COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 06/06 pp 210−221
Volume 4, Number 2, June 2024
DOI: 10.23919/CSMS.2024.0006

Adaptive Memetic Algorithm with Dual-Level Local Search for Cooperative Route Planning of Multi-Robot Surveillance Systems

Hao Cheng, Jin Yi*, Wei Xia, Huayan Pu, and Jun Luo

Abstract: The heightened autonomy and robust adaptability inherent in a multi-robot system have proven pivotal in disaster search and rescue, agricultural irrigation, and environmental monitoring. This study addresses the coordination of multiple robots for the surveillance of various key target positions within an area. This involves the allocation of target positions among robots and the concurrent planning of routes for each robot. To tackle these challenges, we formulate a unified optimization model addressing both target allocation and route planning. Subsequently, we introduce an adaptive memetic algorithm featuring dual-level local search strategies. This algorithm operates independently among and within robots to effectively solve the optimization problem associated with surveillance. The proposed method's efficacy is substantiated through comparative numerical experiments and simulated experiments involving diverse scales of robot teams and different target positions.

Key words: multi-robot system; surveillance; memetic algorithm

1 Introduction

The flourishing development of unmanned systems has led to the liberation of an increasing number of individuals from various complex, dangerous, and unsanitary activities[1–5]. For instance, robots are now employed to consistently and periodically visit specific target points in designated areas, replacing human surveillance behavior for monitoring unexpected events. Moreover, a multi-robot system, as opposed to a single robot, not only reduces task completion time and enhances efficiency but also demonstrates heightened robustness. However, leveraging the potential of a multi-robot system to achieve performance improvements necessitates not only behavioral planning for individual robots but also the implementation of sophisticated strategies to coordinate interactions between robots. In addressing the multi-robot surveillance problem, the primary challenge lies in coordinating and planning the paths of individual robots to traverse all points of interest or cover the entire designated area while minimizing task completion time or reducing the total path length to conserve energy.

Numerous scholars have delved into the intricacies of the multi-robot surveillance problem, and the prevailing approach is the adoption of a partition-based hierarchical planning strategy. Initially, the target area undergoes division into multiple disjoint subareas or subsets through decomposion or clustering methods. Subsequently, algorithms designed for single-robot path planning are applied to formulate an optimal feasible route within each sub-area. Zhang et al.[6] introduced the MSTC* algorithm for multi-robot coverage path planning, incorporating constraints such

• Hao Cheng, Jin Yi, Huayan Pu, and Jun Luo are with the State Key Laboratory of Mechanical Transmission for Advanced Equipment, Chongqing University, Chongqing 400044, China. E-mail: yijin@cqu.edu.cn.
• Wei Xia is with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano 20133, Italy. E-mail: wei.xia@polimi.it.
∗ To whom correspondence should be addressed.
※ This article was recommended by Associate Editor Xinyu Li.
Manuscript received: 2024-03-15; revised: 2024-04-18; accepted: 2024-04-23

as terrain traversability and workload capacity through balanced cut and spiral spanning tree coverage. Chen et al.[7] explored the coverage path planning of autonomous heterogeneous unmanned aerial vehicles (UAVs). To comprehensively explore the solution space, they initially formulated the problem into a mixed-integer linear programming model aimed at minimizing task completion time. A density-based clustering method was then proposed to categorize regions into separate clusters and provide an approximate optimal flight path for each UAV. Tang and Ma[8] pioneered the proposition of a mixed integer programming (MIP) model to optimally address the NP-hard rooted min-max tree cover (RMMTC) problem and proved the optimality ratio of the resulting solution can be 4. Additionally, they introduced two suboptimal heuristics to streamline the model within specified constraints of runtime and memory. Xie et al.[9] initially employed the k-means clustering algorithm to partition surveillance points with varying importance levels into subsets, ensuring workload balance among partitions. Subsequently, an enhanced particle swarm algorithm (PSO) was utilized to plan a visit sequence of surveillance points, minimizing the total distance traveled for each unmanned surface vehicle (USV). Feng et al.[10] investigated the problem of minimizing the number of robots used or maximizing the probabilistic guarantee for sweep-line coverage. They proposed a max-flow based algorithm building on boustrophedon decomposition[11] for solving the allocation task, completing it in less than 2 s for polygonal environments with over 105 vertices. The partition-based hierarchical strategy adeptly coordinates cooperation between robots, preventing conflicts. Furthermore, the transformation of multi-robot path planning into single-robot path planning within each sub-area effectively diminishes the dimensionality of path planning. However, in a multi-robot system, task allocation and path planning are inherently complementary. Path planning must align with task allocation, and the outcomes of path planning can reciprocally reflect the quality of task allocation. Regrettably, the hierarchical planning strategy lacks interaction between these two integral components.

Additionally, through extensive pre-computing and training, learning-based methods can rapidly attain effective solutions when confronted with new instances of problems. Din et al.[12] introduced a dual deep Q-learning (DDQN) based centralized convolutional neural network (CNN) to address the area surveillance problem in agriculture. A custom reward function is devised, incentivizing the exploration of unvisited areas while penalizing undesirable behavior. In cases where the deep Q-network (DQN) is susceptible to the curse of dimensionality, Jana et al.[13] proposed a formulation of the markov decision process (MDP) with a state representation applicable in deep reinforcement learning (DRL) methods such as the DQN. DRL agents can collaboratively devise strategies, showcasing scalability concerning the number of nodes in the field. Moreover, for unknown dynamic environmental factors like agent failures or wind, Tong et al.[14] introduced a distributed reinforcement learning architecture capable of tolerating agent failures and accommodating the addition of supplementary agents to replace failed ones. All agents act independently based on local observations and shared location information. While learning-based methods have demonstrated commendable performance in terms of scalability and time efficiency, their solutions' quality and stability still lag behind traditional methods.

This paper addresses the challenges posed by multi-robot surveillance problems, where a team of robots collaboratively covers all designated target positions continuously. Diverging from a hierarchical approach that initially focuses on task planning (i.e., assigning surveillance targets to different robots) and subsequently on route planning (i.e., determining the visit sequence of surveillance targets for each robot), we propose a novel methodology that concurrently tackles task planning and route planning. Specifically, we employ an encoding strategy to ensure a solution encompassing both surveillance target allocation and route planning information. Additionally, we introduce an adaptive dual-level memetic algorithm (DL-AMA) tailored for this encoding strategy. DL-AMA is designed to minimize the total length traversed by all robots, simultaneously maintaining a low workload difference among robots to ensure balanced task load distribution. The key contributions of this paper are outlined as follows:

(1) Formulation of the multi-robot surveillance problem into a single integrated optimization model to minimize the total travel distance of all robots for energy conservation. The model also seeks to minimize the maximum surveillance length among robots to

reduce the period required for the surveillance mission. Furthermore, we impose constraints on the workload difference among robots, ensuring it remains below a defined threshold to balance task load distribution.

(2) Introduction of a solution encoding strategy that jointly optimizes task allocation and route planning. Subsequently, we propose an adaptive memetic algorithm method featuring dual-level local search strategies. These strategies operate independently, addressing task allocation among robots and route planning within each robot to yield an enhanced solution.

(3) The efficacy and superiority of our proposed method are validated through simulations involving diverse team sizes and varying numbers of surveillance targets. Comparative experiments demonstrate the effectiveness of our approach over alternative methods.

The remainder of this paper is structured as follows. In Section 2, we commence with a detailed depiction of the multi-robot surveillance scenario, followed by an in-depth analysis of the objectives and constraints associated with both task assignment and route planning. Section 3 provides a comprehensive description of the proposed encoding strategy and the adaptive dual-level local search memetic algorithm. The numerical and simulation results stemming from our approach are presented and discussed in Sections 4 and 5. Finally, Section 6 encapsulates concluding remarks and outlines avenues for future work.

2 Problem Formulation

In this section, we will give a detailed description of the scenario of multi-robot surveillance, and analyze the objectives and constraints that contributed to establishing a mathematical optimization model for this problem.

2.1 Environment description

We consider a set of surveillance targets $V=\{v_1,v_2,\ldots,v_M\}$ statically distributed in a certain given area, where M is the number of surveillance targets, and the coordinates of the target v_i is (x_i,y_i). A team of robots $U=\{u_1,u_2,\ldots,u_N\}$ is mandated to collaboratively visit all designated target positions continuously for surveillance purposes, where N is the number of robots. A schematic representation of multi-robot surveillance is shown in Fig. 1. Each robot is assigned a distinct set of disjoint targets P_i, where $P_i\cap P_j=\varnothing,\forall 1\leqslant i,j\leqslant N$, and $V=P_1\cup P_2\cup\cdots\cup P_N$. Subsequently, each robot traverses a path that covers all the assigned targets.

2.2 Goal for task assignment

Concentration within single robot for surveillance targets assignment (C): This criterion emphasizes that the allocated surveillance targets for each robot should exhibit a high degree of concentration. In practical terms, this implies minimizing the distances between surveillance points assigned to the same robot. The objective is to reduce the overall length of robot paths required to cover all surveillance points effectively. The concentration of target assignment C can be expressed as follows:

$$C=\frac{1}{Nd_{\max}}\sum_{u=1}^{N}\frac{J_u}{|P_u|} \tag{1}$$

$$J_u=\sum_{i=1}^{|P_u|}\sum_{j=1}^{|P_u|}\mathrm{dist}(v_i,v_j),u=1,2,\ldots,N \tag{2}$$

where $|P_u|$ is the number of surveillance targets for robot u, $\mathrm{dist}(v_i,v_j)$ is the Euclidean distance between two targets, and $d_{\max}$ is the distance between the farthest two points among all surveillance targets in the

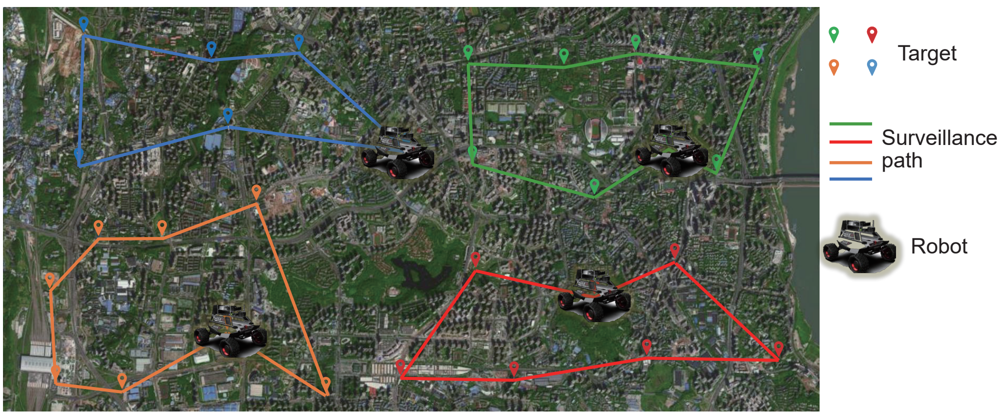


Fig. 1 Schematic diagram of multi-robot surveillance.

area:

$$\mathrm{dist}(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \forall v_i, v_j \in P_u \tag{3}$$

$$d_{\max} = \max_{v_i, v_j} \mathrm{dist}(v_i, v_j), \forall v_i, v_j \in V \tag{4}$$

Dispersion among robots for surveillance targets assignment (D): In contrast to the concentration within a single robot, the dispersion criterion focuses on allocating the set of surveillance points among robots in a dispersed manner. The aim is to minimize conflicts arising from the intersection of surveillance routes between robots. To quantify this dispersion, we employ the distance between the center points of each robot's allocated surveillance targets. The dispersion of surveillance points allocation among robots to be maximized is represented as follows:

$$D = \frac{1}{N d_{\max}} \sum_{m=1}^{N} \sum_{n=1}^{N} \mathrm{dist}(v_c^m, v_c^n) \tag{5}$$

where v_c^m is the center of the assigned targets for robot m, and its coordinate $(x_{v_c^m}, y_{v_c^m})$ can be computed as Eq. (6):

$$\begin{cases} x_{v_c^m} = \dfrac{1}{|P_u|} \displaystyle\sum_{i=1}^{|P_u|} x_i, m = 1, 2, \ldots, N; \\ y_{v_c^m} = \dfrac{1}{|P_u|} \displaystyle\sum_{i=1}^{|P_u|} y_i, m = 1, 2, \ldots, N \end{cases} \tag{6}$$

Task load difference for surveillance targets assignment (T_d): This criterion underscores the importance of maintaining a task load difference for each robot below a specified threshold T_h. The objective is to achieve a balanced distribution of the task load among robots, preventing the overuse of some robots and optimizing the utilization of others. The task load for a robot J_u is defined as the summation of distances between two targets assigned to that robot, as represented by Eq. (2). The task load difference T_d is mathematically expressed as follows:

$$T_d = \sqrt{\sum_{u=1}^{N} \frac{(J_u - \bar{J})^2}{M N d_{\max}}} \tag{7}$$

$$\bar{J} = \frac{1}{N} \sum_{u=1}^{N} J_u \tag{8}$$

2.3 Goal for route planning

Total length of all surveillance paths for route planning (L): This criterion emphasizes that each robot should plan a path traversing the assigned surveillance points as efficiently as possible to conserve energy. The surveillance path length for a single robot is calculated by using Eq. (9). Subsequently, the total surveillance length for all robots is computed through Eq. (10):

$$|l_u| = \sum_{i=1}^{|P_u|} \sum_{j=1}^{|P_u|} \mathrm{dist}(v_i, v_j) X_{i,j}, u = 1, 2, \ldots, N \tag{9}$$

$$L = \frac{1}{M d_{\max}} \sum_{u=1}^{N} |l_u| \tag{10}$$

where $X_{i,j}$ is a binary indicator, and $X_{i,j} = 1$ indicates that a robot moves from v_i to v_j during its surveillance, otherwise, $X_{i,j} = 0$.

Maximum surveillance length among all robots ($L_{\max}$): Given the constant and consistent speed of the robots, the completion time of the surveillance task is contingent on the robot with the longest surveillance route. Therefore, minimizing the maximum path length of surveillance $L_{\max}$ is crucial to reduce task completion time and enhance surveillance efficiency:

$$L_{\max} = \max_{u} |l_u|, u = 1, 2, \ldots, N \tag{11}$$

2.4 Optimization model

In summary, the primary objective of the multi-robot surveillance problem addressed in this paper is to devise a surveillance path l_u for each robot, ensuring comprehensive coverage of all surveillance points within a specified area. The paths are strategically separated to minimize conflicts between robots. Additionally, the overarching goals include minimizing the total surveillance length to conserve energy and reducing the maximum path length to enhance task completion efficiency. The mathematical description can be summarized as follows:

$$\begin{array}{ll} \text{Find} & \{l_1, l_2, \ldots, l_N\}, \\ \text{Minimize} & f = w_1 L + w_1 L_{\max} + w_2 C - w_2 D, \\ \text{Subject to} & T_d \leqslant T_h \end{array} \tag{12}$$

where $w_1, w_2 \in (0, 1)$, and $w_1 + w_2 = 1$ are objective weights for route planning and surveillance targets assignment, respectively.

3 Proposed Method

In this section, we commence by providing a comprehensive description of the encoding strategy, designed to enable a solution to encompass both task allocation and route planning information.

Subsequently, we introduce an adaptive memetic algorithm incorporating a dual-level local search strategy, tailored to enhance the effectiveness of solving this problem.

3.1 Encoding strategy

For continuous algorithms like genetic algorithms (GA) and differential evolution (DE), the solutions obtained are inherently continuous. Thus we introduce a novel strategy to address this characteristic in the context of the multi-robot surveillance problem. Initially, the dimension of the algorithm solution is set to M with the value of each dimension ranging from 1 to $N+1$, where M and N represent the numbers of surveillance targets and robots, respectively. As an illustration, consider the case of 2 robots and 8 surveillance targets; an example algorithm solution is presented in Fig. 2a. Subsequently, floor rounding is applied to each dimension of the solution vector to obtain an integer solution vector. Each surveillance target, represented by each dimension, is then allocated to the corresponding robot based on these integers. As depicted in Fig. 2b, the surveillance points assigned to each robot are 1, 2, 4, 5, 6, 8 and 3, 7, respectively. Finally, the surveillance points assigned to each robot are sorted based on their original values in the algorithm solution, resulting in a visiting sequence of surveillance points for each robot, which also constitutes the surveillance paths. Figure 2c illustrates the final surveillance paths obtained by encoding the example continuous solution vector as 6-2-8-1-5-4 and 7-3.

3.2 DL-AMA algorithm

GA[15] is an evolutionary optimization technique inspired by the process of natural selection. It is used to find approximate solutions to complex optimization and search problems. In GA, a population of potential solutions evolves over generations through selection, crossover, and mutation operations, mimicking the process of survival of the fittest. Through repeated iterations, GA gradually converges towards optimal or near-optimal solutions by favoring the best-performing individuals in the population. Building upon this foundation, Moscato and Norman[16] introduced the memetic algorithm framework, combining the global exploration ability of GA with the exploitation power of a local search strategy. This innovative approach enhances the algorithm's capability to explore a wide search space and effectively handle large-scale problems. As a result, memetic algorithms find wide applications across various domains, including engineering, economics, and computer science[17–22].

In this study, we introduced an adaptive memetic algorithm with dual-level local search strategies (DL-AMA), addressing both task assignment and route planning levels. The algorithm, outlined in Algorithm 1, is primarily implemented through the following six steps:

(1) Initialization: randomly generate Pop_size number of solutions with values range from 1 to N;

(2) Selection: choose outstanding individuals as parents through the tournament selection[23];

(3) Genetic operation: two-point crossover strategy[24] is adapted in this paper. Firstly, two integers

(1) Solution of continuous algorithms

Point	1	2	3	4	5	6	7	8
Value	1.6	1.2	2.4	1.9	1.7	1.1	2.2	1.5

(2) Assign by round value

Robot 1

Point	1	2	4	5	6	8
Round value	1	1	1	1	1	1

Robot 2

Point	3	7
Round value	2	2

(3) Sort by original value

Robot 1

Original value	1.1	1.2	1.5	1.6	1.7	1.9
Path	6	2	8	1	5	4

Robot 2

Original value	2.2	2.4
Path	7	3

Fig. 2 Encoding strategy for multi-robot surveillance.

Algorithm 1 Framework of DL-AMA

Input: Population size: Pop_size; max generation: Max_gen; number of robots: N; solution dimension (number of surveillance targets): M; crossover probability: Pc

Output: Surveillance paths of all robots: $\{l_1, l_2,..., l_N\}$

```
Initialize Population Pop
for gen = 1 to Max_gen do
    for i = 1 to Pop_size/2 do
        dad, mom = Tournament_selection(Pop)
        if random() < P_c then
            /* Two-point crossover */
            cut1, cut2 = random_int(1, M−1)
            child1 = concatenate(dad(0:cut1),
              mom(cut1:cut2), dad(cut2:M))
            child2 = concatenate(mom(0:cut1),
              dad(cut1:cut2), mom(cut2:M))
        end
        child ← child1,child2
    end
    for j = 1 to Pop_size do
        tm = task_level_local_search(child(j,:),P_tm)
        pm(j,:) = path_level_local_search(tm, P_pm)
    end
    Pop = Population_Update(pm, Pop)
    P_tm, P_pm = Adaptive_scheme(gen, Max_gen)
end
```

cut1 and cut2 are randomly generated from the range of 1 to $M-1$ as two intersection points, and then the fragments of cut1 to cut2 from the father and mother are exchanged with each other, while the rest remains unchanged;

(4) Task-level local search: The task assignment level local search is as shown in Algorithm 2, for each dimension of the solution vector, a probability P_{tm} exists to randomly generate a new value within the range of 1 to N. If the newly generated random number differs from the original value after floor rounding, the corresponding surveillance target allocation will change. This process leads to a reassignment of surveillance targets among the robots, aiming to explore more potential surveillance target allocation schemes. An example operation is shown in Fig. 3, where the value 1.7 in the 5-th position of the solution vector is replaced by a randomly generated new value 2.3, consequently, the corresponding surveillance target is reassigned from being allocated to Robot 1 to being allocated to Robot 2.

(5) Path-level local search: The path planning level local search is as shown in Algorithm 3. The local search strategy begins by decoding the solution vector to obtain the surveillance paths of each robot. Then, for each path l_i, there is a probability P_{pm} to generate two random integers cut1 and cut2 from 1 to $|l_i|$, where $|l_i|$ is the number of targets in the surveillance path of the robot. Subsequently, the sequence between positions cut1 and cut2 in the solution vector is reversed, resulting in a new visiting sequence. This process aims to explore better surveillance paths for each robot without altering the surveillance point allocation scheme. An example is illustrated in Fig. 4, the path of Robot 1 is mutated, where the sequence between Position 2 and Position 5 is reversed, after the local search operation, the surveillance path of Robot 1 is changed from {6-2-8-1-4} to {6-4-1-8-2};

Algorithm 2 Task-level local search strategy

Input: Child solution after crossover: child; number of robots: N; solution dimension (number of surveillance targets): M
Output: Operated solution tm

for i =1 to M **do**
  if random() < P_{tm} **then**
    tm(i) = random_int(1, N)
  else
    tm(i) = child(i)
  end
end

Algorithm 3 Path-level local search strategy

Input: Solution after task level local search: tm; number of robots: N; solution dimension (number of surveillance targets): M
Output: Operated solution pm

$l_1, l_2, \ldots, l_N$ = **decoding**(tm)
for i =1 to N **do**
  if random() < P_{pm} **then**
    cut1, cut2 = **random_int**(1, $|l_i|$)
    invert_path = **invert**(tm(cut1, cut2))
    l_i' = **concatenate**(l_i(0:cut1), invert_path, l_i(cut2:))
  else
    $l_i' = l_i$
  end
end
pm = **encoding**($l_1', l_2', \ldots, l_N'$)

(6) Population update: The survivor process selects the worst individual out of k-way individuals (tournament selection) and compares it with child individuals. The better individual will be kept for the next generation.

(7) Adaptive scheme: The above local search probability significantly influences the final solution. In the task assignment level local search strategy, the local search probability should be increased in the early stages to explore more promising task assignment

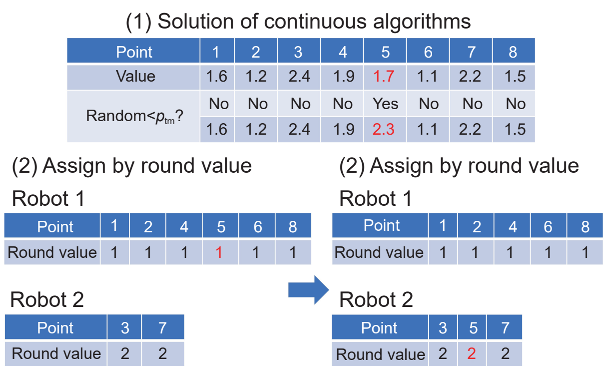


Fig. 3 Task assignment level local search.

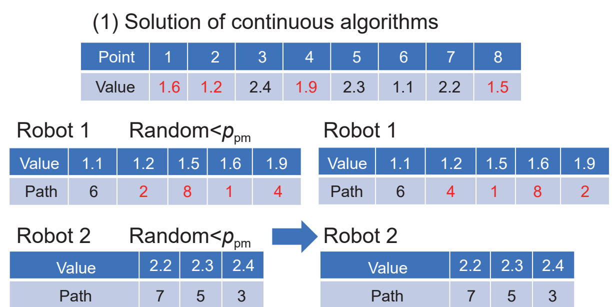


Fig. 4 Path planning level local search.

schemes. However, in the later stages, the local search probability should be reduced to provide a stable assignment plan for route planning. Similarly, the local search probability at the route planning level should gradually increase during the iteration process to enhance the local paths for each robot. This approach aims to strike a balance between exploration and exploitation, allowing the algorithm to find better solutions by exploring different possibilities while exploiting the gained knowledge during the optimization process. The variation process of the two local search probabilities with the iterations is shown in Eqs. (13) and (14), respectively.

$$P_{\mathrm{tm}} = 0.05 - 0.04 \times \sqrt{\frac{\mathrm{gen}}{\mathrm{max\_gen}}} \tag{13}$$

$$P_{\mathrm{pm}} = 0.01 + 0.05 \times \left(\frac{\mathrm{gen}}{\mathrm{max\_gen}}\right)^2 \tag{14}$$

4 Numerical Experiment

In this section, we will verify the effectiveness of the proposed method through comparative experiments with other methods on different numbers of surveillance targets and robot scales.

4.1 Experimental setup

For the experimental setup for multi-robot surveillance scenarios, we selected three problem instances labeled eil51, eil76, and eil101 from the TSPLIB[25] database. These instances serve as the basis for our multi-robot surveillance environments. The number of surveillance target points in three instances are 51, 72, and 102, respectively. All instances share a common scene size of 70 m × 70 m. Moreover, we conducted tests with different scales of robot teams to perform the surveillance task, including 2, 4, 6, and 8 robots. Regarding the algorithm configurations, we opted for two state-of-the-art algorithms: the physics-based algorithm equilibrium optimizer (EO)[26] and the swarm-based algorithm harris hawks optimization (HHO)[27]. Both algorithms, introduced in 2020 and 2019, respectively, have demonstrated superior performance and attracted considerable attention and research. Additionally, we also introduced two classical swarm-based algorithms, PSO[28] and grey wolf optimization (GWO)[29] for comparative purposes.

As for the parameter settings, the objective weights and task assignment and route planning were set as 0.2 and 0.8, respectively. The task load difference threshold, T_h, was configured to 0.5. In addition, the crossover probability P_c in our proposed algorithm was set at 0.95, and the adaptive local search probability followed the formulation in Eq. (13). All the comparative algorithms were executed using the Mealpy library[30] with default parameters. The Python module Mealpy incorporates various population-based meta-heuristic techniques. Furthermore, all algorithms employ a population size of 200 and a maximum iteration count of 500, and each problem instance was independently run for 20 repetitions to ensure the robustness and reliability of the results.

4.2 Experimental result

We initially evaluated our proposed methods with different numbers of surveillance targets. Figure 5 shows the convergence curves of each algorithm in scenarios where the robot team size is 4 and the number of surveillance targets is 51, 76, and 101, respectively. From Fig. 5, it is evident that our method outperforms others, notably surpassing the second-ranked EO algorithm. Specifically, our proposed method excels in obtaining superior surveillance target allocation schemes early in the algorithm iteration. It further optimizes the local surveillance paths of each robot in the later stages of iteration, ultimately converging to a significantly low-cost function value for the multi-robot surveillance task.

To delve deeper into the performance of the proposed method concerning three performance metrics—task energy consumption, task completion time, and task allocation uniformity—we also recorded the total length of surveillance paths (Total), the maximum surveillance path length (Max), and the variance of surveillance path lengths among robots (Std). The results are displayed in Table 1 (Note that the best results among these methods are marked in boldface). From Table 1, it is apparent that our proposed method outperforms other algorithms in terms of energy consumption, task completion time, and task allocation uniformity when solving multi-robot surveillance tasks. Furthermore, as the number of surveillance task points increases while maintaining a constant number of robots, both the total length of surveillance paths and the maximum surveillance path length among robots (i.e., task completion time) also increase. Although the difference in task load between different robots shows an upward trend, the increase is relatively gradual.

Moreover, we conducted tests to assess the performance of the proposed method under the same number of surveillance points with varying robot team sizes. Figure 6 illustrates the comparative results of

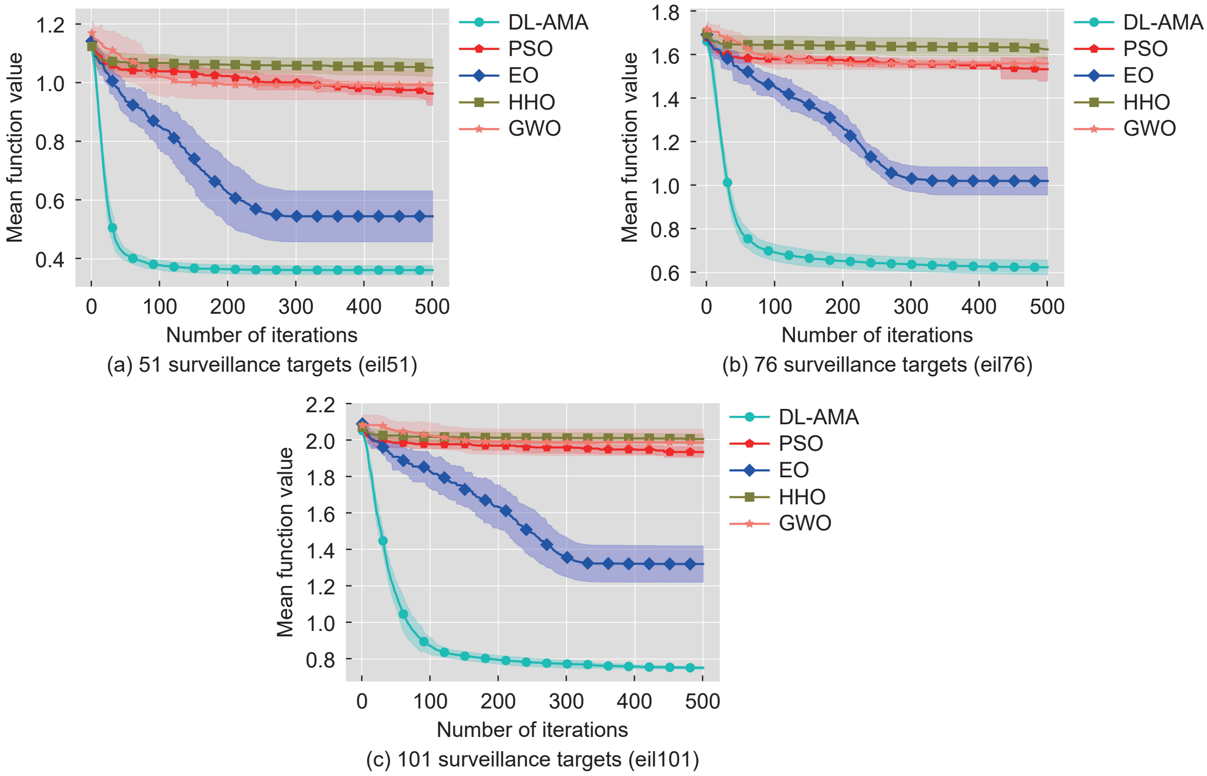


Fig. 5 Mean convergence curves of the compared algorithms.

Table 1 Performance of different methods on three metrics (Total: Sum length of all the robot surveillance paths; Max: Maximum length among all surveillance paths; Std: Standard deviation of all the surveillance paths).

Instance	Method	Total (m)	Max (m)	Std (m)
eil51	PSO	1377.16	383.11	34.24
	EO	700.19	207.21	24.65
	HHO	1303.59	350.39	22.80
	GWO	1241.11	340.494	24.90
	DL-AMA	**463.22**	**125.85**	**9.22**
eil76	PSO	2248.62	634.695	50.05
	EO	1234.04	366.056	41.54
	HHO	1792.01	571.978	44.3
	GWO	1794.21	538.94	34.43
	DL-AMA	**620.07**	**165.92**	**9.04**
eil101	PSO	3142.63	849.4	51.91
	EO	1745.96	502.86	45.01
	HHO	2799.43	793.97	54.07
	GWO	2493.61	816.83	61.71
	DL-AMA	**771.53**	**209.81**	**14.12**

each algorithm across three performance indicators when the number of surveillance targets is fixed at 51, and the robot team size varies between 2, 6, and 8. From Fig. 6, it is evident that, in comparison to other methods, our proposed method consistently achieves the surveillance target allocation scheme with the lowest task load difference across different robot team sizes. Additionally, our method demonstrates superiority in terms of energy consumption and task completion time, as evidenced by achieving both the shortest total surveillance path length and the shortest maximum surveillance path length simultaneously. Furthermore, it was observed that with a constant number of surveillance targets, as the size of the robot team increases, the total surveillance path length and task load difference among robots exhibit a slight increase, while the maximum surveillance path length gradually decreases. This trend is attributed to the increasing robot team size, the average number of surveillance targets allocated to each robot decreases. Consequently, both the task load and surveillance path length for each robot decrease. This collaborative approach among multiple robots enhances efficiency and reduces task completion time.

4.3 Effectiveness of dual-level local search and adaptive scheme

To validate the effectiveness of the dual-level local search strategy and adaptive probability scheme, we conducted ablation experiments. We compared our proposed adaptive dual-level local search strategy-based memetic algorithm (referred to as DL-AMA) with an adaptive MA lacking the task assignment level local search strategy (referred to as No-tm), an adaptive

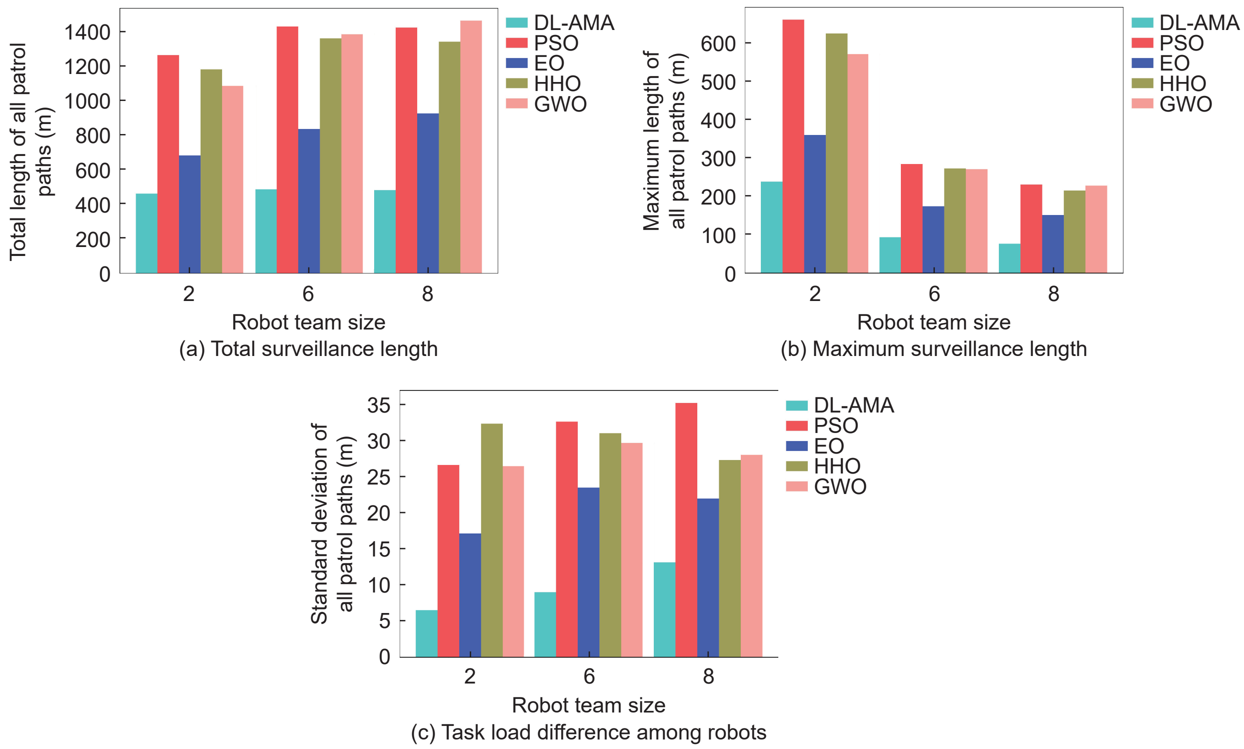


Fig. 6 Performance of the compared algorithms when a fixed number of targets surveilled by different robot team sizes.

MA without path planning level local search strategy (referred to as No-pm), and the dual-level MA without an adaptive scheme (referred to as DL-MA, with the local search probability of both set to 0.025). The comparative results for multi-robot surveillance scenarios involving four robots surveilling 51, 76, and 101 target points are presented in Table 2 (Note that the best results among these methods are marked in boldface).

From Table 2, it can be seen that the MA with both a dual-level local search strategy and an adaptive scheme performs the best in task allocation uniformity, energy consumption, and task completion time. Among them, the adaptive MA without task assignment level local search strategy exhibits the poorest performance, and this gap further expands as the number of surveillance targets increases. This highlights that task assignment plays a crucial role in multi-robot surveillance. If there is no excellent global allocation scheme in advance, even the best local path planning can achieve poor comprehensive performance. The importance of task allocation becomes more prominent when the task scale expands, i.e., when the number of surveillance targets or robot team size expands. In addition, we also noticed that the absence of an adaptive local search probability strategy can lead to a decrease in the quality of planning results. As we mentioned before, there are

Table 2 Performance of proposed method with different strategies on three metrics (Total: Sum length of all the robot surveillance paths; Max: Maximum length among all surveillance paths; Std: Standard deviation of all the surveillance paths).

Instance	Method	Total (m)	Max (m)	Std (m)
eil51	No-tm	775.39	217.95	17.27
	No-pm	469.65	126.51	**7.69**
	DL-MA	464.23	126.47	8.34
	DL-AMA	**463.22**	**125.85**	9.22
eil76	No-tm	1131.57	310.5	21.49
	No-pm	634.12	172.79	11.43
	DL-MA	629.84	173.10	10.70
	DL-AMA	**620.07**	**165.92**	**9.04**
eil101	No-tm	1386.04	386.64	30.34
	No-pm	787.12	214.80	**13.75**
	DL-MA	848.64	234.65	18.62
	DL-AMA	**771.53**	**209.81**	14.12

fluctuations in the importance of task allocation and path planning throughout the optimization process for multi-robot surveillance. In the early stages, global task assignment holds more importance, while in the later stages, local path planning becomes more crucial. This emphasizes the need to quickly acquire suitable surveillance target allocation in the initial phase to establish a stable global task assignment environment and enhance the quality of local route planning in the later stages of optimization.

Furthermore, we conducted a Friedman test on these results to gain a more distinct understanding of the comprehensive performance of each strategy. As depicted in Fig. 7, the results align with the aforementioned findings. The adaptive MA with both strategies consistently achieves the lowest overall rank and the best performance. It is followed by the adaptive MA without path planning-level local search strategy, while the adaptive MA lacking task assignment-level local search strategy ranks the highest and exhibits the poorest performance.

5 Simulation Experiment

To verify the effectiveness of DL-AMA under the practical motion control of the robot, we further conducted simulation experiments. Figures 8a and 8b show the robot simulation model and the simulated surveillance environment, respectively. The robot model is the Agilex robot of type Huntese 2.0, equipped with Livox solid-state lidar, RealSense depth camera, and a vision-enhanced fusion sensor RTK containing an embedded camera and IMU, which can continuously provide high-precision positioning information even in environments without GNSS signals for a long time. The simulation environment is built in Gazebo, with 51 surveillance targets statically distributed in a 35×35 m^2 grid, similar to eil51. The control, navigation, and communication modules for the robot were based on the robotic operating system, and the maximum velocity for all robots was set to 2 m/s.

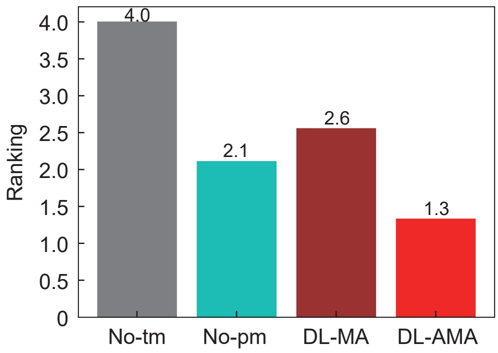
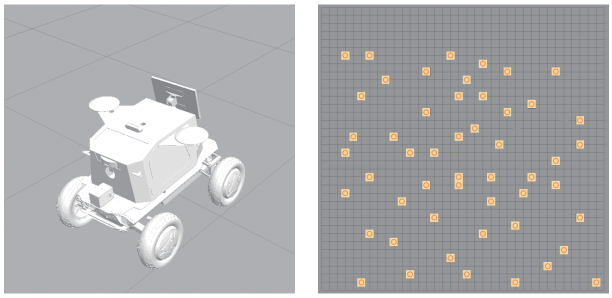


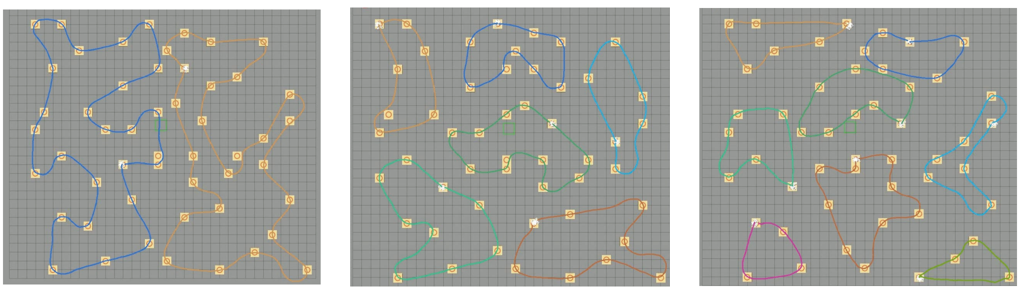
Fig. 7 Average rankings of the proposed method with different strategies by the Friedman test (The lower the ranking, the better the performance).



(a) Robot model (b) Simulated surveillance environment

Fig. 8 Simulation case built in Gazebo.

We conducted simulation tests on scenarios with robot team sizes of 2, 6, and 8. The simulation video can be seen on https://youtu.be/X9EebjDKemc. The final surveillance trajectories of each robot are shown in Fig. 9. It can be observed that for different robot team sizes, the established multi-robot surveillance optimization model can evenly allocate non-overlapping surveillance targets to each robot. This allocation strategy effectively mitigates task load differences and minimizes mutual interference among robots. Furthermore, in terms of path planning, the model successfully generates concise surveillance paths for each robot to traverse their assigned targets, contributing to the reduction of energy consumption and task completion time. In addition, Table 3 also records information about trajectory lengths and task completion time. From the results, as the number of robots performing the surveillance task increases from 2 to 8, the maximum trajectory length gradually decreases. With the speed remaining constant, the task



(a) 2 robots (b) 6 robots (c) 8 robots

Fig. 9 Simulated trajectories for 51 target points patrolled by different robot team sizes.

Table 3 Simulation results on trajectory and completion time.

Team size	Total (m)	Max (m)	Std (m)	Completion time (s)
2	247.16	126.22	2.64	110
6	258.94	50.65	5.99	48
8	262.75	42.49	6.33	40

completion time also gradually decreases, corresponding to a decrease from 110 s to 40 s in this simulation.

6 Conclusion

In this study, we delved into the challenges posed by multi-robot surveillance. To tackle the fundamental issues of task assignment and path planning in multi-robot surveillance, we initially devised a mathematical optimization model that carefully considers the objectives and constraints associated with both task assignment and path planning. Subsequently, we implemented an encoding strategy, enabling a continuous evolutionary algorithm to concurrently address the task assignment and path planning problems. This encoding approach entails rounding and sorting operations on the continuous-valued solution vector, providing essential information for both task assignment and path planning. In conclusion, to effectively address the multi-robot surveillance problem, we introduced an adaptive memetic algorithm founded on a dual-level local search strategy. Through comparative experiments involving various robot team sizes and numbers of surveillance targets, ablation experiments exploring the proposed strategies, as well as simulated experiments, we validated the feasibility and efficacy of our approach.

In future research endeavors, we aim to explore more intricate multi-robot surveillance problems and enhance the efficiency and solution quality, especially in scenarios involving larger surveillance target scales.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 52105244) and Entrepreneurship and Innovation Support Plan of Chongqing for Returned Overseas Scholars (No. cx2023085).

References

[1] D. Shishika, J. Paulos, and V. Kumar, Cooperative team strategies for multi-player perimeter-defense games, *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2738–2745, 2020.

[2] L. Lin and M. A. Goodrich, UAV intelligent path planning for wilderness search and rescue, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 709–714.

[3] P. Ghassemi, D. DePauw, and S. Chowdhury, Decentralized dynamic task allocation in swarm robotic systems for disaster response: Extended abstract, in *Proc. Int. Symp. on Multi-Robot and Multi-Agent Systems* (*MRS*), New Brunswick, NJ, USA, 2019, pp. 83–85.

[4] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, Monitoring and mapping with robot swarms for agricultural applications, in *Proc. 14th IEEE Int. Conf. Advanced Video and Signal Based Surveillance* (*AVSS*), Lecce, Italy, 2017, pp. 1–6.

[5] R. N. Haksar, S. Trimpe, and M. Schwager, Spatial scheduling of informative meetings for multi-agent persistent coverage, *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3027–3034, 2020.

[6] Z. Zhang, J. Yu, J. Tang, Y. Xu, and Y. Wang, MR-TopoMap: Multi-robot exploration based on topological map in communication restricted environment, *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10794–10801, 2022.

[7] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, A clustering-based coverage path planning method for autonomous heterogeneous UAVs, *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25546–25556, 2022.

[8] J. Tang and H. Ma, Mixed integer programming for time-optimal multi-robot coverage path planning with efficient heuristics, arXiv preprint arXiv: 2306.17609, 2023.

[9] J. Xie, R. Zhou, J. Luo, Y. Peng, Y. Liu, S. Xie, and H. Pu, Hybrid partition-based patrolling scheme for maritime area patrol with multiple cooperative unmanned surface vehicles, *J. Mar. Sci. Eng.*, vol. 8, no. 11, p. 936, 2020.

[10] S. W. Feng, T. Guo, and J. Yu, Optimal allocation of many robot guards for sweep-line coverage, arXiv preprint arXiv: 2302.04319, 2023.

[11] H. Choset, Coverage of known spaces: The boustrophedon cellular decomposition, *Auton. Rob.*, vol. 9, no. 3, pp. 247–253, 2000.

[12] A. Din, M. Y. Ismail, B. Shah, M. Babar, F. Ali, and S. U. Baig, A deep reinforcement learning-based multi-agent area coverage control for smart agriculture, *Comput. Electr. Eng.*, vol. 101, p. 108089, 2022.

[13] M. Jana, L. Vachhani, and A. Sinha, A deep reinforcement learning approach for multi-agent mobile robot patrolling, *Int. J. Intell. Robot. Appl.*, vol. 6, no. 4, pp. 724–745, 2022.

[14] C. Tong, A. Harwood, M. A. Rodriguez, and R. O. Sinnott, An energy-aware and fault-tolerant deep reinforcement learning based approach for multi-agent patrolling problems, arXiv preprint arXiv: 2212.08230, 2022.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: The MIT Press, 1992.

[16] P. Moscato and M. G. Norman, A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems, *Parallel Computing and Transputer Applications*, vol. 1, pp. 177–186, 1992.

[17] L. Wei, L. Xing, Q. Wan, Y. Song, and Y. Chen, A multi-objective memetic approach for time-dependent agile earth observation satellite scheduling problem, *Comput. Ind. Eng.*, vol. 159, p. 107530, 2021.

[18] N. R. Sabar, A. Bhaskar, E. Chung, A. Turky, and A. Song, An adaptive memetic approach for heterogeneous vehicle routing problems with two-dimensional loading constraints, *Swarm Evol. Comput.*, vol. 58, p. 100730, 2020.

[19] E. Yağmur and S. E. Kesen, Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches, *Comput. Ind. Eng.*, vol. 161, p. 107649, 2021.

[20] A. B. Krishna, S. Saxena, and V. K. Kamboj, hSMA-PS: A novel memetic approach for numerical and engineering design challenges, *Eng. Comput.*, vol. 38, no. 4, pp. 3513–3547, 2022.

[21] L. Beke, L. Uribe, A. Lara, C. A. C. Coello, M. Weiszer, E. K. Burke, and J. Chen, Routing and scheduling in multigraphs with time constraints—A memetic approach for airport ground movement, *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 474–488, 2024.

[22] A. C. Cinar, A novel adaptive memetic binary optimization algorithm for feature selection, *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13463–13520, 2023.

[23] B. L. Miller and D. E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, *Complex Systems*, vol. 9, no. 3, pp. 193–212, 1995.

[24] H. Ouerfelli and A. Dammak, The genetic algorithm with two point crossover to solve the resource-constrained project scheduling problems, in *Proc. 5th Int. Conf. Modeling, Simulation and Applied Optimization* (*ICMSAO*), Hammamet, Tunisia, 2013, pp. 1–4.

[25] G. Reinelt, TSPLIB—A traveling salesman problem library, *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, 1991.

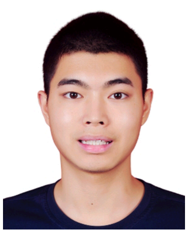
[26] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, *Knowledge-Based Systems*, vol. 191, p. 105190, 2020.

[27] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, Harris Hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019.

[28] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. ICNN'95—Int. Conf. Neural Networks*, Perth, WA, Australia, 1995, pp. 1942–1948.

[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

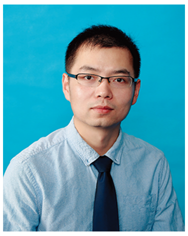
[30] N. Van Thieu and S. Mirjalili, MEALPY: An open-source library for latest meta-heuristic algorithms in Python, *J. Syst. Archit.*, vol. 139, p. 102871, 2023.



Hao Cheng received the BEng degree in mechanical design, manufacturing, and automation from Chongqing University, China in 2016. He is currently pursuing the PhD degree at Chongqing University, China. His main research interests include evolutionary algorithms and their applications in task scheduling for intelligent unmanned systems.

Wei Xia received the PhD degree in civil and environmental engineering from National University of Singapore, Singapore in 2020. He is a postdoctoral researcher at the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Italy. His research interests are focused on the development of efficient optimization and simulation methods for the design and management of large systems.

Jun Luo received the BS and MS degrees in mechanical engineering from Henan Polytechnic University, Jiaozuo, China in 1994 and 1997, respectively, and the PhD degree from Shanghai Jiao Tong University, Shanghai, China in 2000. He is currently a professor at the State Key Laboratory of Mechanical Transmission for Advanced Equipment, Chongqing University, China. His research areas include intelligent unmanned systems planning and control.



Jin Yi received the BS and PhD degrees in industrial engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2012 and 2017, respectively. He was a research fellow at the Department of Industrial System Engineering and Management, National University of Singapore, Singapore from 2017 to 2020. He is currently an associate professor at the State Key Laboratory of Mechanical Transmission for Advanced Equipment, Chongqing University, China. His research interests include evolutionary algorithms and their applications in task assignment and planning for intelligent unmanned systems.

Huayan Pu received the MSc and PhD degrees in mechatronics engineering from Huazhong University of Science and Technology, Wuhan, China in 2007 and 2011, respectively. She is currently a professor at the State Key Laboratory of Mechanical Transmission for Advanced Equipment, Chongqing University, China. Her current research interests include artificial intelligence and intelligent unmanned systems planning and control.