Ye et al.
EURASIP Journal on Advances in Signal Processing (2024) 2024:21
https://doi.org/10.1186/s13634-024-01116-4

EURASIP Journal on Advances
in Signal Processing

RESEARCH

Open Access

Offline and online task allocation algorithms for multiple UAVs in wireless sensor networks

Liang Ye[1*], Yu Yang[1], Weixiao Meng[1], Xuanli Wu[1], Xiaoshuai Li[2] and Rangang Zhu[2]

*Correspondence:
yeliang@hit.edu.cn

[1] Communication Research Center, Harbin Institute of Technology, Harbin 150001, China
[2] School of Electronic Countermeasures, National University of Defense Technology, Hefei 230031, China

Abstract

In recent years, UAV techniques are developing very fast, and UAVs are becoming more and more popular in both civilian and military fields. An important application of UAVs is rescue and disaster relief. In post-earthquake evaluation scenes where it is difficult or dangerous for human to reach, UAVs and sensors can form a wireless sensor network and collect environmental information. In such application scenarios, task allocation algorithms are important for UAVs to collect data efficiently. This paper firstly proposes an improved immune multi-agent algorithm for the offline task allocation stage. The proposed algorithm provides higher accuracy and convergence performance by improving the optimization operation. Then, this paper proposes an improved adaptive discrete cuckoo algorithm for the online task reallocation stage. By introducing adaptive step size transformation and appropriate local optimization operator, the speed of convergence is accelerated, making it suitable for real-time online task reallocation. Simulation results have proved the effectiveness of the proposed task allocation algorithms.

Keywords: Improved immune multi-agent algorithm, Improved adaptive discrete cuckoo algorithm, Task allocation, UAV

1 Introduction

In recent years, UAV (Unmanned Aerial Vehicle) [1] techniques are developing very fast, and UAVs are becoming more and more popular due to their strong maneuverability. When performing tasks in hazardous areas, they do not rely on onboard personnel. They not only are convenient and intelligent to use, but also reduce pilot training costs. Therefore, they are widely used in various application scenarios. With the increasing popularity of UAV applications [2], research issues such as autonomous algorithms, collaborative planning algorithms, and intelligent control algorithms [3] for UAVs are becoming increasingly prominent. Especially the ability of UAVs to perform tasks in dry, harsh or dangerous environments for humans has attracted more and more scientific researchers to conduct research on UAV flight formation algorithms, task allocation algorithms, path planning algorithms, and efficiency evaluation methods. UAVs not only play an increasingly important role in the military field, but also have a high demand in civilian field such as wireless sensor networks. Especially in today's increasingly frequent occurrence of various disasters, UAVs are increasingly being used to monitor disaster scenes, such as mountain fire monitoring, maritime emergency rescue, and medical material transportation.

The main scenario considered in this paper is that after a severe earthquake, the early warning measurement center needs to collect environmental information such as water quality and pressure. Such information is very important for analyzing the disaster situation and giving early warning of possible secondary disasters such as forming of barrier lakes and aftershocks. Due to the threat of aftershocks and the impact of landforms being destroyed, it is not feasible to manually collect information using land transportation facilities. Utilizing the high mobility of UAVs, the early warning measurement center can send a UAV cluster to collect data from distributed or pre-set sensors at task target points. Due to the cluster distribution of wireless sensor nodes, when earthquakes occur on a large-scale, it is difficult to network between each group of sensor nodes. At the same time, there is no support from communication facilities such as base stations, which makes it impossible to gather all information from various locations. In another word, there are far more locations in a large range that require data collection tasks than UAVs. By allocating tasks and planning UAV flight paths reasonably, data collection tasks within the entire earthquake area can be efficiently completed.

Nowadays, there are many researchers studying on task allocation of UAVs. UAV task allocation problem belongs to the task planning problem of multi-agent systems. In multi-agent systems, it is quite difficult to find the optimal or near optimal solution of the task allocation problem. In general, it is proved to be NP-hard, and it is difficult to solve it with conventional numerical methods. The main goal of task allocation, besides achieving overall optimal system performance, can also be to minimize the execution time of tasks, minimize the time of certain agent activities, maximize the number of tasks completed within a specific time, and maximize the robustness of the task allocation process, that is, to ensure the completion of tasks [4]. Due to the fact that optimal overall performance is a vague concept that is difficult to quantify and may depend on each agent, the concept of utility is generally used to estimate the value or cost of task allocation processes on system performance. Yoon, et al. [5] introduced the concept of virtual task to reduce the complexity of the problem, and introduced the Hungarian algorithm to decompose the virtual character to calculate an approximately optimal path. Itshak, et al. [6] proposed an improved bee colony optimization algorithm that utilized distributed swarm intelligence methods to allocate fixed heterogeneous sensors to upcoming unknown tasks to minimize task detection time. They assigned sensors to tasks based on their performance, task priority, and the distance between the sensor and the location where the task was executed. Based on the classic fixed response threshold model, Wu, et al. [7] adopted a bottom-up approach to design new dynamic environmental stimuli, response thresholds, and transfer probabilities under the concept of problem centered and evolutionary solving. A dynamic ant colony labor division model was proposed, which allowed a group of relatively low intelligent agents to perform complex tasks. It had the characteristics of distributed framework, multi-task execution sequence, multi-state, adaptive response threshold, and multi-individual response.

For UAV path planning algorithms, many researchers in this field have proposed effective and feasible planning algorithms that belong to various branches, and have proposed some improved intelligent algorithms based on them. Zeng, et al. [8] first proposed to use the block coordinate descent algorithm to decompose the problem, and use the continuous convex optimization algorithm to obtain the optimal solution of the joint optimization problem of UAV communication resources and track. Later, this idea was successfully applied to various other scenarios, such as UAV communication path planning based on energy efficiency maximization [9], multiple UAV downlink broadcast communication [10], UAV communication physical layer security design [11], UAV mobile edge computing [12], wireless power transmission supported by UAVs [13] and wireless energy carrying communication [14]. It is worth noting that one drawback of alternately updating UAV paths and allocating communication resources is that if the initialization design is improper, it may fall into suboptimal local optima. Therefore, Shen, et al. [15] studied the simultaneous updating of these two variables in certain scenarios by developing new concave lower bound functions, and also introduced the use of alternating direction multiplier techniques to reduce the computational complexity of multi-UAV path design. Liang et al. [16] studied the placement and flight optimization of multiple UAVs for uplink coordination and multi-point communication. Each UAV forwards its signals received from all ground users to the central processor for joint decoding. Due to the limited flight speed of UAVs, utilizing their maneuverability to improve communication performance is generally the most suitable application scenario for delay tolerance. In fact, for UAV platforms serving multiple users, there is a new compromise between communication throughput and access delay. Lyu [17] first studied UAVs with fixed flight trajectories, and then Wu [18] expanded them through joint optimization of UAV trajectories and communication resource allocation in OFDMA (Orthogonal Frequency-Division Multiple Access) systems.

This paper firstly proposes an improved immune multi-agent algorithm for the offline task allocation stage. The proposed algorithm provides higher accuracy and convergence performance by improving the optimization operation. Then, this paper proposes an improved adaptive discrete cuckoo algorithm for the online task reallocation stage. By introducing adaptive step size transformation and appropriate local optimization operator, the speed of convergence is accelerated, making it suitable for real-time online task reallocation. The efficiency of the proposed algorithms is finally proved by numerical simulations.

2 Methods

2.1 Task allocation model

The goal of multi-UAV task allocation [19] is to assign a set of target point sequences to all UAVs under the condition of known UAV formation groups, target point information, and partial environmental information, in order to complete all tasks with the minimum cost and achieve the optimal efficiency.

The UAV clusters in this paper meet the following conditions:

1. Isomorphism: All the UAVs have the same structure and function;
2. Communication guarantee: All the UAVs have established communication links through flight Ad hoc networks;
3. Functionality: All the UAVs are equipped with relevant equipment for data collection.

Assume that the number of UAVs is N, and the set of UAVs is $U=\{U_1, U_2, \ldots, U_N\}$. The number of target points to traverse is M, and the set of tasks is $T=\{T_1, T_2, \ldots, T_M\}$. The task allocation problem meets the following principles:

1. All the UAVs have the same resource and constraints;
2. A single target point can only be accessed once;
3. The UAVs visit all target points with the shortest possible total flight distance;
4. The task allocation of all the UAVs should be as fair as possible;
5. UAVs should try to avoid threat areas as much as possible.

The task allocation problem is a combinatorial optimization problem which meets the following constraints:

1. Maximum flight distance constraint The maximum flight distance of a single UAV is limited by the amount of fuel carried. Assume that the maximum flight distance is $D_{\max}$, and the route of the ith UAV is L_i. Then, the maximum flight distance constraint is expressed as,

$$D(L_i) \leq D_{\max}, \quad \forall i = 1, \ldots, N \tag{1}$$

2. Target traversal constraint All the target points should be assigned only once, and this constraint is expressed as,

$$\sum_{i=1}^{N} \sum_{j=1}^{M_i} T_i^j = T \&\& \sum_{j=1}^{M_i} T_i^j \cap \sum_{j=1}^{M_k} T_k^j = \emptyset, \quad \forall i, \quad j \in N, \quad i \neq j \tag{2}$$

where T_i^j is the jth target point that the ith UAV will visit.

In dynamic scenarios, task allocation should consider the total length of the UAV cluster's execution path, the flight load of each UAV cluster, and the penalty for passing through threat zones. Considering these factors, the fitness function is expressed as,

$$Fitness = \alpha \sum_{i=1}^{N} D(L_i) + \beta \max\left(D(L_i)\right) + \gamma \sum_{i=1}^{N} D_o(L_i) \tag{3}$$

where $\sum_{i=1}^{N} D(L_i)$ is the total length of all the UAVs, $\max\left(D(L_i)\right)$ is the maximum flight distance, and $D_o(L_i)$ is the penalty path for UAVs flying through threat zones. α, β, γ are three weighting coefficients and,

$$\alpha + \beta + \gamma = 1 \tag{4}$$

By minimizing the fitness function, an optimal solution can be obtained for the comprehensive evaluation of the total distance, the task balance, and the flight safety. The final optimization problem is expressed as,

$$\begin{aligned} \min \quad & Fitness \\ s.t. \quad & D(L_i) \leq D_{\max}, \quad \forall i = 1, \ldots, N \\ & \sum_{i=1}^{N} \sum_{j=1}^{M_i} T_i^j = T \&\& \sum_{j=1}^{M_i} T_i^j \cap \sum_{j=1}^{M_k} T_k^j = \emptyset, \quad \forall i, \quad j \in N, \quad i \neq j \end{aligned} \tag{5}$$

2.2 Improved immune multi-agent algorithm for offline task allocation

Based on the concept of artificial immune algorithm [20] and multi-agent system, we propose an improved immune multi-agent algorithm for the offline task allocation stage. By limiting some operations within the neighborhood, the ability of traditional immune algorithms to jump out of local optima is improved, while maintaining the fast convergence speed of traditional immune algorithms in the initial stage. It is also suitable for dynamic and static task allocation issues.

2.2.1 Memory population initialization

The immunological memory operation refers to the concept of memory population in the clonal selection algorithm, and is used to save the optimal part of agents in each evolution process. The size of the memory agent group is limited, so it is necessary to update and eliminate it in real-time during each iteration. The specific steps are as follows:

1. Assuming that the size of the initial memory agent group is N, select k antibody agents with the best fitness in the current antibody agent grid, and try to add them to the memory agent group. If the memory agent group has not reached the pre-set size, this addition is allowed, and the k antibody agents are directly added to the agent group.

2. If the memory agent group is full, or the number of antibody agents that can be accommodated is less than k, the elimination operation is performed. All antibody agents in the current memory agent group are compared with the new k antibody agents for fitness, and the antibody agents with the lowest fitness that exceed the size of the memory agent group are removed from the memory agent group.

The overall process can be expressed as,

$$N_{best} = SelectN(N_{best}, L) \tag{6}$$

where N_{best} is the memory agent group, $SelectN(\cdot)$ means to select the first N agent functions with the best fitness.

2.2.2 Neighborhood cloning operation

The neighborhood cloning operation refers to the antibody cloning operation in the clonal selection algorithm. The antibody cloning operation in the clonal selection algorithm only multiplies the best few antibodies in the population and then mutates, so it is not suitable for maintaining the diversity of the population. Therefore, this paper keeps the cloning process in the neighborhood. For each antibody agent, we select the agent with the best fitness in its neighborhood and its own several agents, multiply and mutate it to a certain extent, and then select the antibody agent with the best fitness to replace the original antibody agent. The neighborhood cloning operation is represented as,

$$A_{i,j} = Select\big(Mutation\big(Clone\big(Select\big(\{A_{i,j}, Neigh.A_{i,j},\}\big)\big)\big)\big) \tag{7}$$

where $Select(\cdot)$ represents selecting the best agent, $Mutation(\cdot)$ represents the mutation operation, and $Clone(\cdot)$ represents the cloning operation.

2.2.3 Neighborhood suppression operation

The neighborhood suppression operator suppresses high affinity antibodies to maintain population diversity. Since the cloning operation in this paper is carried out in the neighborhood, if an antibody agent is the local optimal agent, its excellent fitness will make it the parent of all antibody agent clones in its neighborhood, that is, the antibodies in a small range all preserve the characteristics of the local optimal agent, and the affinity is naturally high, which needs to be suppressed. Therefore, within each neighborhood, neighborhood suppression operations can be represented as,

$$A_{i,j} = \begin{cases} B_{i,j}, \ if \ A_{i,j} \in Neigh.\{A_{i,j}\} \\ A_{i,j}, \ otherwise \end{cases} \tag{8}$$

where $B_{i,j}$ is a randomly generated new antibody agent. Since the neighborhood range is small, for each agent, if there is an identical agent in the neighborhood, it will be directly replaced with the randomly generated antibody agent to maintain the diversity of the agent group.

2.2.4 Population crossover operation

The population crossover operation exchanges certain gene loci of two randomly selected parent antibodies according to certain rules with a certain probability, thereby exchanging information and generating two new offspring. The objects selected for the crossover operation in this algorithm are different. For each antibody agent, a roulette-wheel-based selection method is used to select another parent agent in the entire grid for cross operation, and then replace the original agent with the better offspring after cross operation. This operation not only provides an opportunity for each antibody agent, regardless of its excellence, to intersect, improve the diversity of the population, conduct global search, but also has mutual motivation between excellent individuals, ensuring the direction of evolution. If the crossover operator uses a partially matched crossover operator, the population crossover operation can be represented as,

$$A_{i,j} = \begin{cases} PMX\left(A_{i,j}, Roulette(L)\right), & if \ \ unifrnd(0,1) < (times - i)/times \\ A_{i,j}, & otherwise \end{cases} \tag{9}$$

where $Roulette(L)$ represents the roulette choice function for the entire grid L, *unifrnd*(0,1) represents a random number that follows a uniform distribution between 0 and 1, *times* represents the overall number of iterations, and i indicates the current number of iterations. The probability of population crossover operations gradually decreases as the number of iterations increases, with the aim of accelerating population convergence and preventing the loss of too much useful information due to excessive crossover.

2.2.5 Self-learning operation

The self-learning operation refers to the self-learning ability of agents in multi-agent systems. In this paper, the multi-agent is the antibody agent group, and its most representative individual is the individual with the best fitness. Through cloning and mutation of the individual, selecting the optimal individual after mutation to replace the original optimal agent, and completing local search in a small range, the antibody agent group will further evolve. Self-learning operations is represented as,

$$A_{best} = Select(Mutation(Clone(A_{best}))) \tag{10}$$

2.2.6 Partial 2-opt operator

The 2-opt operator was first used in the traveling salesman problem, of which the goal is to find a Hamiltonian circuit with the minimum sum of weights. The traveling salesman problem is one of the most widely studied problems in the field of combinatorial optimization. This problem is NP-hard. There is no constant approximation of polynomial time for the traveling salesman problem if no additional assumptions are added. Researchers are committed to designing a better approximation algorithm for the traveling salesman problem in metric space. The 2-opt optimization algorithm is one of them, which is the core of the classic Lin Kernighan algorithm. The 2-opt optimization algorithm starts from an arbitrary feasible solution and repeatedly searches. If a crossing edge is found, the encoding in the middle of this pair of crossing edges is reversed. This substitution operation is performed until the local optimal is reached, and the local optimal solution is finally used as the algorithm's solution. The 2-opt algorithm is expressed as follows.

Algorithm 2-opt local optimization algorithm

```
Input: feasible solution, distance matrix D between nodes
Output: 2-opt optimized feasible solution
    For a = 1:length(Solution)-2
        For b = a+2: length(Solution)-1      % Traverse the entire feasible solution
            If D(a,a+1)+D(b,b+1)>D(a,b)+D(a+1,b+1)    % If cross edges are found
                Solution(a+1:b)= Solution(b:-1:a+1)
            End
        End
    End
```

Although it is not possible to directly perform 2-opt operations on the overall antibody, for the target points under each UAV, partial 2-opt operations can be used to accelerate the convergence of the algorithm. The specific operations are as follows,

$$\text{Solution} = \left\{ \begin{array}{l} 2-opt\left(\text{Solution}\left[1:\left\lfloor\dfrac{N}{M}\right\rfloor\right]\right), ..., \\ ..., 2-opt\left(\text{Solution}\left[M*\left\lfloor\dfrac{N}{M}\right\rfloor:N\right]\right) \end{array} \right\} \tag{11}$$

where $2 - opt(\cdot)$ represents performing 2-opt optimization, N is the number of target points, M is the number of drones, and $\{\cdot\}$ represents the splicing of various parts.

2.3 Improved cuckoo algorithm for online task allocation

The previous section proposes an improved offline task allocation algorithm, which is characterized by the introduction of a large number of optimizations to ensure the accuracy of the optimization results. The time complexity is large, so the calculation time is long, and the requirements for the real-time scenes cannot be met. Therefore, this section proposes an improved cuckoo optimization algorithm, which has strong ability to jump out of local optimization and a lower complexity.

2.3.1 Target point exchange operator

The target point exchange operator mainly completes the approximation of large step random walks in the Levy distribution. The main idea of the target point exchange operator is to evaluate the excellent state indicators of each target point, identify one or more target points with the worst indicators under a UAV in this generation of feasible solutions, and remove them from the UAV's target point sequence and place them in the candidate pool. Then several UAVs are selected in a random order, selecting one or more target points from the candidate pool that have the best performance for them into their own target point sequence. The evaluation index selected in this section is the average value of the sum of distances from the target point to all points in the UAV target point sequence, which is given as,

$$value(i) = \frac{1}{K-1}\sum_{\substack{j=1\\ j\neq i}}^{K} Distance\left(i,j\right) \tag{12}$$

where $value(i)$ represents the evaluation index, i represents the target point label, K represents the number of target points of the UAV, and $Distance(i,j)$ represents the distance between the target points. So, the target point exchange operator can be expressed as,

$$\begin{aligned} &[rest, pool] = delete\_worst(Solution) \\ &Solution = reselect(rest, pool) \end{aligned} \tag{13}$$

where $delete\_worst(\cdot)$ denotes the function of removing the worst objective point, and $reselect(rest, pool)$ denotes reselecting the target point function in the candidate pool.

2.3.2 Partial 2-opt operator

For online task allocation algorithms, real-time performance is an important indicator that needs to consider the constraints of computing time and computer power. For the initial stage of population evolution, the use of partial 2-opt operators is of great benefit to evolution, because it can replace all local cross paths and generate paths with no or less loops, which can greatly accelerate the convergence of the algorithm. However, as a search operator with time complexity of $O(n^3)$ in the overall algorithm, it will consume more time. In the later stage of algorithm iteration, almost all loops are removed by local or global optimization operators, so not performing 2-opt optimization with a certain probability will not reduce the algorithm's optimization ability. So, drawing inspiration from the sigmoid function, the probability set by the following equation determines whether to perform 2-opt optimization.

$$P_2 = 0.2 + 0.5 * \frac{1}{1 + e^{-4\times \frac{times/2 - i\_num}{times}}} \tag{14}$$

where *times* represents the overall number of iterations, *times-i_num* represents the remainder iterations. The probability of 2-opt execution has decreased from around 64% to around 26%, saving a lot of computation time.

2.3.3 Adaptive optimization proportion strategy

In general discrete cuckoo algorithms, the proportion of executing short step, medium step, and long step jumps is generally fixed, set to 20%, 30%, and 50%, respectively. This not only maintains local evolution, but also maintains a good ability to jump out of local optima. However, in the later stage of evolution, the population benefits little from long step jumps, and only a small range of local optimizations are needed to approach the optimal solution. Therefore, this paper proposes an adaptive optimization step probability adjustment strategy. According to the above-mentioned analysis, the probability of short steps should be adjusted from 20% to around 50%, and the probability of long steps should be adjusted from 50% to around 20%. Using the sigmoid function, the probability of short step jumps is expressed as,

$$P_{short} = 0.1 + \frac{1}{1 + e^{-2.2 + 1.36 * \frac{times - i\_num}{times}}} \tag{15}$$

and the probability of long step jumps is expressed as,

$$P_{long} = 0.6 - \frac{1}{1 + e^{-2.2 + 1.8 * \frac{times - i\_num}{times}}} \tag{16}$$

where times represents the total number of iterations, and *times-i_num* represents the remaining iterations, so the probability of medium step jumps is expressed as,

$$P_{mid} = 1 - P_{short} - P_{long} \tag{17}$$

2.3.4 Learning strategies for discovering bird nests

The discovery of bird nests is similar to the dropout strategy in machine learning, which preserves population diversity by randomly discarding some feasible solutions. The probability of bird nests being discovered in the improved algorithm is set to a constant value of 20%. After the bird's nest is discovered, the generated feasible solution is discarded and the strategy of learning from the optimal solution of this generation is activated. After three medium steps, an optimal individual is selected to replace the position of the generated feasible solution.

The improved cuckoo algorithm for online task allocation is expressed as follows,

Algorithm Improved cuckoo algorithm

```
Input: i_num=0; N; times % Initial number of iterations, population size and number of iterations
Output: allocation result
L = Init(); % Randomly generate initial solution
Fitarr = calFit(L); % Calculate the fitness of all initial solutions
while i_num ≤ times
for (each A_i in L)
    P = rand(0,1);
    if 0<P<P_short    % Generate 2-step neighborhood
        Solution = best(2_step_neighbour(A_i))
    else if P_short < P < P_short + P_mid    % Generate 3-step neighborhood
        Solution = best(3_step_neighbour(A_i))
    else    % Generate target point exchange neighborhood
        [rest, pool] = delete_worst(Solution)
        Solution = reselect(rest, pool)
    end
    if rand(0,1) < P_2
        Solution = {2-opt(Solution[1:⌊N/M⌋]),...,
                    ...,2-opt(Solution[M*⌊N/M⌋:N])} % 2-opt optimization
    end
    A_i = Solution; % Update population
    if rand(0,1)<0.2 % Bird's nest discovered, learning new solutions
        A_i = best(3_step_neighbour(A_best));
    end
end
end
```

The time complexity of the improved adaptive cuckoo algorithm proposed in this section is $O(times \times N)$. Compared with the offline allocation algorithm in the previous section, it has greater complexity reduction and faster running time.

3 Results and discussion

3.1 Improved immune multi-agent algorithm for offline task allocation

The simulation parameters are set as follows: The grid parameter is 6, the size of the agent group is 36, the memory population size is 36, the number of iterations is 5000, and the clone multiplication factor is 10. Assuming that the task is distributed in a three-dimensional space of 50 km × 50 km × 400 m, with 4 UAVs and 30 target points, the truncation vector is [7, 14, 21]. We set a cylindrical meteorological threat zone with a radius of 3 km and a height of 400 m in the scenario, and the center of the circle is located at the horizontal coordinates [33 km, 20 km]. Figure 1 shows an example of the flight routes of the UAVs and the location of the threat zone. The circles on the routes are the start points of the UAVs, and the triangles are the target points.

3.1.1 Offline task allocation without threat or energy constraint

Since there is no threat zone, the penalty path weighting coefficient $\gamma = 0$. α should be slightly greater than β, and $\alpha + \beta + \gamma = 1$, so we set $\alpha = 0.7$ and $\beta = 0.3$, experimentally.

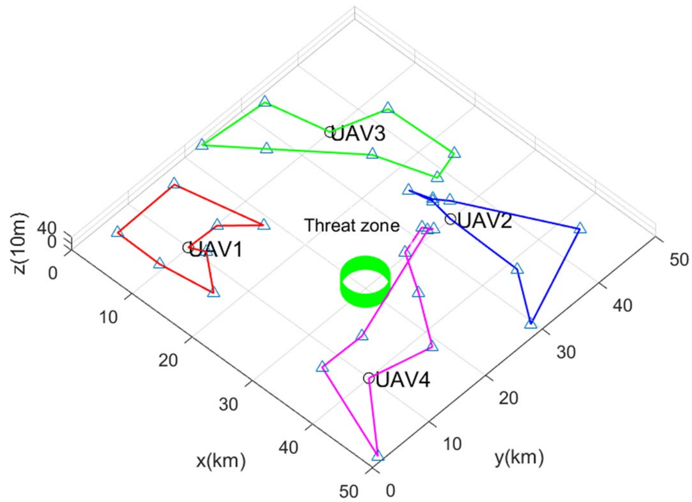

Fig. 1 An example of the simulation scenario. UAV1: the flight route of UAV 1. UAV2: the flight route of UAV 2. UAV3: the flight route of UAV 3. UAV4: the flight route of UAV 4. Threat zone: the cylindrical meteorological threat zone

We compare our proposed algorithm with the improved clonal selection algorithm [21]. Table 1 shows the numerical results.

The known optimal solution of this problem is 74.6146 by using the proof by exhaustion in mathematics to traverse all the existing feasible solutions. Next, we shall analyze the optimization performance of the proposed immune multi-agent algorithm.

From Table 2, we can see that the iterations all reach the global optimal solution, and there is no deviation between the optimal solution and the average solution, which is suitable for offline task allocation.

Table 1 Distance comparison of the offline task allocation algorithms with no constraint (unit: km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	58.3190	27.4975	71.9817	116.7276	274.5259	116.7276	0
Proposed offline algorithm	58.3190	65.2507	70.6570	85.5261	279.7528	85.5261	0

Table 2 Fitness function of the improved immune multi-agent algorithm with no constraint

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of average solution
Fitness function	74.6146	74.6146	74.6146	0%	0%

Table 3 Distance comparison of the offline task allocation algorithms with energy constraint (km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	103.0753	41.7938	94.4393	94.6392	333.9476	103.0753	0
Proposed offline algorithm	67.7887	65.4857	68.4093	67.9007	269.5844	68.4093	0

Table 4 Fitness function of the improved immune multi-agent algorithm with energy constraint

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of optimal solution
Fitness function	67.7000	67.7000	71.5029	0%	5.6%

3.1.2 Offline task allocation with energy constraint

When the UAV has energy constraints, the task allocation problem can be changed to the following statement: when the base calculates the time when the UAV must return due to insufficient energy, the completed task set is recorded as T_c, and the returning UAV set is recorded as U_{return}, then the problem becomes to allocate the tasks in the remaining task set $T - T_c$ to the UAVs in the remaining set $U - U_{return}$. Since there is also no threat zone, the penalty path weighting coefficient $\gamma = 0$. α should be slightly greater than β, and $\alpha + \beta + \gamma = 1$, so we set $\alpha = 0.7$ and $\beta = 0.3$, experimentally. We compare our proposed algorithm with the improved clonal selection algorithm, and Table 3 shows the numerical results.

By using the proof by exhaustion in mathematics to traverse all the existing feasible solutions, we can know that the optimal solution of the problem is 67.7000. Next, we analyze the optimization performance of the proposed immune multi-agent algorithm. The result is given in Table 4.

In the proposed algorithm, on average, eight out of 30 iterations have reached the global optimal solution. For offline planning algorithms, parallel data centers on the ground can definitely be used for operations to achieve numerical optimal solutions. For a more complex multi-objective optimization problem, it is already excellent, and the optimal solution has no deviation degree. The average solution deviation degree is small, and the algorithm performance improvement is significant.

3.1.3 Offline task allocation with threat and energy constraints

In the case of meteorological threat, the task allocation problem can be changed to the following statement: keep away from meteorological threat as far as possible while ensuring the shortest distance and load balance. Assume that a cylindrical meteorological threat zone with a radius of 3 km and a height of 400 m appears, and the center of the circle is located at the horizontal coordinates [33 km, 20 km]. Due to the dominant factor in the threat zone, the penalty path weighting coefficient $\gamma = \frac{1}{3}$.

Table 5 Distances of the offline task allocation algorithms with threat and energy constraints (km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	66.8262	110.5447	81.3587	90.9541	349.6837	110.5447	0
Proposed offline algorithm	58.3190	65.2507	70.6570	90.9069	285.1336	90.9069	0

α should be slightly greater than β, so we set $\alpha = \frac{7}{15}$ and $\beta = \frac{1}{5}$, experimentally. We compare our proposed algorithm with the improved clonal selection algorithm, and Table 5 shows the numerical results.

Next, we will analyze the optimization performance of immune multi-agent algorithms. In the simulations, three out of 30 iterations have reached the global optimal solution. The details are given in Table 6.

For offline planning algorithms, parallel data centers on the ground can definitely be used for operations to achieve numerical optimal solutions. For a more complex multi-objective optimization problem, it is already excellent, and the optimal solution is unbiased, with a small deviation from the average solution and a significant improvement in algorithm performance.

3.2 Improved cuckoo algorithm for online task allocation

3.2.1 Online task allocation without threat or energy constraint

Since there is no threat zone, the penalty path weighting coefficient $\gamma = 0$. α should be slightly greater than β, and $\alpha + \beta + \gamma = 1$, so we set $\alpha = 0.7$ and $\beta = 0.3$, experimentally. We compare the performances of the improved clonal selection algorithm [21], the offline task allocation algorithm proposed in Sect. 2.2, and the online task allocation algorithm proposed in Sect. 2.3. The simulation result is given in Table 7.

Table 6 Fitness function of the improved immune multi-agent algorithm with both constraints

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of optimal solution
Fitness function	75.0640	75.0640	76.8237	0%	2.3%

Table 7 Distances of the online and offline task allocation algorithms with no constraint (km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	58.3190	27.4975	71.9817	116.7276	274.5259	116.7276	0
Proposed offline algorithm	58.3190	65.2507	70.6570	85.5261	279.7528	85.5261	0
Proposed online algorithm	58.3190	65.2507	70.6570	85.5261	279.7528	85.5261	0

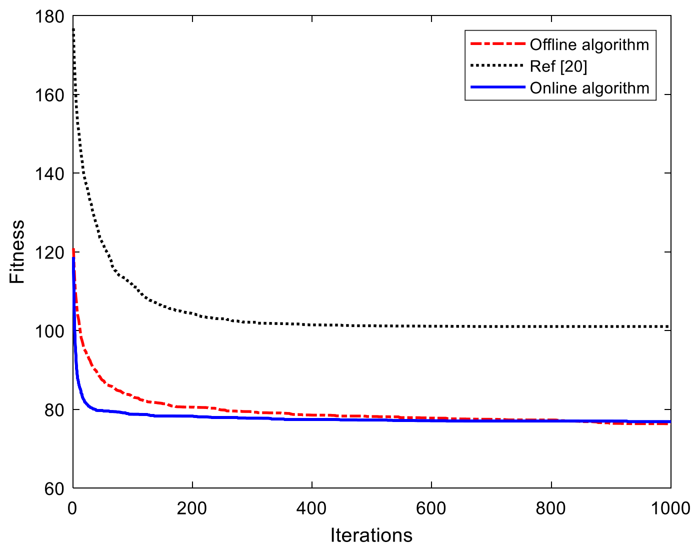

Fig. 2 Convergence of fitness functions of the task allocation algorithms with no constraint. Offline algorithm: the fitness function value of the proposed offline algorithm varies with the number of iterations. Reference [20]: The fitness function value of the algorithm which was proposed in Reference [20] varies with the number of iterations. Online algorithm: The fitness function value of the proposed online algorithm varies with the number of iterations

Table 8 Fitness function of the improved cuckoo algorithm with no constraint

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of optimal solution
Fitness function	74.6146	74.6146	76.8631	0%	3%

From Table 7, although our proposed algorithm has a slightly longer total distance than the improved clonal selection algorithm, it has a much better balance in task allocation, i.e., the difference between the maximum distance and the minimum distance is much smaller. Figure 2 shows the fitness function curves of the three algorithms. It can be seen that the online task allocation algorithm has the fastest convergence speed, so it is suitable for real-time scenarios.

The proposed online task allocation algorithm (the improved cuckoo algorithm) has a smaller maximum distance than the algorithm in [21]. Then we test the overall performance of the improved cuckoo algorithm, and the simulation result is given in Table 8.

In the simulations, 6 out of 30 iterations have reached the global optimal solution. If the number of target points changes, the online task allocation algorithm can give out a good solution within limited time.

3.2.2 Online task allocation with energy constraint

In the case of energy constraints on UAVs, the online task allocation problem can be expressed as follows: the UAV's own system realizes that there is a risk of energy shortage when performing several tasks and must return. The completed task set is expressed as Tc, and the returning UAV is expressed as U_{return}. Then, the problem becomes to assign

Table 9 Distances of the online and offline task allocation algorithms with energy constraint (km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	103.0753	41.7938	94.4393	94.6392	333.9476	103.0753	0
Proposed offline algorithm	67.7887	65.4857	68.4093	67.9007	269.5844	68.4093	0
Proposed online algorithm	67.7887	65.4857	65.2353	69.8132	268.3229	69.8132	0

tasks in T-Tc to the UAVs in U-U_{return}. Since there is also no threat zone, the penalty path weighting coefficient $\gamma=0$. α should be slightly greater than β, and $\alpha+\beta+\gamma=1$, so we set $\alpha=0.7$ and $\beta=0.3$, experimentally. Assume that UAV 2 must return after completing Task 5. The simulation result is given in Table 9.

The improved cuckoo algorithm shows better fairness than the algorithm in [21], however, a little worse than the offline algorithm in Sect. 2.2, but the convergence speed of the online algorithm is much faster than that of the offline one, as shown in Fig. 3.

The overall performance with energy constraint is given in Table 10.

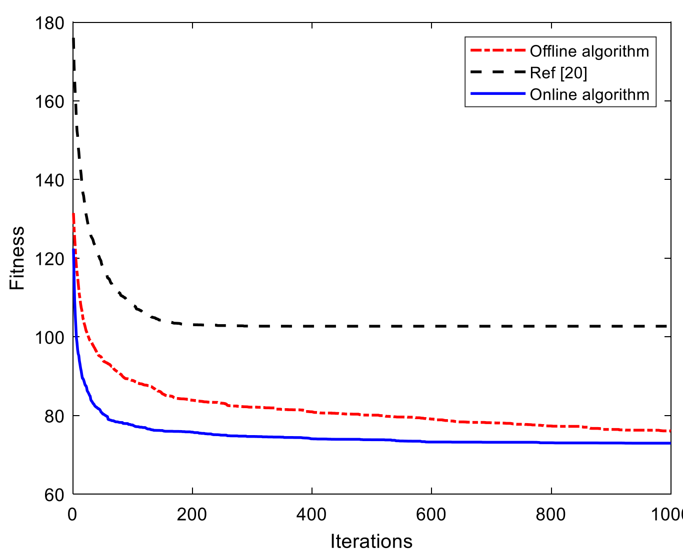

Fig. 3 Convergence of fitness functions of the task allocation algorithms with energy constraint. Offline algorithm: the fitness function value of the proposed offline algorithm varies with the number of iterations. Reference [20]: The fitness function value of the algorithm which was proposed in Reference [20] varies with the number of iterations. Online algorithm: The fitness function value of the proposed online algorithm varies with the number of iterations

Table 10 Fitness function of the improved cuckoo algorithm with energy constraint

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of optimal solution
Fitness function	67.7000	67.9005	72.9760	0.3%	7.7%

The deviation degree of the optimal solution for 30 iterations is 7.7%. Although the performance declines a little, the calculation speed is fast; and thus, the algorithm is suitable for online task allocation where scenarios that require a temporary return flight often occurs.

3.2.3 Online task allocation with threat and energy constraint

When there is a meteorological threat, the task allocation problem can be expressed as follows: while ensuring the shortest distance and load balance, try to stay as far away from the meteorological threat as possible. The meteorological threat zone is the same with that in Sect. 3.1.3. Due to the dominant factor in the threat zone, the penalty path weighting coefficient $\gamma = \frac{1}{3}$. α should be slightly greater than β, and $\alpha+\beta+\gamma=1$, so we set $\alpha = \frac{7}{15}$ and $\beta = \frac{1}{5}$, experimentally. The simulation result is given in Table 11.

The convergence speeds of the three algorithms are given in Fig. 4. It can be seen that the proposed online algorithm has better real-time performance than the offline algorithm.

Although the optimal fitness is 67.9005, which is about 0.3% lower than the optimal solution, the total distance index and maximum distance index are similar to those of the immune multi-agent algorithm, proving that the improved algorithm performs equally well in the problem of poor performance of clone selection algorithms. The overall performance with energy constraint and threat zone is given in Table 12.

The average solution without deviation degree for 30 iterations is 4.9%, which is only a 2.6% decrease in performance compared with that of the improved immune multi-agent algorithm. However, the speed improvement is significant, and it is suitable for online task allocation with threat and energy constraints. This algorithm can be used to obtain a high-quality solution in a limited time for scenarios where UAVs need to return temporarily.

According to the simulation results above, both the proposed offline task allocation algorithm and the proposed online task allocation algorithm have their advantages and disadvantages. The proposed offline task allocation algorithm, i.e., the improved immune multi-agent algorithm, can achieve better balance among all the UAVs, and both the total distance and the maximum distance are smaller, but the convergence speed of the algorithm is slower, so it is suitable for task allocation in the offline stage. On the other hand, the proposed online task allocation algorithm, i.e., the improved

Table 11 Distances of the online and offline task allocation algorithms with both constraints (km)

Algorithm	UAV1 distance	UAV2 distance	UAV3 distance	UAV4 distance	Total distance	Maximum distance	Threat penalty
Reference [21]	66.8262	110.5447	81.3587	90.9541	349.6837	110.5447	0
Proposed offline algorithm	58.3190	65.2507	70.6570	86.4723	280.6991	86.4723	0
Proposed online algorithm	58.3190	65.2507	70.6570	87.2897	281.5164	87.2897	0

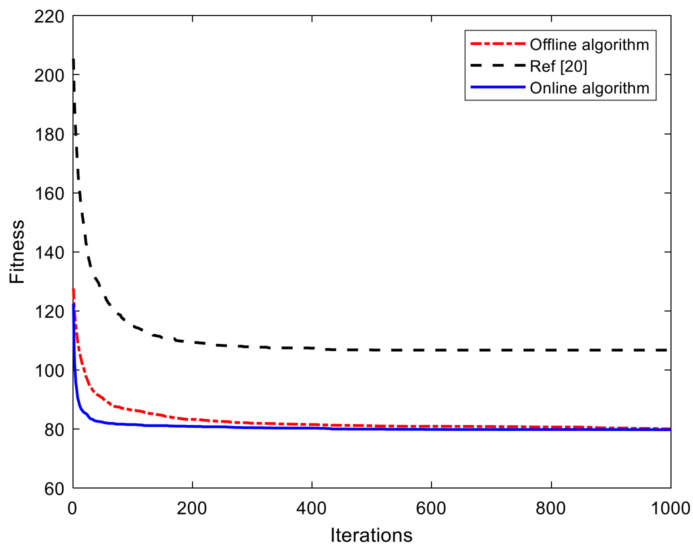

Fig. 4 Convergence of the fitness functions of the task allocation algorithms with both constraints. Offline algorithm: the fitness function value of the proposed offline algorithm varies with the number of iterations. Reference [20]: The fitness function value of the algorithm which was proposed in Reference [20] varies with the number of iterations. Online algorithm: The fitness function value of the proposed online algorithm varies with the number of iterations

Table 12 Fitness function of the improved cuckoo algorithm with both constraints

Index	Known optimal solution	Algorithm optimal solution	Algorithm average solution	Deviation degree of optimal solution	Deviation degree of optimal solution
Fitness function	75. 0640	75.4523	78.9049	0.3%	4.9%

cuckoo algorithm, has a faster convergence speed, but the total distance and the maximum distance are larger, so it is suitable for task allocation in the online stage. In a complete flight task, from before taking off to completing the last mission, both the two algorithms are applied.

4 Conclusions

For scenarios where multiple UAVs collect sensor data from large-scale wireless sensor nodes after earthquakes, this paper proposes two task allocation algorithms for different stages.

In response to the high accuracy requirements of offline task allocation algorithms, this paper first establishes a general mathematical model for data collection task allocation problems, and then proposes an improved immune multi-agent algorithm as a solution for offline task allocation problems, striving to improve the accuracy of the algorithm and plan the global optimal task allocation scheme. The accuracy and effectiveness of the algorithm have been demonstrated through simulation analysis.

In response to the issue of higher real-time requirements for online task allocation algorithms, this paper proposes an improved adaptive discrete cuckoo algorithm as the allocation algorithm for online tasks. By introducing adaptive step size transformation and appropriate local optimization operators, the convergence speed of the algorithm is accelerated, making it suitable for real-time scenarios of online reassignment. Through runtime analysis and algorithm optimization characteristics analysis, it has been proven that it achieves fast convergence performance with minimal loss of accuracy.

Abbreviations

NP	Non-deterministic Polynomial
OFDMA	Orthogonal Frequency-Division Multiple Access
UAV	Unmanned Aerial Vehicle

Acknowledgements
The authors would like to thanks all the people who have helped in the projects.

Author contributions
LY proposed the idea, supervised the project, and wrote this manuscript. YY improved the idea. WM and XW provided project support and guidance. XL and RZ provided real scenario suggestions. All authors read and approved the final manuscript.

Funding
This research was funded by China Chile ICT "the Belt and Road" Joint Laboratory under grant number SQ2020YFE020708 and National Natural Science Foundation of China under grant number 61971161.

Availability of data and materials
The data and materials are custom designed according to real scenarios.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 24 November 2023 Accepted: 21 January 2024
Published online: 03 February 2024

References

1. X. Liu, Y. Yu, B. Peng et al., RIS-UAV enabled worst-case downlink secrecy rate maximization for mobile vehicles. IEEE Trans. Veh. Technol. **72**(5), 6129–6141 (2023)
2. X. Liu, Y. Yu, F. Li et al., Throughput maximization for RIS-UAV relaying communications. IEEE Trans. Intell. Transp. Syst. **23**(10), 19569–19574 (2022)
3. Jiang, M.: Binocular stereo vision intelligent control framework for the landing position of quadrotor UAV based on fuzzy control. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES), pp. 1348–1352 (2021)
4. X. Liu, B. Lai, B. Lin et al., Joint communication and trajectory optimization for multi-UAV enabled mobile internet of vehicles. IEEE Trans. Intell. Transp. Syst. **23**(9), 15354–15366 (2022)
5. Yoon, S., Kim, J.: Fleet size optimization and collaborative route planning for multi-vehicle task allocation. AIAA Scitech 2019 Forum (2019)
6. T. Itshak, J. Aleksandar, N. Shimon et al., A Modified distributed bees algorithm for multi-sensor task allocation. Sensors **18**(3), 759 (2018)
7. Wu, H., Li, H., Xiao, R., et al.: Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. Phys. Stat. Mech. Appl. S0378437117308166 (2017).
8. Y. Zeng, R. Zhang, T.J. Lim, Throughput maximization for UAV-enabled mobile relaying systems. IEEE Trans. Commun. **64**, 4983–4996 (2016)
9. X. Liu, Z. Liu, B. Lai et al., Fair energy-efficient resource optimization for multi-UAV enabled internet of things. IEEE Trans. Veh. Technol. **72**(3), 3962–3972 (2023)
10. G. Zhang, Q. Wu, M. Cui et al., Securing UAV communications via joint trajectory and power control. IEEE Trans. Wirel. Commun. **18**, 1376–1389 (2018)
11. X. Zhou, Q. Wu, S. Yan et al., UAV-enabled secure communications: joint trajectory and transmit power optimization. IEEE Trans. Veh. Technol. **68**(4), 4069–4073 (2019)
12. Cao, X., Xu, J., Zhang, R.: Mobile edge computing for cellular-connected UAV: computation offloading and trajectory optimization. In: 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–5 (2018)
13. J. Xu, Y. Zeng, R. Zhang, UAV-enabled wireless power transfer: trajectory design and energy optimization. IEEE Trans. Wirel. Commun. **99**, 1–1 (2017)
14. Xie, L., Xu, J., Zhang, R.: Throughput maximization for UAV-enabled wireless powered communication networks. IEEE Internet Things J. 1 (2018)

15. C. Shen, T.H. Chang, J. Gong et al., Multi-UAV interference coordination via joint trajectory and power control. Optim. Control **68**, 843–858 (2020)
16. L. Liang, S. Zhang, Z. Rui, CoMP in the sky: UAV placement and movement optimization for multi-user communications. IEEE Trans. Commun. **99**, 5645–5658 (2018)
17. J. Lyu, Y. Zeng, R. Zhang, Cyclical multiple access in UAV-aided communications: a throughput-delay tradeoff. IEEE Wirel. Commun. Lett. **5**(6), 600–603 (2016)
18. Q. Wu, R. Zhang et al., Common throughput maximization in UAV-enabled OFDMA systems with delay consideration. IEEE Trans. Commun. **66**(12), 6614–6627 (2018)
19. Hasan, S.Y., Hüseyin, G., Hakan, Ç., et al.: Optimal task allocation for multiple UAVs. In: 2020 28th Signal Processing and Communications Applications Conference (SIU) (2020)
20. Zhang, Y., Liu, C.: An immune algorithm for network data security detection. In: 2021 17th International Conference on Computational Intelligence and Security (CIS), pp. 247–251 (2021)
21. Y. Quan, Y. He, Clonal selection algorithm for multi robot task assignment and scheduling. J. South China Univ. Technol. **49**(5), 102–110 (2021)

Publisher's Note