi-objective marine predator algorithm. *Comput. Methods Appl. Mech. Eng.* **385**, 114029. <https://doi.org/10.1016/j.cma.2021.114029> (2021).
7. Alp, S., Dehkharghani, R., Akan, T. & Bhuiyan, M. A. N. MOBRO: Multi-objective battle royale optimizer. *J. Supercomput.* **80**, 5979–6016. <https://doi.org/10.1007/s11227-023-05676-4> (2024).
8. Jameel, M. & Abouhawwash, M. Multi-objective mantis search algorithm (MOMSA): A novel approach for engineering design problems and validation. *Comput. Methods Appl. Mech. Eng.* **422**, 116840. <https://doi.org/10.1016/j.cma.2024.116840> (2024).
9. Coello, C. A. C. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comput. Intell. Mag.* **1**, 28–36. <https://doi.org/10.1109/MCI.2006.1597059> (2006).
10. Sun, S. Z., Wang, C. X., Wang, Y., Zhu, X. H. & Lu, H. C. Multi-objective optimization dispatching of a micro-grid considering uncertainty in wind power forecasting. *Energy Rep.* **8**, 2859–2874. <https://doi.org/10.1016/j.egyr.2022.01.175> (2022).
11. Khishe, M., Orouji, N. & Mosavi, M. R. Multi-objective chimp optimizer: An innovative algorithm for multi-objective problems. *Expert Syst. Appl.* <https://doi.org/10.1016/j.eswa.2022.118734> (2023).
12. Saini, N. & Saha, S. Multi-objective optimization techniques: A survey of the state-of-the-art and applications multi-objective optimization techniques. *Eur. Phys. J.-Spec. Top.* **230**, 2319–2335. <https://doi.org/10.1140/epjs/s11734-021-00206-w> (2021).
13. Zhang, J. W. & Zhang, X. Y. Adaptive wireless network multi-objective optimization algorithm based on image synthesis. *EURASIP J. Image Video Process.* <https://doi.org/10.1186/s13640-018-0289-3> (2018).
14. He, Y.-L. et al. A Novel MOWSO algorithm for microgrid multi-objective optimal dispatch. *Electric Power Syst. Res.* **232**, 110374. <https://doi.org/10.1016/j.epsr.2024.110374> (2024).
15. Upadhyay, P., Chhabra, J. K. & IEEE. In 2016 1st India International Conference On Information Processing (IICIP) (2016).
16. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evolut. Comput.* **1**, 67–82. <https://doi.org/10.1109/4235.585893> (1997).
17. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
18. Kennedy, J. & Eberhart, R. In *Proceedings of ICNN'95 - International Conference on Neural Networks*. 1942–1948 vol. 1944.
19. Yue, Y. W., Peng, Y. & Wang, D. C. Deep learning short text sentiment analysis based on improved particle swarm optimization. *Electronics* <https://doi.org/10.3390/electronics12194119> (2023).
20. Elhani, D. et al. Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification. *Expert Syst. Appl.* <https://doi.org/10.1016/j.eswa.2023.120411> (2023).
21. Zhu, J., Liu, J. H., Chen, Y. X., Xue, X. S. & Sun, S. H. Binary restructuring particle swarm optimization and its application. *Biomimetics* <https://doi.org/10.3390/biomimetics8020266> (2023).
22. Mohamed, R. E., Hunjet, R., Elsayed, S. & Abbass, H. Connectivity-aware particle swarm optimisation for swarm shepherding. *IEEE Trans. Emerging Topics Comput. Intell.* **7**, 661–683. <https://doi.org/10.1109/tetc.2022.3195178> (2023).
23. Elsheikh, A. H. & Abd Elaziz, M. A. Review on applications of particle swarm optimization in solar energy systems. *Int. J. Environ. Sci. Technol.* **16**, 1159–1170. <https://doi.org/10.1007/s13762-018-1970-x> (2018).
24. Mirjalili, S., Saremi, S., Mirjalili, S. M. & Coelho, L. D. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst. Appl.* **47**, 106–119. <https://doi.org/10.1016/j.eswa.2015.10.039> (2016).
25. Got, A., Zouache, D. & Moussaoui, A. MOMRFO: Multi-objective Manta ray foraging optimizer for handling engineering design problems. *Knowl.-Based Syst.* **237**, 107880. <https://doi.org/10.1016/j.knosys.2021.107880> (2022).
26. Khodadadi, N., Abualigah, L. & Mirjalili, S. Multi-objective Stochastic Paint Optimizer (MOSPO). *Neural Comput. Appl.* **34**, 18035–18058. <https://doi.org/10.1007/s00521-022-07405-z> (2022).
27. Chen, Y. et al. An improved multi-objective particle swarm optimization with mutual information feedback model and its application. *Arab. J. Sci. Eng.* **47**, 9405–9421. <https://doi.org/10.1007/s13369-021-06178-2> (2022).
28. Panagant, N., Pholdee, N., Bureerat, S., Yildiz, A. R. & Mirjalili, S. A Comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems. *Arch. Comput. Methods Eng.* **28**, 4031–4047. <https://doi.org/10.1007/s11831-021-09531-8> (2021).
29. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **6**, 182–197. <https://doi.org/10.1109/4235.996017> (2002).
30. Zhang, Q. & Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolut. Comput.* **11**, 712–731. <https://doi.org/10.1109/TEVC.2007.892759> (2007).
31. Coello, C., Toscano Pulido, G. & Lechuga, M. Handling multiple objectives with particle swarm optimization. *Evolut. Comput. IEEE Trans.* **8**, 256–279. <https://doi.org/10.1109/TEVC.2004.826067> (2004).
32. Akbari, R., Hedayatzadeh, R., Ziari, K. & Hassanizadeh, B. A multi-objective artificial bee colony algorithm. *Swarm Evolut. Comput.* **2**, 39–52. <https://doi.org/10.1016/j.swevo.2011.08.001> (2012).
33. Mirjalili, S., Jangir, P. & Saremi, S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Appl. Intell.* **46**, 79–95. <https://doi.org/10.1007/s10489-016-0825-8> (2017).
34. Yapici, H. & Cetinkaya, N. A new meta-heuristic optimizer: Pathfinder algorithm. *Appl. Soft Comput.* **78**, 545–568. <https://doi.org/10.1016/j.asoc.2019.03.012> (2019).
35. Yi, J. H. et al. Behavior of crossover operators in NSGA-III for large-scale optimization problems. *Inf. Sci.* **509**, 470–487. <https://doi.org/10.1016/j.ins.2018.10.005> (2020).
36. Premkumar, M. et al. MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting. *IEEE Access* **9**, 3229–3248. <https://doi.org/10.1109/access.2020.3047936> (2021).
37. Deng, W. et al. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **585**, 441–453. <https://doi.org/10.1016/j.ins.2021.11.052> (2022).
38. Ganesh, N. et al. A novel decomposition-based multi-objective symbiotic organism search optimization algorithm. *Mathematics* <https://doi.org/10.3390/math11081898> (2023).
39. He, Y. L. et al. A Novel MOWSO algorithm for microgrid multi-objective optimal dispatch. *Electric Power Syst. Res.* <https://doi.org/10.1016/j.epsr.2024.110374> (2024).
40. Jiang, J. H., Wu, J. Q., Luo, J. M., Yang, X. & Huang, Z. L. MOBCA: Multi-objective besiege and conquer algorithm. *Biomimetics* <https://doi.org/10.3390/biomimetics9060316> (2024).
41. Sahoo, S. K. et al. An arithmetic and geometric mean-based multi-objective moth-flame optimization algorithm. *Cluster Comput. J. Netw. Softw. Tools Appl.* <https://doi.org/10.1007/s10586-024-04301-0> (2024).
42. Pandya, S. B., Kalita, K., Jangir, P., Ghadai, R. K. & Abualigah, L. Multi-objective geometric mean optimizer (MOGMO): A novel metaphor-free population-based math-inspired multi-objective algorithm. *Int. J. Comput. Intell. Syst.* <https://doi.org/10.1007/s44196-024-00420-z> (2024).
43. Deng, W. et al. MOQEA/D: Multi-objective QEA With decomposition mechanism and excellent global search and its application. *IEEE Trans. Intell. Transp. Syst.* <https://doi.org/10.1109/tits.2024.3373510> (2024).
44. Vo, N., Tang, H. Y. & Lee, J. H. A multi-objective Grey Wolf-Cuckoo Search algorithm applied to spatial truss design optimization. *Appl. Soft Comput.* <https://doi.org/10.1016/j.asoc.2024.111435> (2024).
45. Li, G. S. & Zhou, T. A multi-objective particle swarm optimizer based on reference point for multimodal multi-objective optimization. *Eng. Appl. Artif. Intell.* <https://doi.org/10.1016/j.engappai.2021.104523> (2022).

46. Houssein, E. H., Saad, M. R., Ali, A. A. & Shaban, H. An efficient multi-objective gorilla troops optimizer for minimizing energy consumption of large-scale wireless sensor networks. *Expert Syst. Appl.* <https://doi.org/10.1016/j.eswa.2022.118827> (2023).
47. Arrieta, A. in *Genetic and evolutionary computation conference (GECCO)*, 1317–1326 (2022).
48. Qu, D., Xiao, H. L., Chen, H. F. & Li, H. Y. An improved differential evolution algorithm for multi-modal multiobjective optimization. *PeerJ Comput. Sci.* <https://doi.org/10.7717/peerj.cs.1839> (2024).
49. Mirjalili, S., Jangir, P., Mirjalili, S. Z., Saremi, S. & Trivedi, I. N. Optimization of problems with multiple objectives using the multi-verser optimization algorithm. *Knowl.-Based Syst.* **134**, 50–71. <https://doi.org/10.1016/j.knosys.2017.07.018> (2017).
50. Mirjalili, S. et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002> (2017).
51. Fu, S. et al. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems. *Artif. Intell. Rev.* **57**, 134. <https://doi.org/10.1007/s10462-024-10716-3> (2024).
52. Fu, S. et al. Improved dwarf mongoose optimization algorithm using novel nonlinear control and exploration strategies. *Expert Syst. Appl.* **233**, 120904. <https://doi.org/10.1016/j.eswa.2023.120904> (2023).
53. Derrac, J., García, S., Molina, D. & Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut. Comput.* **1**, 3–18. <https://doi.org/10.1016/j.swevo.2011.02.002> (2011).
54. Bu, K., Zhao, Y. & Zheng, X. Optimization design for foundation pit above metro tunnel based on NSGA2 genetic algorithm. *J. Railway Sci. Eng.* <https://doi.org/10.19713/j.cnki.43-1423/u.T20200287> (2020).

Acknowledgements

This work was supported by the Education and Teaching Reform Fund of Central University of Finance and Economics (No. 2022ZXJG21), Exploration of the School-wide Elective Curriculums for Cultivating Global Economic Governance Talents for International Organizations, the fundamental research fund for the central universities, project number: A24JBW700030.

Author contributions

All authors contributed to the presented study. Conceptualization was provided by Jinyan Shao (J.S). Formal analysis was performed by Yuan Lu (Y.L). Methodology was performed by Yi Sun (Y.S). Validation of results was performed by Lei Zhao (L.Z). The first draft of the manuscript was written by J.S and Y.L. The review and editing were provided by Y.S. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.L.

Reprints and permissions information is available at www.nature.com/reprints.

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025