scientific reports

Check for updates

OPEN

An improved artemisinin algorithm for task allocation in heterogeneous robot systems for chemical inspection

Chaofan Li[1,2], Qiang Liu[1,2]✉, Mingqiang Yin [1,2], Xianming Lang[1,2] & Guihua Bo[1,2]

The task allocation of multiple robot chemical inspection plays a vital role in enhancing the inspection efficiency of robots, and it is crucial for the timely detection of hazardous factors in chemical enterprises and the elimination of potential production risks. In this paper, an integer programming model of multiple heterogeneous inspection robot task assignment (MHRTA) problem is established based on the consideration of the constraints on the applicability of sensors carried by robots to inspection tasks. Given the NP-hard nature of the MHRTA problem, this study proposes a multiple strategy enhanced artemisinin algorithm to solve the proposed MHRTA model. Improving the position update strategy of artemisinin molecules in the comprehensive elimination phase of the artemisinin algorithm by incorporating concepts from the slime mold algorithm, and incorporates a nonlinear curve as the probability factor during the later consolidation phase, while introducing self-adaptive *t* distribution mutation to enhance the quality of solutions. Furthermore, Considering the discrete combinatorial optimization characteristics of the MHRTA problem, a two-layer encoding scheme is adopted to create a connection between the encoding space and the solution space, specifically for addressing multiple robot task allocation and inspection Hamiltonian routing, a variable neighborhood search strategy is embedded in the artemisinin optimizer to improve its computational efficiency. In eight test cases, the proposed method was evaluated against other algorithms and CPLEX solvers. The experimental results verified that the proposed method has strong optimization ability and stability in solving MHRTA problems.

Keywords Chemical plant inspection, Multiple heterogeneous robots, Task allocation, Hamiltonian routing, Artemisinin Optimization Algorithm

The intricate nature of chemical and petrochemical facilities necessitates substantial efforts to prevent the unintended release of hazardous substances and to safeguard personnel, workers, and the environment from accidents. Both designers and operators must pay greater attention to developing innovative solutions to ensure enhanced safety levels. Otherwise, failures or containment losses from process equipment could result in severe accidental scenarios[1]. The equipment of chemical enterprises operates in complex environments, which contain many potential risk points. Each risk point needs to be inspected at regular intervals to prevent major safety accidents from occurring[2]. The continuous advancement of artificial intelligence and robotic technologies has enabled inspection robots, through the integration of advanced sensors and algorithms, to replace manual detection of hazardous factors in chemical equipment such as gas leaks, liquid spillage, droplet formation, fluid leakage, and abnormal temperature fluctuations[3]. The diversity of hazardous factors in chemical enterprises necessitates robotic systems to be equipped with multiple types of sensing devices. In order to control the budget and manage the payload limitations of robots, chemical companies often deploy different types of robots and customize different sensor configurations according to specific inspection requirements. The different terrain environments in chemical plants, including flat surfaces, stairs, and pipelines, impose specific mobility requirements on robots to effectively navigate different operating areas. Therefore, large chemical plants often require heterogeneous robot systems such as wheeled robots, quadruped robots, Unmanned Aerial Vehicles, etc. for collaborative inspection.

[1]School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China. [2]Chaofan Li, Qiang Liu, Mingqiang Yin, Xianming Lang and Guihua Bo contributed equally to this work. ✉email: qiangliu@lnpu.edu.cn

Multi-heterogeneous robot chemical inspection task allocation is one of the key technologies for robot application in chemical inspection. The task allocation significantly affects the inspection efficiency of robots, and is crucial for timely inspection of hazardous factors in chemical enterprises and elimination of production risks and hidden dangers. Nevertheless, the issue of task assignment for multiple inspection robots is inherently an NP Hard problem. Considering the applicability of the terrain of the inspection task location to the robot's ability to pass through, as well as the constraints of different inspection tasks on the robot's ability to carry sensors, it brings new challenges to the design of related solving algorithms[4,5]. It is challenging for accurate algorithms to solve NP-hard problems within polynomial time. When tackling complex combinatorial optimization problems, Metaheuristic algorithms are characterized by their rapid convergence, low algorithmic complexity, and high solution accuracy. However, they also have the challenge of designing efficient encoding schemes to map encoding to solution space, and how to avoid falling into local optimal problems[6,7]. Therefore, it is crucial to study the task allocation of multi heterogeneous robot chemical inspection and design efficient metaheuristic algorithms to promote the improvement of robot inspection efficiency and timely eliminate chemical production risks and hidden dangers.

This paper studies an application scenario in which multiple heterogeneous robots perform inspection tasks in a chemical industrial park, and considers the constraints of the location and terrain of the inspection tasks on the robot's ability to pass through, as well as whether the sensors equipped on the robots are suitable for inspection tasks. In the chemical industrial park environment, each robot is fitted with a particular sensor designed to measure and record data from inspection tasks that are spread across various locations. There are heterogeneous robot systems such as wheeled mobile robots, quadruped robots, and drones in the chemical industrial park, and each robot carries different sensors for performing inspection tasks such as sound anomaly inspection, temperature anomaly inspection, and toxic gas inspection. Combining the comprehensive challenges of collaborative optimization of robot inspection task allocation and inspection Hamiltonian routing, the multiple heterogeneous inspection robot task allocation (MHRTA) problem has multidimensional complexity and is essentially a combinatorial optimization problem. When the quantities of robots and tasks rise, the number of solutions will grow exponentially, leading to the problem becoming NP-hard. This challenge considers the complexity of the application of multi heterogeneous robot systems in practical scenarios and proposes a solution for task allocation and inspection Hamiltonian routing collaborative optimization in heterogeneous robot systems.

In recent years, numerous scholars have introduced solution methods tailored to the multiple-robot task planning problem. Generally, these algorithms can be categorized into three groups: exact methods, market-based algorithms, and metaheuristics inspired by biological theories. Exact methods are widely employed to address combinatorial optimization problems. For example, Miloradović et al.[8] proposed a mathematical model for the extended colored traveling salesman problem and demonstrated its superiority compared to existing models. Yin et al.[9] introduced a two-stage stochastic programming framework to characterize the electric vehicle battery recycling problem, which was subsequently reformulated as an equivalent mixed-integer linear programming model. To mitigate the computational challenges associated with a large sample size, the authors proposed a sample-based decomposition algorithm that efficiently computes both the upper and lower bounds of the problem, ensuring high solution quality and computational efficiency. Wen et al.[10] designed a unified optimization model for representing multi-mission task allocation problems, enabling the application of various solvers to different problem instances. The exact methods works well for scenarios involving a limited number of robots, simple tasks. However, when there is an increase in the number of robots, added task complexity, and more constraints, exact methods face significant challenges due to higher computational demands and extended processing times. Market-based cooperative algorithms also show significant potential[11,12]. Quinton et al.[13] investigated the impact of communication degradation on market-driven task assignment in dynamic surveillance environments and introduced an innovative connectivity factor within the bidding evaluation formula. Bai et al.[14] developed a multi-level system to address the task scheduling challenge for multiple robots under temporal constraints, facilitating continuous refinement of operational strategies through inter-agent communication. The market mechanism relies on continuous communication among robots to maintain consistent bidding information. Additionally, since auctions typically involve straightforward task types that naturally match one robot to one task, traditional market-based approaches prove inadequate for assigning the more complex tasks encountered in chemical inspection. Metaheuristic algorithms, known for their strong adaptability and low computational complexity, are extensively applied in solving complex combinatorial optimization problems in multi-robot task planning[15]. For instance, Yan et al.[16] developed an intelligent ocean task allocation and path planning approach based on an enhanced genetic algorithm. Hua et al.[17] addressed the multi-task assignment challenge for large-scale UAVs by integrating Particle Swarm Optimization with evolutionary algorithms.

Due to the advantages of metaheuristic algorithm in convergence speed, algorithm complexity, and solution accuracy compared to exact algorithms and market methods in solving MHRTA problems, this paper chooses it as the algorithm for solving this problem. Various metaheuristic algorithms, including Differential Evolution (DE)[18], Chimpanzee Optimization Algorithm (ChOA)[19], Dung Beetle Optimizer (DBO)[20], Northern Goshawk Optimization (NGO)[21], Energy Valley Optimizer (EVO)[22], Dream Optimization Algorithm (DOA)[23], and Artemisinin Algorithm (AO)[24] has been effectively utilized in solving combinatorial optimization problems as well as diverse engineering optimization challenges. According to the theory of no free lunch, it is meaningful to continuously improve existing optimization algorithms to surpass current ones[25]. Hussien et al.[26] proposed a white whale optimization approach that combines transformation factor strategy and Gaussian local mutation improvement. Khalid et al.[27] developed an improved Emperor Penguin algorithm that integrates multiple parameter adaptation strategies with selection probability. AO is a novel metaheuristic algorithm that has recently attracted significant attention. Compared with traditional metaheuristic algorithms such as particle swarm optimization (PSO)[28] and genetic algorithm (GA)[29], the AO algorithm is characterized by its strong global

search capability, which allows it to efficiently explore the solution space and avoid local optima–a common challenge in complex optimization tasks like MHRTA. Additionally, the AO algorithm has fewer parameters and a simpler structure, reducing the need for extensive manual parameter tuning. The AO has achieved successful applications across diverse fields, including indoor positioning[30], prediction of interpersonal relationship sensitivity[31], and cost optimization of wind energy collection systems[32].In these fields, the AO algorithm has shown an excellent ability to balance global exploration and local search, which is crucial for achieving high-quality solutions in complex environments. Given these advantages, According to the theory of no free lunch[25]. We improved the traditional AO algorithm and designed corresponding encoding schemes to apply it to solve the MHRTA problem.

The primary contributions of this study can be summarized as follows:

(1) For the MHRTA problem, we have formulated an integer programming (IP) model that incorporates constraints on the operational capabilities of heterogeneous robots across diverse terrain environments, as well as constraints on the applicability of sensors mounted on the robots for different tasks. The correctness of the model was verified using the CPLEX solver.

(2) A two-layer encoding scheme has been adopted to establish a connection between the encoding space and the solution space, specifically addressing the challenges of multi-robot task allocation and inspection Hamiltonian routing. This approach effectively decouples the robot task allocation from the inspection Hamiltonian routing.

(3) An improved artemisinin algorithm (IAO), incorporating three enhanced strategies, has been developed to boost the optimization accuracy of the artemisinin optimizer. The effectiveness of the IAO algorithm was validated through comparisons with other algorithms using both benchmark test functions and CEC2014 test functions.

(4) Considering the NP-hard nature of the MHRTA problem, we propose an algorithm that integrates a variable neighborhood search strategy into the IAO algorithm (IAO-v) to enhance local search capability when solving the MHRTA problem.

The paper is structured as follows. Section 2 outlines a survey of existing research on multi-robot task allocation. Section 3 details the problem model under consideration, including the underlying assumptions, and establishes the corresponding Integer Programming (IP) formulation. Section 4 introduces the AO algorithm and the Improved Artemisinin Optimization (IAO) algorithm, which integrates three enhanced strategies. Section 5 presents a two-layer encoding scheme, explains its approach to handling constraints, and outlines the IAO-v algorithm along with the solution process for the MHRTA problem addressed in this study. Section 6 is dedicated to experimental analyses, featuring optimization comparisons using benchmark functions and CEC2014 test functions, as well as simulation experiments and comparative analyses on inspection case studies. Finally, Section 7 concludes the paper by summarizing the research contributions.

Related work

Many researchers have obtained significant results in the applications of multi-robot systems. To address the challenge of decoupling task allocation and path planning for multiple inspection robots, scholars have partitioned the process into two frameworks. First, inspection tasks are divided into several sub-areas using clustering algorithms or specific rules, after which tasks in each sub-area are assigned to individual robots. Subsequently, the operational task sequence for each robot within its sub-area is planned, followed by the generation of inspection routes for each robot. For instance, Zhu et al.[33] utilized the k-means algorithm to cluster inspection nodes and assigned the resulting clusters to robots for inspection; they also proposed a boundary deployment strategy to optimize the positioning of boundary nodes. In an innovative enhancement to the K-means++ clustering process, Yuan et al.[34] introduced a maximum cluster size constraint. This crucial modification ensures that the clustering results are well-matched with the actual carrying capacity of the robots. It effectively resolves the issue of task overload on a single robot to make multiple trips–thereby significantly improving the solution's practicality. In terms of specific rules, Huang et al.[35] developed a decoupled Traveling Salesman Problem (TSP)-based algorithm to manage task allocation and path planning for multiple unmanned aerial vehicles conducting pesticide spraying in densely forested hilly terrain, effectively decomposing the multi-UAV task allocation and path planning into several TSPs. Le et al.[36]introduced an algorithm that integrates routing and scheduling based on defined time-frames, aiming to address the limitations of traditional methods which treat these tasks separately in multi-AGV systems, which restrict flexibility in path selection and collision avoidance. This algorithm integrates routing and scheduling within a single framework, thereby eliminating the constraints associated with independent execution. Additionally, Guo et al.[37] introduced a multi-robot task scheduling method that employs a discrete encoding approach to decouple task allocation from path planning.

Researchers worldwide have developed a variety of optimization objectives and constraints that address the practical requirements of multi-robot industrial inspection, task allocation, and path planning, and establishing corresponding mathematical models. For instance, lu et al.[38] conceptualized agricultural plots as task nodes, incorporating workload attributes into each node to account for the limited workload capacity associated with each inspection task. They introduced a reinforcement learning-driven attention mechanism for policy optimization that aims to balance robot workload, ensuring they are neither overloaded nor underutilized; however, this network did not account for inter-robot collisions. In contrast, nguyen et al.[39] achieved improved estimation of travel times and sequential arrival times in multi-AGV systems by incorporating the robots load status and adaptive acceleration of arrival time sequences, thereby enabling a more precise prediction of inter-robot collisions. Zhang et al.[40] examined the problem of assigning tasks to multiple robots while considering both time window and precedence constraints. They proposed a graph-based batch processing framework (BMRTA), which effectively solves the multi-robot task allocation problem under the combined constraints of time windows and task precedence. In practical industrial applications, enhancing inspection efficiency, ensuring safety, and

balancing workloads often necessitates the simultaneous optimization of multiple objectives. Consequently, researchers in the multi-robot inspection domain have proposed multi-objective optimization models and algorithms. For instance, Majumder et al.[41] developed a novel multi-objective algorithm that optimizes both robot task completion time and fuel consumption to address challenges in path planning and task allocation, ultimately yielding more efficient and energy-saving multi-robot inspection systems. Wei et al.[42] reformulated the collaborative multi-robot task allocation challenge as a multi-objective variant of the multiple traveling salesman problem. Their approach seeks to reduce both the cumulative distance traveled by all robots and the maximum workload borne by any single robot. Similarly, Luo et al.[43] introduced a multi-objective optimization model for chemical safety inspections that incorporates robotic gas detection sensitivity, thereby reducing travel distances while enhancing the detection capabilities for hazardous gas leaks.

In large-scale, complex, and high-risk industrial environments, the necessity for multiple heterogeneous robots to achieve functional complementarity, collaborative perception, and parallel operation during inspection processes is increasingly evident. In response, researchers have begun to investigate scenarios in which heterogeneous robot systems perform inspection tasks and have developed corresponding algorithms. For example, Bai et al.[14] addressed tasks with temporal logic constraints by considering scenarios that involve cooperative tasks requiring the collaboration of two robots. Similarly, Ye et al.[44] devised a two-phase strategy that categorizes construction tasks into distinct types. Their task allocation method considered not only the capabilities of individual robots but also the aggregated potential of robot teams, thereby ensuring that each task group was assigned the most suitable team. Additionally, Chakraa et al.[45] established a mathematical model that accounts for the sensor limitations present in robots during inspection tasks . Chakra et al.[46] examined the collision issue in multi-robot systems involving heterogeneous robots and tasks, and they proposed a clustering-based task allocation strategy for addressing large-scale homogeneous tasks and isomorphic robot allocation challenges.

Table 1 summarizes and compares the characteristics of relevant papers in the literature, including the types of problems considered, the types of algorithms used. The full names of the relevant abbreviations are shown in Supplementary Table S1 in the Appendix. Figure 1 summarizes recent accomplishments in the field of multi-robot task allocation, highlighting the significant contributions of researchers over the past few years. Usually the focus of research is on homogeneous robots, and there is relatively little research on the collaborative optimization problem of task allocation and inspection Hamiltonian routing for multi heterogeneous inspection robots. In addition, the design form of operators is usually discrete, and the accuracy and convergence speed of solving algorithms still need to be improved.

Description of the MHRTA problem

Description and assumptions

The robot chemical inspection task allocation problem, which takes the robot inspection travel cost as the optimization objective and considers the constraints of terrain pass ability for robots at the inspection task locations and the applicability of sensors carried by robots for the inspection tasks, is represented as: In a chemical industrial park, there are m heterogeneous robots and n inspection tasks, where $m, n \in Q = \{1, 2, 3, \ldots\}$, and $m < n$. An oriented graph $G(T, E)$ is defined, where $T = \{0, 1, 2, \ldots, n\}$ is the set of inspection tasks and the robot warehouse, Here, 0 represents the robot warehouse. Each edge in the directed graph must satisfy $i, j \in E$ with $i \neq j$ The set E is associated with the weights d_{ij} where d_{ij} denotes the travel cost between two inspection tasks i and j, since each inspection can only be executed by a robot that satisfies the relevant conditions, the set of inspection tasks is divided into m non-empty subsets, The inspection task set T_k is used to represent the inspection tasks that each robot is capable of performing. where k warehouse the index of the robot, $\forall k \in Z = \{1, 2, 3 \ldots, n\}$. The robot's route matrix is warehouse as $X = [x_{i,j}]$, All robots need to

Author(s)	Application field	Types of robots	Algorithm type	Algorithm
Zhu. et al(33)	PPMR	Homogeneous	Metaheuristic	K-means+GWO
Yuan. et al(34)	MRTA	Homogeneous	Metaheuristic	K-means+PSA
Huang. et al(35)	MAWM	Homogeneous	Metaheuristic	GA
Le. et al(36)	VMP	Homogeneous	HHA	TFRS
Guo. et al(37)	MARTA	Homogeneous	Metaheuristic	CDABC
Lu. et al(38)	MRTA	Homogeneous	Reinforcement learning	Reinforcement learning
nguyen. et al(39)	MARTA	Homogeneous	HHA	A2CAT-VT2N
Zhang. et al(40)	MRTA	Homogeneous	Metaheuristic	BMRTA
Majumder. et al(41)	MRTA	Homogeneous	Metaheuristic	TLOA
Wei. et al(42)	MRTA	Homogeneous	Metaheuristic	PSO
Luo. et al(43)	CSIPO	Homogeneous	Metaheuristic	GOA
Bai. et al(14)	MHRTP	Heterogeneous	HHA	HCPF
Ye. et al(44)	MRCTA	Heterogeneous	Metaheuristic	GA
Chakraa. et al(45)	MHRTA	Hemogeneous	Metaheuristic	HFBS+2-Opt
Chakraa. et al(46)	MHRTA	Heterogeneous	Metaheuristic	GA

Table 1. Introduction to related work.

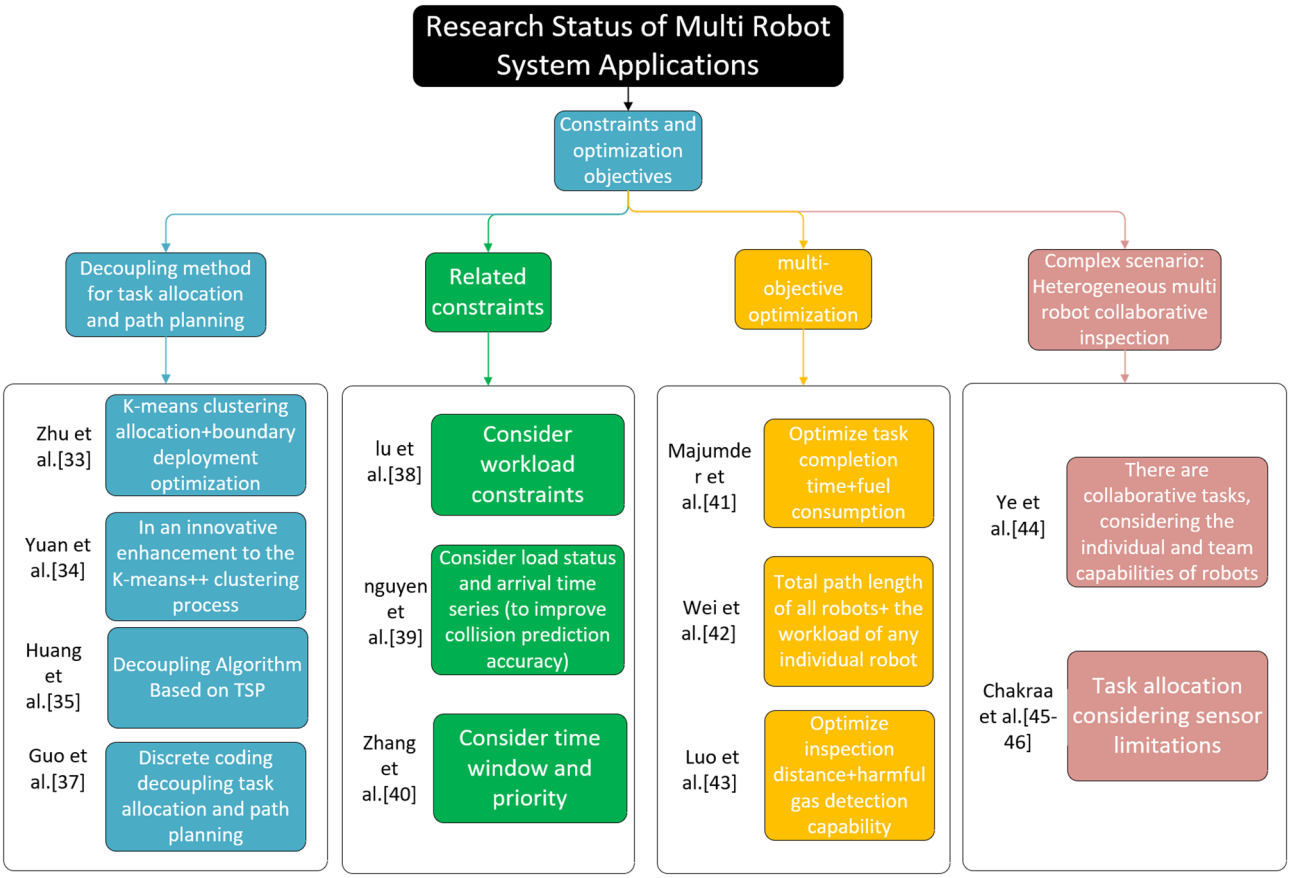


Fig. 1. Recent accomplishments in the field of multi-robot task allocation.

depart from the robot warehouse to carry out inspection tasks and eventually return to the robot warehouse. Each inspection task needs to be assigned to one robot for execution. Fig. 2 illustrates an example of a robot chemical inspection simulation environment. In the example, the chemical enterprise has two wheeled robots equipped with different sensors and a quadruped robot. The types of inspection tasks are divided into two categories: inspection of oil storage tanks and inspection of distillation towers. Among them, the inspection of distillation towers, which requires climbing stairs, can only be carried out by the quadruped robot. The objective is to allocate appropriate inspection tasks to each robot and to plan a Hamiltonian routing for the robots that begins and ends at the warehouse, with the goal of reducing the robots' travel cost. The study operates under the following assumptions:

(1) The battery level of the robot in this article is sufficient to complete one inspection task (i.e., the battery level is sufficient).

(2) The priority of each inspection task is the same, and there is no time window constraint for individual tasks.

Mathematical model

The symbols employed in this section are detailed in Appendix Supplemental Table S2. In this study, we present an integer linear programming model designed to minimize robot inspection travel costs while accounting for the constraints associated with terrain passability at various task locations and the suitability of sensors for different inspection tasks in chemical plants. The resulting model is formulated as follows:

$$\text{MIN}\left\{\sum_{i,j\in T} x_{ijk}\times d_{ij}\mid i\neq j, k\in\mathbb{Z}\right\} \tag{1}$$

Subject to constraints.

$$\sum_{i=1}^{n} x_{0ik}=\sum_{i=1}^{n} x_{i0k}=1\quad \forall k\in\mathbb{Z} \tag{2}$$

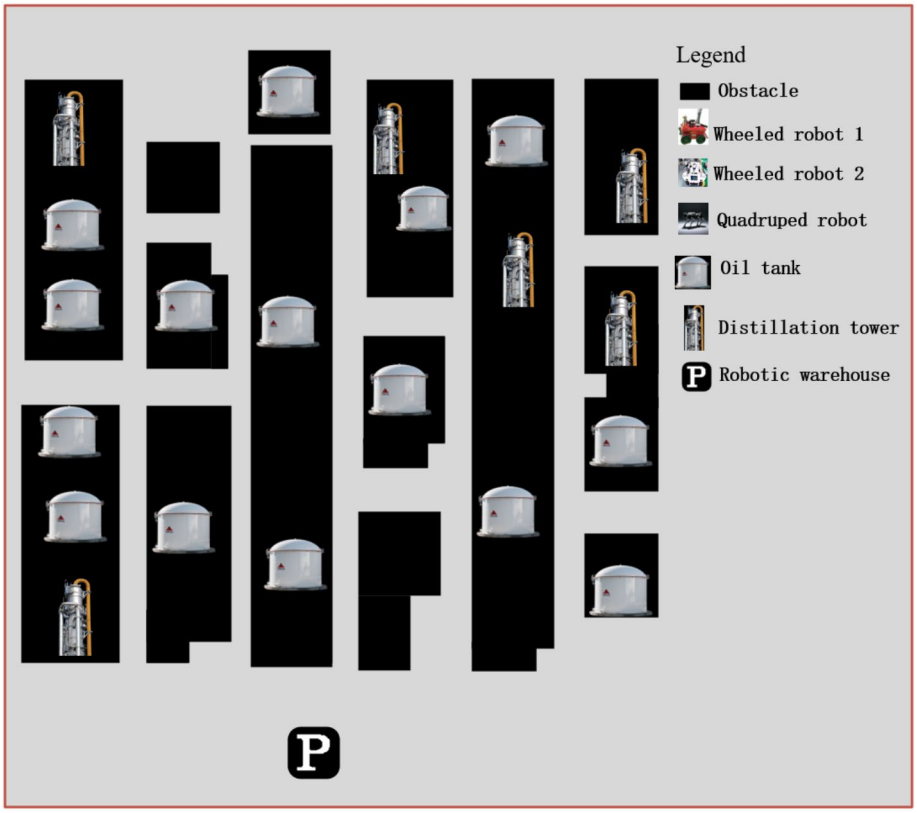


Fig. 2. Example of a robotic chemical plant inspection simulation environment.

$$\sum_{i}\sum_{j} x_{ijk} = 0 \qquad i \in T_k, j \in T \setminus (0 \cup T_k), k \in \mathbb{Z} \tag{3}$$

$$\sum_{i}\sum_{j} x_{ijk} = 0 \qquad i \in T \setminus (0 \cup T_k),\ j \in T_k,\ k \in \mathbb{Z} \tag{4}$$

$$\sum_{j=0}^{n}\sum_{k=1}^{m} x_{ijk} = 1 \qquad i \in T \setminus \{0\}, j \in T, i \neq j, k \in \mathbb{Z} \tag{5}$$

$$\sum_{j=0}^{n}\sum_{k=1}^{m} x_{ijk} = \sum_{j=0}^{n}\sum_{k=1}^{m} x_{jik} = 1 \qquad i \in T, k \in \mathbb{Z}, i \in T, i \neq j \tag{6}$$

$$u_{ik} - u_{jk} + (n-1) \cdot x_{ijk} \leq n \quad \forall i, j \in T_k, k \in \mathbb{Z} \tag{7}$$

The optimization objective in Eq. (1) seeks to minimize the travel cost incurred by the robots during inspection tasks. Constraint (2) ensures that each robot commences its route from the robot warehouse and returns there upon completing its assigned inspection task. Constraints (3) and (4) guarantee that the robots do not engage in inspection tasks beyond their capacity limits. Constraint (5) mandates that each inspection task, except for the robot warehouse, is performed only once. Constraint (6) requires that after a robot arrives at an inspection point and completes the corresponding task, it must depart from that point and proceed to the next designated inspection site. Finally, Constraint (7) prohibits the formation of sub-inspection routes, thereby ensuring that each robot's itinerary constitutes a single loop starting and ending at the robot warehouse.

Improved AO

Traditional AO

The Artemisinin Algorithm[24] is a novel metaheuristic optimization method inspired by the distribution and function of artemisinin in the human body. By emulating the natural mechanisms for detecting and combating the malaria pathogen, the algorithm seeks to determine the optimal solution to a given problem. In this framework, each candidate solution is conceptualized as the "position of an artemisinin molecule," and the search space is analogous to the environment occupied by the malaria pathogen. This process is mathematically represented by the following equations:

(1) Comprehensive Elimination Phase of AO: Simulating the initial stage of artemisinin molecules in malaria treatment, In this phase, a probability factor r_1 is introduced. When $r_1 < K$, it enters the comprehensive elimination phase. Eq.(8) represents the update method for the position of artemisinin molecules in this phase:

$$\begin{cases} a_{i,j}^{t+1} = a_{i,j}^{t} + c \cdot a_{i,j}^{t}(-1)^{t} & \text{if } r_1 < 0.5 \text{ and } r_2 < K \\ a_{i,j}^{t+1} = a_{i,j}^{t} + c \cdot a_{\text{best},j}^{t}(-1)^{t} & \text{if } r_1 < 0.5 \text{ and } r_2 < K \end{cases} \tag{8}$$

$$K = 1 - \frac{t^{1/6}}{t_{\max}^{1/6}} \tag{9}$$

$$c = 1 \cdot e^{-(4t/t_{\max})} \tag{10}$$

In the Eqs.(8)-(10), $a_{i,j}^{t}$ and $a_{i,j}^{t} + 1$ represent the positions of the artemisinin molecules before and after the update, respectively. $a_{best,j}^{t}$ denotes the current optimal position of the artemisinin molecule; t_{max}, t decibels represent the maximum and current iteration times. c represents the drug concentration factor. The random variables r_1 and r_2 follow a uniform distribution over the interval [0, 1].

(2) Local Elimination Phase: In the local elimination phase, the patient continues to receive a low dose of artemisinin medication to make sure that all malaria parasites are completely eliminated. Eq.(11) represents the update method for the position of artemisinin molecules in this phase:

$$a_{i,j}^{t+1} = a_{b3,j}^{t} + d \cdot (a_{b1,j}^{t} - a_{b2,j}^{t}), \quad \text{if } r_3 < f_{\text{norm}}(i) \tag{11}$$

$$f_{\text{norm}}(i) = \frac{[f(i) - \min(f)]}{[\max(f)] - \min(f)} \tag{12}$$

$$b_1, b_2, b_3 \sim U(1, N), \quad b_1 \neq b_2 \neq b_3 \tag{13}$$

In the Eqs.(11)-(13), $f_{norm}(i)$ represents the normalized fitness, min(f) indicates the minimum fitness, (f_i) represents the fitness value of the artemisinin molecule vector at the current iteration, max(f) denotes the maximum fitness. d is a random coefficient within the interval [0.1, 0.6], The random variable (r_3) follow a uniform distribution over the interval [0, 1].

(3) Later Consolidation Phase: In this phase, it is assumed that parasites still exist within the human body, and there is still a possibility of malaria relapse. In this phase, the method for updating the position of artemisinin molecules is represented by Eq.(14).

$$\begin{cases} a_{i,j}^{t+1} = a_{i,j}^{t} & \text{if } r_4 < 0.05 \\ a_{i,j}^{t+1} = a_{\text{best},j}^{t} & \text{if } r_5 < 0.2 \end{cases} \tag{14}$$

In which (r_4) and (r_5) are random numbers ranging between [0, 1].

Due to space limitations, a detailed introduction to the AO algorithm can be found in reference[24].

Improved strategy

This study proposes an Improved Artemisinin optimizer (IAO) that integrates three strategies. Improving the position update strategy of artemisinin molecules in the comprehensive elimination phase of the AO algorithm by incorporating concepts from the slime mold algorithm(SMA). and designing a nonlinear curve as the probability factor in the later consolidation phase to balance the AO's search capability. Additionally, self-adaptive t distribution mutation is incorporated to decrease the likelihood of the AO getting stuck in local optima.

The SMA's idea for updating positions

Because comprehensive clearing phase of the AO algorithm relies on random symbols t for updating the positions of artemisinin molecules, this perturbation method lacks directionality and may lead to low search efficiency. And the update step size is completely controlled by the attenuation factor c. In the later stage of iteration, if the disturbance is too small, it is easy to fall into local optima (see the introduction of AO algorithm for specific formulas). In 2020, Li et al.[47] proposed the slime mold algorithm by simulating the behavior changes of slime molds in nature. This new meta heuristic algorithm has the advantages of strong robustness, fast optimization speed, and simple parameter tuning. In the slime mold algorithm, the weight W is inversely proportional to the fitness t, achieving elite guided search and allowing the algorithm to search towards the elite region. The oscillation amplitude t is randomly selected within the range of -a to a (a variable related to the number of iterations), which can enhance the ability of artemisinin molecules to expand into new search regions. We solve the problem of low search efficiency in the comprehensive clearance stage of the AO algorithm by introducing adaptive weights and oscillation behavior of the slime mold algorithm. The improved position update formula for the comprehensive elimination phase of AO, incorporating concepts from the slime mould algorithm, is presented as follows:

$$a_{i,j}^{t+1} = \begin{cases} X_b(t) + V_b \cdot (W \cdot X_A(t)) - X_B(t) & \text{if } r < p, r_6 < K \\ V_C \cdot X(t) & \text{if } r \geq p, r_6 < K \end{cases} \tag{15}$$

$$K = 1 - \left(\frac{t^{1/6}}{T_{\max}}\right)^{1/6} \tag{16}$$

$$a = \arctan\left(-\left(\frac{t}{T_{\max}}\right) + 1\right) \tag{17}$$

$$p = \tanh|S(i) - DF| \tag{18}$$

In the Eqs.(15)-(18), Eq.(15) describes the slime mould's foraging process, which represents the position update strategy in this phase; $a_{i,j}^{t} + 1$ denotes the newly generated position of the slime mould ; The parameter v_b takes random values within the interval [-a, a]. The random variables r_6 follow a uniform distribution over the interval [0,1]; parameter V_c decreases linearly during the foraging process; X_b represents the individual position where the odor concentration is the highest. t denotes the current iteration count; X indicates the current position; X_A and X_B represent any two arbitrary positions of the slime mould; W indicates the current mass of the slime mould individual; T_{max} denotes the maximum iteration number; Eq.(18), S_i denotes the fitness value of the slime mould, where i signifies the population index, while D_F corresponds to the optimal fitness value. Moreover, the mathematical expression for the slime mould's mass parameter W under varying food concentration conditions is formulated as follows:

$$W(S_r) = \begin{cases} 1 + r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right) & \text{if } i \le N/2 \\ 1 - r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right) & \text{else} \end{cases} \tag{19}$$

$$s_r = \text{sort}(S) \tag{20}$$

In the Eqs.(19)-(20), $N/2$ represents half of the total number of slime molds; b_F and w_F are used to represent the best and worst fitness, respectively; s_r represents the sorted fitness sequence. *sort* denotes the ascending order of the swarm's fitness.

Design a nonlinear probability factor

In the later consolidation stage, the original AO algorithm used fixed probabilities (0.05 and 0.2) to determine whether to update the position of artemisinin molecules. 0.05 and 0.2 are empirical values that cannot adapt to the characteristics of different problems, lack adaptability, and limit the search ability of the AO algorithm. By simulating the phenomenon of decreasing recurrence probability over time in malaria treatment, the AO algorithm has a relatively high mutation probability in the early stages of iteration, ensuring that the algorithm has a certain exploration ability in the early stages. A nonlinear curve D is designed as the probability factor, due to the non-linear probability factor of the design decreasing with the increase of iteration times, the algorithm also has a certain local search ability in the later stages of iteration. Eqs.(21) and (22) represent the movement methods of artemisinin molecules in the later consolidation phase of the improved AO algorithm.

$$D = a \times \sqrt{\left(1 - \frac{t}{t_{\max}}\right)^{\frac{1}{10}}} \tag{21}$$

$$a_{i,j}^{t+1} = a_{\text{best},j}^{t}, \text{ if rand} < D \tag{22}$$

In the Eqs.(21)-(22), D represents the probability factor determining whether to execute the consolidation phase. $a_{i,j}^{t} + 1$ denotes the updated position of the artemisinin molecule, $a_{best,j}^{t}$ is the current best position of the artemisinin molecule, and $a \in [0, 1]$ serves as the coefficient of the probability factor, which controls both the decay magnitude and initial probability. The nonlinear convergence curves with coefficients a=0.2, 0.8, and 0.5 are shown in Fig. 3.

Introducing a self-adaptive t distribution

In the later stage of AO algorithm iteration, artemisinin molecules quickly gather near the current optimal position, greatly reducing the search ability of the population and making it easy to fall into the local optimal solution. Many scholars have chosen to introduce mutation operators for perturbation. Among the many mutation operators, t-distribution is widely used due to its advantages of Gaussian mutation and Cauchy mutation. Although AO itself performs well in global exploration and convergence speed, its local development ability needs to be strengthened when dealing with complex multimodal problems. For the t-distribution, when $t(m \to \infty) \to N(0,1)$, $N(0,1)$ is a Gaussian distribution, and $t(m = 1) = C(0,1)$, $C(0,1)$ is a Cauchy distribution. The relationship between the t-value and the Gaussian and Cauchy distributions is shown in Fig. 3. Therefore, adaptive t-distribution mutation can dynamically adjust the intensity of mutation according to the iteration process during algorithm operation. Early iteration mutation, similar to Cauchy mutation, exhibits heavy tailed characteristics, and its random sampling values are relatively easier to obtain larger values, enhancing the ability to global exploration and escape from local optima. During the iteration process, the algorithm's mutation strategy tends towards a transitional form that balances exploration and exploitation capabilities. As the number of iterations increases, the degrees of freedom of the t-distribution increase. Similar to the Gaussian distribution, its random sampling values are then more likely to obtain smaller values. Therefore, introducing an adaptive t-distribution enhances the local exploration capability of the AO algorithm. The t-distribution[48] has the following probability density function:

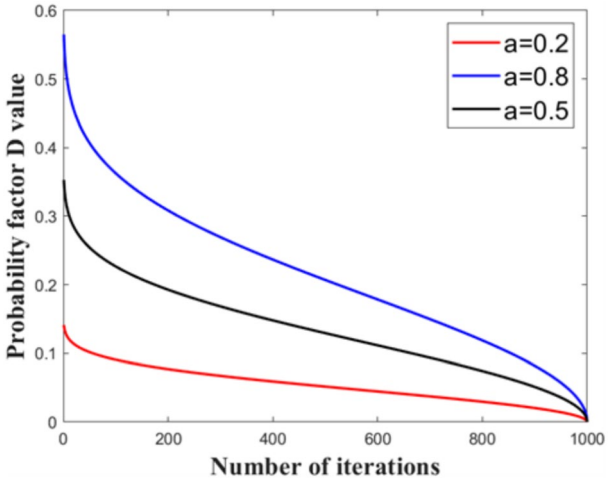


Fig. 3. The curve of the probability factor *D* varying with the number of iterations.

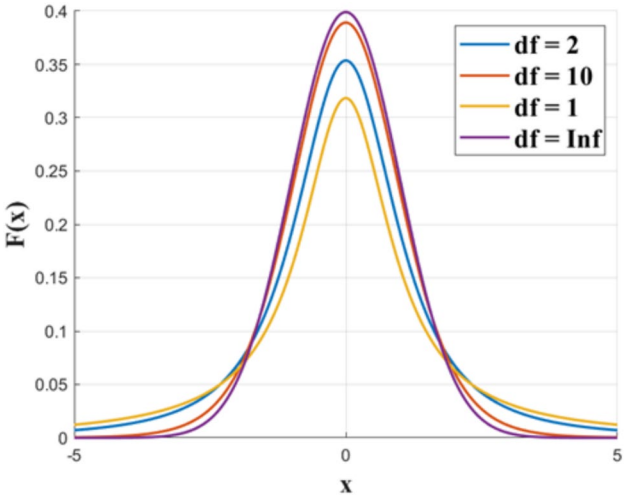


Fig. 4. The t-distribution, Gaussian distribution and Cauchy distribution curves.

$$p_t(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\cdot\Gamma\left(\frac{m}{2}\right)}\cdot\left(1+\frac{x^2}{n}\right)^{-\frac{n+1}{2}}, -\infty < x < \infty \tag{23}$$

In the Eq.(23), $\Gamma\left(\frac{n+1}{2}\right) = \int_0^{+\infty} x^{\left(\frac{n+1}{2}-1\right)} e^{-x}\, dx$ signifies the second kind of the Euler integral.

The number of iterations plays a role in tuning the degrees of freedom in the t-distribution, which helps to balance local search and global exploration. For the t-distribution, when $t(m \to \infty) \to N(0,1), \quad N(0,1)$ is the Gaussian distribution, $t(m=1) = C(0,1)$, $C(0,1)$ is the Cauchy distribution. The distribution of the Cauchy distributions and both the t and Gaussian distribution is shown in Fig. 4.

An self-adaptive *t* distribution-based mutation strategy is applied to individually perturb the positions of each Artemisinin molecule, with the specific formulations detailed as follows:

$$a_i^{t+1} = a_i^t + a_i^t \cdot t((\text{iter}) \tag{24}$$

In the Eq.(24), $a_i^t + 1$ represents the perturbed artemisinin molecule position, a_i^t the original position, and *t*(*iter*) is a parameter of the same size as the number of iterations.

While the self-adaptive *t* distribution perturbs the current position of the artemisinin molecule, it does not ensure that the fitness value of the perturbed position is superior to that of the original position. Consequently, a greedy strategy is incorporated, with the selection formula given as follows:

$$newa_i^{t+1} = \begin{cases} a_i^{t+1} & \text{if } f(a_i^{t+1}) < f(a_i^t) \\ a_i^t & \text{if } f(a_i^{t+1}) \geq f(a_i^t) \end{cases} \tag{25}$$

In the Eq.(25), $f(a_i^t)$ denotes the original fitness, $f(a_i^{t+1})$ represents the fitness of the perturbed molecule, and $newa_i^{t+1}$ indicates the new position information of the artemisinin molecule following the greedy selection.

The pseudocode of the improved Artemis Optimization (IAO) algorithm, which integrates three strategies, is shown in Algorithm 1.

Require: Establish an objective function f, the Population size N, agent population $A_{N,D}$ Max fitness evaluation $MaxF$ and evaluate their fitness f_{norm}. Dimension D.
Ensure: Current optimal X_b, the fitness of the optimal A_{best}.

```
Find the A_best
for each iteration f = 1 : MaxF do
    Calculate the K using Eq. (9)
    for each agent i = 1 : N do
        for For each dimension j = 1 : D do
            if rand < K then
                Update the a_{i,j} using Eq. (15)
            end if
            Update the a_{i,j} using Eq. (11)
        end for
        Update the a_{i,j} using Eq. (22)
    end for
    Calculate the f_norm
    Update the A_{N,D} and find the X_b.
    Self-adaptive t distribution perturbation by Eq. (24)
    Calculate the f_norm
    Update the A_{N,D} and find the X_b by Eq. (25)
end for
Output the optimal results of objective function: X_b and A_best, which represent decision variables and fitness, respectively.
```

Algorithm 1. IAO algorithm.

The IAO-v algorithm for solving the MHRTA problem

The AO algorithm is only suitable for continuous optimization problems, whereas the MHRTA is a discrete combinatorial optimization problem. Therefore, setting up an appropriate encoding scheme to discretize the artemisinin molecules of the AO algorithm is the primary issue. Considering the applicability of heterogeneous robots to perform different inspection tasks, a two-level encoding scheme is adopted to establish a mapping between the encoding and the solution space. To confront the NP-hard challenge of the cooperative optimization problem for task allocation and Hamiltonian routing-based inspection in heterogeneous multi-robot systems, a variable neighborhood search strategy[49] is integrated into the IAO algorithm to boost its local search capability in the later stages when solving the MHRTA problem.

Encoding scheme

Reference[50] introduces a transformation mechanism that maps continuous algorithms onto the solution space of the discrete flexible workshop scheduling problem to assign workpieces to machines. In this study, we employ this conversion mechanism to establish a mapping between the encoding scheme and the inspection tasks allocated to the robot. Additionally, the LPV (Large Position Value) rule[51] is applied to project the encoding into the solution space of the robot inspection Hamiltonian circuit. The overall problem is divided into two subproblems: (1) assigning pending inspection tasks to robots and (2) determining the sequence in which these tasks are executed, thereby establishing a Hamiltonian routing for robotic inspection paths. Each artemisinin molecule is represented using a randomized two-level encoding scheme with both levels of equal length. The encoding format is detailed in Equation (26).

$$X = \begin{Bmatrix} x_{11} & x_{12} \cdots x_{1l} \\ x_{21} & x_{22} \cdots x_{2l} \end{Bmatrix} \tag{26}$$

In which, each element in X takes any value within the range $[\varepsilon, \varepsilon]$. The value of ε can be any appropriate integer, l represents the number of inspection tasks in the chemical plant. Assuming there are 12 inspection tasks in the chemical park, the length of the vector of the artemisinin molecule is 12, and Assuming that the value of ε is 10, then each element in X takes any value within the interval [10,10] and is stored in a certain order.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
Allocated robot	2.1	2.2	-1.2	2.8	0.5	-1.3	2.0	1.6	5.8	-6.4	-3.1	-2.1
Sequence sorting	-0.1	1.1	-0.5	0.3	1.5	-0.9	-1.2	2.1	0.5	-1.3	2.2	1.6

Fig. 5. An example of the two-layer encoding scheme.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
Applicable robot	[1,2,3]	[1,2]	[1]	[1,2,3]	[1]	[1,2,3]	[1,2,3]	[1,2,3]	[1,2,3]	[1,2,3]	[1,2]	[1,2,3]

An example of the candidate robot set for an inspection task

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
First-level encoding	-9.2	6.3	3.8	2.8	8.2	-5.1	9.6	1.6	5.8	-6.4	2.5	-2.1
Assigned robot	1	2	1	2	1	1	3	2	3	1	2	2

Fig. 6. The decoding example of the first-layer encoding.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
second-level encoding	2.1	2.2	-1.2	2.8	0.5	-1.3	2.0	1.6	5.8	-6.4	-3.1	-2.1
Sequence sorting	9	10	5	11	6	4	8	7	12	1	2	3

Fig. 7. The decoding example of the second-layer encoding.

Fig. 5 demonstrates an example of the encoding scheme. In which, A_i A denotes the pending inspection task numbered *i*. Specifically, the first layer of the artemisinin molecule vector *X* indicates which robot is assigned to each inspection task, while the second layer specifies the order in which the assigned robot performs the tasks.

According to Equation (27), the first layer of the artemisinin molecular vector is transformed into an index representing the available robots for the inspection task, thereby determining the identification numbers of the robots assigned to execute the task.

$$z_i = floor(\frac{x_{1i} + \varepsilon}{2\varepsilon}(s_i)) + 1 \tag{27}$$

In the Eq.(27), *floor* represents the down rounding operation, i denotes the number of the inspection task, ε can be any appropriate integer value, S_i represents the number of robots suitable for inspection task *i*. $z_i \in [1, s_i]$ indicates the index of the robot assigned to inspection task *i* from the set of available robots for that task. In other words, the robot at position z_i in the robot set is designated to perform inspection task *i*. Fig. 6 illustrates an example of decoding the first-layer encoding.

We employ the Large Position Value (LPV) rule[51] to map the encoding into the solution space corresponding to the Hamiltonian loop for robot inspection. Based on Eq. (27), which assigns each inspection task to a specific robot, the second layer of the encoding–representing the artemisinin molecule–is sorted in ascending order by size. This ordering reflects the sequence in which the robot executes the inspection tasks, thereby establishing the Hamiltonian routing for the robotic inspection paths. Assuming all inspection tasks are allocated to a single robot, Fig. 7 illustrates the decoding of the second encoding layer. Algorithm 2 presents the pseudo-code for decoding using the two-layer encoding approach.

Require: The number of inspection tasks in the chemical park n, The number of robots m, The number of robots that can be selected for the inspection task with task number i is S_i, Initialize parameter ε, Save the two-dimensional matrix z of robot-task applicability.

Ensure: Robots assigned for each task b, Each robot executes the assigned task sequence E.

Encode according to Eq. (26)
******Decoding of the first layer encoding to obtain the assigned robots for each inspection task.*******
for Each robot $a = 1 : n$ **do**
  b_a = Calculate the robots assigned for each inspection task Eq. (27)
end for
******Decoding of the second layer encoding to obtain the order in which each inspection task is executed.*******
for Each robot $a = 1 : m$ **do**
  E_a = find $(b == a)$. % Select the indices of elements in vector b that are equal to a.
  E_a = sort (E_a). % Sort the vector E in ascending order.
end for
Output Robots assigned for each task b, and Each robot executes the assigned task sequence E.

Algorithm 2. Two layer encoding algorithm.

A description of constraint handling

In Section 3 of this paper, dedicated to the MHRTA problem formulation, we provide a detailed description of the optimization objectives and constraints. The following exposition comprehensively explains how these constraints are addressed during the decoding process.

1) Constraint Handling for Eq.(2): Constraint Eq.(2) ensures that the beginning point of each robot is the warehouse and returns to the warehouse after completing the inspection task. A detailed example of constraint handling is illustrated in Fig. 8. Assuming that inspection tasks $[A_1, A_2, \ldots, A_{A_{14}}]$ are assigned to robot number 1, the order in which robot number 1 performs the inspection tasks can be obtained by decoding according to the encoding scheme. To enforce the constraint that each robot departs and returns to the robot warehouse after completing all assigned inspection tasks, the visit order of the robot warehouse (A_0) is set to the first and last positions, while the execution order of other tasks is incrementally assigned sequential indices starting from 2. This ensures that the solution meets the specified constraints.

2) Handling of constraints Eq.(3) and Eq.(4): Constraints (3) and (4) ensure that robots do not perform tasks beyond their capability limits, According to Eq.(24), s_i represents the number of robots available for inspection task i, and j indicates the index of the robot assigned to inspection task i within the set of available robots for that task, and $z_i \in [1, s_i]$ indicates the index of the robot assigned to inspection task i within the set of available robots for that task. Therefore, the robot assigned to each inspection task is selected from the set of robots capable of performing the task, thus avoiding the generation of infeasible solutions under this constraint.

3) Handling of constraints Eqs.(5)-(7): Eq.(5) ensures that each inspection task, except for the warehouse, is performed only once, and guarantees that after completing an inspection task, the robot must leave and proceed to the next inspection point. Eq.(6) ensures that after each robot arrives at an inspection point and completes the task, it must leave that point and move to the next inspection point. Eq.(7) prohibits the formation of sub-inspection routing, ensuring that the path of each robot forms a unique routing starting from and returning to the warehouse. Fig. 9 further elaborates on how the encoding scheme handles the constraints specified in Eqs. (5)-(7). As shown in Fig. 9, 12 inspection tasks are allocated to 3 robots. Since each inspection task is uniquely assigned to a robot via the first-layer encoding, and the second-layer encoding ensures a unique execution order for each robot's assigned tasks during decoding, the inspection route of each robot forms a Hamiltonian routing

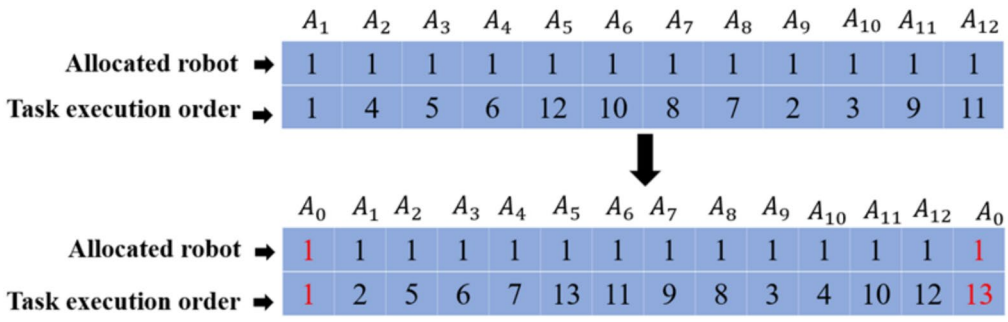


Fig. 8. Equation (2) Constraint handling example.

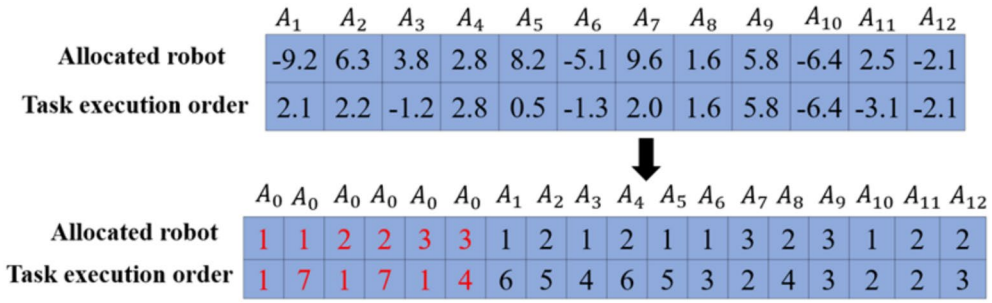


Fig. 9. Overall Encoding Scheme Decoding.

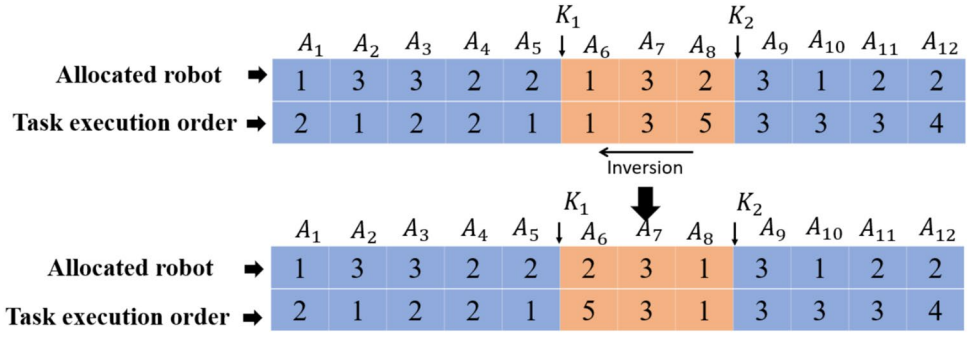


Fig. 10. Example of the inversion process.

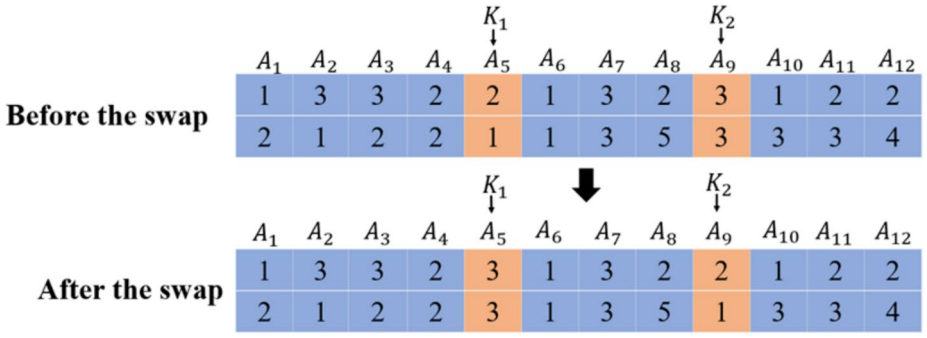


Fig. 11. Example of the inversion process.

that starts from the robot warehouse, visits all designated inspection points, and finally returns to the warehouse. Thus, the constraints in Eqs.(5)-(7) are satisfied.

Introducing variable neighborhood search

The IAO algorithm introduced in this study incorporates several parameters during its search process, including the search step size and the drug concentration attenuation index. These parameters affect the algorithm's performance, and their optimal values vary with the scale of the solution space. Consequently, there is an increased risk of the algorithm becoming trapped in a local optimum when addressing the MHRTA problem. To mitigate this issue, we propose an enhanced version of the IAO algorithm, termed IAO-V, which integrates a variable neighborhood search strategy. In each iteration of the IAO algorithm, elite individuals–constituting the top 10% by fitness value–are selected, and a variable neighborhood search is executed on these elite individuals to refine the solutions. Three distinct neighborhood structures are applied to the decoded vector.

1) Neighborhood Structure N_1: Inversion Operation, for the parent individual, randomly generate two distinct integers k_1 and k_2 within the range $[1, A_n]$ as inversion points, where A_n denotes the total number of inspection tasks. An illustration of this inversion operation is provided in Fig. 10.

2) Neighborhood Structure N_2: Swap Operation, for the parent individual, randomly generate two distinct integersk_1 and k_2 within the range $[1, A_n]$ as swap points, The elements at these two swap points are then exchanged. Fig.11 illustrates an example workflow of this operation.

3) Neighborhood Structure N_3: Mutation operation. For the parent individual, randomly generate an integer k_1 within the range $[1, A_n]$, The element in the first-layer encoding is then randomly reassigned to a feasible solution, i.e., altering the robot assigned to task k_1. Fig. 12 shows an example of the mutation operation.

The Variable Neighborhood Search is embedded into the IAO algorithm, as detailed in Algorithm 3. Assuming the population size of IAO is *N*, the dimension of the algorithm, i.e. the number of tasks, is *n*, and the number of robots is *m*, then according to Algorithm 2, the time complexity for calculating the population fitness value can be obtained as $T_1 = O(N(m+n))$. Assuming the maximum iteration number is *MaxF*, the time complexity of the main loop of IAO algorithm is $T_2 = O(MaxF \cdot N \cdot (m+n))$, so the time complexity of IAO can be obtained as $T = T_1 + T_2 = O(MaxF \cdot N \cdot (m+n))$. It can be seen that the IAO algorithm has the same time complexity as traditional AO.

Due to the fact that the variable neighborhood search extracts the top 10% of elite solutions from the IAO algorithm, its population size is $N \cdot 10\%$. The time complexity of the variable neighborhood search is $T_3 = O(N \cdot 10\% \cdot \lambda_{\max} \cdot q_{max} \cdot (m+n))$. Since q_{max} and $\lambda_{\max}$ are generally small constants of 10 and 3, the time complexity of the variable neighborhood strategy is $T3 = O(N \cdot (m+n))$. Since IAO-v embeds a variable neighborhood search strategy in the IAO algorithm, the time complexity of IAO-v is $T^{'} = T \cdot T_3 = O(MaxF(N^2 + n^2 + m^2) = O(N^2) + n^2 + m^2)$.

Require: Parameters initializing: Elite solutions ranked in the top 10% of IAO algorithm X, Maximum number of perturbations $q_{\max}$, Number of neighborhood structures $\lambda_{\max}$,The number of elite individuals P.
Ensure: The optimal results of objective function: X_b and F_b.
Main Loop
for Each iteration $i = 1$ **:** P **do**
  for Each perturbation $j = 1$ **:** $q_{\max}$ **do**
    Set the perturbation selection variable λ to 1.
    while Current disturbance variable $\lambda < \lambda_{\max}$ **do**
      Apply neighborhood structure N_λ: $X' = N_\lambda(X)$.
      if Fitness improved after N_λ **then**
        Update population: $X = X'$, Update fitness f_x.
      end if
      Otherwise Break.
      $\lambda = \lambda + 1$.
    end while
  end for
  Update the F_b and find the X_b.
end for
Output the optimal results of objective function: X_b and F_b, which represent decision variables and fitness, respectively.

Algorithm 3. Variable neighborhood search algorithm.

The process of solving the MHRTA problem using the IAO-v algorithm

Figure 13 illustrates the process of addressing the multi-robot chemical inspection task allocation and Hamiltonian routing optimization problem using the IAO-v algorithm. Following the method architecture proposed in[45], the solution process is divided into three phases: the preprocessing phase, the IAO-v algorithm phase, and the return phase. In the preprocessing phase, operational data from the chemical plant's robotic inspection environment is utilized as input. This includes the quantities of various robot types and inspection

Fig. 12. Example of the Mutation process.

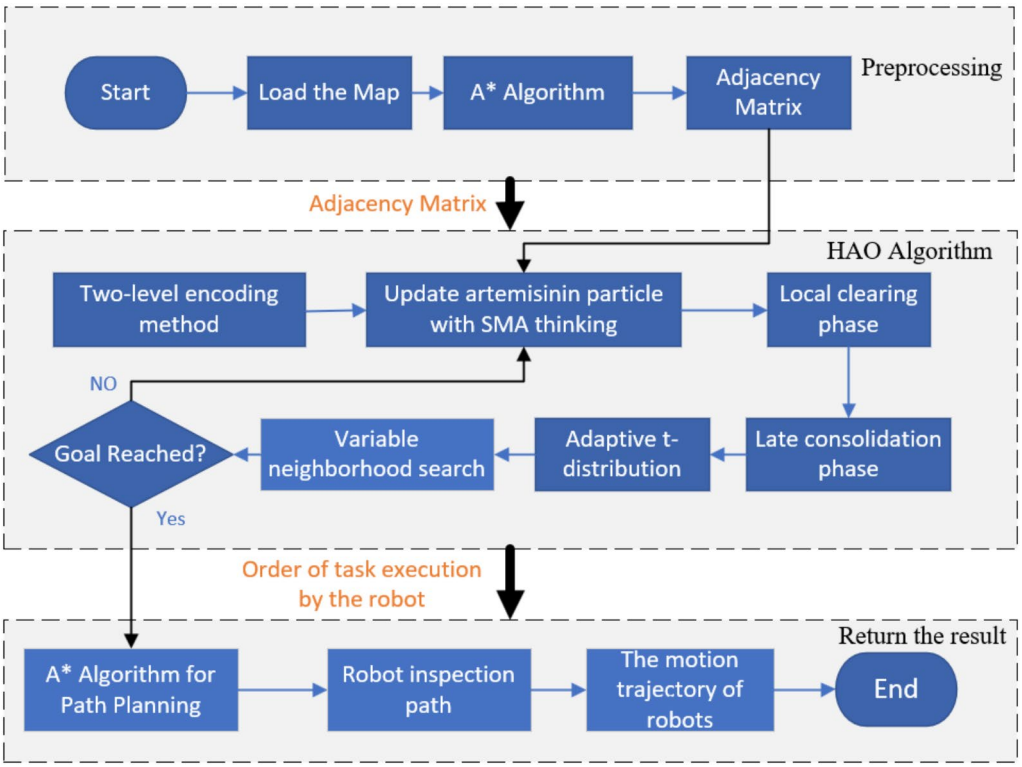


Fig. 13. Overall Flowchart of the Algorithm.

tasks, as well as the locations of charging stations and task points. The facility map is loaded and represented as a grid graph, and the A* algorithm[52] is employed to compute the shortest path costs for all inspection task pairs, with the results organized into an adjacency matrix. In the IAO-v algorithm phase, the computed adjacency matrix, together with information regarding the numbers of different robots and tasks, is used as input with the objective of minimizing the total travel cost for the robots. In this phase, the IAO-v algorithm is applied to enhance the fitness value. Finally, in the return phase, the task allocation and Hamiltonian routing solution derived from the improved AO algorithm, along with the path planning results from the A* algorithm, are integrated to compute the movement trajectory for each robot.

Experimental analysis

Numerical example experimental comparison

Experimental setup

Experimental Setup: The population size N was established at 30, and the maximum number of iterations Max_iter was set to 1000. All experiments were executed on the MATLAB R202a platform using a computer equipped with an Intel Core i7-13700H CPU operating at 2.4 GHz and 16.0 GB of RAM. Each algorithm was independently executed 30 times under identical environmental conditions.

Performance comparison of different improvement strategies

To evaluate the effect of different strategies on the Artemisinin optimizer, the AO algorithm, IAO algorithm, AO algorithm that introduced slime mold algorithm to change the method of updating artemisinin molecules (SAO), the AO algorithm with non-linear probability factor (NAO), and the AO algorithm with self-adaptive t distribution (AAO) are compared on the CEC2014 test functions[53] for optimization performance analysis. In particular, the probability factor coefficient a is set to 0.2 for both IAO and NAO. The selected benchmark functions are listed in Table Table 2, where CEC01 and CEC02 are unimodal functions, CEC06 and CEC10 are multimodal functions, CEC17 and CEC18 are hybrid functions, and CEC27 and CEC30 are composition functions, all with a dimension $d = 30$.

Table 3 demonstrates that, among the eight benchmark functions, the SAO algorithm yields results that are closer to the theoretical optimum than those produced by the AO algorithm. With the exception of the unimodal function CEC01, the NAO algorithm exhibits superior optimization capability compared with the AO algorithm across all tested functions. Similarly, the AAO algorithm outperforms the AO algorithm on all test functions, except for the multimodal function CEC10. Notably, the IAO algorithm consistently provides enhanced optimization performance relative to the AO algorithm across all eight benchmark functions, thereby indicating the effectiveness of the integrated multi-strategy enhancements applied to the AO algorithm. The convergence curves of AO algorithm improved by different strategies are shown in Fig. 14.

Number	Dimensions	Characteristics	Domain
CEC01	30	Unimodal	[-100,100]
CEC02	30	Unimodal	[-100,100]
CEC06	30	Multimodal	[-100,100]
CEC10	30	Multimodal	[-100,100]
CEC17	30	Hybrid	[-100,100]
CEC18	30	Hybrid	[-100,100]
CEC27	30	Compositional	[-100,100]
CEC30	30	Compositional	[-100,100]

Table 2. Introduction to selected CEC2014 functions.

Function	Statistical Results	AO	SAO	NAO	AAO	IAO
CEC01	Mean	1.46E+07	**7.10E+06**	1.61E+07	1.23E+07	1.16E+07
	Std	9.56E+06	4.89E+06	7.79E+06	6.03E+06	5.59E+06
	Rank	4	1	5	2	3
CEC02	Mean	2.04E+06	1.35E+04	2.72E+04	3.74E+05	**1.11E+04**
	Std	9.44E+05	9.72E+03	1.41E+04	2.93E+05	7.42E+03
	Rank	5	2	3	4	1
CEC06	Mean	6.08E+02	6.07E+02	**6.01E+02**	6.05E+02	**6.01E+02**
	Std	2.74E+00	2.20E+00	6.44E-01	2.75E+00	1.07E+00
	Rank	5	4	1	3	2
CEC10	Mean	1.35E+03	**1.22E+03**	2.34E+03	1.43E+03	1.24E+03
	Std	1.55E+02	1.15E+02	5.20E+02	1.56E+02	1.85E+02
	Rank	4	1	5	3	2
CEC17	Mean	3.32E+06	1.14E+06	**9.37E+05**	1.03E+06	8.97E+05
	Std	2.85E+06	6.87E+05	6.35E+05	6.55E+05	9.10E+05
	Rank	5	4	2	3	1
CEC18	Mean	1.67E+06	7.41E+03	5.87E+03	5.37E+04	**4.40E+03**
	Std	1.34E+06	5.32E+03	5.72E+04	4.38E+03	3.50E+03
	Rank	5	3	2	4	1
CEC27	Mean	3.28E+03	3.24E+03	3.08E+03	**2.90E+03**	**2.90E+03**
	Std	1.09E+02	9.26E+01	7.82E+01	0.00E+00	0.00E+00
	Rank	5	4	3	1	1
CEC30	Mean	1.38E+04	7.81E+03	1.07E+04	**3.92E+03**	4.46E+03
	Std	6.02E+03	2.24E+03	5.73E+03	1.95E+03	1.71E+03
	Rank	5	3	4	1	2

Table 3. Performance comparison of improvement strategies.

Comparison of the IAO algorithm with other algorithms

To further evaluate the optimization capability of the IAO algorithm, we selected IAO for comparison with the AO Algorithm , Rime Optimization Algorithm (RIME)[54], Grey Wolf Optimization Algorithm (GWO)[55], and Enterprise Development Algorithm (ED)[56]. Parameter configurations were set as follows: the soft frost parameter w of RIME was assigned its default value of 5, set the number of associated individuals in ED to 3, and the probability factor coefficient a of IAO was fixed at 0.2. Twenty-three benchmark functions from reference[57] were employed to evaluate the optimization capabilities. Specifically, functions $f_1 - f_7$ represent unimodal functions, $f_8 - f_{13}$ represent complex multimodal functions, and $f_{14} - f_{23}$ represent fixed-dimension multimodal functions.

As shown in Table 4, the IAO algorithm performs well on functions $f_1 - f_7$. It is important to highlight that the IAO reaches the theoretical best value for functions $f_1 - f_4$. For functions $f_5 - f_7$, although IAO does not achieve the theoretical optimum in terms of average convergence accuracy, it demonstrates closer proximity to the theoretical optimum in both average accuracy and best-found solutions compared to other algorithms, along with the lowest standard deviation. This indicates the algorithm's strong local exploitation capability, validating the optimization speed and effectiveness of the IAO algorithm. For functions $f_8 - f_{13}$, the IAO algorithm achieves smaller mean values and standard deviations in its 30 independent runs compared to other algorithms, except for function f_{13}. Notably, it attains the theoretical optimum in terms of average convergence accuracy for function f_9, which validates the IAO algorithm's global search capability and stability.

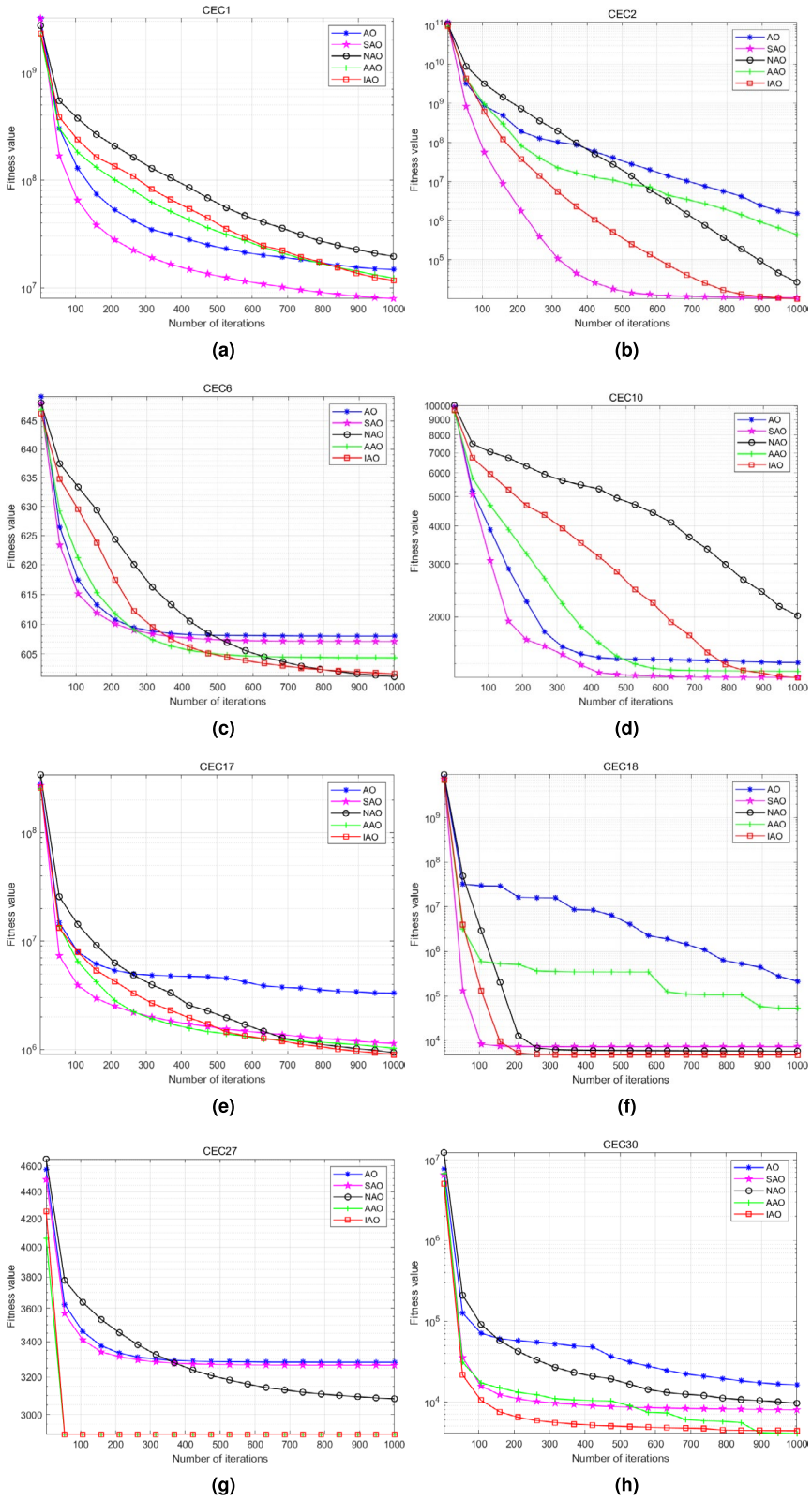


Fig. 14. Comparative plot of convergence curves for various improvement strategies.

For functions $f_{14} - f_{23}$, although IAO does not match the stability and optimization capability of RIME, GWO, or ED in some test functions, it generally outperforms the original AO algorithm across most cases.

The Wilcoxon rank-sum test is a nonparametric statistical method that is effective in detecting complex data distribution patterns. To comprehensively demonstrate the superiority of IAO, statistical analysis methods are

Function	AO		RIME		GWO		ED		IAO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	4.3E-33	8.5E-33	3.8E-01	1.4E-01	1.1E-58	3.5E-58	5.2E-06	1.1E-05	**0.0E+00**	**0.0E+00**
f_2	4.8E-23	1.2E-22	4.3E-01	2.8E-01	9.6E-35	9.9E-35	2.8E-03	3.3E-03	**0.0E+00**	**0.0E+00**
f_3	1.3E-03	3.2E-03	3.6E-02	9.6E+01	1.9E-14	5.7E-14	6.8E+02	4.1E+02	**0.0E+00**	**0.0E+00**
f_4	5.2E-05	3.5E-05	3.4E+00	1.6E+00	1.9E-14	2.2E-14	1.4E+01	3.5E+00	**0.0E+00**	**0.0E+00**
f_5	2.7E+01	1.3E+01	3.6E+02	5.9E+02	2.7E+01	7.5E-01	9.0E+01	4.2E+01	**2.6E+01**	**3.6E-01**
f_6	8.2E-05	2.5E-04	3.9E-01	2.1E-01	6.4E-01	4.0E-01	1.3E-05	3.4E-05	**9.8E-07**	**6.7E-07**
f_7	1.4E-03	7.1E-04	2.0E-02	8.1E-03	7.8E-04	3.5E-04	4.5E-02	1.9E-02	**4.9E-05**	**3.6E-05**
f_8	-1.1E+04	3.2E+02	-1.1E+04	4.6E+02	-6.0E+03	8.6E+02	-9.9E+03	4.0E+02	**-1.2E+04**	**2.4E+02**
f_9	**0.0E+00**	**0.0E+00**	4.8E+01	1.0E+01	3.1E-01	1.7E+00	2.5E+01	3.8E+00	**0.0E+00**	**0.0E+00**
f_{10}	2.0E-14	9.3E-15	1.6E+00	5.6E-01	1.6E-14	4.4E-15	2.5E+00	6.8E-01	**4.4E-16**	**3.0E-31**
f_{11}	6.5E-03	1.8E-02	5.3E-01	1.6E-01	3.0E-03	8.8E-03	2.9E-02	3.3E-02	**0.0E+00**	**0.0E+00**
f_{12}	5.6E-04	1.7E-03	1.4E+00	1.2E+00	4.3E-02	3.0E-02	7.6E-02	2.5E-01	**3.1E-05**	**1.2E-04**
f_{13}	1.9E-01	2.1E-01	5.6E-02	2.8E-02	5.0E-01	2.1E-01	**2.7E-02**	**5.6E-02**	1.0E+00	5.9E-01
f_{14}	3.4E+00	3.9E+00	10.0E-01	5.5E-13	3.6E+00	3.8E+00	**10.0E-01**	**3.4E-16**	1.5E+00	1.8E+00
f_{15}	6.4E-03	1.3E-02	5.4E-03	1.3E-02	3.0E-03	6.9E-03	1.0E-03	3.1E-04	**3.9E-04**	**1.3E-04**
f_{16}	**-1.0E+00**	**0.0E+00**	-1.0E+00	1.6E-08	-1.0E+00	4.66E-09	**-1.0E+00**	**0.0E+00**	**-1.0E+00**	**0.0E+00**
f_{17}	4.0E-01	1.7E-03	4.0E-01	6.0E-09	4.0E-01	2.8E-07	**4.0E-01**	**1.1E-16**	4.0E-01	1.1E-04
f_{18}	4.8E+00	6.9E+00	3.0E+00	1.4E-07	3.0E+00	9.5E-06	3.0E+00	4.5E-16	3.0E+00	1.3E-08
f_{19}	-2.9E+00	1.5E-03	-3.9E+00	1.4E-07	-3.9E+00	2.1E-03	-3.9E+00	2.7E-15	-3.9E+00	3.5E-05
f_{20}	-3.3E+00	5.5E-02	-3.3E+00	5.5E-02	-3.3E+00	8.5E-02	-3.3E+00	1.4E-15	-3.3E+00	5.4E-02
f_{21}	-6.3E+00	3.2E+00	-8.5E+00	2.4E+00	-9.7E+00	1.6E+00	-8.6E+00	2.2E+00	**-1.0E+01**	**2.1E-02**
f_{22}	-5.3E+00	3.5E+00	-9.4E+00	2.2E+00	**-1.0E+01**	**3.1E-04**	-8.5E+00	2.5E+00	-1.0E+01	1.4E+00
f_{23}	-6.4E+00	3.7E+00	-9.6E+00	2.0E+00	-1.0E+01	1.8E+00	-9.0E+00	2.3E+00	**-1.1E+01**	**7.9E-03**

Table 4. Comparison of optimization capabilities for Benchmark functions.

employed to analyze each simulation result. Perform the Wilcoxon rank-sum test comparing IAO against the ED, GWO, ED, and RIME algorithms on benchmark test functions, and calculate the corresponding p-values. As indicated by Table 5, The symbols "$+$", "$=$", and "$-$" represent that the optimization performance of IAO is better than, equal to, and worse than other algorithms, respectively ,the p-values of the Wilcoxon rank-sum test for IAO are mostly less than 5%. This statistically indicates the advantage of IAO in optimizing benchmark test functions.

Simulation of inspection cases and algorithm comparison

Simulation of inspection cases

To evaluate the effectiveness of the proposed approach for addressing the Multi-Heterogeneous Robot Task Allocation (MHRTA) challenge, simulations were conducted within a robotic inspection environment, as depicted in Fig. 1. Considering the constraints imposed by terrain passability at inspection locations and the availability of heterogeneous robot types, we formulated an optimization objective aimed at minimizing the total travel costs for the robot fleet. Implemented in MATLAB, the proposed method optimally assigns inspection tasks to individual robots and determines the corresponding Hamiltonian routes for their inspection paths. The resulting multi-heterogeneous robot inspection routes are illustrated in Fig. 15. In the chemical plant, three robots R_1, R_2, and R_3 are required to perform 21 inspection tasks. R_1 and R_2 are wheeled robots, while R_3 is a quadruped robot. The inspection route for robot R_1 and R_2 are represented by a dashed line and a dotted line, respectively, whereas the routes for robots R_3 is depicted with a solid line, inspection points are indicated by circles and triangles, with triangles denoting distillation columns. Due to their particular environmental constraints, tasks at distillation columns can exclusively be executed by the more robust R_3 robot. In contrast, circles, which represent oil storage tanks, indicate tasks that may be performed by any of the three robots. Additionally, black areas denote obstacles, and the pentagram at the bottom of the figure represents the warehouse, serving as both the starting and ending point for all robots. The total travel cost computed using the proposed method is 262.99 meters. ?

Comparative analysis of inspection test cases

Due to the confidentiality of the layout information of the chemical plant, we have synthesized eight sets of test cases based on our understanding and knowledge of the chemical plant layout to assess the performance of the IAO-v when addressing the MHRTA problem. With the parameters detailed configurations shown in Table 6. In the test cases, where M_T represents the number of tasks, M_R the number of robots. N_T represents the

Function	AO	RIME	GWO	ED
f_1	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_2	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_3	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_4	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_5	3.02E-11	3.02E-11	8.15E-11	1.07E-07
f_6	5.48E-11	3.01E-11	3.01E-11	5.11E-01
f_7	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_8	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_9	NaN	1.21E-12	2.15E-02	1.21E-12
f_{10}	9.62E-13	1.21E-12	5.19E-13	1.21E-12
f_{11}	2.16E-02	1.21E-12	4.19E-02	1.21E-12
f_{12}	4.61E-10	3.02E-11	3.02E-11	1.53E-05
f_{13}	7.12E-09	3.69E-11	2.01E-04	4.50E-11
f_{14}	4.45E-04	7.84E-02	7.28E-04	1.93E-10
f_{15}	1.78E-10	1.09E-10	1.27E-02	1.76E-10
f_{16}	NaN	1.21E-12	1.21E-12	NaN
f_{17}	1.89E-04	5.94E-02	3.03E-02	1.93E-10
f_{18}	8.72E-11	2.60E-11	2.64E-12	1.61E-01
f_{19}	2.84E-04	1.43E-05	5.86E-06	1.21E-12
f_{20}	4.64E-01	3.18E-03	2.64E-01	1.21E-12
f_{21}	2.43E-05	7.62E-03	5.32E-03	2.80E-09
f_{22}	5.19E-07	5.30E-01	2.07E-02	1.27E-07
f_{23}	5.61E-05	6.95E-01	1.71E-01	2.11E-10
+/=/-	19/3/1	17/4/2	19/1/3	16/1/6

Table 5. Wilcoxon rank sum test results.

number of different task types, and each N_T is an array of 1 row and 8 columns, with each column representing a task type. Therefore, a total of 8 task types are considered. Reference[45] considers 5 task types, including fire detection, specific area photography, etc. This paper also adds three task types: voiceprint detection, required robot passing ability, and valve opening and closing operation. Take Case 1 in Table 6: N_T indicates that Case 1 contains 0 tasks of the first type, 3 tasks each of the second and third task types, 5 tasks of the fourth type, and so on for the remaining task types. N_R represents the number of different robot types, and each N_R is an array of 1 row and 5 columns, with each column representing a robot type. Taking Case 1 in Table 6 as an example, N_R indicates that there is 1 robot in the first category, 1 robot each in the second and third categories, and so on for the remaining categories. The applicability of different types of robots to various types of tasks is shown in Table 7, The symbol ✓ indicates that this type of robot is suitable for the corresponding task type, while the opposite symbol × indicates that it is not suitable.

To evaluate the impact of population size, iteration times on solving the MHRTA problem. Sensitivity analysis is conducted by setting the population size and iteration times to different values. With all other parameters held constant, Case 6 was executed. As illustrated in Fig. 16, the figure delineates how the interplay between population size and iteration count influences the performance of both AO and ED algorithms when addressing MHRTA problems. It can be intuitively seen from Fig. 16 that as the population size and iteration times increase, The average value becomes better and eventually tends to stabilize. The reason is that the increase in population size improves search efficiency, while the increase in iteration times leads to an increase in search frequency and subsequent search accuracy.

To investigate the impact of varying ε values on solving MHRTA problems, ε was selected from the set [5, 8, 15, 18, 20]. Sensitivity analysis was performed using test cases 6 and 7 to evaluate the results obtained by various algorithms under different ε values. With a population size of 100 and an iteration limit of 1000, Tables 8 and 9 present the computational outcomes for test cases 6 and 7, respectively, corresponding to different values of ε. Analysis of these tables, together with the box plots of different ε values shown in Fig. 17, indicates that ε has little effect on algorithm performance. This is attributed to the fact that each element in the encoding is confined to the interval $[-\varepsilon,\varepsilon]$. Upon initializing the population, the encoding elements are uniformly distributed within this range, and new individuals are generated by performing addition and multiplication operations that remain within $[-\varepsilon, \varepsilon]$. Therefore, the magnitude of the new individuals is determined by comparative evaluation rather than direct mapping to the solution space.

To assess the effect of the number of relevant individuals incorporated in the ED algorithm for addressing the MHRTA problem, we conducted a sensitivity analysis by varying this number across the set [1, 2, 3, 4, 5,

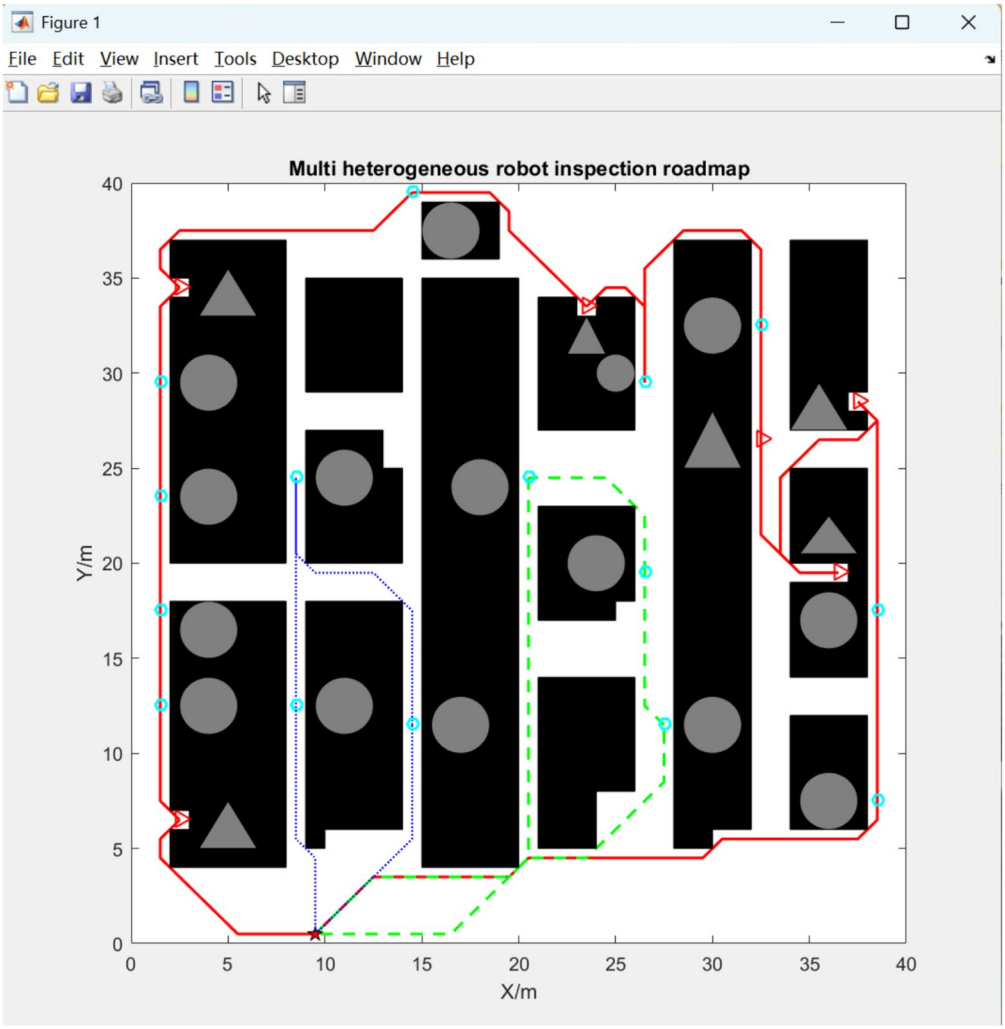


Fig. 15. Multi-Heterogeneous robot inspection route map.

Test Case	M_R	M_T	N_R	N_T
Case 1	3	20	[1,1,1,0,0]	[0,3,3,5,3,6,0,0]
Case 2	3	30	[1,1,1,0,0]	[0,5,5,6,5,9,0,0]
Case 3	3	46	[1,1,1,0,0]	[0,10,9,10,6,11,0,0]
Case 4	4	20	[1,1,1,1,0]	[0,3,3,5,3,6,0,0]
Case 5	4	30	[1,1,1,1,0]	[0,3,3,7,3,6,5,3]
Case 6	4	46	[1,1,1,1,0]	[0,10,9,10,6,11,0,0]
Case 7	5	30	[1,1,1,1,1]	[3,3,4,5,3,3,3,6]
Case 8	5	50	[1,1,1,1,1]	[5,7,8,7,5,5,5,8]

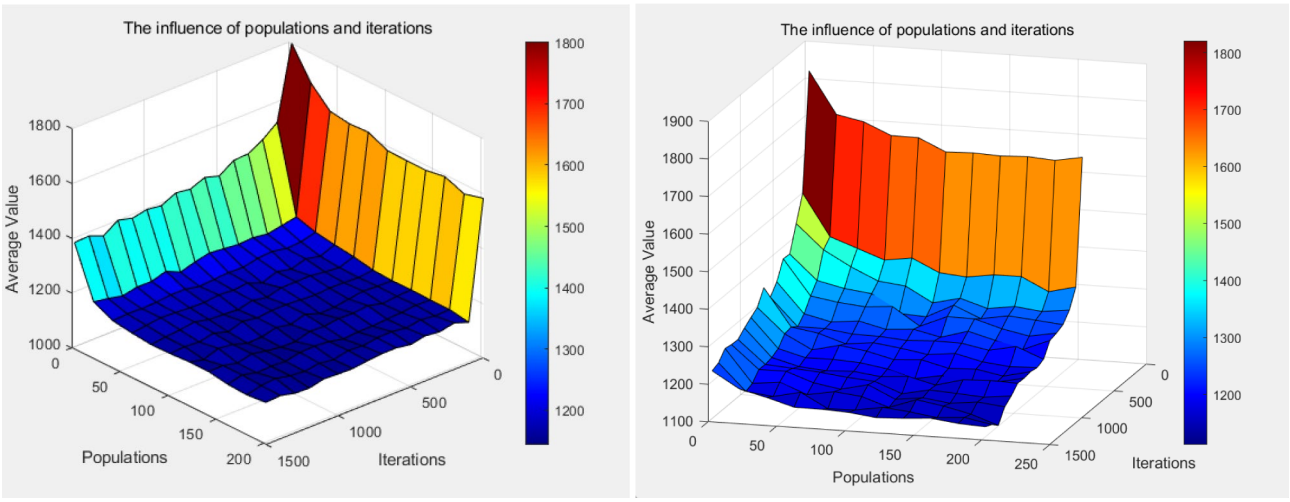
Table 6. Test case parameters.

6]. In these experiments, the iteration count was fixed at 1000 and the population size at 100. Under otherwise identical conditions, tests were performed in both Case 6 and Case 7, with the results presented in Table 10. As shown in the table, increasing the number of associated individuals leads to a deterioration in solution quality. Consequently, for subsequent comparative experiments, we set the number of associated individuals to 1 and compared its performance with the algorithm proposed in this study.

The improved artemisinin optimizer proposed in this paper is compared with other novel and classical metaheuristic algorithms as well as the CPLEX exact algorithm solver. GA and PSO adopted discrete encoding methods from the literature[45], And other algorithms use the encoding scheme in this paper and set ε to 10. The population size is 100 and the number of iterations is 1000. Each algorithm was independently executed 10 times for comparison, with the results presented in Table 11, GA represents genetic algorithm, and PSO

Robot Type	Task Type							
	Type 1	Type 2	Type3	Type 4	Type 5	Type 6	Type 7	Type 8
Type 1	×	×	×	✓	✓	✓	×	×
Type 2	×	✓	×	×	×	✓	×	✓
Type 3	×	×	✓	✓	×	✓	✓	×
Type 4	×	×	×	✓	×	×	✓	✓
Type 5	✓	×	×	×	×	×	✓	✓

Table 7. Suitability of robots for task execution.



(a) (b)

Fig. 16. The influence of populations and iterations.

εvalue	AO		ED		GWO		IAO		IAO-v	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
5	1.06E+03	5.89E+01	1.29E+03	4.52E+01	1.22E+03	9.03E+01	9.66E+02	3.63E+01	9.38E+02	3.00E+01
8	1.08E+03	5.95E+01	1.29E+03	4.22E+01	1.23E+03	6.81E+01	9.78E+02	5.72E+01	9.54E+02	4.74E+01
15	1.06E+03	5.59E+01	1.29E+03	4.35E+01	1.23E+03	7.89E+01	9.76E+02	4.25E+01	9.36E+02	2.12E+01
18	1.07E+03	7.27E+01	1.30E+03	4.45E+01	1.23E+03	1.02E+02	9.76E+02	4.67E+01	9.43E+02	3.22E+01
20	1.06E+03	5.36E+01	1.28E+03	4.67E+01	1.25E+03	8.37E+01	9.62E+02	3.58E+01	9.47E+02	3.37E+01

Table 8. Analysis of the effect of ε Value on Algorithm performance in case 6.

εvalue	AO		ED		GWO		IAO		IAO-v	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
5	1.16E+03	3.66E+01	1.17E+03	3.52E+01	1.22E+03	2.81E+01	1.13E+03	3.77E+01	1.13E+03	3.83E+01
8	1.17E+03	3.12E+01	1.17E+03	3.21E+01	1.22E+03	2.25E+01	1.13E+03	4.32E+01	1.12E+03	4.12E+01
15	1.17E+03	3.34E+01	1.17E+03	3.65E+01	1.22E+03	4.86E+01	1.13E+03	3.69E+01	1.12E+03	4.03E+01
18	1.17E+03	3.03E+01	1.16E+03	3.82E+01	1.22E+03	2.76E+01	1.13E+03	3.49E+01	1.11E+03	3.94E+01
20	1.16E+03	3.07E+01	1.16E+03	3.16E+01	1.22E+03	2.68E+01	1.13E+03	3.75E+01	1.12E+03	4.19E+01

Table 9. Analysis of the Effect of ε Value on Algorithm Performance in Case 7.

represents particle swarm optimization algorithm. AO represents the basic artemisinin optimization algorithm without any improvement mechanism, IAO denotes the improved AO algorithm with three strategies integrated, IAO-v indicates the IAO algorithm with the variable neighborhood search strategy embedded. Unit specifies the measurement units in the table, where *m* represents meters (length units), *s* represents seconds (time units), Bold

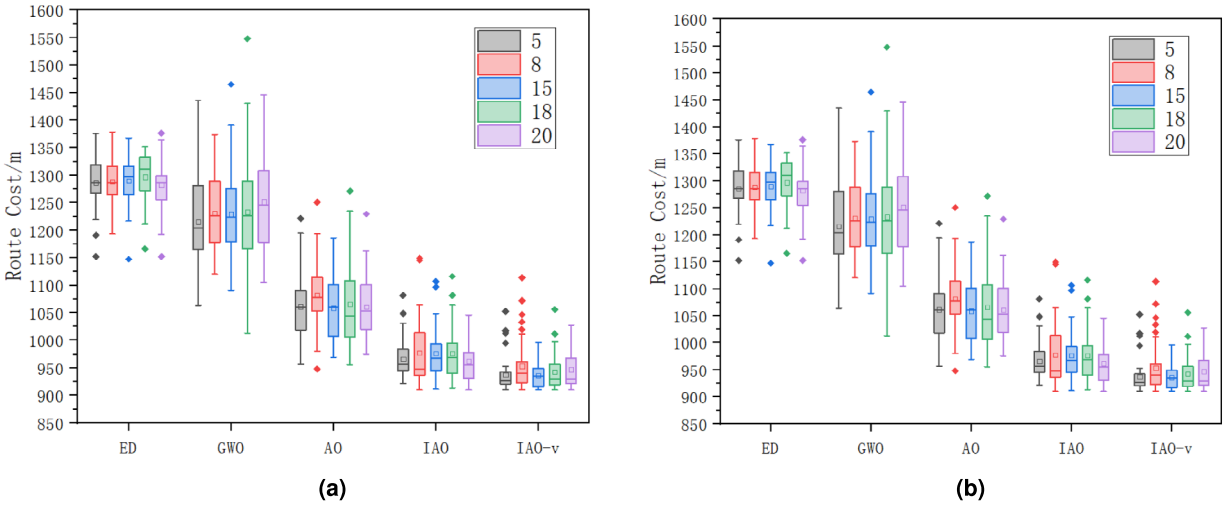


Fig. 17. The influence of parameter ε on the algorithm.

Number of associated individuals	Case6		Case7	
	Avg	Std	Avg	Std
1	1.25E+03	4.14E+01	1.15E+03	2.89E+01
2	1.26E+03	4.00E+01	1.17E+03	2.94E+01
3	1.28E+03	4.80E+01	1.17E+03	3.68E+01
4	1.31E+03	3.98E+01	1.18E+03	3.40E+01
5	1.32E+03	3.86E+01	1.19E+03	4.20E+01
6	1.29E+03	8.09E+01	1.20E+03	3.35E+01

Table 10. The impact of ED parameter values.

indicates the optimal result for the same example, and only one algorithm has achieved this optimal result. After conducting sensitivity analysis on the relevant parameters of each algorithm, a basis is provided for setting the algorithm parameters in the experiment. The number of associated individuals in the ED algorithm is taken as 1. The crossover probability and mutation probability of GA are equal to 0.08. The individual learning factor and social learning factor of PSO are both 0.15, and the inertia weight is 0.5.

According to the data presented in Table 11, the improved Artemisinin Optimizer (IAO) algorithm exhibits superior mean values and lower standard deviations over 10 independent runs when compared to the AO, ED, and GWO algorithms, as well as the classical GA and PSO algorithms across all eight cases. This result highlights the enhanced effectiveness of the IAO integrated with three strategies in addressing the MHRTA problem. Although the CPLEX solver is capable of obtaining optimal solutions for cases 1, 2, and 4 within a relatively short time, it requires more time for case 2 relative to the metaheuristic algorithms. Moreover, owing to the NP-hard characteristics of the MHRTA challenge in the remaining test cases, the CPLEX solver struggles to compute solutions within polynomial time, with its computation time being capped at 540 seconds. Comparative results between CPLEX and the IAO-V algorithm indicate that IAO-V outperforms CPLEX in both computational time and solution accuracy. Furthermore, IAO-V achieves higher precision than other metaheuristic algorithms across all eight test cases analyzed in this study. These findings demonstrate that integrating a Variable Neighborhood Search strategy within the Artemisinin Optimization framework effectively enhances local search capabilities and overall performance in solving the MHRTA problem. Nevertheless, due to the inclusion of this strategy, the IAO-V algorithm requires more computational time compared to the original IAO algorithm.

A box plot is a statistical chart that depicts the dispersion of a dataset, and it can reflect the stability of optimization outcomes. To validate the robustness of the IAO-V algorithm, we analyze box plots that compare the results of seven algorithms across eight test cases. Figure 18 presents the performance of each algorithm over varying test case sizes. In small-scale test cases, the IAO-V algorithm exhibits a substantially smaller interquartile range and lower average values compared to the other algorithms, with its median positioned closer to the lower quartile. In test case 7, although the interquartile range of the IAO-V algorithm is larger than that of the others, both its median and average values remain lower. These results indicate that the IAO-V algorithm demonstrates high stability and enhanced search capability in solving the MHRTA problem.

To comprehensively demonstrate the superiority of IAO-V. The Wilcoxon rank-sum test is conducted on the results of IAO-V for eight different cases, comparing them with the results of ED, GWO, AO, and IAO algorithms, and the corresponding p-values are calculated. As indicated by Table 12, The symbols "+", " =",

Test Case	N_R	K_T	Results	GA	PSO	ED	GWO	AO	IAO	IAO-v	CPLEX	Unit
Case 1	3	20	Best	792.1	786.2	670.3	670.3	670.3	670.3	670.3	670.3	*m*
			Avg	837.8	843.2	673.7	696.1	677.7	675.8	671.4	**670.3**	*m*
			Time	13.68	24.60	11.66	26.76	11.32	12.44	111.12	5.97	s
Case 2	3	30	Best	1164.2	1141.5	821.6	843.3	823.0	821.6	821.6	821.6	*m*
			Avg	1253.3	1233.4	891.1	897.6	845.5	833.9	824.3	**821.6**	*m*
			Time	16.23	34.10	13.6	34.44	14.2	59.08	139.18	199.14	*s*
Case 3	3	46	Best	1725.1	1675.4	1208.1	1149.7	1048.6	1008.6	**981.2**	1049.2	*m*
			Avg	1842.0	1841.2	1261.8	1236.3	1084.5	1034.4	**1008.3**	1050.2	*m*
			Time	23.5	50.05	17.75	46.6	19.86	26.61	182.69	540	*s*
Case 4	4	20	Best	793.0	838.9	674.6	675.9	674.6	674.6	674.6	**660.06**	*m*
			Avg	843.2	920.6	676.2	701.1	688.9	686.9	674.7	**660.06**	*m*
			Time	14.67	26.23	12.5	28.27	12.32	23.96	119.93	4.05	*s*
Case 5	4	30	Best	1168.6	1226.7	779.1	801.4	729.4	728.5	**728.5**	793.6	*m*
			Avg	1275.8	1337.4	866.8	879.0	815.0	774.8	**745.5**	793.6	*m*
			Time	18.3	35.41	14.92	37.8	15.19	61.91	149.04	540	*s*
Case 6	4	46	Best	1658.2	1729.0	1220.1	1007.0	994.2	937.1	**917.4**	953.9	*m*
			Avg	1890.9	1894.8	1293.7	1186.4	1058.8	972.2	**937.6**	953.9	*m*
			Time	23.56	51.55	19.49	50.42	20.85	87.98	197.44	540	*s*
Case 7	5	30	Best	1294.6	1345.4	1140.5	1157.8	1148.1	1074.1	**1062.0**	1126.4	*m*
			Avg	1395.4	1451.7	1170.1	1206.5	1175.1	1155.3	**1125.1**	1126.4	*m*
			Time	18.324	36.17	15.34	37.36	16.3	66.90	154.13	540	*s*
Case 8	5	50	Best	2024.7	2045.5	1595.4	1483.2	1303.2	1232.3	**1223.8**	1291.2	*m*
			Avg	2233.0	2199.3	1638.0	1606.6	1399.6	1266.9	**1243.0**	1294.2	*m*
			Time	25.37	56.23	20.63	53.05	22.43	95.42	211.69	540	*s*

Table 11. Comparison results of test case.

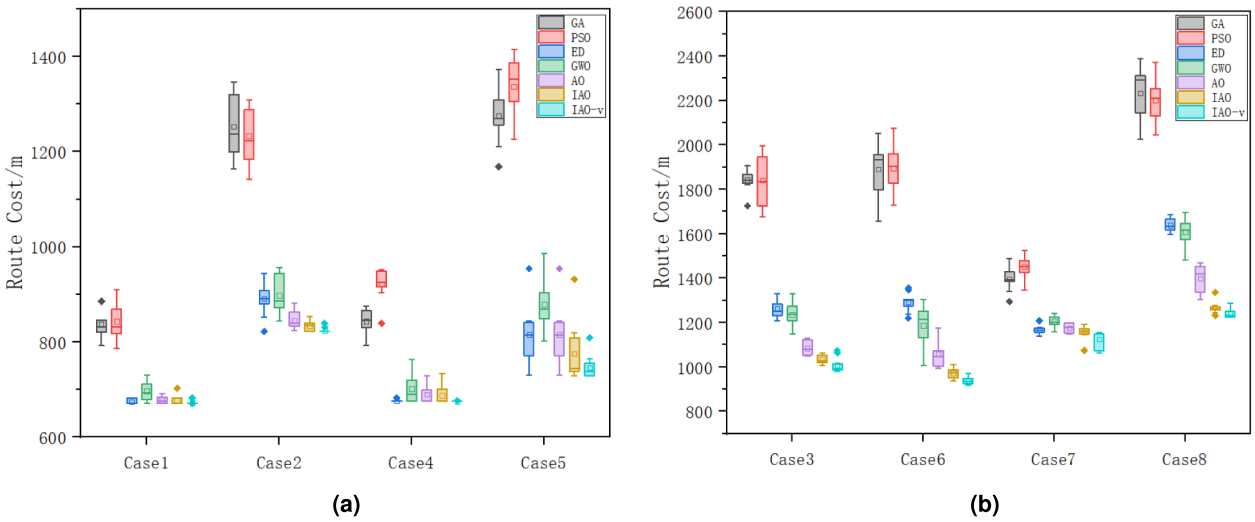


Fig. 18. Distribution of results from 10 independent runs.

and "−" represent that the optimization performance of IAO-V is better than, equal to, and worse than other algorithms, respectively. ,the p-values of the Wilcoxon rank-sum test for IAO-V are mostly less than 5%. This statistically demonstrates the advantage of IAO-V in optimizing MHRTA problems, further demonstrating its robustness.

Conclusions

This paper introduces a method for the coordinated optimization of task allocation and Hamiltonian path planning for multiple heterogeneous robots using an improved Artemisinin optimizer. While the traditional AO algorithm is designed for continuous optimization, the Multiple Heterogeneous Robot Task Allocation

Test Case	ED	GWO	AO	IAO
Case 1	$5.17E-01$	$3.72E-04$	$4.92E-02$	$3.32E-02$
Case 2	$1.02E-03$	$1.49E-04$	$1.13E-03$	$2.26E-02$
Case 3	$1.83E-04$	$1.83E-04$	$1.31E-03$	$3.12E-02$
Case 4	$2.55E-01$	$1.33E-04$	$6.62E-04$	$2.29E-03$
Case 5	$3.13E-04$	$3.13E-04$	$2.40E-03$	$1.37E-01$
Case 6	$1.83E-04$	$1.83E-04$	$1.83E-04$	$3.61E-03$
Case 7	$2.78E-03$	$1.81E-04$	$2.46E-03$	$8.98E-03$
Case 8	$1.82E-04$	$1.82E-04$	$1.82E-04$	$4.51E-02$
+/=/-	8/0/0	8/0/0	8/0/0	8/0/0

Table 12. Compared with the IAO-v Wilcoxon signed-rank test.

(MHRTA) problem presents a discrete combinatorial optimization challenge. To address this, we adopt a two-layer encoding scheme that links the encoding space with the solution space, specifically targeting multiple-robot task allocation and Hamiltonian routing. Moreover, to mitigate the risk of the conventional AO algorithm becoming trapped in local optima, we propose an enhanced version that integrates three key strategies: incorporating concepts from the slime mold algorithm to modify the position update method of artemisinin molecules during the comprehensive elimination phase, introducing a self-adaptive t-distribution to boost global search capabilities, and employing a variable neighborhood search strategy to improve local search performance in the later stages of optimization, thereby addressing the NP-hard nature of the MHRTA problem.

Experiments conducted using several CEC2014 functions along with 23 benchmark functions demonstrate that the IAO algorithm outperforms alternative methods in terms of both performance and consistency. Additionally, in studies involving eight heterogeneous robot inspection scenarios, evaluations against other metaheuristic algorithms and the CPLEX solver reveal that the IAO-v algorithm exhibits superior optimization capabilities and stability in addressing the MHRTA problem. Future research will focus on the following key areas:

(1)We further emphasize the practical application requirements of chemical industrial parks by considering the domino effects of initial accidents and quantifying risk values across different zones. To address these challenges, we develop multi-objective optimization models and corresponding algorithms designed to minimize losses arising from accidents in these facilities.

(2) Incorporating a variable neighborhood search strategy into the IAO algorithm enhances the accuracy in solving MHRTA problems; however, it also moderately increases the algorithm's time complexity. Future research will focus on developing new algorithms that improve accuracy without incurring additional computational costs.

(3) This study does not capture dynamic production changes, particularly uncertainty factors such as the random arrival of inspection tasks and robot malfunctions. Future research will incorporate stochastic elements into the task assignment process to enhance responsiveness in dynamic settings.

Data availability

The datasets and code for this study are available from the corresponding author on reasonable request.

Received: 26 May 2025; Accepted: 20 November 2025
Published online: 01 December 2025

References

1. Vianello, C., Milazzo, M. F. & Maschio, G. Cost–benefit analysis approach for the management of industrial safety in chemical and petrochemical industry. *J. Loss Prev. Process. Ind.* **58**, 116–123 (2019).
2. Gómez-Rosal, D. A. et al. A smart robotic system for industrial plant supervision. In *2023 IEEE SENSORS*, 1–4 (2023).
3. Vesentini, F., Di Persio, L. & Muradore, R. A brownian–markov stochastic model for cart-like wheeled mobile robots. *Eur. J. Control.* **70**, 100771 (2023).
4. Wen, C. & Ma, H. An efficient two-stage evolutionary algorithm for multi-robot task allocation in nuclear accident rescue scenario. *Appl. Soft Comput.* **152**, 111223 (2024).
5. Liu, W. et al. An effective hybrid genetic algorithm for the multi-robot task allocation problem with limited span. *Expert. Syst. with Appl.* **280**, 127299 (2025).
6. Xiao, Y., Cui, H., Khurma, R. A. & Castillo, P. A. Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems. *Artif. Intell. Rev.* **58**, 84 (2025).
7. Wang, C.-H., Pan, Q.-K., Li, X.-P., Sang, H.-Y. & Wang, B. A multi-objective teaching-learning-based optimizer for a cooperative task allocation problem of weeding robots and spraying drones. *Swarm Evol. Comput.* **87**, 101565 (2024).
8. Miloradović, B., Çürüklü, B., Ekström, M. & Papadopoulos, A. V. Gmp: A genetic mission planner for heterogeneous multirobot system applications. IEEE Transactions on Cybern. *J. Clean. Prod.* **52**, 10627–10638 (2021).
9. Yin, M., Zhou, Y., Qian, X., Liu, Q. & Guo, X. Optimizing Multi-Period Green Electric Vehicle Battery Recycling Network: Integrating Winner Determination within a 4PL Framework. *IEEE Transactions on Engineering Management.* under review. (2025)
10. Wen, C. & Ma, H. An indicator-based evolutionary algorithm with adaptive archive update cycle for multi-objective multi-robot task allocation. *Neurocomputing* **593**, 127836 (2024).
11. Milot, A., Chauveau, E., Lacroix, S. & Lesire, C. Market-based multi-robot coordination with htn planning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* , 2606–2612 (2021).

12. Gautier, P., Laurent, J. & Diguet, J.-P. Comparison of market-based and dqn methods for multi-robot processing task allocation (mrpta). In *2020 fourth IEEE international conference on robotic computing (IRC)*, 336–343 (2020).
13. Quinton, F., Grand, C. & Lesire, C. Communication-preserving bids in market-based task allocation. In *In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13708–13713 (2022).
14. Bai, R., Zheng, R., Xu, Y., Liu, M. & Zhang, S. Hierarchical multi-robot strategies synthesis and optimization under individual and collaborative temporal logic specifications. *Robotics Auton. Syst.* **153**, 104085 (2022).
15. Li, W.-H., Zhao, T. & Dian, S.-Y. A multi-robot coverage path planning method based on genetic algorithm. In *2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, 13–18 (2021).
16. Yan, M., Yuan, H., Xu, J., Yu, Y. & Jin, L. Task allocation and route planning of multiple uavs in a marine environment based on an improved particle swarm optimization algorithm. *EURASIP J. on Adv. Signal Process.* **2021**, 1–23 (2021).
17. Hua, X. et al. Research on many-to-many target assignment for unmanned aerial vehicle swarm in three-dimensional scenarios. *Comput. & Electr. Eng.* **91**, 107067 (2021).
18. Liu, D. et al. Function value ranking aware differential evolution for global numerical optimization. *Swarm Evol. Comput.* **78**, 101282 (2023).
19. Khishe, M. & Mosavi, M. R. Chimp optimization algorithm. *Expert Syst. Appl.* **149**, 113338 (2020).
20. Xue, J. & Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **79**, 7305–7336 (2023).
21. Enriquez, E. A. T., Mendoza, R. G. & Velasco, A. C. T. Philippine eagle optimization algorithm. *IEEE Access* **10**, 29089–29120 (2022).
22. Azizi, M., Aickelin, U., Khorshidi, A. H. & Baghalzadeh Shishehgarkhaneh, M. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Sci. Reports* **13**, 226 (2023).
23. Lang, Y. & Gao, Y. Dream optimization algorithm (doa): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **436**, 117718 (2025).
24. Yuan, C. et al. Artemisinin optimization based on malaria therapy: Algorithm and applications to medical image segmentation. *Displays* **84**, 102740 (2024).
25. Sasmal, Buddhadev et al. A comprehensive survey on aquila optimizer. *Arch. Comput. Methods Eng.* **30**, 4449–4476 (2023).
26. Hussien, A. G., Khurma, R. A., Alzaqebah, A., Amin, M. & Hashim, F. A. Novel memetic of beluga whale optimization with self-adaptive exploration–exploitation balance for global optimization and engineering problems. *Soft Comput.* **27**, 13951–13989 (2023).
27. Khalid, O. W., Isa, N. A. M. & Lim, W. H. Self-adaptive emperor penguin optimizer with multi-strategy parameter adaptation mechanism for complex optimization problems. *Alex. Eng. J.* **120**, 657–686 (2025).
28. Koessler, E. & Almomani, A. Hybrid particle swarm optimization and pattern search algorithm. *Optim. Eng.* **22**, 1539–1555 (2021).
29. Chen, D. Application of improved genetic algorithms in path planning. *J. Internet Technol.* **25**, 1091–1099 (2024).
30. Liu, Z., Song, S., Chen, J. & Hou, C. Enhanced wifi/pedestrian dead reckoning indoor localization using artemisinin optimization-particle swarm optimization-particle filter. *Electronics* **13**, 3366 (2024).
31. Tian, Y., Pan, X., Zhou, X., Liu, L. & Wei, D. Interpersonal sensitivity prediction based on multi-strategy artemisinin optimization with fuzzy k-nearest neighbor. *J. Bionic Eng.* **22**, 1484–1505 (2025).
32. Zhang, X., Li, J. & Guo, Z. Region-partitioned obstacle avoidance strategy for large-scale offshore wind farm collection system considering buffer zone. *Energy* **313**, 133758 (2024).
33. Zhu, C., Zhang, J. & Yang, J. Grain condition inspection: one improved grey wolf algorithm for nodes deployment optimization combining with path planning of multi-inspection robots. *Soft Comput.* **28**, 11971–11986 (2024).
34. Yuan, Y., Yang, P., Jiang, H. & Shi, T. A multi-robot task allocation method based on the synergy of the K-Means++ algorithm and the particle Swarm algorithm. *Biomimetics.* **9**, 694 (2024).
35. Huang, J. et al. A pesticide spraying mission allocation and path planning with multicopters. *IEEE Trans. Aerosp. Electron. Syst.* **60**, 2277–2291 (2024).
36. Le, T. et al. Integrating Both Routing and Scheduling Into Motion Planner for Multivehicle System. *IEEE Can. J. Electr. Comput. Eng.* **46**, 56–68 (2023).
37. Guo, H., Miao, Z., Ji, J. & Pan, Q. An effective collaboration evolutionary algorithm for multi-robot task allocation and scheduling in a smart farm. *Knowledge-Based Syst.* **289**, 111474 (2024).
38. Lu, Z. et al. A reinforcement learning-based optimization method for task allocation of agricultural multi-robots clusters. *Comput. Electr. Eng.* **120**, 109752 (2024).
39. Nguyen, T. P. et al. Towards sustainable scheduling of a multi-automated guided vehicle system for collision avoidance. *Comput. Electr. Eng.* **120**, 109824 (2024).
40. Zhang, L. et al. Energy efficient multi-robot task allocation constrained by time window and precedence. *IEEE Trans. Autom. Sci. Eng.* **22**, 18162–18173 (2025).
41. Majumder, A., Majumder, A. & Bhaumik, R. Teaching–learning-based optimization algorithm for path planning and task allocation in multi-robot plant inspection system. *Arab. J. for Sci. Eng.* **46**, 8999–9021 (2021).
42. Wei, C., Ji, Z. & Cai, B. Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach. *IEEE Robot. Autom. Lett.* **5**, 2530–2537 (2020).
43. Luo, S. et al. Chemical safety inspection path optimization problems using improved multi-objective discrete growth optimization algorithm. *Processes* **13**, 1445 (2025).
44. Ye, X., Guo, H. & Luo, Z. Two-stage task allocation for multiple construction robots using an improved genetic algorithm. *Autom. Constr.* **165**, 105583 (2024).
45. Chakraa, H., Leclercq, E., Guérin, F. & Lefebvre, D. A centralized task allocation algorithm for a multi-robot inspection mission with sensing specifications. *IEEE Access* **11**, 99935–99949 (2023).
46. Chakraa, H. et al. Integrating collision avoidance strategies into multi-robot task allocation for inspection. *Trans. Inst. Meas. Control* **47**, 1466–1477 (2025).
47. Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **111**, 300–323 (2020).
48. Zhang, H., Huang, Q., Ma, L.& Zhang, Z. Sparrow search algorithm with adaptive t distribution for multi-objective low-carbon multimodal transportation planning problem with fuzzy demand and fuzzy time. *Expert Syst. Appl.* **238**, 122042 (2024).
49. Xu, Y., Zhang, M., Yang, M. & Wang, D. Hybrid quantum particle swarm optimization and variable neighborhood search for flexible job-shop scheduling problem. *J. Manuf. Syst.* **73**, 334–348 (2024).
50. Yuan, Y. & Xu, H. Flexible job shop scheduling using hybrid differential evolution algorithms. *Comput. Ind. Eng.* **65**, 246–260 (2013).
51. Yuan, Y. & Xu, H. An integrated search heuristic for large-scale flexible job shop scheduling problems. *Comput. Oper. Res.* **40**, 2864–2877 (2013).
52. Ahmed, G., Sheltami, T. & Yasar, A. Optimal path recommendation in dynamic traffic networks using the hybrid Tabu-A* algorithm. *Transportation Research Part E: Logistics and Transportation Review.* **204**, 104414 (2025).
53. Liang, J. J., Qu, B. Y., Suganthan, P. N. et al. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Report, Nanyang Technol. Univ. Singapore* **635,** 2014 (2013).

54. Su, H. et al. Rime: A physics-based optimization. *Neurocomputing* **532**, 183–214 (2023).
55. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. engineering software.* **69**, 46–61 (2014).
56. Truong, D.-N. & Chou, J.-S. Metaheuristic algorithm inspired by enterprise development for global optimization and structural engineering problems with frequency constraints. *Eng. Struct.* **318**, 118679 (2024).
57. Wang, T. L. et al. Cuckoo catfish optimizer: a new meta-heuristic optimization algorithm. *Artif. Intell. Rev.* **58**, 326 (2025).

Acknowledgements

This work was supported by Basic Science Research Program for the Education Department of Liaoning Province [grant number JYTMS20231434]; the Liaoning Provincial Science and Technology Innovation Project in the Field of Artificial Intelligence [grant number 2023JH26/10300013]; the Natural Science Foundation of Liaoning Province [grant number 2024-BS-227].

Author contributions

Chaofan Li: Creation of models, Methodology, Writing- Original draft preparation, Data curation, Software code; Qiang Liu: Methodology,Supervision, Writing - Review & Editing, Validation; Mingqiang Yin: Conceptualization, Methodology, Creation of models, Supervision, Formal analysis; Xianming Lang: Supervision, Validation; Guihua BO: Provide revision Manuscript writing suggestions.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-30056-8.

Correspondence and requests for materials should be addressed to Q.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.