mathematics

MDPI

Article

A Unified Multi-Objective Optimization Framework for UAV Cooperative Task Assignment and Re-Assignment

Xiaohua Gao [1], Lei Wang [1], Xichao Su [2], Chen Lu [3,4,5], Yu Ding [3,4,5], Chao Wang [3,4,5,*], Haijun Peng [6] and Xinwei Wang [6,*]

[1] School of Mathematical Science, Dalian University of Technology, Dalian 116024, China
[2] Department of Airborne Vehicle Engineering, Naval Aeronautical and Astronautical University, Yantai 264001, China
[3] Science and Technology on Reliability and Environmental Engineering Laboratory, Beijing 100191, China
[4] Institute of Reliability Engineering, Beihang University, Beijing 100191, China
[5] School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
[6] State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China
* Correspondence: wangchaowork@buaa.edu.cn (C.W.); wangxinwei@dlut.edu.cn (X.W.)

Abstract: This paper focuses on cooperative multi-task assignment and re-assignment problems when multiple unmanned aerial vehicles (UAVs) attack multiple known targets. A unified multi-objective optimization framework for UAV cooperative task assignment and re-assignment is studied in this paper. In order to simultaneously optimize the losses and benefits of the UAVs, we establish a multi-objective optimization model. The amount of tasks that each UAV can perform and the number of attacks on each target are limited according to the ammunition capacity of each UAV and the value of each target. To solve this multi-objective optimization problem, a multi-objective genetic algorithm suitable for UAV cooperative task assignment is constructed based on the NSGA-II algorithm. At the same time, a selection strategy is used to assist decision-makers in choosing one or more solutions from the Pareto-optimal front. Moreover, to deal with emergencies such as UAV damage and to detect of new targets, a task re-assignment algorithm based on the contract network protocol (CNP) is developed. It can be implemented in real-time while only slightly sacrificing the ability to seek the optimal solution. Simulation results demonstrate that the methods developed in this paper are effective.

Keywords: unmanned aerial vehicle; cooperative task assignment; multi-objective optimization; genetic algorithm; contract network protocol

MSC: 90C29

check for updates

Citation: Gao, X.; Wang, L.; Su, X.; Lu, C.; Ding, Y.; Wang, C.; Peng, H.; Wang, X. A Unified Multi-Objective Optimization Framework for UAV Cooperative Task Assignment and Re-Assignment. *Mathematics* **2022**, *10*, 4241. https://doi.org/10.3390/math10224241

Academic Editor: Ioannis G. Tsoulos

Received: 15 September 2022
Accepted: 4 November 2022
Published: 13 November 2022

1. Introduction

Unmanned aerial vehicles (UAVs) refer to aircraft without pilots; such aircraft can fly autonomously or can be remotely controlled by an operator [1]. They can completely reduce casualties and costs when performing high-risk missions [2]. At present, UAVs have become very popular in many fields, such as infrastructure inspection, coastal border surveillance, military applications, and others fields [3–6]. In the increasingly complex battlefield situation, a single UAV cannot quickly adapt to the changing battlefield environment due to lack of information interaction. At the same time, the effectiveness of a single UAV in perform tasks is not high due to the limited ammunition capacity of a single UAV. However, when a team composed of multiple UAVs performs tasks cooperatively it can overcome the shortcomings of a single UAV. The UAV team can share information and fully allocate internal resources, allowing tasks to be completed efficiently [7,8].

In order to take the advantage of multiple UAVs when performing tasks realize improvements in efficiency, cooperative task assignment is important. In recent years, results have been achieved in research on task assignment in different fields [9,10]. According to the different factors considered by researchers, the task assignment problem of multiple UAVs can be categorized into different models, and many related algorithms have been proposed as well. Swarm intelligence optimization algorithms are widely used in this field, particularly genetic algorithms [11]. The UAV team can be composed of either homogeneous UAVs or heterogeneous UAVs. For cooperative task assignment of homogeneous UAVs considering the limitations on the total flight distance of UAVs, Wang et al. [12] established a combinatorial optimization problem with the total weighted cost of target value and distance cost as the objective, and presented an improved genetic algorithm based on the beetle antennae search algorithm. Venugopalan et al. [13] presented a team search-based decentralized task assignment scheme for homogeneous UAVs. Velhal et al. [14] formulated the restricted airspace protection problem as a multi-UAV spatio-temporal multi-task allocation problem, and proposed a modified consensus-based bundled auction method to solve it. For cooperative task assignment of heterogeneous UAVs, Fatemeh Afghah et al. [15] proposed a coalition formation approach to solve the problem of adversary target detection and subsequent task completion. Schwarzrock et al. [16] proposed a method to increase the amount of tasks performed within the problem of task allocation among agents representing UAVs. Taking the minimization of the task execution time of UAVs as the objective, Ye et al. [17] established a task assignment model and proposed a modified genetic algorithm with a multi-type-gene chromosome encoding method. Considering a coupled task allocation and path planning problem, Yan et al. [18] proposed a task allocation algorithm and a cooperative particle swarm algorithm. Uncertain factors are considered in the cooperative task assignment problem as well. Considering the parameter and time-sensitive uncertainties in the task assignment problem, Chen et al. [19] proposed an algorithm that combines the interior point method and the modified two-part wolf pack search algorithm. Jia et al. [20] established a two-stage stochastic programming model of the cooperative task assignment problem incorporating the stochastic velocities of UAVs, and proposed a novel metaheuristic based on a modified genetic algorithm.

However, only a single goal is considered by the above task assignment problems. In order to simultaneously optimize the losses and benefits of the UAV team, it is necessary to study the multi-objective optimization problem for multi-UAV task assignment. NSGA-II and its variants are widely used in the study of multi-objective optimization for UAV mission assignment [21,22]. Cheng et al. [23] considered the multi-objective optimization of task assignment, with minimization of cost and maximization of the value of destroyed targets regarded as the objectives. Taking into account the relationship between the UAVs and the ground control stations, Cristian et al. [24] proposed a new multi-objective genetic algorithm for solving complex mission planning problems be formulating mission planning as a constraint satisfaction problem [25]. Chen et al. [26] studied the task assignment problem for UAVs with different sensor capacities, and proposed a modified multi-objective symbiotic organism search algorithm. Wang et al. [27] considered a high-dimensional multi-objective optimization problem containing four objectives for task assignment, then used an improved multi-objective quantum-behaved particle swarm optimization algorithm to solve the problem. Pohl et al. [28] developed an innovative algorithm for multi-UAV mission routing. Phiboon et al. [29] studied multi-fidelity multi-objective airfoil design optimization for fixed-wing UAVs. However, the above studies have not explained how decision-makers are to choose a solution from the Pareto-optimal front. At the same time, the above literature does not consider the emergencies that may occur on the battlefield. The complexity of the battlefield environment inevitably causes emergencies; for example, damage to UAVs, the appearance of new targets, etc. The problem of UAV task re-assignment needs to be considered when such emergencies occur. Unlike the general task assignment problem, task re-assignment in emergencies must be completed in a short time. Therefore, the task re-assignment problem has higher requirements with respect to the calculation speed of the algorithm. A contract network algorithm [30] based on the auction mechanism has been applied to the real-time task assignment problem. Zhen et al. [31] proposed an improved contract network protocol-based cooperative target assignment scheme to deal with heterogeneous overloading and time sequence problems. Zhang et al. [32] established a model of real-time assignment of tasking based on reconnaissance benefits and reconnaissance costs, and proposed an improved contract network algorithm. Xiang et al. [33] studied the cooperation target assignment of multiple agents, and proposed an improved contract network protocol according to the characteristics and restrictions of target assignment.

The contributions of the present article are provided as follows:

(1) Based on the idea of the NSGA-II algorithm, an algorithm suitable for solving the multi-objective optimization problem of multi-UAV task assignment is presented, and the encoding format and genetic operators therein are specially designed.

(2) A method for aiding commanders in choosing an operation plan from among the Pareto solution set is provided.

(3) A highly efficient CNP-based algorithm is developed for real-time task re-assignment in emergencies.

The remainder of this paper is organized as follows. Section 2 presents a multi-objective optimization model of cooperative task assignment. Section 3 provides the multi-objective optimization strategy. Section 4 provides the method of selecting solutions from the Pareto solution set. The problem of task re-assignment in emergencies is studied in Section 5. Numerical examples are provided in Section 6. Finally, Section 7 concludes the paper.

2. Multi-Objective Optimization Model of Cooperative Task Assignment

For convenience, let $I_n := \{1, 2, \cdots, n\}$, $\bar{I}_n := \{0\} \cup I_n$, $n \in \mathbb{N}_+$. On the battlefield, UAVs with attack capabilities are required to attack multiple known targets in coordination to improve efficiency. Assume that N_U $(N_U \in \mathbb{N}_+)$ UAVs coordinately attack N_T $(N_T \in \mathbb{N}_+)$ targets in the combat area. Let $U := \{U_1, U_2, \cdots, U_{N_U}\}$ be the set of UAVs, where U_i $(i \in I_{N_U})$ represents the i-th UAV. The target set is recorded as $T := \{T_1, T_2, \cdots, T_{N_T}\}$, where T_j $(j \in I_{N_T})$ represents the j-th target. When a UAV attacks a target, the UAV may be destroyed, and different UAVs have different probabilities of being destroyed when they attack different targets. Let P_{ij}, K_{ij} respectively denote the probability that U_i $(i \in I_{N_U})$ and T_j $(j \in I_{N_T})$ are destroyed when U_i attacks T_j. Let V_{T_j} $(j \in I_{N_T})$ and W_{U_i} $(i \in I_{N_U})$ represent the value of T_j and U_i, respectively.

Because the previous single-objective optimization cannot achieve simultaneous optimization of two conflicting objectives, i.e., simultaneous optimization of the costs and benefits in terms of UAVs, it is necessary to establish a multi-objective optimization model for the cooperative task assignment problem. Considering the respective probabilities of UAVs and targets being destroyed, the following two objectives are used to maximize the value of the destroyed targets while incurring the minimum cost in damaged UAVs.

(i) Maximizing the total value of the targets destroyed by UAVs:

$$\max \; f_1(x) = \sum_{i=1}^{N_U} \sum_{j=1}^{N_T} K_{ij} V_{T_j} x_{ij}, \tag{1}$$

(ii) Minimizing the total cost of the damaged UAVs:

$$\min \; f_2(x) = \sum_{i=1}^{N_U} \sum_{j=1}^{N_T} P_{ij} W_{U_i} x_{ij}, \tag{2}$$

where $x_{ij} \in \{0, 1\}$, $i \in I_{N_U}$, and $j \in I_{N_T}$. If $x_{ij} = 1$, this means that U_i attacks T_j,; otherwise, T_j is not attacked by U_i.

The UAVs have a limited amount of ammunition, which makes it impossible to allocate more tasks to each UAV than its ammunition capacity. Targets have different values, and are divided into high-value targets and low-value targets. In this paper, the values of enemy targets are known in advance. Multiple attack missions are assigned to the same high-value target in order to increase the probability of success. To decrease the time that UAVs must stay within the enemy's threat range, the following assumption is made.

Hypothesis 1 (H1). *Different tasks for the same target are performed by different UAVs.*

The implicit constraint in **H1** is that the same UAV can only attack the same target once. This constraint not being taken into account can cause the limited ammunition to be distributed unevenly, which can affect efficiency. For this reason, the number of high-value targets to be attacked needs to be limited. According to the above description, the task assignment problem needs to satisfy the following constraints.

(i) The amount of tasks assigned to each UAV cannot exceed its own ammunition capacity:

$$\sum_{j=1}^{N_T} x_{ij} \leq n_i, \ \ i \in I_{N_U}, \tag{3}$$

(ii) The number of tasks for each target is limited:

$$\sum_{i=1}^{N_U} x_{ij} \leq m_j, \ \ j \in I_{N_T}, \tag{4}$$

(iii) The same UAV can only attack the same target once:

$$a_{ij} \leq 1, \ \ i \in I_{N_U}, \ j \in I_{N_T}, \tag{5}$$

where $n_i \ (i \in I_{N_U})$ represents the ammunition capacity of U_i, $m_j \ (j \in I_{N_T})$ represents the maximum number of T_j being attacked, and $a_{ij} \ (i \in I_{N_U}, \ j \in I_{N_T})$ represents the number of tasks performed by U_i on T_j.

Let $F(x) := \big(-f_1(x), f_2(x) \big)^\top$. The multi-objective optimization problem of the cooperative task assignment (CTAMOP) is expressed as follows.

$$\begin{aligned} \text{(CTAMOP)} \ \ \min \ & F(x) \\ s.t. \ & \sum_{j=1}^{N_T} x_{ij} \leq n_i, \\ & \sum_{i=1}^{N_U} x_{ij} \leq m_j, \\ & a_{ij} \leq 1, \ \ i \in I_{N_U}, \ j \in I_{N_T}, \\ & x_{ij} \in \{0,1\}, \ i \in I_{N_U}, \ j \in I_{N_T}. \end{aligned} \tag{6}$$

3. Multi-Objective Optimization Strategy

In order to solve CTAMOP, an improved multi-objective genetic algorithm based on the NSGA-II algorithm [34] is constructed in this section. Based on the characteristics of the task assignment problem, the chromosome encoding method along with the crossover and mutation operators are specially designed and constructed.

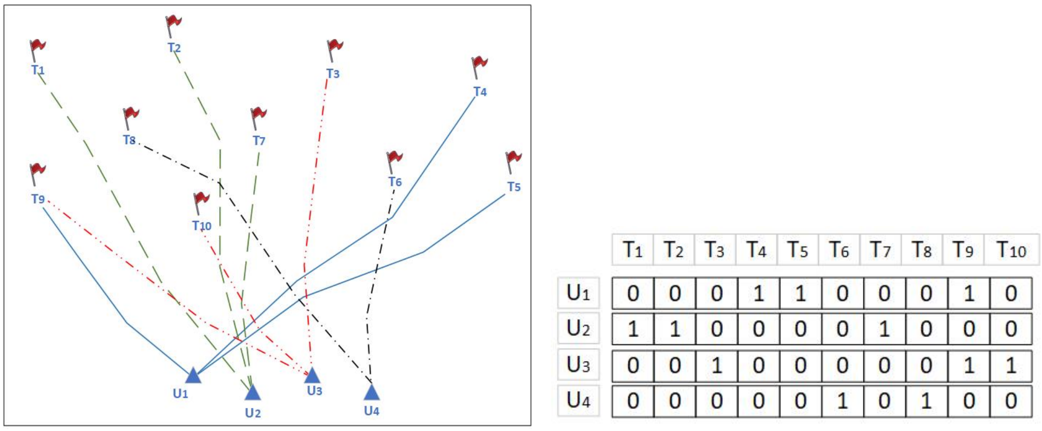
3.1. Chromosome Encoding

The task assignment problem has two characteristics. First, the problem is that multiple UAVs may attack multiple targets; second, whether $U_i \ (i \in I_{N_U})$ attacks $T_j \ (j \in I_{N_T})$ is represented by 0 and 1. Considering the above factors, we use the binary matrix encoding method to encode the chromosomes in order to accurately describe the situation of UAVs performing tasks.

For the scenario of N_U UAVs attacking N_T targets, the generated chromosomes should be $N_U \times N_T$ order matrices, which only contain 0 and 1 elements. Constraints (3)–(5) need to be satisfied, i.e., the sum of the i-th row $(i \in I_{N_U})$ of each generated matrix is less than or equal to n_i, and the sum of the j-th $(j \in I_{N_T})$ column is less than or equal to m_j. The i-th row represents the situation of the targets attacked by U_i, and the j-th column represents the situation of UAVs attacking T_j.

The chromosome encoding method is suitable for all combat situations, i.e., when ammunition is sufficient and when ammunition is insufficient. In order to ensure randomness, the chromosomes are generated in the following way. First, a UAV U_i $(i \in I_{N_U})$ and a target T_j $(j \in I_{N_T})$ are randomly selected; then, it is determined whether constraints (3)–(5) are satisfied. If both constraints are satisfied, a variable (0 or 1) is randomly assigned to x_{ij}; otherwise, $x_{ij} = 0$. This process is executed repeatedly. When all UAVs do not satisfy constraint (3) or all targets do not satisfy constraint (4) or (5), the process is terminated. At the same time, all x_{ij} $(i \in I_{N_U},\ j \in I_{N_T})$ that have not been assigned a value receive a value of 0.

Here, a specific example is provided to illustrate how chromosomes are encoded considering the case of $N_U = 4$ and $N_T = 10$. The ammunition capacity of UAVs is $n_1 = n_2 = n_3 = 3$ and $n_4 = 2$, and the maximum number of targets that can be attacked is $m_1 = m_2 = \cdots = m_{10} = 2$. For the scenario in Figure 1a, T_4, T_5, T_9 are assigned to U_1, T_1, T_2, T_7 are assigned to U_2, T_3, T_9, T_{10} are assigned to U_3, and T_6, T_8 are assigned to U_4. The values of the corresponding positions of the chromosome are set to 1, and the remaining positions are set to 0, as shown in Figure 1b.



	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
U1	0	0	0	1	1	0	0	0	1	0
U2	1	1	0	0	0	0	1	0	0	0
U3	0	0	1	0	0	0	0	0	1	1
U4	0	0	0	0	0	1	0	1	0	0

(**a**) Assignment of tasks (**b**) Chromosome form

Figure 1. Assignment of tasks and the chromosome encoding method.

In the evolution process, the offspring are composed of the retained elite individuals and the individuals obtained by crossover and mutation operations. The crossover and mutation operators play key roles. Because the chromosomes are binary matrix, it is necessary to design the crossover and mutation operators for algorithm.

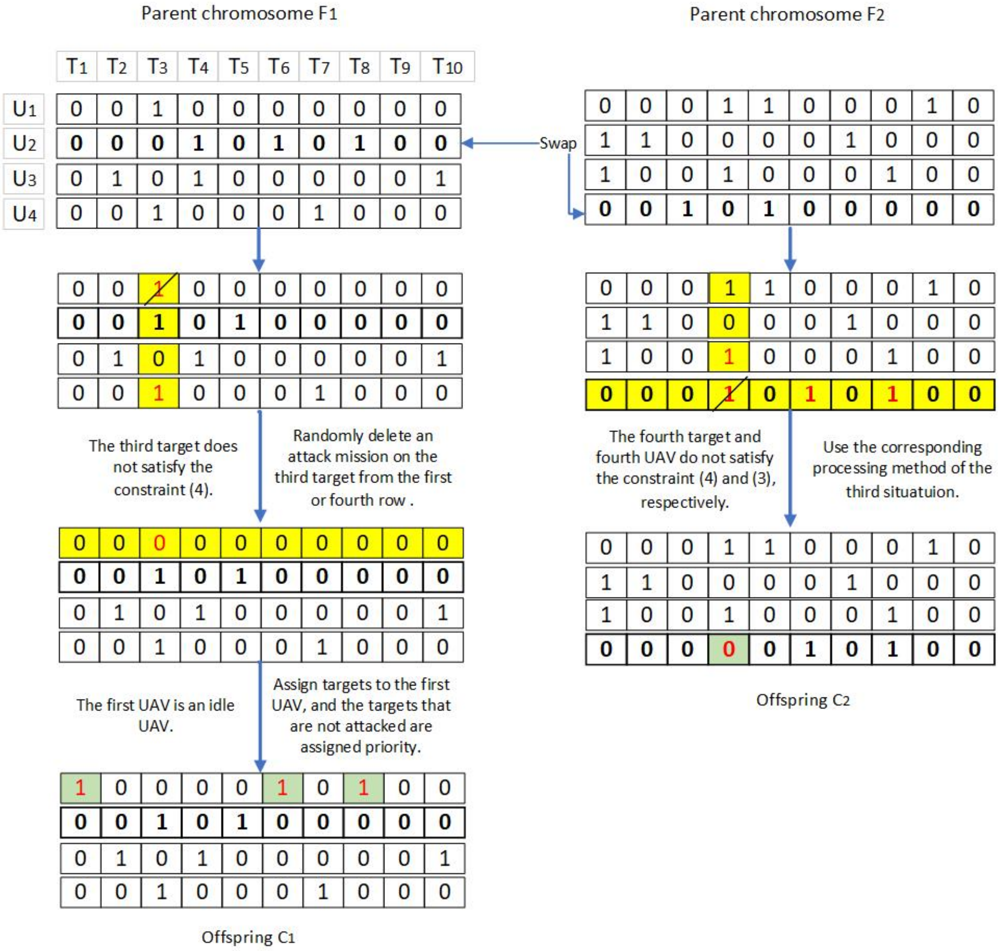
3.2. The Crossover Operator

Two chromosomes F_1 and F_2 are selected from the parents, then a crossover operation is performed on F_1 and F_2 with crossover probability P_c. The specific crossover steps are as follows. First, the l_1-th $(l_1 \in I_{N_U})$ row and the l_2-th $(l_2 \in I_{N_U})$ row from F_1 and F_2, respectively, are randomly selected. Then, the l_1-th row of F_1 and the l_2-th row of F_2 are swapped to obtain two new chromosomes O_1 and O_2. If the task assignment of the obtained chromosome satisfies constraints (3) and (4) and there is no idle UAV, then the obtained chromosome is the crossover offspring. It is common that O_1 and O_2 do not satisfy the constraints, and the following cases may exist as well.

- Task assignment satisfies constraint (4) while not satisfying constraint (3). If the l_1-th $(l_1 \in I_{N_U})$ row of O_1 does not satisfy constraint (3), that is, the number of tasks assigned to U_{l_1} exceeds the ammunition capacity n_{l_1} of U_{l_1}, we sort the tasks of U_{l_1} according to the number of tasks attacked, then delete the corresponding tasks from the task set of U_{l_1} according to their number, from high to low. When the l_1-th $(l_1 \in I_{N_U})$ row satisfies constraint (3), then the operation is stopped.
- Task assignment satisfies constraint (3) while not satisfying constraint (4). If the j-th $(j \in I_{N_T})$ column of O_2 does not satisfy constraint (4), that is, the number of attacks on T_j exceeds the upper limit m_j, the attack task of T_j is randomly deleted from the rows that have not been exchanged. If T_j satisfies the constraint (4), then the operation is stopped.
- Task assignment satisfies neither constraint (3) nor (4). In this case, the same method as in case 1 is first used to change the chromosome and then to determine whether constraint (4) is satisfied. If constraint (4) is not satisfied, then the method from case 2 is used to change the chromosome.

If both constraints (3) and (4) are satisfied and there is an idle UAV, then as many tasks as possible are assigned to the idle UAV under the premise that constraints (3)–(4) are satisfied; targets that have not yet been attacked are prioritized.

The example provided in Section 3.1 is used to illustrate the construction of the crossover operator. The two selected parent chromosomes are shown in Figure 2. Let the rows randomly selected from F_1 and F_2 be the second and fourth rows, respectively. It can be seen from Figure 2 that T_3 (i.e., the third column marked in yellow) in the offspring chromosome obtained by F_1 does not satisfy constraint (4). As this scenario belongs to the first case, an attack task of T_3 is randomly deleted from the first or fourth row. If the task of U_1 (i.e., the first row marked in yellow) is randomly deleted, U_1 becomes an idle UAV. Thus, as many tasks as possible are assigned to U_1 under the premise that the constraints are satisfied, and the crossover offspring C_1 can be obtained. Here, T_4 and U_4 (i.e., the fourth column and fourth row marked in yellow) in the offspring chromosome obtained by F_2 fail to satisfy constraints (4) and (3), respectively. This belongs to the third case. According to the method used in the third case, the attack task of U_4 is deleted. Then, as the chromosome satisfies the constraints and there are no idle UAVs, we have the crossover offspring C_2.



Parent chromosome F1

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
U1	0	0	1	0	0	0	0	0	0	0
U2	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**
U3	0	1	0	1	0	0	0	0	0	1
U4	0	0	1	0	0	0	1	0	0	0

Parent chromosome F2

0	0	0	1	1	0	0	0	1	0
1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0
0	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**

0	0	$\cancel{1}$	0	0	0	0	0	0	0
0	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0	1	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	0	0

The third target does not satisfy the constraint (4).

Randomly delete an attack mission on the third target from the first or fourth row .

0	0	0	0	0	0	0	0	0	0
0	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0	1	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	0	0

The first UAV is an idle UAV.

Assign targets to the first UAV, and the targets that are not attacked are assigned priority.

1	0	0	0	0	1	0	1	0	0
0	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0	1	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	0	0

Offspring C1

0	0	0	1	1	0	0	0	1	0
1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0
0	**0**	**0**	$\cancel{1}$	**0**	**1**	**0**	**1**	**0**	**0**

The fourth target and fourth UAV do not satisfy the constraint (4) and (3), respectively.

Use the corresponding processing method of the third situatuion.

0	0	0	1	1	0	0	0	1	0
1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0
0	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**

Offspring C2

Figure 2. Example of crossover operator.

3.3. The Mutation Operator

A mutation operation is performed on crossover offspring with a mutation probability of P_m. There are three situations that may occur with respect to the crossover offspring, and different mutation operations are used for different situations.

- If there are targets in the chromosome that have not been attacked and there are UAVs with ammunition that can perform tasks, then the mutation operation seeks to assign the targets to these UAVs under the premise that constraint (3) is satisfied.
- If there are targets in the chromosome that have not been attacked and the UAVs have no remaining ammunition, then the mutation operation randomly selects the task sets of two UAVs from the chromosome and exchanges them.
- If all targets are attacked, then the task sets of two UAVs from the chromosome are randomly selected and exchanged.

3.4. The Improved Multi-Objective Genetic Algorithm

Based on the methods used to construct the chromosomes, crossover operators and mutation operators, the following multi-objective optimization algorithm (i.e., Algorithm 1) suitable for UAV cooperative task assignment is developed.

Algorithm 1 Multi-objective optimization algorithm.

```
1: Initialize the values of the following parameters: the population size $N$, crossover
   probability $P_c$, mutation probability $P_m$, and maximum number of evolutions $G$;
2: $g \Leftarrow 0$;
3: Randomly generate an initial population $P_g$ of size $N$. The chromosome encoding
   method and chromosome generation method are based on the methods described in
   Section 3.1;
4: while g < G do
5:     Calculate the objective function values of each chromosome in population $P_g$ using
       (1) and (2);
6:     Sort population $P_g$ using the fast non-dominated sorting approach, then determine
       the front of each chromosome;
7:     Perform crossover operations on the selected parents according to the crossover
       probability $P_c$, then mutate the obtained crossover offspring with probability $P_m$. Proceed
       to the next step when a progeny population $S_g$ with $N$ chromosomes is obtained;
8:     Combine the parent population $P_g$ with the offspring population $S_g$ to obtain a new
       population $Q_g$; then, the size of population $Q_g$ is $2N$;
9:     Perform the process in lines 5–6 on population $Q_g$ to obtain the front of each
       chromosome in $Q_g$;
10:    Select chromosomes from the front;
11:    while the number of chromosomes selected is less than $N$ do
12:        First, the chromosome is selected from the first front, then, the chromosome is
           selected from the second front, and so on;
13:        if the number of chromosomes required is less than the number of chromosomes
           in the $l$-th front then
14:            Calculate the crowded distances;
15:            Select the chromosomes based on the crowding distance from large to small;
16:        end if
17:    end while
18:    $g \Leftarrow g + 1$ and $q \Leftarrow g$
19:    if $g = G$ then
20:        Sort $P_g$ using the fast non-dominated sorting approach, then output the chromosomes
           in the first front;
21:    end if
22: end while
```

4. Selection Strategy

Because the decision-maker needs to select one or more solutions from the Pareto solution set in order to perform specific operations, it is necessary to have a strategy for selecting non-dominated solutions from the Pareto-optimal front. Let f_j denote the j-th objective function, n denote the number of objective functions, and m denote the number of non-dominated solution on the Pareto-optimal front. The specific selection steps are shown in Algorithm 2.

The specific value of α_j $(j \in I_n)$ depends on the degree of preference of the decision-maker for the objective function f_j. Let $C = \{c_1, c_2, \cdots, c_n\}$, where c_j $(j \in I_n)$ represents the degree of preference of the decision-maker for f_j. The rules for setting the value of α_j $(j \in I_n)$ are as follows:

- If $c_{j_1} > c_{j2} > \cdots > c_{jn}$, then $\alpha_{j_1} > \alpha_{j2} > \cdots > \alpha_{jn}$, and $\sum\limits_{l=j_1}^{j_n} \alpha_l = 1, j_i \in I_n,\ i \in I_n$.
- If $c_{j_1} = c_{j_2}$, then $\alpha_{j_1} = \alpha_{j2}$, $\forall\ j_1,\ j2 \in I_n$. In particular, if $c_{j_1} = c_{j_2} = \cdots = c_{jn}$, then $\alpha_{j_1} = \alpha_{j_2} = \cdots = \alpha_{jn} = \frac{1}{n}$.

Finally, the flowchart diagram of the improved genetic algorithm together with the selection strategy is shown in Figure 3.

Algorithm 2 The strategy for selecting solutions.

```
1: Convert all objective functions $f_j\ (j \in I_n)$ into a form that minimizes or maximizes the function $F_j\ (j \in I_n)$.
2: Solve the problem using the improved NSGA-II algorithm to obtain the Pareto-optimal front.
3: If the dimensions of any two objective functions are different or the orders of magnitude of the values of any two objective functions are different, then normalize the objective function values; otherwise, the objective function values are not normalized. The $j$-th objective function of the $i$-th non-dominated solution on the Pareto-optimal front is denoted as $\bar{F}_j^i\ (j \in I_n,\ i \in I_m)$.
4: Weight and sum each group of objective function values obtained in step 3 to obtain the following set:
       $$S = \left\{ S_i | S_i = \sum_{j=1}^{n} \alpha_j \bar{F}_j^i,\ \sum_{j=1}^{n} \alpha_j = 1,\ i \in I_m \right\}.$$
5: In step 1, if the objective functions are transformed into the form of seeking the minimum value, then the solution corresponding to the minimum value in set $S$ obtained in step 4 is selected; otherwise, the solution corresponding to the maximum value in set $S$ is selected.
```

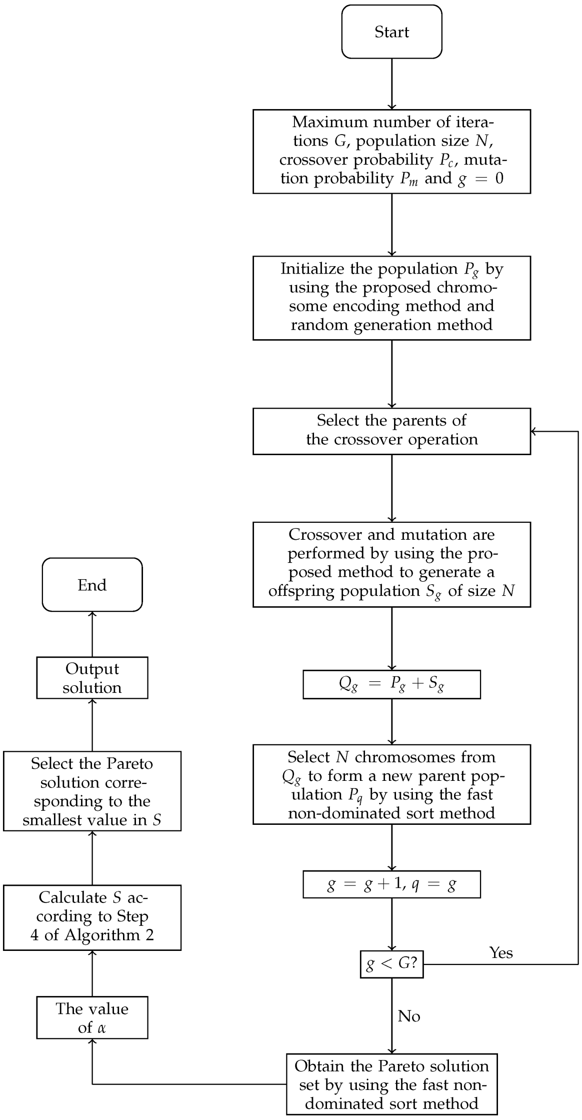

Start
Maximum number of iterations G, population size N, crossover probability P_c, mutation probability P_m and $g = 0$
Initialize the population P_g by using the proposed chromosome encoding method and random generation method
Select the parents of the crossover operation
Crossover and mutation are performed by using the proposed method to generate a offspring population S_g of size N
$Q_g = P_g + S_g$
Select N chromosomes from Q_g to form a new parent population P_q by using the fast non-dominated sort method
$g = g + 1, q = g$
$g < G?$
Yes
No
Obtain the Pareto solution set by using the fast non-dominated sort method
The value of α
Calculate S according to Step 4 of Algorithm 2
Select the Pareto solution corresponding to the smallest value in S
Output solution
End

Figure 3. The flowchart of task assignment.

5. Task Re-Assignment For Emergencies

The task assignment process is carried out on the enemy targets found on the battlefield. However, there are many uncertain factors due to the complexity of the battlefield environment. For example, after assigning the discovered targets, new enemy targets may be found on the battlefield again, or certain UAVs in the task assignment may suddenly malfunction and be unable to continue performing combat tasks. Both new targets and targets in the task sets of damaged UAVs need to be re-assigned. There are two methods of assigning tasks. The first method is to assign all tasks according to the current health status of the UAVs, and the second is to assign tasks that need to be assigned based on the obtained task assignment plan. In combat, efficiency issues are more important when solving unexpected situations. Compared with the first method, the second method takes less time and has a higher task assignment efficiency.

In this paper, the re-assignment strategy for new tasks is constructed based on the idea of the contract network protocol and the characteristics of the task assignment problem. Smith first proposed the idea of a contract network protocol [30]. The principle idea of a contract network protocol is to assign tasks through a process of tendering and bidding between agents. There are three types of agents in a contract network protocol: the tender agent, bidding agent, and winning agent. In the problem of UAV task re-assignment, the tender agent is the reconnaissance UAV that discovers a new task or the UAVs that becomes damaged, the bidding agents are UAVs with the ability to perform the newly available tasks, and the winning agent is the UAV corresponding to the bid with the best function value.

5.1. Task Re-Assignment Model

Based on the above description, there are two trigger conditions for task reassignment: (1) new enemy targets are discovered and (2) one or more UAVs are damaged. Suppose that the number of new targets found on the battlefield is s_1, and the number of destroyed UAVs is s_2. Let $\bar{T}$ denote the set of targets that need to be assigned,

$$\bar{T} := \{T_{N_T+1}, T_{N_T+2}, \cdots, T_{N_T+s_1}\},$$

where T_{N_T+i} $(i \in I_{s_1})$ represents the i-th task that needs to be assigned. Let $\bar{U}$ denote the set of UAVs that can perform tasks,

$$\bar{U} := \{U_{l_1}, U_{l_2}, U_{l_{N_U-s_2}}\} = U - \{U_{\bar{l}_1}, U_{\bar{l}_2}, \cdots, U_{\bar{l}_{s_2}}\},$$

where $U_{\bar{l}_j}$ $(j \in I_{s_2})$ represents the $\bar{l}_j$-th damaged UAV. Based on (1) and (2), an objective function is constructed with the following form:

$$\begin{aligned} \max \bar{f} \quad &= \alpha_1 \sum_{i=l_1}^{l_{N_U-s_2}} K_{i(N_T+j)} V_{T_{N_T+j}} x_{i(N_T+j)} \\ &+\alpha_2 \sum_{i=l_1}^{l_{N_U-s_2}} \left(1 - P_{i(N_T+j)}\right) W_{U_i} x_{i(N_T+j)} - \alpha_1 \sum_{i=l_1}^{l_{N_U-s_2}} K_{ij_r} V_{T_{j_r}} x_{ij_r} \\ &-\alpha_2 \sum_{i=l_1}^{l_{N_U-s_2}} \left(1 - P_{ij_r}\right) W_{U_i} x_{ij_r}, \end{aligned} \tag{7}$$

where $j \in I_{s_1}$ and T_{j_r} $(j_r \in I_{N_T+j-1})$ represents the target replaced by T_{N_T+j}; T_{j_r} is only considered in an interchange contract, which is described along with sales contracts in the next section. The values of α_1 and α_2 are the same as the values of α_1 and α_2 in Algorithm 2 in Section 4. Thus, the model of task re-assignment (TRAM) is as follows:

$$
\begin{aligned}
(\text{TRAM})\ \max\ & \bar{f} \\
s.t.\ & \sum_{\tilde{j}=1}^{N_T+j} x_{i\tilde{j}} \le n_i, \\
& \sum_{i=l_1}^{l_{N_U-s_2}} x_{i(N_T+j)} \le m_j, \\
& a_{i(N_T+j)} \le 1,\ \ i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}, \\
& x_{i(N_T+j)} \in \{0,1\},\ \ i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\},
\end{aligned} \tag{8}
$$

where $j \in I_{s_1}$.

5.2. Task Re-Assignment Algorithm

In the CNP-based algorithm, sales contracts and interchange contracts are considered. The idea of a sales contract is that a new task is added to the task set of the bidding agent. The specific form of a sales contract of a UAV U_i for a target T_{N_T+j} is represented as follows:

$$< U_i,\ T_{N_T+j},\ 0,\ \bar{f} >, \tag{9}$$

where, $i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}$, $j \in I_{s_1}$. The sales contract (9) indicates that the revenue obtained by adding target T_{N_T+j} to the task set of U_i is $\bar{f}$. An interchange contract, on the other hand, replaces a target in the UAV task set with a target that needs to be assigned. The specific form of an interchange contract of a UAV U_i for a target T_{N_T+j} is represented as follows:

$$< U_i,\ T_{N_T+j},\ T_{j_r},\ \bar{f} >, \tag{10}$$

where $i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}$, $j \in I_{s_1}$, and $j_r \in I_{N_T+j-1}$. The interchange contract (10) indicates that the revenue obtained by replacing T_{j_r} in the task set of U_i with T_{N_T+j} is $\bar{f}$. If UAV U_i ($i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}$) has ammunition remaining, then U_i can execute both the sales contract and the interchange contract; otherwise, U_i can only execute the interchange contract. The steps for assigning target T_{N_T+j} ($j \in I_{s_1}$) are provided in Algorithm 3.

Algorithm 3 Task re-assignment algorithm T_{N_T+j} ($j \in I_{s_1}$).

```
1: Initialize the information of UAV $U_i$, the values of parameters $K_{i(N_T+j)}$, $P_{i(N_T+j)}$, $V_{T_{N_T+j}}$ ($i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}$, $j \in I_{s_1}$), and the number of iterations of interchange contract $G_{ic}$.
2: Calculate the remaining ammunition $R_i^m$ ($i \in \{l_1, l_2, \cdots, l_{N_U-s_2}\}$) of $U_i$.
3: If $R_i^m \neq 0$, then $U_i$ executes the interchange contract and sales contract for $T_{N_T+j}$; otherwise, $U_i$ only executes the interchange contract for $T_{N_T+j}$. Then, bidding agent $U_i$ chooses the contract with the largest value of $\bar{f}$.
4: The bidding agent evaluates the received contracts and selects the contract with the largest value of $\bar{f}$ as the winning contract, then broadcasts the information of winning agent.
```

The flowchart diagram of the CNP-based algorithm is shown in Figure 4.

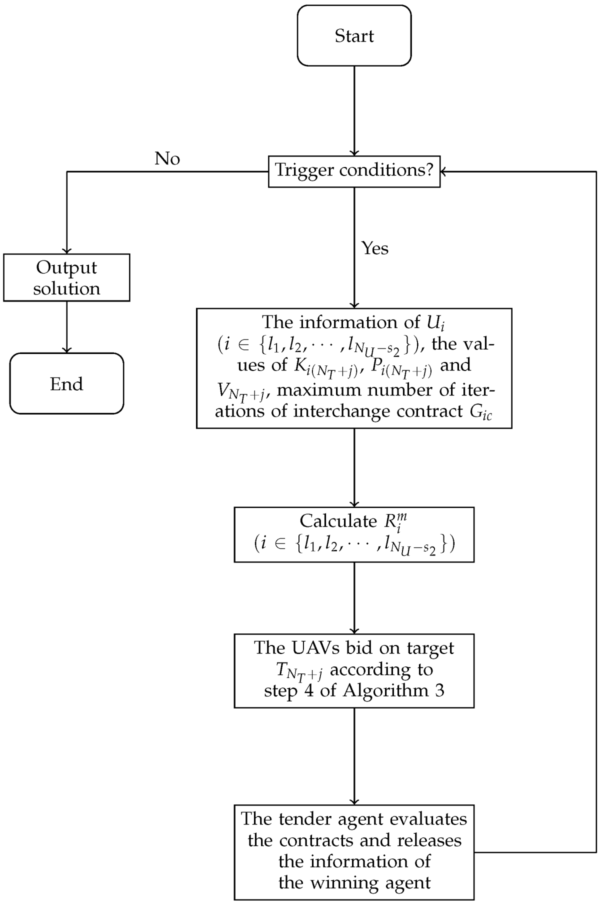

Start
Trigger conditions?
No
Yes
Output solution
End
The information of U_i ($i \in \{l_1, l_2, \cdots, l_{N_U - s_2}\}$), the values of $K_{i(N_T+j)}$, $P_{i(N_T+j)}$ and V_{N_T+j}, maximum number of iterations of interchange contract G_{ic}
Calculate R_i^m ($i \in \{l_1, l_2, \cdots, l_{N_U - s_2}\}$)
The UAVs bid on target T_{N_T+j} according to step 4 of Algorithm 3
The tender agent evaluates the contracts and releases the information of the winning agent

Figure 4. The flowchart of task re-assignment

6. Numerical Experiments

In order to verify the effectiveness of Algorithms 1–3, simulation examples of various battle situations are analyzed in this section. Examples of the improved multi-objective genetic algorithm and selection strategy are provided in Section 6.1. In Sections 6.1.1 and 6.1.2, examples where the total amount of ammunition is less than or equal to the total number of tasks are considered, and the effectiveness of Algorithm 1 is verified. In Section 6.1.3, a large-scale example where the total amount of ammunition is greater than the total number of tasks is provided. Examples of task re-assignment in emergencies are laid out in Section 6.2. In Section 6.2.1, an example is used to illustrate the process of Algorithm 3. A large-scale example considering emergencies is provided in Section 6.2.2 to verify the effectiveness and advantages of Algorithm 3. In Section 6.2.3, we analyze why the CNP-based method is not directly used to solve the task assignment problem. All numerical experiments were implemented using Python 3.8 on a computer with an Intel Core i5-10210U CPU @ 1.60GHz, 2.11 GHz, and 4.00 GB RAM. The code for the algorithms can be found at the URL (accessed on 6 November 2022) https://github.com/gaoxh-github/multi-objetive-task-assignment-source-code. For the different examples provided in this paper, the reader only needs to change the corresponding parameters in the code.

6.1. Test of the Improved Multi-Objective Genetic Algorithm

6.1.1. Case 1: Total Amount of Ammunition = Total Number of Tasks

First, we consider a case in which $N_U = 4$ and $N_T = 8$. The ammunition capacity of each UAV is $n_i = 2$ ($i \in I_4$), and the upper limit of each target being attacked is $m_j = 1$ ($j \in I_8$). When U_i ($i \in I_4$) attacks T_j ($j \in I_8$), the probability P_{ij} of U_i being destroyed and the probability K_{ij} of T_j being destroyed are shown in Table 1. The values of U_i and T_j are shown in Table 2. To verify the effectiveness of Algorithm 1, we compare it with the Multiple Objective Particle Swarm Optimization (MOPSO) algorithm using the example in this section. In Algorithm 1, the crossover probability is $P_c = 0.8$ and the mutation probability is $P_m = 0.2$. In MOPSO, the values of the parameters are $\omega = 0.7298$, $c_1 = 1.49618$ and $c_2 = 1.49618$. The population size and maximum number of iterations of the two algorithms are $N = 100$ and $G = 200$, respectively.

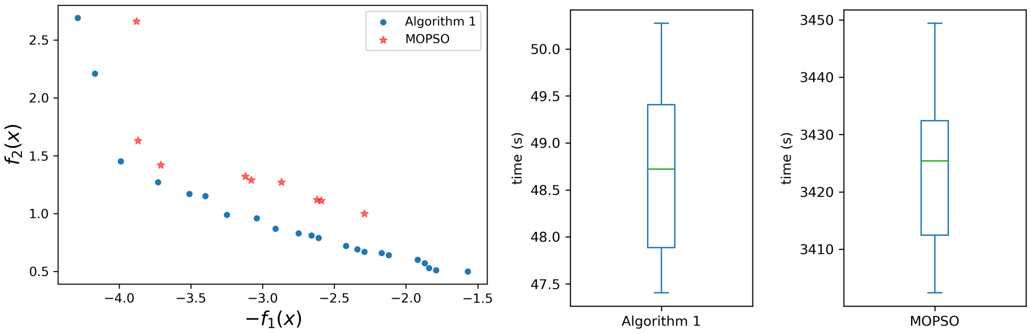
The comparison results of the obtained Pareto-optimal fronts and CPU runtime are shown in Figure 5. According to the definition of the Pareto solution [34], it can be seen from Figure 5a that all the non-dominated solutions obtained by Algorithm 1 dominate the non-dominated solution obtained by MOPSO. As can be seen from Figure 5b, the CPU runtime of GA-CTAP is significantly shorter than that of MOPSO. The convergence curves of the objective functions are shown in Figure 6. Clearly, thte GA-CTAP algorithm achieves better convergence performance compared to MOPSO. For convenient description, the task assignment corresponding to the non-dominated solutions A, B, and C are provided in Table 3.

Table 1. Probabilities of UAVs and targets being destroyed.

		T$_1$	**T$_2$**	**T$_3$**	**T$_4$**	**T$_5$**	**T$_6$**	**T$_7$**	**T$_8$**
U$_1$	P_{1j}	0.16	0.65	0.14	0.16	0.44	0.14	0.35	0.25
	K_{1j}	0.5	0.8	0.3	0.4	0.5	0.6	0.6	0.7
U$_2$	P_{2j}	0.14	0.16	0.44	0.14	0.65	0.16	0.45	0.35
	K_{2j}	0.8	0.4	0.4	0.8	0.7	0.6	0.8	0.6
U$_3$	P_{3j}	0.44	0.14	0.16	0.16	0.14	0.65	0.35	0.18
	K_{3j}	0.6	0.5	0.8	0.3	0.8	0.3	0.7	0.7
U$_4$	P_{4j}	0.65	0.14	0.16	0.44	0.08	0.16	0.55	0.48
	K_{4j}	0.6	0.4	0.2	0.3	0.3	0.6	0.5	0.6

Table 2. Values of UAVs and targets.

Target	T$_1$	T$_2$	T$_3$	T$_4$	T$_5$	T$_6$	T$_7$	T$_8$
Value	0.62	0.65	0.68	0.7	0.73	0.78	0.81	0.85
UAV	U$_1$	U$_2$	U$_3$	U$_4$				
Value	0.8	1.1	0.9	1.3				



(**a**) Pareto-optimal front of case 1

(**b**) CPU runtime

Figure 5. Comparison of results.

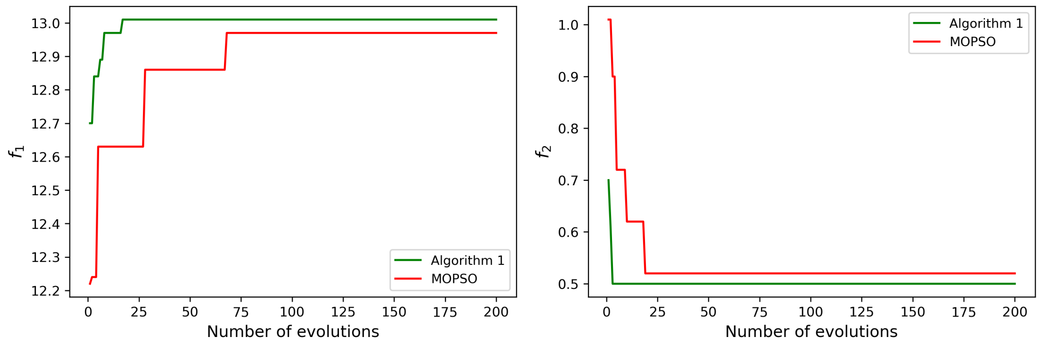


Figure 6. Convergence curves of objective functions for case 1.

Table 3. Specific task assignment of non-dominated solutions *A*, *B*, *C*.

	A	**B**	**C**
x	0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 1	0 0 0 0 0 1 0 0
	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0
	0 0 1 0 1 0 0 0	0 0 1 0 1 0 0 0	0 1 0 0 0 0 0 1
	0 1 0 0 0 1 0 0	0 0 0 0 0 1 0 0	0 0 0 0 1 0 0 0

Case 1 considers a situation in which the total amount of ammunition is equal to the number of tasks. However, in actual combat it is possible that the number of tasks is more or less than the amount of ammunition.

6.1.2. Case 2: Total Amount of Ammunition < Total Number of Tasks

In this section, we analyze a case in which four UAVs attack twenty targets. For convenience of calculation, it is assumed that each UAV has the same amount of ammunition $n_i = 4$ $(i \in I_4)$. At the same time, in order to simplify the description, the parameter information of multiple UAVs carrying out attacking tasks on the first eight targets takes the values in Tables 1 and 2. When U_i $(i \in I_4)$ attacks T_j $(j \in \{9, 10, \cdots, 20\})$, the probability P_{ij} of U_i being destroyed, the probability K_{ij} of T_j being destroyed, and the value of T_j $(j \in \{9, 10, \cdots, 20\})$ are provided in Table 4.

Table 4. Probabilities of UAVs and targets being destroyed and target values.

		T$_9$	**T$_{10}$**	**T$_{11}$**	**T$_{12}$**	**T$_{13}$**	**T$_{14}$**	**T$_{15}$**	**T$_{16}$**	**T$_{17}$**	**T$_{18}$**	**T$_{19}$**	**T$_{20}$**
U$_1$	P_{1j}	0.21	0.15	0.32	0.18	0.46	0.23	0.55	0.14	0.35	0.16	0.26	0.25
	K_{1j}	0.7	0.6	0.5	0.6	0.4	0.6	0.4	0.3	0.3	0.8	0.3	0.7
U$_2$	P_{2j}	0.30	0.24	0.51	0.44	0.44	0.30	0.42	0.15	0.16	0.18	0.08	0.48
	K_{2j}	0.6	0.7	0.7	0.3	0.3	0.5	0.3	0.5	0.3	0.7	0.3	0.6
U$_3$	P_{3j}	0.53	0.16	0.44	0.68	0.14	0.25	0.50	0.48	0.14	0.09	0.13	0.60
	K_{3j}	0.7	0.4	0.6	0.5	0.3	0.5	0.4	0.6	0.3	0.8	0.4	0.8
U$_4$	P_{4j}	0.65	0.30	0.42	0.08	0.50	0.18	0.48	0.15	0.16	0.20	0.15	0.60
	K_{4j}	0.7	0.6	0.6	0.7	0.4	0.7	0.6	0.5	0.4	0.8	0.4	0.8
Target	value	0.72	0.78	0.62	0.65	0.76	0.88	0.63	0.7	0.68	0.82	0.75	0.81

The obtained Pareto-optimal front is shown in Figure 7. It can be seen that the non-dominated solutions are evenly distributed and the population diversity is good. Figure 8 shows the changes of the maximum value of f_2 and the minimum value of f_1 in each iteration over the course of the entire iteration when $n_i = 4$ $(i \in I_4)$. It can be seen that as the evolutionary algebra increases, the values of the two functions converge.

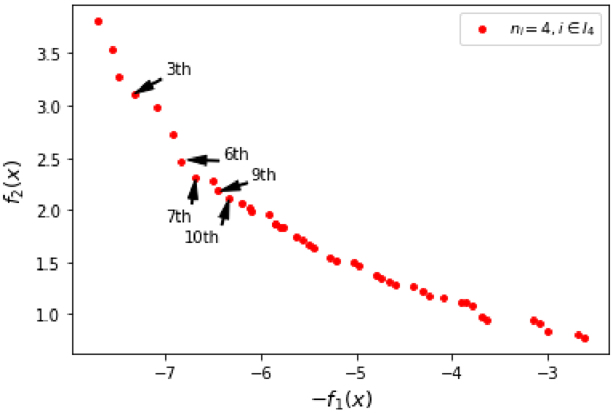


Figure 7. Pareto-optimal front of case 2.

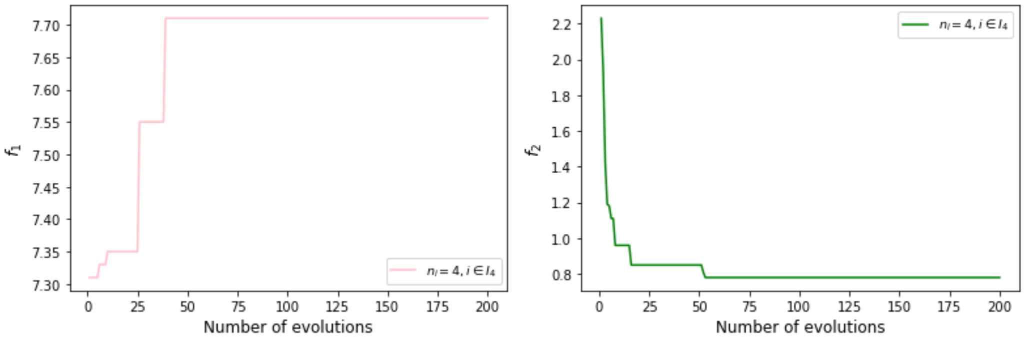


Figure 8. Convergence curves of objective functions for case 2.

The process of selecting solutions from the Pareto solution set using the selection strategy is as follows. Let $S := \{S_0, S_1, \cdots, S_{44}\}$ represent the result obtained in Step 4 of Algorithm 2, where $S_i = \alpha_1 F_1^i + \alpha_2 F_2^i$, $F_1^i = -f_1^i$, $F_2^i = f_2^i$ $(i \in \bar{I}_{44})$, α_1, and α_2 represent the weights and (F_1^i, F_2^i) represents the i-th non-dominated solution in the Pareto-optimal front. Based on this example and the description of weights in Section 4, the rules for setting the values of α_i $(i \in I_2)$ are as follows:

- If $c_1 > c_2$, then $\alpha_1 \in (0.5, 1]$.
- If $c_1 = c_2$, then $\alpha_1 = \alpha_2 = 0.5$.
- If $c_1 < c_2$, then $\alpha_1 \in [0, 0.5)$.

For the Pareto-optimal front shown in Figure 7, the solutions from the upper left corner to the lower right corner are the 0th to the 44th non-dominated solutions. In this example, the objective function values are not normalized because the dimensions of f_1 and f_2 are the same, and the order of magnitude of the values of f_1 and f_2 is the same at $2.62 \leq f_1 \leq 7.71$ and $0.78 \leq f_2 \leq 3.81$, respectively. Assuming the weights $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, the results obtained by selection strategy are shown in Table 5.

As can be seen from Table 5, the sixth and seventh non-dominated solutions on the Pareto-optimal front are the best choices, while the 44th non-dominated solution is the worst choice. For the sake of simplifying the description, only the non-dominated solution information and the specific task assignment corresponding to the first five choices are shown. The positions of the five solutions are shown in Figure 7, and the corresponding specific task assignments are shown in Table 6.

The above description represents the process of selecting solutions of the multi-objective task assignment problem containing two objective functions. For general multi-objective optimization problems, the process of selecting solutions from the Pareto-optimal front is similar to the above process, and the values of weights are set according to the rules in Section 4.

Table 5. The order of selection of non-dominated solutions.

S_i	−2.185	−2.185	−2.135	−2.11	−2.105	−2.105	−2.1	−2.1	−2.065	−2.055	−2.05	−2.05
i	6	7	9	10	3	8	2	5	11	13	4	12
S_i	−2.01	−1.99	−1.99	−1.98	−1.975	−1.97	−1.95	−1.94	−1.925	−1.915	−1.9	−1.86
i	1	15	16	14	17	18	0	19	20	21	22	23
S_i	−1.85	−1.765	−1.755	−1.715	−1.7	−1.675	−1.655	−1.565	−1.545	−1.53	−1.465	−1.395
i	24	25	26	27	28	29	30	31	32	33	34	35
S_i	−1.37	−1.355	−1.355	−1.34	−1.105	−1.09	−1.08	−0.935	−0.92			
i	36	37	38	39	40	41	42	43	44			

Table 6. The information of the first five non-dominated solutions

	$(-f_1, f_2)$	x
6th	(−6.84, 2.47)	$\begin{matrix} 0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&0 \end{matrix}$
7th	(−6.68, 2.31)	$\begin{matrix} 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&0 \end{matrix}$
9th	(−6.45, 2.18)	$\begin{matrix} 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&0 \end{matrix}$
10th	(−6.33, 2.11)	$\begin{matrix} 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0 \end{matrix}$
3th	(−7.32, 3.11)	$\begin{matrix} 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&1 \end{matrix}$

Let M_{U_i} ($i \in I_{N_U}$) denote the task set of UAV U_i. According to the meaning of chromosomes, the task assignment information corresponding to the five non-dominated solutions in Table 6 is provided in Table 7.

Table 7. The task assignment schemes of the solutions in Table 6.

	UAV	Task Set		UAV	Task Set
6th	U_1	$M_{U_1} = \{T_8, T_9, T_{10}, T_{11}\}$	7th	U_1	$M_{U_1} = \{T_7, T_8, T_9, T_{10}\}$
	U_2	$M_{U_2} = \{T_1, T_4, T_6\}$		U_2	$M_{U_2} = \{T_1, T_4, T_6\}$
	U_3	$M_{U_3} = \{T_2, T_3, T_5, T_7\}$		U_3	$M_{U_3} = \{T_2, T_3, T_5, T_{13}\}$
	U_4	$M_{U_4} = \{T_{12}, T_{14}, T_{16}\}$		U_4	$M_{U_4} = \{T_{12}, T_{14}, T_{16}\}$
9th	U_1	$M_{U_1} = \{T_7, T_8, T_9, T_{10}\}$	10th	U_1	$M_{U_1} = \{T_7, T_8, T_9, T_{10}\}$
	U_2	$M_{U_2} = \{T_1, T_4, T_6\}$		U_2	$M_{U_2} = \{T_1, T_4, T_6\}$
	U_3	$M_{U_3} = \{T_2, T_3, T_5\}$		U_3	$M_{U_3} = \{T_2, T_3, T_5, T_{13}\}$
	U_4	$M_{U_4} = \{T_{12}, T_{14}, T_{16}\}$		U_4	$M_{U_4} = \{T_{12}, T_{14}\}$
3th	U_1	$M_{U_1} = \{T_7, T_8, T_9, T_{10}\}$			
	U_2	$M_{U_2} = \{T_1, T_4, T_6\}$			
	U_3	$M_{U_3} = \{T_2, T_3, T_5, T_{13}\}$			
	U_4	$M_{U_4} = \{T_{12}, T_{18}, T_{19}, T_{20}\}$			

6.1.3. Case 3: Total Amount of Ammunition > Total Number of Tasks

In this section, a large-scale task assignment example of 15 UAVs attacking 100 targets is considered. The ammunition capacity of each UAV is $n_i = 7$ $(i \in I_{15})$, and the upper limit of each target being attacked is $m_j = 1$ $(j \in I_{100})$. Considering the length of the paper, the values of parameters V_{T_j}, W_{U_i}, P_{ij}, and K_{ij} $(i \in I_{15},\ j \in I_{100})$ are not provided. The values of these parameters can be downloaded from the website (accessed on 6 November 2022) https://github.com/gaoxh-github/Values-of-parameters. In Algorithm 1, the values of the parameters are as follows: $N = 100$, $G = 200$, $P_c = 0.8$, $P_m = 0.2$. The obtained Pareto-optimal front is shown in Figure 9. The average time for this scale of experiment is about 60 min.

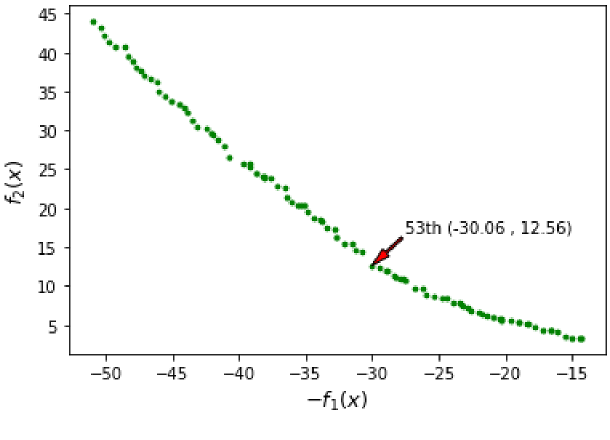


Figure 9. Pareto-optimal front of case 3.

Algorithm 2 is used to select the solution from the Pareto solution set in Figure 9. In Algorithm 2, the values of weights α_1 and α_2 are $\alpha_1 = \alpha_2 = 0.5$. The obtained results are shown in Table 8. It can be seen from Table 8 that the 53rd non-dominated solution should be selected. The information of the 53rd non-dominated solution is shown in Figure 9 and the task assignment scheme of the 53rd non-dominated solution is shown in Table 9.

To further verify the effectiveness of the proposed task assignment algorithm combined with the solution selection strategy, in this section we compare the algorithms developed in this paper with the well-developed Gurobi optimization solver. There are three methods for solving multi-objective optimization problems in the Gurobi solver, namely, Blend, Hierarchical, and a combination of these two methods. In this paper, after obtaining the Pareto front of the task assignment problem using Algorithm 1, the decision-maker can be assisted in selecting a solution from the set of non-dominated solutions based on

Algorithm 2, which is constructed based on the weights. Considering the idea behind the construction of Algorithm 2, we compared Algorithm 1 and Algorithm 2 with the Blend method in Gurobi in order to better reflect the comparison results. The comparison process is as follows. First, given nine different sets of weights, the solution of the example provided in this section is solved for each set of weights using Blend in Gurobi. Second, the Pareto front of the example is solved using Algorithm 1, then the solution is selected from the Pareto front under each set of weights using Algorithm 2. Finally, the solutions obtained by the two methods are compared for each case of weights. Table 10 shows the results of the comparison between Gurobi and the algorithms in this paper under nine sets of weights. As can be seen in Table 10, for each set of weights, the solutions developed this paper and those obtained by Gurobi are not dominated by each other, i.e., both are non-dominated solutions of the problem. Although it cannot be proven that the algorithms in this paper are better than the Gurobi solver, it can be seen that the proposed algorithms are not inferior to Gurobi in terms of solution quality.

Table 8. The order of selecting non-dominated solutions.

S_i	−8.75	−8.56	−8.555	−8.55	−8.54	−8.525	−8.515	−8.51	−8.46	−8.44	−8.375	−8.355
i	53	64	54	58	55	57	62	56	59	61	60	51
S_i	−8.355	−8.35	−8.275	−8.165	−8.14	−8.12	−8.095	−8.03	−7.995	−7.935	−7.92	−7.91
i	65	49	63	48	66	52	68	50	67	46	73	70
S_i	−7.88	−7.87	−7.865	−7.83	−7.78	−7.745	−7.745	−7.715	−7.715	−7.665	−7.655	−7.62
i	43	69	72	71	47	74	75	44	45	76	42	38
S_i	−7.61	−7.55	−7.485	−7.46	−7.405	−7.375	−7.325	−7.315	−7.155	−7.13	−7.13	−7.12
i	39	37	40	77	41	80	78	79	30	26	35	33
S_i	−7.045	−7.04	−7.025	−6.995	−6.975	−6.95	−6.915	−6.81	−6.805	−6.775	−6.605	−6.56
i	31	81	32	36	29	27	34	83	82	28	84	85
S_i	−6.525	−6.51	−6.42	−6.34	−6.335	−6.28	−6.19	−6.19	−6.165	−6.125	−6.085	−5.995
i	25	86	87	24	20	23	22	88	89	21	19	91
S_i	−5.965	−5.85	−5.815	−5.675	−5.605	−5.58	−5.57	−5.53	−5.52	−5.48	−5.06	−5.02
i	90	18	92	15	14	17	16	93	94	13	10	11
S_i	−4.965	−4.92	−4.825	−4.49	−4.4	−4.285	−4.2	−3.935	−3.915	−3.57	−3.465	
i	12	9	8	7	6	4	3	2	5	1	0	

Table 9. Task assignment of the 53rd non-dominated solution in Figure 9.

UAV	Task Set	UAV	Task Set
U_1	$M_{U_1} = \{T_8, T_{27}, T_{29}, T_{41}, T_{48}, T_{96}, T_{98}\}$	U_2	$M_{U_2} = \{T_{19}, T_{44}, T_{64}, T_{65}, T_{79}, T_{86}\}$
U_3	$M_{U_3} = \{T_{22}, T_{24}, T_{62}, T_{80}, T_{89}, T_{91}\}$	U_4	$M_{U_4} = \{T_{16}, T_{33}, T_{43}, T_{71}\}$
U_5	$M_{U_5} = \{T_{34}\}$	U_6	$M_{U_6} = \{T_3, T_{68}\}$
U_7	$M_{U_7} = \{T_{10}, T_{23}, T_{78}, T_{83}, T_{85}\}$	U_8	$M_{U_8} = \{T_{21}, T_{45}\}$
U_9	$M_{U_9} = \{T_5, T_{11}, T_{32}, T_{36}, T_{54}, T_{57}, T_{75}\}$	U_{10}	$M_{U_{10}} = \{T_7, T_{66}\}$
U_{11}	$M_{U_{11}} = \{T_{73}\}$	U_{12}	$M_{U_{12}} = \{T_{13}, T_{38}, T_{74}, T_{76}, T_{77}, T_{93}\}$
U_{13}	$M_{U_{13}} = \{T_6, T_9\}$	U_{14}	$M_{U_{14}} = \{T_{99}\}$
U_{15}	$M_{U_{15}} = \{T_{28}, T_{52}, T_{55}, T_{84}, T_{88}\}$		

Table 10. Comparison results of the proposed algorithms and the multi-objective method in Gurobi.

Weight	(α_1, α_2)	$(0.1, 0.9)$	$(0.2, 0.8)$	$(0.3, 0.7)$	$(0.4, 0.6)$	$(0.5, 0.5)$
Gurobi Algorithm 1 and 2	$(-f_1, f_2)$	$(-8.61, 0.02)$ $(-15.01, 3.38)$	$(-8.61, 0.05)$ $(-15.01, 3.38)$	$(-19.26, 3.22)$ $(-20.27, 5.52)$	$(-19.26, 3.22)$ $(-25.97, 8.85)$	$(-19.26, 3.22)$ $(-30.06, 12.56)$
Weight	(α_1, α_2)	$(0.6, 0.4)$	$(0.7, 0.3)$	$(0.8, 0.2)$	$(0.9, 0.1)$	
Gurobi Algorithm 1 and 2	$(-f_1, f_2)$	$(-19.26, 3.22)$ $(-40.73, 26.47)$	$(-19.31, 3.32)$ $(-50.94, 44.01)$	$(-19.38, 3.5)$ $(-50.94, 44.01)$	$(-19.41, 3.68)$ $(-50.94, 44.01)$	

6.2. Test of the CNP-Based Task Re-Assignment Algorithm

6.2.1. Case 1: A Small-Scale Scenario Involving the Task Re-Assignment Problem

In order to introduce the specific steps of Algorithm 3 in detail, the process of task re-assignment is explained based on the sixth non-dominated solution selected in Section 6.1.2. Suppose that four new targets T_i $(i \in \{21, 22, 23, 24\})$ are found on the battlefield. The probability of UAVs successfully attacking these targets, the probability of UAVs being destroyed, and the values of these new targets are shown in Table 11. In Algorithm 3, the value of the parameter G_{ic} is 10 and $\alpha_1 = \alpha_2 = 0.5$.

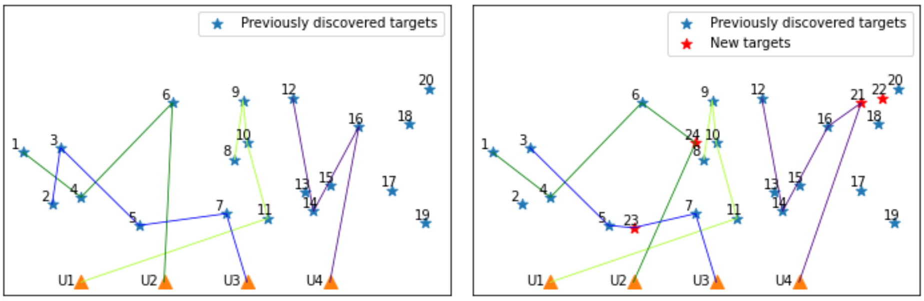
Table 11. Probabilities of UAVs and new targets being destroyed and the value of new targets.

		T_{21}	T_{22}	T_{23}	T_{24}			T_{21}	T_{22}	T_{23}	T_{24}
U_1	P_{1j}	0.3	0.22	0.44	0.17	U_2	P_{1j}	0.21	0.29	0.41	0.15
	K_{1j}	0.58	0.86	0.45	0.82		K_{1j}	0.51	0.95	0.55	0.92
U_3	P_{1j}	0.34	0.2	0.21	0.32	U_4	P_{1j}	0.41	0.33	0.46	0.23
	K_{1j}	0.69	0.86	0.54	0.82		K_{1j}	0.64	0.94	0.62	0.95
Target	value	0.8	0.6	0.75	0.65						

Targets T_i $(i \in \{21, 22, 23, 24\})$ are assigned using Algorithm 3; the bidding process is shown in Table 12. The results of task re-assignment are shown in Figure 10, where Figure 10a shows the results for the initial assignment and Figure 10b the results for task re-assignment in emergencies. From Figure 10 and Table 12, the following can be determined. The winning agents of T_{21} and T_{22} are U_4 and U_2, respectively, and the contracts of U_4 and U_2 are all sales contracts. The winning agents of T_{23} and T_{24} are U_3 and U_2, respectively, and the contracts of U_3 and U_2 are all interchange contracts. In the contract of U_3 and U_2, the replaced targets are T_2 and T_{22}, respectively, while in the bidding processes of T_2 and T_{22}, no UAV bids for T_2 or T_{22}.

Table 12. The bidding process and results.

	T_{21}		T_{22}		Winning contract
U_1	-		-		-
U_2	$< 2, 21, 0, 0.6385 >$		$< 2, 22, 0, 0.6755 >$		$< 2, 22, 0, 0.6755 >$
U_3	$< 3, 21, 2, 0.0235 >$		$< 3, 22, 2, 0.0685 >$		-
U_4	$< 4, 21, 0, 0.6395 >$		-		$< 4, 21, 0, 0.6395 >$
	T_{23}		T_{24}		Winning contract
		T_2		T_{22}	
U_1	-	-	$< 1, 24, 11, 0.001 >$	-	-
U_2	-	-	$< 2, 24, 22, 0.0455 >$	-	$< 2, 24, 22, 0.0455 >$
U_3	$< 3, 23, 2, 0.0085 >$	-	-	-	$< 3, 23, 2, 0.0085 >$
U_4	-	-	-	-	



(**a**) The result of initial task assignment (**b**) The result of task re-assignment

Figure 10. Results of initial task assignment and task re-assignment in emergencies.

6.2.2. Case 2: A Large-Scale Scenario involving the Task Re-Assignment Problem

The primary advantage of the CNP-based method is that it can shorten the time requried for task assignment. The effectiveness of this method can be fully reflected in large-scale calculation examples. In a large-scale problem, to assign all the current tasks again takes a lot of time. However, if only new tasks are assigned using the CNP-based method, a large amount of time can be saved. In this section, the large-scale calculation example in Section 6.1.3 is used to discuss the effectiveness of the task re-assignment method in emergencies.

Suppose that ten new targets are found. The values of the parameters V_{T_j}, P_{ij}, K_{ij} $(i \in I_{15},\ j \in \{101, 102, \cdots, 110\})$ of the new targets can be downloaded from the website (accessed on 6 November 2022) https://github.com/gaoxh-github/Values-of-parameters. In Algorithm 3, the value of the parameter G_{ic} is 50. The results of the assignment of ten new targets based on the original assignment results are shown in Table 13. It can be seen from Table 13 that the task sets of UAVs U_4, U_6, U_{11}, and U_{14} have changed. The task assignment results obtained using Algorithms 1 and 2 to assign all 110 targets are shown in Table 14. Through calculation, the values of S_1 corresponding to the task assignment schemes in Tables 13 and 14 are -8.91 and -9.42, respectively. The performance of the assignment scheme obtained using Algorithms 1 and 2 to assign all tasks is better than the performance of the assignment scheme obtained using Algorithm 3 to assign new tasks. However, the calculation time of Algorithm 3 is lower than that of Algorithm 1 combined with Algorithm 2. Figure 11 shows the time of ten task re-assignment experiments under these two assignment methods. It can be seen from Figure 11 that Algorithm 3 requires less time to solve the problem of task re-assignment in emergencies than Algorithm 1 combined with Algorithm 2. Therefore, in emergency situations, Algorithm 3 can perform real-time task re-assignment by slightly sacrificing the ability to seek global optimization.

Table 13. Results of task re-assignment using Algorithm 3.

UAV	Task Set	UAV	Task Set
U_1	$M_{U_1} = \{T_8, T_{27}, T_{29}, T_{41}, T_{48}, T_{96}, T_{98}\}$	U_2	$M_{U_2} = \{T_{19}, T_{44}, T_{64}, T_{65}, T_{79}, T_{86}\}$
U_3	$M_{U_3} = \{T_{22}, T_{24}, T_{62}, T_{80}, T_{89}, T_{91}\}$	U_4	$M_{U_4} = \{T_{16}, T_{33}, T_{43}, T_{71}, T_{103}, T_{105}, T_{106}\}$
U_5	$M_{U_5} = \{T_{34}\}$	U_6	$M_{U_6} = \{T_3, T_{68}, T_{110}\}$
U_7	$M_{U_7} = \{T_{10}, T_{23}, T_{78}, T_{83}, T_{85}\}$	U_8	$M_{U_8} = \{T_{21}, T_{45}\}$
U_9	$M_{U_9} = \{T_5, T_{11}, T_{32}, T_{36}, T_{54}, T_{57}, T_{75}\}$	U_{10}	$M_{U_{10}} = \{T_7, T_{66}\}$
U_{11}	$M_{U_{11}} = \{T_{73}, T_{107}, T_{109}\}$	U_{12}	$M_{U_{12}} = \{T_{13}, T_{38}, T_{74}, T_{76}, T_{77}, T_{93}\}$
U_{13}	$M_{U_{13}} = \{T_6, T_9\}$	U_{14}	$M_{U_{14}} = \{T_{99}, T_{101}, T_{102}, T_{104}, T_{108}\}$
U_{15}	$M_{U_{15}} = \{T_{28}, T_{52}, T_{55}, T_{84}, T_{88}\}$		

Table 14. Results of task re-assignment using Algorithm 1 and Algorithm 2.

UAV	Task Set	UAV	Task Set
U_1	$M_{U_1} = \{T_3, T_{27}, T_{106}\}$	U_2	$M_{U_2} = \{T_{108}\}$
U_3	$M_{U_3} = \{T_{14}, T_{38}, T_{39}, T_{79}, T_{80}, T_{94}\}$	U_4	$M_{U_4} = \{T_{44}, T_{57}, T_{77}, T_{84}\}$
U_5	$M_{U_5} = \{T_8\}$	U_6	$M_{U_6} = \{T_{22}\}$
U_7	$M_{U_7} = \{T_5, T_9, T_{17}, T_{34}, T_{68}, T_{74}, T_{75}\}$	U_8	$M_{U_8} = \{T_{86}\}$
U_9	$M_{U_9} = \{T_{11}, T_{48}, T_{65}, T_{73}, T_{98}\}$	U_{10}	$M_{U_{10}} = \{T_{40}, T_{52}, T_{69}\}$
U_{11}	$M_{U_{11}} = \{T_{55}, T_{66}, T_{87}\}$	U_{12}	$M_{U_{12}} = \{T_6, T_{21}, T_{32}, T_{76}, T_{100}, T_{101}, T_{103}\}$
U_{13}	$M_{U_{13}} = \{T_{24}, T_{89}\}$	U_{14}	$M_{U_{14}} = \{T_{83}\}$
U_{15}	$M_{U_{15}} = \{T_1, T_{10}, T_{33}, T_{45}, T_{78}, T_{93}, T_{99}\}$		

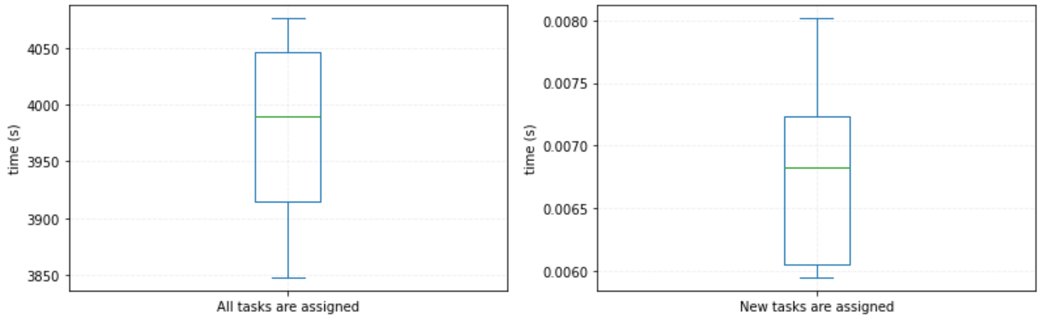


Figure 11. CPU runtime required for the task re-assignment process. (**a**) Algorithm 1 combined with Algorithm 2. (**b**) Algorithm 3.

6.2.3. Case 3: Attempting to Solve the Task Assignment Problem Directly Using the CNP-Based Method

Considering that Algorithm 3 can quickly solve the task re-assignment problem, it may be possible to use this algorithm directly to solve the task assignment problem. An analysis of this problem is provided in this section. In the task assignment problem before the battle, the main goal of assignment is to find an assignment plan that achieves better performance. Taking the 15 UAVs attacking 100 targets in Section 6.1.3 as an example. Let $\alpha_1 = \alpha_2 = 0.5$, the calculation time of Algorithm 3 is about 1 second. The task assignment plan obtained using the CNP-based algorithm is shown in Table 15. The values of S_1 corresponding to the task assignment schemes in Tables 9 and 15 are -8.75 and -5.189, respectively. Obviously, the calculation result of the improved multi-objective genetic algorithm combined with selection strategy is much better than the calculation result of the CNP-based method. For the task assignment problem, the CNP-based method is much faster than the improved multi-objective genetic algorithm combined with selection strategy. However, the result obtained by the improved multi-objective genetic algorithm combined with selection strategy is much better than the result obtained by the CNP-based algorithm. Therefore, according to the above analysis, Algorithms 1 and 2 should be used in the task assignment phase with high requirements for the performance of the task assignment scheme, and Algorithm 3 should be used in the task re-assignment phase in emergencies with high requirements related to assignment time.

Table 15. The result of Case 3 in Section 6.1.3 based on the CNP-based method.

UAV	Task Set	UAV	Task Set
U_1	$M_{U_1} = \{T_{33}, T_{40}, T_{48}, T_{52}, T_{55}, T_{64}, T_{65}\}$	U_2	$M_{U_2} = \{T_{72}, T_{73}, T_{75}, T_{76}, T_{77}, T_{78}, T_{79}\}$
U_3	$M_{U_3} = \{T_{99}, T_{100}\}$	U_4	$M_{U_4} = \{T_5, T_9, T_{10}, T_{14}, T_{16}, T_{21}, T_{22}\}$
U_5	$M_{U_5} = \{T_{36}, T_{44}, T_{46}, T_{53}, T_{54}, T_{60}, T_{61}\}$	U_6	$M_{U_6} = \{T_1, T_6, T_{18}, T_{24}, T_{26}, T_{29}, T_{31}\}$
U_7	$M_{U_7} = \{T_{74}, T_{80}, T_{82}, T_{84}, T_{85}, T_{88}, T_{96}\}$	U_8	$M_{U_8} = \{T_2, T_{42}, T_{49}, T_{50}, T_{58}, T_{59}, T_{69}\}$
U_9	$M_{U_9} = \{T_{43}, T_{51}, T_{57}, T_{62}, T_{66}, T_{68}, T_{70}\}$	U_{10}	$M_{U_{10}} = \{T_{15}, T_{17}, T_{19}, T_{20}, T_{25}, T_{30}, T_{32}\}$
U_{11}	$M_{U_{11}} = \{T_{23}, T_{27}, T_{28}, T_{34}, T_{35}, T_{38}, T_{39}\}$	U_{12}	$M_{U_{12}} = \{T_{81}, T_{86}, T_{87}, T_{90}, T_{91}, T_{97}, T_{98}\}$
U_{13}	$M_{U_{13}} = \{T_{37}, T_{41}, T_{45}, T_{47}, T_{56}, T_{63}, T_{67}\}$	U_{14}	$M_{U_{14}} = \{T_3, T_4, T_7, T_8, T_{11}, T_{12}, T_{13}\}$
U_{15}	$M_{U_{15}} = \{T_{71}, T_{83}, T_{89}, T_{92}, T_{93}, T_{94}, T_{95}\}$		

7. Conclusions

In this paper, we have provided a unified multi-objective optimization framework for the cooperative task assignment and re-assignment of multiple UAVs. First, we propose a multi-objective optimization problem in which the minimization of the cost and the maximization of the benefits are regarded as the objectives. To solve the problem, a multi-objective genetic algorithm suitable for UAV cooperative task assignment is proposed and the encoding format and genetic operators in the proposed algorithm are specially designed. Then, we provide a selection strategy to facilitate the choice of an operation plan from the Pareto solution set by the decision-maker. Finally, taking into account the possible emergencies in the complex combat environment, the task re-assignment problem in emergencies before the battle is studied and a task re-assignment algorithm based on a contract network protocol is proposed. Simulation examples are used to verify the effectiveness of the proposed algorithms.

When the battlefield environment is more complex, a single target may contain multiple different tasks. For multiple types of tasks, heterogeneous UAVs have higher efficiency in performing their tasks compared to homogeneous UAVs. At the same time, there may be multiple obstacles in the environment that affect UAV flight. In addition, it is important to study the performance of the algorithm. Therefore, in our next work we intend to focus on the performance of the algorithm and task assignment of heterogeneous UAVs in complex environments containing obstacles.

Author Contributions: Data curation, Y.D.; Funding acquisition, L.W. and X.W.; Investigation, X.S. and Y.D.; Methodology, X.G., L.W., C.W. and X.W.; Project administration, L.W. and X.W.; Resources, C.L., C.W. and X.W.; Supervision, C.L., Y.D., C.W. and H.P.; Validation, X.S.; Writing—original draft, X.G.; Writing—review and editing, X.G. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Plan (2021YFB3302501); the National Natural Science Foundation of China (12102077); and the Fundamental Research Funds for the Central Universities (DUT22LAB305, DUT22RC(3)010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Contact the first/corresponding author please.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Newcome, L.R. *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*; Library of Flight Series, AIAA: Reston, VA, USA, 2004. [CrossRef]
2. Sarris, Z. Survey of UAV applications in civil markets (June 2001). In Proceedings of the IEEE Mediterranean Conference on Control and Automation, Dubrovnik, Croatia, June 2001. Available online: https://www.researchgate.net/publication/229091536_Survey_of_UAV_applications_in_civil_markets_June_2001 (accessed on 6 November 2022).

3. Nonami, K.; Kendoul, F.; Suzuki, S.; Wang, W, Nakazawa, D. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*; Springer: Tokyo, Japan, 2010. Available online: https://dl.acm.org/doi/abs/10.5555/1941802 (accessed on 6 November 2022).
4. Qu, X.B.; Zhang, W.G.; Wang, X.G. Research of UAVs' attack strategy under uncertain condition. *Flight Dyn.* **2015**, *33*, 381–384. Available online: http://en.cnki.com.cn/Article_en/CJFDTOTAL-FHLX201504021.htm (accessed on 6 November 2022).
5. Máthé, K.; Busoniu, L. Vision and Control for UAVs: A Survey of General Methods and of Inexpensive Platforms for Infrastructure Inspection. *Sensors* **2015**, *15*, 14887–14916. [CrossRef] [PubMed]
6. Turner, I.L.; Harley, M.D.; Drummond, C.D. UAVs for coastal surveying. *Coast. Eng.* **2016**, *114*, 19–24. [CrossRef]
7. Fu, X.; Feng, P.; Gao, X. Swarm UAVs Task and Resource Dynamic Assignment Algorithm Based on Task Sequence Mechanism. *IEEE Access* **2019**, *7*, 41090–41100. [CrossRef]
8. Zhen, Z.; Xing, D.; Gao, C. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. *Aerosp. Sci. Technol.* **2018**, *76*, 402–411. [CrossRef]
9. Zhang, L.; Sun, J.; Guo, C.; Zhang, H. A Multi-swarm Competitive Algorithm Based on Dynamic Task Allocation Particle Swarm Optimization. *Arab. J. Sci. Eng.* **2018**, *43*, 8255–8274. [CrossRef]
10. Issac, T.; Silas, S.; Rajsingh, E.B. Prototyping a Scalable P-system-Inspired Dynamic Task Assignment Algorithm for a Centralized Heterogeneous Wireless Sensor Network. *Arab. J. Sci. Eng.* **2020**, *45*, 10353–10380. [CrossRef]
11. Takahashi, M.; Kita, H. A crossover operator using independent component analysis for real-coded genetic algorithms. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Seoul, Korea, 27–30 May 2001; Volume 1, pp. 643–649. [CrossRef]
12. Wang, Z.Y.; Wang, B.; Wei, Y.L.; Liu, P.F.; Zhang, L. Cooperative multi-task assignment of multiple UAVs with improved genetic algorithm based on beetle antennae search. In Proceedings of the 2020 39th Chinese Control Conference, Shenyang, China, 27–29 July 2020; pp. 1065–1610. [CrossRef]
13. Venugopalan, T.K.; Subramanian, K.; Sundaram, S. Multi-UAV task allocation: A team-based approach. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 45–50. [CrossRef]
14. Velhal, S.; Sundaram, S. Restricted airspace protection using multi-UAV spatio-temporal multi-task allocation. *arXiv* **2020**, arXiv:2011.11247.
15. Afghah, F.; Zaeri-Amirani, M.; Razi, A.; Chakareski, J.; Bentley, E. A coalition formation approach to coordinated task allocation in heterogeneous UAV networks. In Proceedings of the 2018 Annual American Control Conference, Milwaukee, WI, USA, 27–29 June 2018; pp. 5968–5975. [CrossRef]
16. Schwarzrock, J.; Zacarias, I.; Bazzan, A.L.; Fernandes, R.Q.D.A.; Moreira, L.H.; de Freitas, E.P. Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence. *Eng. Appl. Artif. Intell.* **2018**, *72*, 10–20. [CrossRef]
17. Ye, F.; Chen, J.; Tian, Y.; Jiang, T. Cooperative Multiple Task Assignment of Heterogeneous UAVs Using a Modified Genetic Algorithm with Multi-type-gene Chromosome Encoding Strategy. *J. Intell. Robot. Syst.* **2020**, *100*, 615–627. [CrossRef]
18. Yan, F.; Zhu, X.; Zhou, Z.; Tang, Y. Real-time task allocation for a heterogeneous multi-UAV simultaneous attack. *Sci. Sin. Informationis* **2019**, *49*, 555–569. [CrossRef]
19. Chen, Y.; Yang, D.; Yu, J. Multi-UAV Task Assignment With Parameter and Time-Sensitive Uncertainties Using Modified Two-Part Wolf Pack Search Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2853–2872. [CrossRef]
20. Jia, Z.; Yu, J.; Ai, X.; Xu, X.; Yang, D. Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. *Aerosp. Sci. Technol.* **2018**, *76*, 112–125. [CrossRef]
21. Singh, M.K.; Choudhary, A.; Gulia, S.; Verma, A. Multi-objective NSGA-II optimization framework for UAV path planning in an UAV-assisted WSN. *J. Supercomput.* **2022**, 1–35.
22. Zhu, J.; Wang, X.; Huang, H.; Cheng, S.; Wu, M. A NSGA-II Algorithm for Task Scheduling in UAV-Enabled MEC System. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 9414–9429. [CrossRef]
23. Cheng, C.; Wu, X.Q.; Liu, M.; Chen, M. Research on task allocation for UCAVs cooperatively attacking multiple targets. *J. Jilin Univ. (Inf. Sci. Ed.)* **2012**, *30*, 609–615.
24. Ramirez-Atencia, C.; Bello-Orgaz, G.; R-Moreno, M.; Camacho, D. Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms. *Soft Comput.* **2016**, 21, 4883–4900. [CrossRef]
25. Ramirez-Atencia C, Camacho D () Constrained multi-objective optimization for multi-UAV planning. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 2467–2484. [CrossRef]
26. Chen, H.-X.; Nan, Y.; Yang, Y. Multi-UAV Reconnaissance Task Assignment for Heterogeneous Targets Based on Modified Symbiotic Organisms Search Algorithm. *Sensors* **2019**, 19, 734. [CrossRef]
27. Wang, J.-F.; Jia, G.-W.; Lin, J.-C.; Hou, Z.-X. Cooperative task allocation for heterogeneous multi-UAV using multi-objective optimization algorithm. *J. Central South Univ.* **2020**, *27*, 432–448. [CrossRef]
28. Pohl, A.J.; Lamont, G.B. Multi-objective UAV mission planning using evolutionary computation. In Proceedings of the 2008 Winter Simulation Conference, Miami, FL, USA, 7–10 December 2008; pp. 1268–1279. [CrossRef]
29. Phiboon, T.; Khankwa, K.; Petcharat, N.; Phoksombat, N.; Kanazaki, M.; Kishi, Y.; Bureerat, S.; Ariyarit, A. Experiment and computation multi-fidelity multi-objective airfoil design optimization of fixed-wing UAV. *J. Mech. Sci. Technol.* **2021**, 35, 4065–4072. [CrossRef]

30. Smith, R.G. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 1980, 29, 1104–1113.
31. Zhen, Z.; Wen, L.; Wang, B.; Hu, Z.; Zhang, D. Improved contract network protocol algorithm based cooperative target allocation of heterogeneous UAV swarm. *Aerosp. Sci. Technol.* **2021**, *119*, 107054. [CrossRef]
32. Zhang, K.W.; Zhao, X.L.; Li, Z.Z.; Zhao, B.X.; Xiao, Z.H. Real-time reconnaissance task assignment of multi-UAV based on improved contract network. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering, Beijing, China, 23–25 October 2020.
33. Xiang, J.Q.; Dong, X.W.; Li, Q.D.; Ren, Z. Cooperation target assignment of missiles based on multi-agent technique and improved contract net protocol. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference, Xiamen, China, 10–12 August 2018. [CrossRef]
34. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, 6, 182–197. [CrossRef]