

Article

Multi-UAV Reconnaissance Task Assignment for Heterogeneous Targets Based on Modified Symbiotic Organisms Search Algorithm

Hao-Xiang Chen * , Ying Nan and Yi Yang 

Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; nanying@nuaa.edu.cn (Y.N.); nuaa_yang@nuaa.edu.cn (Y.Y.)

* Correspondence: chenhaoxiang@nuaa.edu.cn

Received: 22 January 2019; Accepted: 5 February 2019; Published: 12 February 2019



Abstract: This paper considers a reconnaissance task assignment problem for multiple unmanned aerial vehicles (UAVs) with different sensor capacities. A modified Multi-Objective Symbiotic Organisms Search algorithm (MOSOS) is adopted to optimize UAVs' task sequence. A time-window based task model is built for heterogeneous targets. Then, the basic task assignment problem is formulated as a Multiple Time-Window based Dubins Travelling Salesmen Problem (MTWDTSP). Double-chain encoding rules and several criteria are established for the task assignment problem under logical and physical constraints. Pareto dominance determination and global adaptive scaling factors is introduced to improve the performance of original MOSOS. Numerical simulation and Monte-Carlo simulation results for the task assignment problem are also presented in this paper, whereas comparisons with non-dominated sorting genetic algorithm (NSGA-II) and original MOSOS are made to verify the superiority of the proposed method. The simulation results demonstrate that modified SOS outperforms the original MOSOS and NSGA-II in terms of optimality and efficiency of the assignment results in MTWDTSP.

Keywords: UAV; reconnaissance task assignment; sensor; Pareto dominance determination; symbiotic organisms search

1. Introduction

Unmanned aerial vehicles (UAVs) have played increasingly important roles in military reconnaissance during the last decade. When dealing with multiple reconnaissance tasks [1], it is often required that UAVs form teams and work collaboratively, which necessitates cooperative task assignments [2]. The purpose of cooperative task assignments is to allocate necessary tasks and determine the appropriate task execution sequence to the UAVs to maximize overall performance. The battlefield situation and the range of permissible performance of the UAV set are commonly considered in the cooperative task assignments.

The cooperative task assignment problem (CTAP) is usually considered as NP-Hard Problems. In some prior works, the basic CTAP was formulated as a mixed-integrated linear programming (MILP) problem [3], which aims to obtain an optimal assignment solution. Decentralized approaches were taken to deal with the dynamic, unexpected situations by using the information obtained by itself or through communications with neighboring UAVs [4]. With the intention of reducing the computational burden, optimization algorithms such as genetic algorithms (GA) were adopted to obtain suboptimal solutions [5]. Shaferman et al. [6] took UAVs' dynamic constraints into consideration to ensure flyable trajectories and applied a stochastic search method to improve assignment solutions. Gyeongtaek et al. [7] put forward a market-based distributed assignment

algorithm for timing constrained tasks and determine the allocation sequence by the estimated time of arrival (ETA) of each UAV. In the works mentioned above, UAVs' paths to the task positions were not precisely calculated during the process of assignment.

In the actual assignment planner, the path for the UAVs to perform the tasks should also be taken into consideration. Thus, in some other prior studies, the basic CTAP were formulated as a travelling salesman problem (TSP) [8], which aims to find the optimal scheme for a salesman to visit all the cities with the shortest path. For the original TSP, the length between two cities is usually considered as Euclidean length. According to the dynamic characteristics and constraints of UAVs, Dubins vehicle model and Dubins path were introduced into the CTAP for route length computation [9]. Thus, the CTAP was formulated as a Dubins travelling salesman problem (DTSP) [10]. Furthermore, Zhang et al. [11] took the effective range of UAVs' sensors into consideration and formulated the CTAP as a Dubins travelling salesman problem with neighborhood (DTSPN). Wang et al. [12] considered the multi-UAV reconnaissance task allocation as an extended multiple DTSP, where the visit paths to the heterogeneous targets must meet specific constraints due to the targets' feature. Since the reconnaissance tasks are time-sensitive, the execution time required will affect UAVs' choices of tasks and flight paths. Considering the time windows of the tasks (targets), Karabulut et al. [13] formulated the TSP with a service time containing ready time and due time, which was so called TSPTW. Nunes et al. [14] explained the different circumstance, under which the time window was took as a hard temporal constraint and a soft temporal constraint.

As for the subject of CTAP, the literature mentioned above concentrated on the heterogeneity features and kinematic constraints of UAVs, but the features of targets were ignored or considered to be homogeneous. In this paper, we focus on the reconnaissance task allocation problem for UAVs, considering ground targets with heterogeneous features and sizes are considered. To describe the CTAP, a novel cooperative reconnaissance task assignment model for multi-UAV is formulated: multiple time window-based Dubins travelling salesmen problem (MTWDTSP). Compared with the MILP form, MTWDTSP takes into account the influence of various flight paths on the assignment results; as for compared with common TSP form, MTWDTSP could deal with the effect of different time windows on task sequence. In this particular problem, heterogeneous reconnaissance targets are also presented as point, strip, and surface targets, according to the features of targets and the performance of UAV sensor. To accomplish the reconnaissance of each target, UAVs must cover all the heterogeneous targets with the airborne sensors.

The MTWDTSP is a typical NP-hard multi-objective optimization problem. Group intelligent optimization methods (such as Bayesian approach [15]) are employed to solve the problem with independent task points in a static environment within limited space, which can reduce the complexity of modeling and calculation. In recent years, heuristic and bio-inspired algorithms based group intelligent optimization methods have been adopted to solve the multi-objective optimization problem, including the simulated annealing algorithm [16], the Tabu search algorithm [17], genetic algorithms (NSGA-II) [18,19], the artificial fish school algorithm [20], ant colony algorithm [21] and the particle swarm optimization (niching PSO) [22–25]. These algorithms perform better than most traditional mathematical techniques in solving these problems, because they do not require substantial gradient information. Traditional group intelligent optimization algorithms can obtain locally optimal solutions for low-dimensional problems. As the number of UAVs and tasks (targets) grows, the traditional group intelligent optimization algorithms can easily premature and fall into local optimum. In order to search over the designed possible solution spaces as much as possible, Cheng and Prayogo [26] first proposed in 2014 a new meta-heuristic optimization algorithm known as symbiotic organisms search (SOS). In this paper, we introduce a modified multi-objective symbiotic organisms search (MOSOS) [27] to solve the MTWDTSP with dynamic time window constraints, task type constraints, and UAV sensor constraints. To improve the performance of MOSOS, a globally adaptive parasitism parameter [28] is used to increase convergence speed. Pareto optimal solution set is introduced to improve the variety of population and increase the probability of converging to the optimal solution. Also, a double-chain

chromosomes encoding is adopted to pre-process the designed solution space for the efficiency of the algorithm.

2. Mathematical Model of the MTWDTSP

In this section, the models of sensors, targets, and UAVs are established. The paths of covering targets with different shapes are also designed to calculate the estimated reconnaissance time.

2.1. Modeling of the UAV and Sensor

In the particular problem, a set of heterogeneous targets need to be reconnoitered by different types of UAVs subject to several types of constraints. In this article, Dubins model for UAV [10] is introduced with the following basic assumptions: (1) the velocity of each UAV is constant; (2) the UAVs perform the reconnaissance task at given altitudes; (3) the UAVs fly at different altitudes without inter-UAV collision; (4) the flight time of each UAV is limited. At this point, the spatial dimension of the UAV can be reduced from three dimensions to two dimensions (X_U - Z_U plane):

$$S_U = (x, z, \psi, V_U, r_{\min}) \quad (1)$$

$$\begin{cases} \dot{x} = V_U \cos \psi \\ \dot{z} = V_U \sin \psi \\ \dot{\psi} = c \cdot V_U / r_{\min} \end{cases} \quad (2)$$

where x and z represent the UAV's coordinates in plane space, V_U represents for the speed, r_{\min} represents the minimum turning radius, and ψ is the flight yaw angle.

Airborne sensors are required to reconnoiter ground targets. It is assumed that the reconnaissance task on this target is finished when a ground target is fully covered by the sensor's field of view. In this study, a sensor's field of view is considered as a circle region with the constant opening angle on the ground, neglecting the influence of altitude and noise, as illustrated in Figure 1. In Figure 1, H_U is the flight altitude, W_U is the detection width of the sensor, the axis X_U - Z_U - Y_U denotes the body/dynamic system of the UAV.

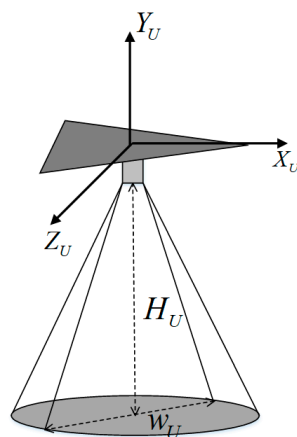


Figure 1. Body/dynamic system and airborne sensor of UAV.

2.2. Modeling of Heterogeneous Ground Targets

In general, one of the fundamental features is the target shape. To facilitate the determination of the reconnaissance duration and path, the shape of a target can be divided into point targets, strip targets, and surface targets. Due to the different sizes of targets, the same UAV will have different detection times for each type of target. Assuming that the detection time is always lower than the time window, then the following provisions are made to determine a suitable target detection time:

- (1) For point targets, the default detection time is a fixed duration: $t_d = t_{Dec}$;
- (2) For strip targets, the UAV detects targets along the direction of strip extension (shown as Figure 2), and the detection duration is $t_d = L_{st}/V_U$, where L_{st} represents the extension length of the target and V_U is the UAV's speed.
- (3) For surface targets, the UAV must fly along the "Z" path to detect the entire area. Considering the irregularity of the detection area, as shown in Figure 3a, an equal interval rotation method should first be performed to find the smallest circumscribed rectangle (SCR) of the irregular area. Then the circumscribed rectangle should be divided into several smaller rectangles, and finally, the UAV can detect the whole area along the "Z" path. Denote the SCR as $H_{min} = L_{sf} * W_{sf}$, where L_{sf} and W_{sf} represent the length and width, respectively, shown in Figure 3b. Therefore, the detection duration of surface targets is $t_d = \frac{1}{V_U} \left(L_{sf} \cdot \lceil W_{sf}/w_U \rceil + \pi w_U \cdot \lfloor W_{sf}/w_U \rfloor \right)$, where w_U is the detection width of a single UAV, $\lceil \cdot \rceil$, $\lfloor \cdot \rfloor$ represent for the upward and downward rounding operators.

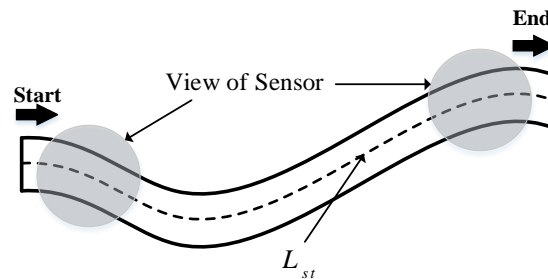


Figure 2. Reconnaissance of strip targets.

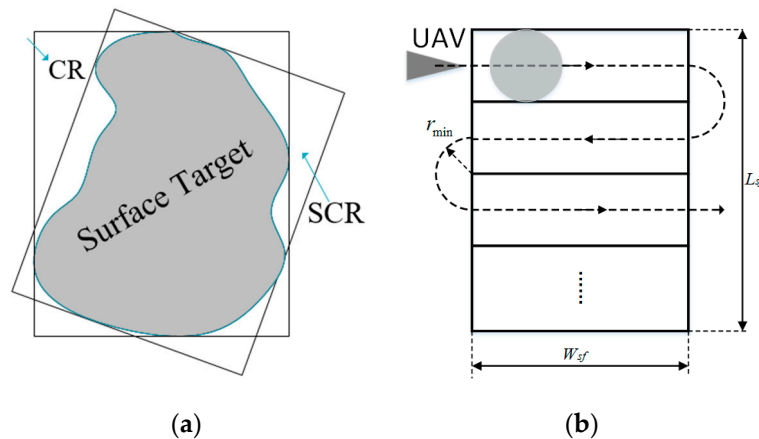


Figure 3. Reconnaissance of surface targets. (a) Smallest circumscribed rectangle of surface targets; (b) Reconnaissance path.

It is worth pointing out that the way to select the entry point of the surface or the strip and the generation of Dubins-Path may be found in [12].

The second feature of heterogeneous target is the detectable time. The time attribute of the reconnaissance task means that the target has a fixed time window (as shown in Figure 4): $T_{task} = (t_{begin}, t_{end})$, where t_{begin} is the earliest time available for detection and t_{end} is the latest time available, which is considered as a hard-constraint. Denote Get_{uav} as the arrival time of the UAV, and then the waiting time (shown in Figure 5.) before execution t_c can be calculated as follows:

$$t_c = \begin{cases} t_{begin} - Get_{uav} & \text{if } t_{begin} > Get_{UAV} \\ 0 & \text{if } t_{begin} \leq Get_{UAV} < t_{end} \\ -1 & \text{else} \end{cases} \quad (3)$$

where $t_c = -1$ means the failure of this task.

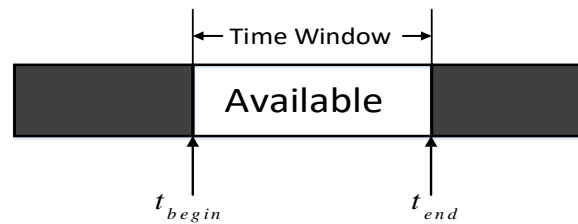


Figure 4. Time window of the reconnaissance task.



Figure 5. Schematic diagram of the waiting time and arrival time. (a) Arrive before available; (b) Arrive when available.

2.3. Assignment Model of Reconnaissance Tasks for Multi-UAVs

Task assignment problem for multiple cooperative UAVs can be described as follows: for N_U UAVs and N_T reconnaissance tasks of geographically dispersed targets, under the decision variable $X|_{N_U}^{N_T}$, the whole group of the UAVs is able to complete all tasks satisfying all types of constraints $C_{Asn}(c_1, c_2 \dots c_{N_C})$ while ensuring that the overall performance indicator J_{Asn} will be optimized. The objective of the task assignment is to minimize the total cost (performance indicator), which can be expressed as:

$$\begin{cases} X|_{N_U}^{N_T} = \underset{X}{\operatorname{argmin}}(J_{Asn}) \\ C_{Asn}(c_1, c_2 \dots c_{N_C}) \leq 0 \end{cases} \quad (4)$$

Considering the reconnaissance model, the number of the UAVs, the type of restrictions, and the time window of the task itself, certain tasks may not be successfully implemented. Therefore, determining how to prioritize the important tasks and make the total flight distance as short as possible become the optimization goals of the problem. To describe the above progress in a mathematical form, the decision variable of task assignments can be given as follows:

$$X|_{N_U}^{N_T} = \left\{ \chi_{ij}^{kp} \mid i = 1, 2 \dots N_U; j \in \mathbf{T}; k = 1, 2 \dots N_S; p = 1, 2 \dots N_P \right\} \quad (5a)$$

$$\chi_{ij}^{kp} = \begin{cases} 1, & \text{UAV } i \text{ for Task } j \\ 0, & \text{else} \end{cases} \quad (5b)$$

where $\mathbf{T} = \langle 1, 2 \dots N_T \rangle$ denotes the set of the number for task points, k denotes the type of UAV, p denotes the p th drone of the k th type, and the number of all reconnaissance tasks is N_T . Therefore, the decision variable has the following matrix form:

$$X|_{N_U}^{N_T} = \begin{bmatrix} a_{00} & a_{01} & \cdots & \cdots & \cdots & a_{0T_{\max}} \\ a_{10} & a_{11} & \cdots & \cdots & \cdots & a_{1T_{\max}} \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ a_{T_{\max}0} & a_{T_{\max}1} & \cdots & \cdots & \cdots & a_{T_{\max}T_{\max}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ 1 & \vdots & & 0 & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & \cdots & 0 \end{bmatrix} \quad (6)$$

Given that the base node number is 0, it can be seen that the element has the form of a sparse matrix. In this study, considering that each task reconnoiters only once at most, there is at most one nonzero item per row in the sparse matrix.

2.4. Mathematical Model of the Performance Indicator (PI) and Constraints

In this problem, the constraints of assignment involve two parts: physical and logical. Physical constraints are related to UAVs' flight characteristics and task properties, while logical constraints focus on the requirements of each type of task and the strategies for UAVs to implement the reconnaissance tasks. Some of the fundamental constraints are as follows [29,30]:

- (1) All UAVs start from the base and eventually return to the base.
- (2) Each task can be completed no more than once.
- (3) The assignment scheme should match the time window and imaging requirements of each individual task.
- (4) Each UAV can be assigned only one task at the same time period.
- (5) The total time for each UAV to implement the tasks should not be longer than the longest flight time.

The details of the constraints are described as below:

(1) PIs based on task assignment:

$$\min J_1(X) = \lambda(N_U + 2 - S(X)) + (1 - \lambda)N(X) \quad (7)$$

$$S(X) = \sum_{k=1}^{N_k} \sum_{p=1}^{N_p} \sum_{i=1}^{N_U} \sum_{j=1}^{N_T} w_j \chi_{ij}^{kp} \quad (8)$$

$$N_U = \sum_{k=1}^{N_s} N_p \quad (9)$$

where $S(X)$ represents the total weight of the tasks that have been assigned, w_j reflects the importance of the j th reconnaissance task, N_U is the total number of all UAVs, and $N(X)$ is the number of the employed UAVs to execute the assigned tasks.

The PI (7) is designed in a flexible form with the purpose to let the task assignment model choose the proper objective function to optimize depending on the conditions of UAVs and tasks. λ represents the adjustment parameter, and divides $\min J_1(X)$ into two parts: when $\lambda = 1$, $\min J_1(X) = \min(N_U + 2 - S(X))$, which indicates that the PI chooses to optimize the maximum weight of assigned tasks; when $\lambda = 0$, $\min J_1(X) = \min N(X)$, which indicates that the PI chooses to optimize the minimum number of the employed UAVs.

Defining the symbolic functions as the following form:

$$\text{sgn}(\chi) = \begin{cases} 1 & \chi = 1 \\ 0 & \textit{else} \end{cases} \quad (10)$$

so that:

$$N(X) = \sum_{i=1}^{N_U} \text{sgn}\left(\sum_{j=1}^{N_T} \chi_{ij}^{kp}\right) \quad (11)$$

If the value of $\text{sgn}(\chi)$ is kept within the range of 0 to 1, the following condition is satisfied:

$$\lambda = \begin{cases} 1, & S(X) < 1 \\ 0, & S(X) = 1 \end{cases} \quad (12)$$

Then, it can be ensured that the PI optimizes the maximum number of tasks, and after all the tasks are successfully executed, the number of UAVs is minimized at the same time. Since $N(X) \leq N_U$, the following must hold:

$$J_1(X_1) = N(X_1) < J_1(X_2) = N_U + 2 - S(X_2) \quad (13)$$

This equation is a universal calculation formula for all the possible decision variables X . Therefore, when designing the task weight coefficient w , it is necessary to satisfy the following requirement:

$$\sum_{j=1}^{N_T} w_j = 1 \quad (14)$$

Denote $w = [w_1, w_2, w_3, \dots, w_{N_T}]$, and $w' = [0, w]$ represents the extension vector of a weight vector. Then, there is the following expression:

$$S(X) = \sum_{k=1}^{N_k} \sum_{p=1}^{N_p} \chi^{kp} w'^T \leq 1 \quad (15)$$

If and only if the non-zero row element of the matrix of all elements in decision variable X is N_T , then all targets have been successfully reconnoitered.

(2) PI based on flight distance:

$$\min J_2(X) = \sum_{k=1}^{N_s} \sum_{p=1}^{N_p} \sum_{i \in T} \sum_{j \in T} Dis_{ij} \chi_{ij}^{kp} \quad (16)$$

where Dis_{ij} represents the distance from node i to node j . The total optimization PI can be written as follows:

$$\min J(X) = f(J_1(X), J_2(X)) \quad (17)$$

The PI must satisfy the following constraints [29]:

(a) Each target is scouted for only once:

$$\sum_{k=1}^{N_s} \sum_{p=1}^{N_p} \sum_{i \in T} \sum_{j \in T} \chi_{ij}^{kp} - 1 \leq 0 \quad (18)$$

- (b) The number of UAVs cannot exceed the maximum number in the base:

$$\sum_{p=1}^{N_p} \text{sgn}\left(\sum_{j=1}^{N_T} \chi_{ij}^{kp}\right) - N_k \leq 0 \quad (19)$$

- (c) Each UAV departs from the base and eventually returns to the base (as mentioned above, the base is noted as 0): $\forall j = 1, 2, 3, \dots, N_T$, and since $\chi_{ij}^{kp} = 1$, then:

$$\begin{cases} \sum_{l=1}^{N_T} x_{0l}^{kp} = 1 \\ \sum_{l=1}^{N_T} x_{l0}^{kp} = 1 \end{cases} \quad (20)$$

- (d) Time window constraints: Define the time window of task j : $[ms_j, mo_j]$, T_j indicates when the UAV will detect the target; therefore, the constraint can be given as follows:

$$ms_j \leq T_j \leq mo_j \quad (21)$$

$$T_j = T_i + Ser_i^k + t_{ij}^k + t_{Cj} \quad (22)$$

where Ser_i^k is the reconnaissance time of the k th type of UAV, t_{ij}^k indicates the time for the k th type of UAV to fly from node i to node j , and t_{Cj} is the wait time before executing the task.

- (e) Sensor constraints of tasks: The minimum sensor accuracy required for the task j is $SenT_j$, the sensor accuracy of the k th type of UAV is $SenV_j^k$, and the constraint can be given as follows:

$$SenV_j^k - SenT_j \leq 0 \quad (23)$$

- (f) The total flight time of each UAV cannot exceed the maximum flight time:

$$\sum_{i=0}^{N_T} \sum_{j=1}^{N_T} \chi_{ij}^{kp} \left(t_{ij}^k + t_{Cj} + Ser_j^k \right) + x_{last-0}^{kp} t_{last-0}^k - TL_k \leq 0 \quad (24)$$

where node 0 represents the base by default, $last$ represents the last reconnaissance task point of the UAV, and the latter item t_{last-0}^k represents the time to return from the last task point.

3. Modified SOS for MTWDTSP

The original MOSOS algorithm works on the cooperative behavior seen among organisms in nature. During the search process, each organism benefits from continuous interaction with others in three different phases [24]: mutualism, commensalism and parasitism. Mutualism allows the two sides to benefit from each other; commensalism benefits one party, while the other party is not affected; parasitism benefits one party, and the other party suffers. These three phases are adopted from the most common symbioses used by organisms to increase their fitness and survival advantage over the long term. The mechanisms for updating the best organism are triggered after one generation of organisms has completed its three phases. The phases are repeated until the stopping criterion is achieved. The original MOSOS enjoys some advantages such as simplicity in operation, few control parameters, good stability, and strong optimization ability, but there are several shortcomings such as early maturity and delay in later-search [30]. As for the proposed problem, combining with the motivation to improve the diversity and quality of the solution, the accuracy of the algorithm convergence, and reduce the computational complexity, several modifications are introduced to the original MOSOS algorithm [31]. A "Pareto dominance determination" approach [32] is adopted to

preprocess the performance indicator of the proposed problem to improve the diversity of non-inferior solutions. Meanwhile, globally adaptive approach is used to improve speed, accuracy and convergence characteristics of the original MOSOS.

3.1. Double-Chain Encoding of the Decision Variable

Considering the constraints of the multi-UAV reconnaissance task assignment problem, a double-chain encoding method is developed, where both chromosomes are encoded by integers. In Section 2, the decision variable is given by Formulas (4)–(6); and then the individual variable dimensions for each particle are as follows:

$$N_p \times N_k \times T_{max} \times T_{max} \tag{25}$$

Considering the dimension, the number of individuals and the iteration equations, undoubtedly, there will be a complication of the problem and a slow convergence of the algorithm. As noted above, each element of the decision variable χ is in the form of a sparse matrix. If the sparse matrix is compressed and the location of nonzero elements is stored, then the particle’s variable dimension will be compressed as follows:

$$N_p \times N_k \times M \times 2 \times 2 \tag{26}$$

where M represents the number of nonzero elements in each sparse matrix. Compared to the original coding, the decision space is greatly compressed, but it is still large. It can be imagined that there will still be a large number of zero matrices in the optimal decision variable χ .

Considering the constraint functions (a), (b) and (c) proposed in Section 2.4, the double-chain encoding is used from the perspective of the task point, and the decision code string is then mapped to the task sequence of each UAV. For example, for the assignment of six tasks, there are currently three types of UAVs, each of which has 3, 2, and 3 UAVs, so that if $T_{UAV} = (i,j)$ represents the j th drone of the i th type, the example can be easily constructed as in Table 1. It is clear that for the UAV (1,1), the UAV passes through mission points 3 and 5, with mission point 5 flown to first and mission point 3 flown to last. A detailed decoding table is constructed as shown in Table 2.

Table 1. Example of UAV Task Assignment.

Task Number	1	2	3	4	5	6
T_{UAV}	(2,1)	(3,2)	(1,1)	(2,1)	(1,1)	(3,1)
Task Order	1	1	2	2	1	1

Table 2. Task Assignment Decoding.

T_{UAV}	(1,1)	(2,1)	(3,1)	(3,2)
Task Timing	5-3	1-4	6	2

Considering the uniform dimensions of the double-chain encoding, $T_{UAV} = (i,j)$ is reduced dimensionally so that the following formulas are satisfied:

$$\text{If } N = \sum_{k=1}^{N_k} n_k \text{ and } \|T_{UAV} = (i,j)\|_1 = M, \tag{27a}$$

$$\text{then } \begin{cases} i = N_p \\ j = n_{N_k} - \left(\sum_{p=1}^{N_p} n_p - M \right) \\ \sum_{p=1}^{N_p-1} n_p < M \leq \sum_{p=1}^{N_p} n_p \end{cases} \tag{27b}$$

In summary, the decision variables of $2 \times N_T$ dimension can be defined to describe the mission assignment status of the UAV, and N_T represents the number of tasks. The assignment scheme generated by this encoding determines that each task point is visited by a UAV, but in reality, due to constraints such as the time window, there will be situations where certain task points cannot be executed. M can be set to zero to indicate that a UAV was not available to execute the task.

3.2. Pareto Dominance Determination and Optimal Solution Set

According to the mathematical model constructed in Section 2, the task assignment problem can be summarized as a nonlinear multi-objective optimization (minimum) problem:

$$\begin{aligned} \min f(\mathbf{X}) &= (f_1(\mathbf{X}), f_2(\mathbf{X}), f_3(\mathbf{X}) \dots f_n(\mathbf{X})) \\ \mathbf{X} \in \Omega \in R^n & \text{ s.t } c_i(\mathbf{X}) \leq 0 \end{aligned} \quad (28)$$

where $\mathbf{X} = (\chi_1, \chi_2, \chi_3, \dots, \chi_n)$ is the n -dimensional decision vector in R^n space and $c_i(\mathbf{X}) \leq 0$ ($i = 1, 2, 3, \dots, p$) represents the constraints of the optimization problem. The series of sub-PIs may conflict with each other; this nonlinear multi-objective optimization problem will produce multiple solutions. French economist V. Pareto proposed the concept of Pareto solution set for comparing the different feasible solutions. Assuming that for the minimization optimization problem, there are two feasible solutions X_1 and X_2 , if the following holds:

$$\forall f_i(\mathbf{X}), f_i(\mathbf{X}_1) \leq f_i(\mathbf{X}_2) \text{ and } \exists J_k(\mathbf{X}), f_i(\mathbf{X}_1) < f_i(\mathbf{X}_2) \quad (29)$$

Then, X_1 dominates X_2 , and X_1 is called a noninferior solution, and X_2 is called an inferior solution. If solutions X_1 and X_2 are not dominated by each other, then both solutions are noninferior solutions and are added to the noninferior solution set. This solution set is the so-called ‘‘Pareto solution set’’ and is often referred to as the optimal frontier (as shown in Figure 6).

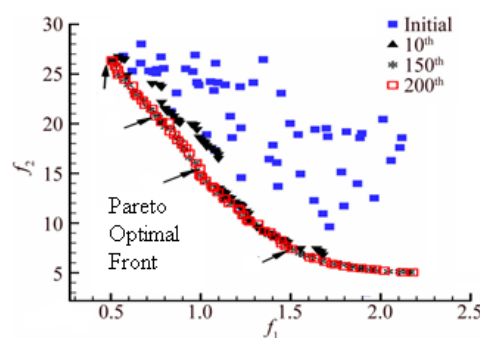


Figure 6. Distribution of Pareto optimal front.

Figure 6 illustrates the process of minimizing the optimization of the two indicator functions. It can be clearly seen that as the number of iterations increases, the distribution of the optimal solution set tends to increasingly form a curve, and the red squares indicate the best frontier after 200 iterations. Then, using the artificial knowledge, the feasible solutions in the set are analyzed and compared based on different weights of PIs.

This paper focuses on the assignment problem for UAVs with obstacles and threats in the flight zone and presents a modified SOS algorithm based on Pareto dominance determination inspired by traditional evolutionary algorithms and PSO. At the same time, a suitable coding operator is designed to effectively address all types of constraints to solve the task assignment problem.

3.3. Modified SOS with Global Adaptive Scaling Factor

Specific steps of the SOS algorithm can be expressed as follows:

Step 1. Initialize the population: First, generate N_b random individual “biologicals” according to Equation (30); each “biological” is an initial solution.

$$X_i = l_b + rand(1, D_X) \cdot (u_b - l_b) \quad (30)$$

where X_i represents the i th ($i = 1, 2, \dots, N_b$) “biological” in the ecosystem, D_X is the dimension of the solution, and $rand(1, D_X)$ is a $1 \times D_X$ dimension scaling factor vector. u_b and l_b are upper and lower bounds of the search space, respectively.

Step 2. Mutualism: At this stage a “biological” X_j is picked randomly from the population to interact with X_i and produce mutual benefits so that each progresses toward the optimal solution. X_i and X_j generate new solutions of X_i and X_j according to Equation (31). In the formula, X_{best} is the optimal individual in the ecosystem, V_M is a symbiotic vector and represents the middle point of the two individuals, and BF is the benefit factor; since each individual has a different degree of benefit, BF takes a value of 1 or 2.

$$\begin{aligned} X_{i_{new}} &= X_i + rand(0, 1) \cdot (X_{best} - V_M \cdot BF_1) \\ X_{j_{new}} &= X_j + rand(0, 1) \cdot (X_{best} - V_M \cdot BF_2) \\ V_M &= \frac{X_i + X_j}{2} \end{aligned} \quad (31)$$

Step 3. Globally Adaptive Parasitism [31]: For the original MOSOS algorithm, only one individual can benefit in a symbiotic relationship, while another individual is not affected. For the i th individual X_i , one individual X_j is randomly selected from other individuals. The original MOSOS algorithm always uses the search strategy of “current-to-best.” Because of the randomness of the scaling factor, the search equation limits the guiding effect of the global optimal value, resulting in insufficient convergence accuracy and a long convergence time, which does not meet the requirement of the task assignment problem. For this reason, this paper proposes an adaptive hybrid search strategy with global optimal guidance. The specific form is as follows:

$$X_{i_{new}} = X_i + (1 - \mu)(X_i - X_j) + \mu(X_{best} - X_j) \quad (32a)$$

$$\mu = \mu_{\min} + (\mu_{\max} - \mu_{\min}) \frac{G_I}{G_{\max}} \quad (32b)$$

where $\mu \in [0, 1]$ is the scaling factor; μ_{\max} , μ_{\min} are the maximum and minimum scaling factors, respectively; G_{\max} is the maximum number of cycles; and G_I is the current number of iterations. This strategy maintains the difference between individuals by adding a differential disturbance factor. It simultaneously introduces an adaptive scaling factor. The initial cycle has a smaller μ . The algorithm emphasizes the global search, reduces the tendency to move toward a single point in the search space, and prevents the algorithm from falling into a local minimum. With the progress of the algorithm, μ gradually increases, the guiding role of the current optimal individual is strengthened, the leading algorithm advances, and the search process is no longer just a complete random search but instead is more purposeful and directional.

3.4. Optimal Solution Selection of the Proposed Algorithm

The core of the MOSOS algorithm is to determine the global optimal solution X_{best} according to Pareto dominance determination, and only one set of noninferior solutions can be identified. Since the optimal solution must exist in the noninferior solution set, in this study the external noninferior profile C of the “biological” population is constructed for the selection of global optimal solution X_{best} . Refer to the selection technique used in [29], solutions in the noninferior solution set is sorted by their dominance firstly, then compared with the external noninferior profile C , and replace the dominated solutions in profile C with better ones.

At the meantime, since each iteration of the MOSOS algorithm will inevitably generate many new solutions, the scale of external non-inferior profile will increase dramatically. With the purpose to

distribute the optimal front edge as evenly as possible, it is necessary to eliminate a part of the densely distributed individuals and avoid falling into local optimum. Therefore, the external noninferior profile need to be “pruned” during the iterations. Refer to the crowded function defined in NSGA-II, an individual density function for sorting the solutions with normalized PIs and constraints is developed in this study as follows:

$$Den_i = \frac{1}{\|(J_{i+1,1} - J_{i-1,1})\|_2 + \|(J_{i+1,2} - J_{i-1,2})\|_2} \quad (33)$$

As shown in Figure 7, $\|(J_{i+1,1} - J_{i-1,1})\|_2, \|(J_{i+1,2} - J_{i-1,2})\|_2$ are the length and width of the quadrilateral, and the density of the individual i is reciprocal of their sum. Which reflects, the larger $\|(J_{i+1,1} - J_{i-1,1})\|_2, \|(J_{i+1,2} - J_{i-1,2})\|_2$, the lower density of individual i .

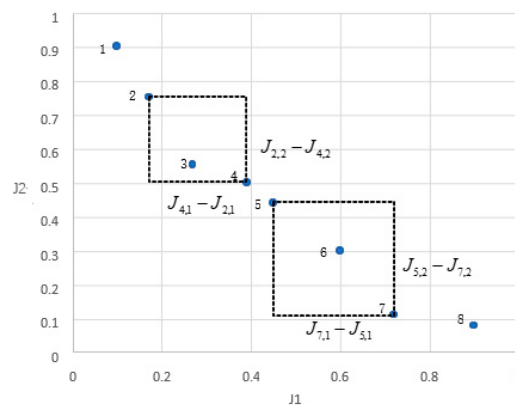


Figure 7. Density function between individuals.

Considering the efficiency of the proposed algorithm, the size of the external non-inferior profile C is limited to less than $1/5$ of the size of the ecosystem. The following operations are required during the iterations of MOSOS: when the number of non-inferior solutions in the profile C exceeds its maximum size, calculate the density of each non-inferior solution. Then eliminate the individuals with the highest density until the number of non-inferior solutions does not exceed the largest scale of profile C .

In the end, in order to let individuals to track the position of X_{best} quickly and accelerate the convergence of the algorithm, the individual with the lowest density in the external non-inferior profile C is selected as the global optimal solution X_{best} . The algorithm of optimal solution selection is given as follows:

Algorithm 1: Algorithm of the optimal solution selection.

- Step 1. Initialize all the “biological” individual variables and calculate their PI values and constraint function values.
 - Step 2. According to Section 3.1, all optimal noninferior solutions are added to the external nontrivial file C .
 - Step 3. Execute the SOS search iteration formula to obtain a new “biological” individual X and calculate its index function value and constraint function value.
 - Step 4. Compare all the solutions in the new X with the external noninferior files C according to the criterion. If the new solution dominates the solution in C , then the inferior solution in C is deleted and the new solution is added to C ; if the new solution and the solution in C do not dominate each other, the new solution is also added to C ; if the new solution is dominated by the C solution, nothing is done.
 - Step 5. Conduct pruning operations on the non-inferiority external profile C , and select X_{best} .
 - Step 6. Repeat Steps 2–5 until the SOS algorithm reaches the maximum number of iterations.
-

4. Numerical Simulation Results and Analysis

The proposed method for task assignments was tested via numerical simulation experiments. The numerical experiments were created by using Visual C++ (Ver. 10.0), and the figures illustrating

the results were created using MATLAB. The entire simulation was performed on a workstation consisting of a 3.5 GHz Intel Core-i7 CPU and 16 GB of physical RAM running 64-bit Microsoft Windows 10.

A total of 12 tasks were set in the battlefield. The parameter settings for all task points are shown in Table 3. The minimum turning radius r_{min} is set as 1 km, and the detection width W_U is set as 2 km.

Table 3. Parameters of Task Point Parameters.

Task No.	Position/km	Time Win./s	Sensor Req.	Shape	Size/km	Weight Para.
1	(22,55)	(100,300)	3	Point	-	0.04587509
2	(47,80)	(100,200)	3	Strip	4.2	0.10454398
3	(80,13)	(100,200)	3	Strip	5.1	0.12473234
4	(36,46)	(250,500)	2	Surface	6*2	0.13173999
5	(50,50)	(250,500)	2	Strip	4.0	0.09815578
6	(75,50)	(250,500)	2	Point	-	0.03243373
7	(33,75)	(750,1000)	1	Strip	5.6	0.11667218
8	(50,75)	(750,1000)	1	Point	-	0.09324776
9	(75,75)	(650,1000)	1	Surface	8*4	0.15580116
10	(63,72)	(500,750)	2	Point	-	0.02624827
11	(13,45)	(1100,1300)	2	Point	-	0.03795899
12	(25,20)	(560,1500)	2	Point	-	0.03259074

The distribution map of all task points is shown in Figure 8. Circles represent for the point targets, short strips and rectangles stand for strip targets and surface targets, respectively.

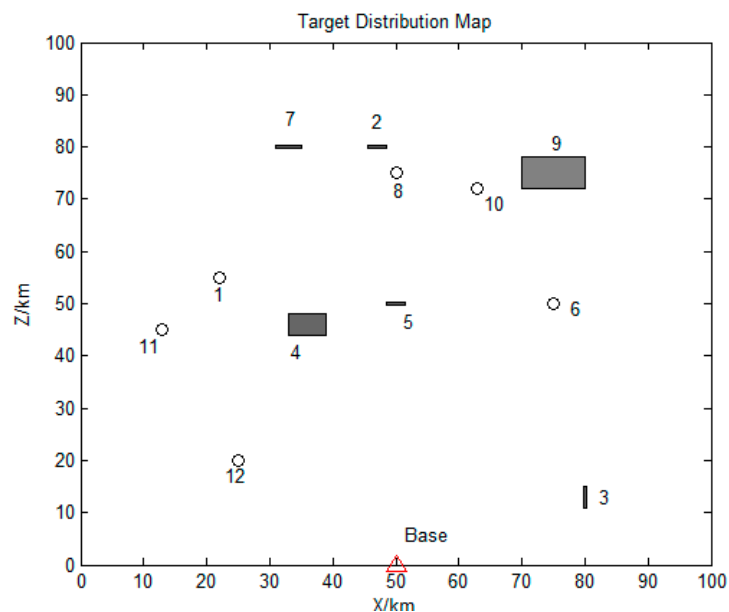


Figure 8. Distribution map of all task points.

There are the categories of UAVs on the battlefield, located at the starting positions (50, 0), of which parameters are shown in Table 4.

Table 4. Parameters of UAVs.

Category	Max. Speed	Max. Cru. Time	Sensor Level
1	230 m/s	800 s	3
2	310 m/s	420 s	3
3	170 m/s	1100 s	2

The MOSOS algorithm is set to have a population size of 30; the number of termination iterations G_{max} is 200, with a maximum external profile size of 30 and an external file of “biological” individual size of 5, scaling factors $\mu_{max} = 0.87$ and $\mu_{min} = 0.13$.

Scenario 1: One UAV of each type, with a total of 3 UAVs are employed in the base for task assignment. Since the number of UAVs is much smaller than the number of tasks, UAVs are unable to complete of the tasks, then the adjustment parameter λ is set to 1 in this scenario. Figure 9 displayed the distribution of PIs of the initial positions of the population and the optimal frontier distribution of the initial generation. Figure 10 shows the evolution of the history frontier distribution in the iterative process, including the initial generation, the 50th generation, the 100th generation, and the 200th generation.

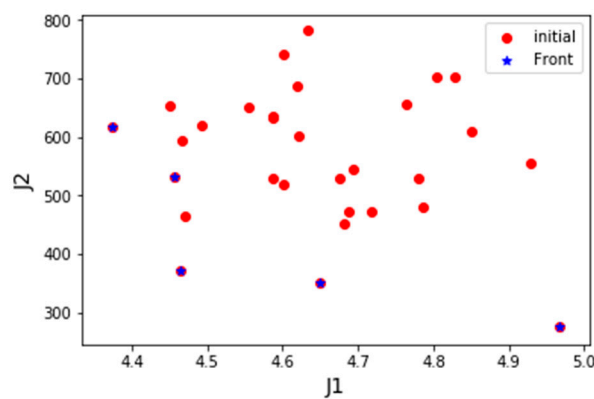


Figure 9. Initial “biological” performance index distribution of Scenario 1.

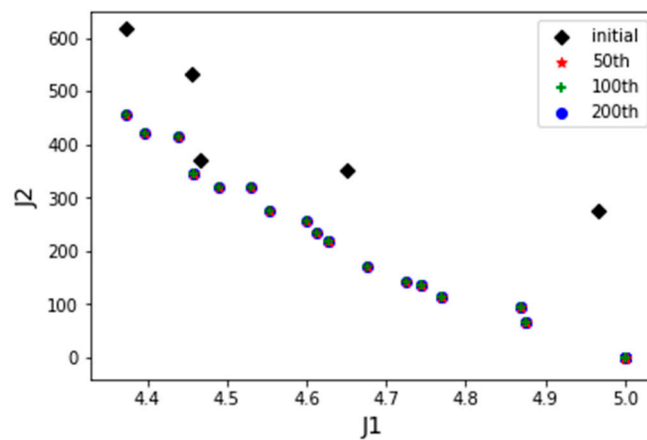


Figure 10. The best frontier distribution in history of Scenario 1.

From the perspective of implementing tasks more economically in the real combat, it is expected that existing UAVs could implement the tasks as much as possible, or implement all the tasks with less UAVs. Thus, the final feasible solution is selected as the one with the smallest value of PI J_1 , then code of assignment variables to the task is constructed as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 3 & 2 & 2 & 0 & 0 & 3 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

The serial number and detailed flight sequence of each UAV is shown in Table 5, while the paths of each employed UAV are displayed in Figure 11. It can be seen that all the UAVs are employed to execute tasks with significant weight under permitted conditions. Besides, the reconnaissance sequences of the assigned targets for the UAVs are in a good order, and the total flight distance

is 456.19 km. Thus, proposed method can obtain satisfied task allocation results for multi-UAV cooperative reconnaissance problems on heterogeneous targets.

Table 5. Flight Sequence of Each UAV in Scenario 1.

UAV No.	UAV Category	Task Sequence
1	1	5-4
2	2	9-8
3	3	7-12

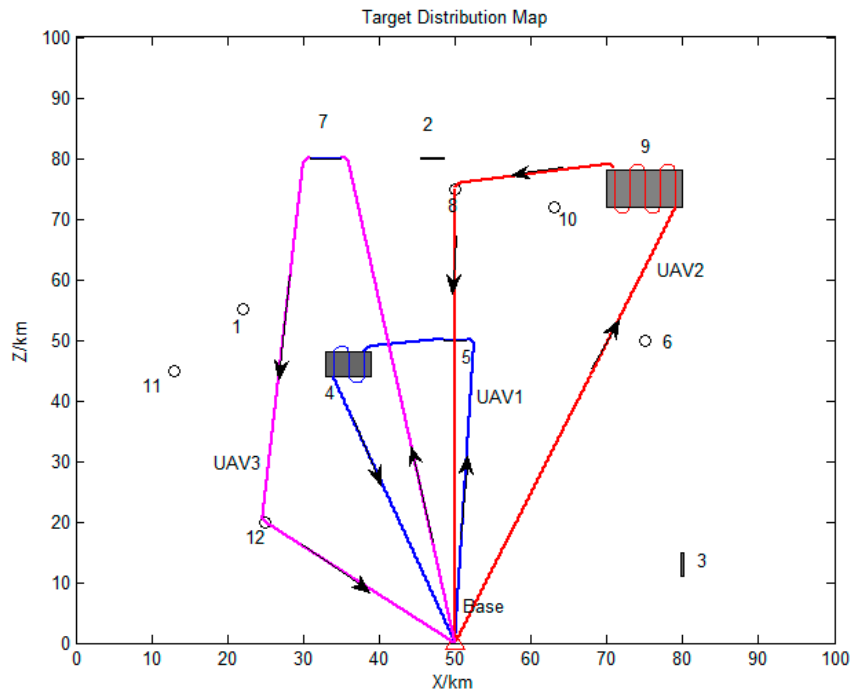


Figure 11. Paths of all UAVs in Scenario 1.

Scenario 2: Three UAVs of each type, with a total of 9 UAVs are prepared in the base for task assignment. Figure 12 displayed the distribution of PIs of the initial positions of the population and the optimal frontier distribution of the initial generation.

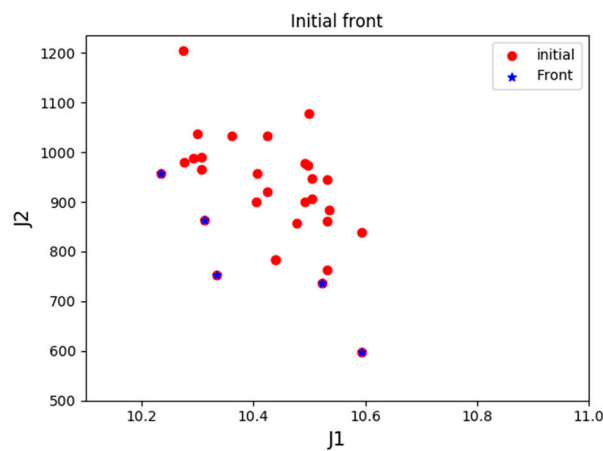


Figure 12. Initial “biological” performance index distribution in Scenario 2.

Figure 13 shows the evolution of the history frontier distribution in the iterative process, each point in the graph represents the best non-inferior solution to the number of iterations. Since there

exists solutions for UAVs to complete all the tasks, the adjustment parameter λ is set to 0 in the latter stage of optimization. As seen in the figure, with the increase in the number of iterations of the algorithm, the external archive is continuously approaching the best frontier, indicating that the algorithm is convergent. The front-end distribution of all generations of noninferior solutions is uniform, which reflects the distribution of the real optimal frontier.

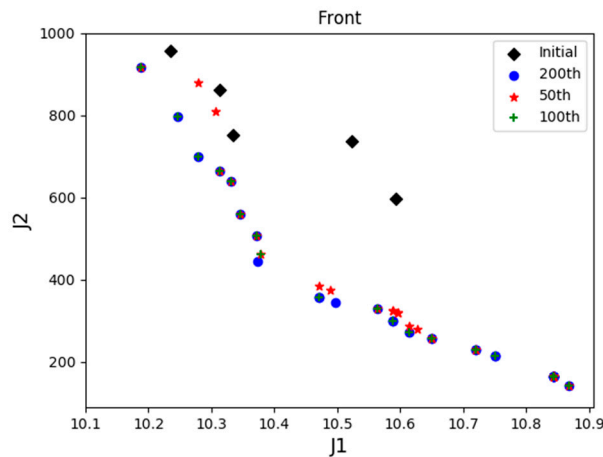


Figure 13. The best frontier distribution in history in Scenario 2.

If taking the most reconnaissance tasks completed as the main PI, that is, the index function J_1 , then in this simulation process, a total of 6 drones were deployed to perform reconnaissance tasks of all 9 UAVs, and a total of 12 task points were visited. The code of assignment variables to the task is constructed as follows:

$$\begin{bmatrix} 2 & 1 & 5 & 2 & 6 & 5 & 9 & 2 & 3 & 1 & 6 & 5 \\ 2 & 2 & 1 & 1 & 1 & 2 & 1 & 3 & 1 & 1 & 2 & 3 \end{bmatrix}$$

The detailed flight sequence of each UAV is shown in Table 6, while the paths of each employed UAV are displayed in Figure 14. It can be seen that all the targets are visited by the employed UAVs. The total flight distance of all employed UAVs is 937.52 km. Thus, it is proved once again that the proposed method can obtain satisfied task allocation results for multi-UAV cooperative reconnaissance problems on heterogeneous targets.

With the intention to verify the superiority of the proposed method (IMOSOS) in Section 3, improved MOPSO (IMOPSO) [33], NSGA-II [12] and original MOSOS with the same simulation conditions and stopping criterion are implemented for comparison. The initial parameters of IMOPSO are set as: $w_{max} = 0.9$; $w_{min} = 0.4$; factor $C1 = C2 = 1.454$. The initial parameters of NSGA-II are set as: probabilities of crossover and mutation are 0.9 and 0.1, respectively; weight factors α and β of sub-objectives are both set to be 0.5. The initial parameters of original MOSOS is the same as the proposed method. The final feasible solutions of all the algorithms are selected as the one with the smallest value of PI_{J_1} .

Table 6. Flight Sequence of Each UAV in Scenario 2.

UAV No.	UAV Category	Task Sequence
1	1	10-2
2	1	4-1-8
3	1	9
5	2	3-6-12
6	2	5-11
9	3	7

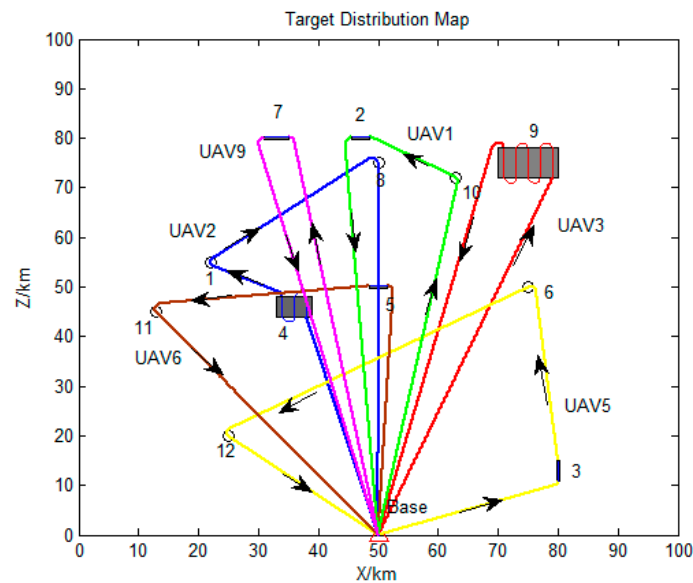


Figure 14. Paths of employed UAVs.

Monte-Carlo simulations on Scenario 2 are carried out, as the 12 tasks mentioned above with random positions. And the statistical results of 200 runs displayed in Table 7. Among the results are the best, worst, and average (i.e. Avg.) value of the PIs proposed in Section 3. Also, the compute efficiency of each algorithm is given in Table 7. Table 8 shows the improvement ratio of IMOSOS compared with other algorithms. From the statistical results of the referred indicators, the IMOSOS performs better in searching optimal solutions to the certain task assignment problem than the other three algorithms. Meanwhile, according to the average results of the referred indicators, the improved MOSOS can provide stable solutions as the three algorithms. Thus, with the help of problem formulation of MTWDTSP and modification to the original MOSOS, this proposed method enjoys certain superiority and efficiency for multi-UAV reconnaissance task allocation problems with heterogeneous targets.

Table 7. Statistical results of Monte-Carlo simulations on Scenario 2.

Indicator	J1			J2 (km)			Compute Efficiency (s)		
	Best	Worst	Avg.	Best	Worst	Avg.	Best	Worst	Avg.
IMOSOS	6	8	6.47	937.52	1089.71	956.45	9.2	17.5	12.7
MOSOS	6	9	7.33	941.27	1146.49	1017.92	14.3	22.9	17.1
NSGA-II	6	9	7.21	932.31	1123.56	998.41	16.4	26.6	21.3
IMOPSO	6	8	7.21	939.26	1109.56	979.9	12.6	20.9	15.3

Table 8. Improvement ratio of IMOSOS compared with other algorithms.

Indicator	J1	J2	Compute Efficiency
MOSOS	11.7%	6.1%	25.7%
NSGA-II	10.2%	4.2%	40.2%
IMOPSO	10.2%	2.0%	16.9%

Three main aspects have considerable influence on the computational efficiency of the algorithm in the simulation process: the number of “biological” individual, the iterations of the algorithm and the scale of the external non-inferior archive. The increased number of task points and the number of the UAVs have little effect on the computational efficiency of the algorithm. The encoding rules ensure that the number of task points is linearly related to the calculation time of the algorithm. If more tasks, UAVs or bases are considered in this particular problem, in order to solve the problem effectively,

the size of encoded decision variables and iterations may increase correspondingly. Therefore, for the problem of multi-UAV multi-reconnaissance tasks, the proposed method can provide a valid suboptimal solution.

5. Conclusions

This paper considers a reconnaissance task assignment problem for multi-UAVs as a multi-objective, multi-constraint nonlinear optimization problem. A model of cooperative UAVs and reconnaissance tasks of heterogeneous ground targets with time windows is built to describe this problem. To solve the task assignment problem, two PIs were constructed and need to be optimized. When dealing with the multi-UAV task assignment optimization model, this paper adopts the modified MOSOS algorithm based on the Pareto dominance determination and the global adaptive operation. To address the various constraints of the task assignment model, the decision variables designed from each UAV itself are mapped to the task point of view. Therefore, the double-chain encoding for modified MOSOS inherently meets the constraint requirements of the task assignment model and greatly reduces the dimensionality of the decision variables, thereby speeding up the convergence of the algorithm. The global best external archive and the best external individual archive are designed in this study. Furthermore, this study defines the density function of each individual, thus ensuring that the solution to the problem meets the requirements of diversity and homogeneity. Finally, numerical simulation results and statistical results of Monte-Carlo simulations are proposed to verify the superiority and efficiency of the introduced method.

Author Contributions: H.-X.C. wrote the original article, designed experiments and analyzed simulation results; Y.N. reviewed the original article and provided part of original simulation data; Y.Y. provided funding project and part of original simulation data.

Funding: This work was supported by funding of Jiangsu Innovation Program for Graduate Education, KYLX15 0319, and the Fundamental Research Funds for the Central Universities.

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

Data Availability: The values of some of the simulation parameters are available online (literatures), and other data is original from the authors of this paper.

Abbreviations

x	The UAV's coordinates in plane space
z	The UAV's coordinates in plane space
V_U	Speed of the UAV
r_{min}	Minimum turning radius of the UAV
ψ	The flight yaw angle of the UAV
H_U	The flight altitude of the UAV
W_U	The detection width of the sensor
t_d	Target detection time
t_{Dec}	Detection time of the point target
L_{st}	Extension length of the strip target
L_{sf}	Length of the smallest circumscribed rectangle
W_{sf}	Width of the smallest circumscribed rectangle
T_{Task}	Time window of the task
t_{begin}	The earliest time available for detection
t_{end}	The latest time available for detection
t_c	The waiting time before execution
N_U	Number of UAVs
N_T	Number of reconnaissance tasks
$X _{N_U}^{N_T}$	The decision variable
J_{Asn}	Performance indicator of the task assignment problem

C_{Asn}	Constraint of the task assignment problem
X_{ij}^{kp}	Element of the decision variable
T	The set of the number for task points
N_k	The number of types of the UAV
N_p	The number of the UAV of the k th type
J_1	Performance indicator of tasks/UAVs
λ	Adjustment parameter
$S(X)$	The total weight of the tasks that have been assigned
$N(X)$	The number of the employed UAVs to execute the assigned tasks
$\text{sgn}(\cdot)$	Symbol function
w	Weight vector
J_2	Performance indicator of flight distance
Dis_{ij}	The distance from task i to task j
$SenT_j$	The minimum sensor accuracy required for the task j
$SenV_j^k$	The sensor accuracy of the k th type of UAV
X_i	The i th ($i = 1, 2, \dots, N_b$) "biological" in the ecosystem
D_X	The dimension of the solution
BF	The benefit factor
μ	The scaling factor
G_{max}	The maximum number of cycles
G_I	The current number of iterations
C	The external noninferior profile of the noninferior solution set

References

- Chandler, P.R.; Pachter, M.; Rasmussen, S.J.; Schumacher, C. Multiple task assignment for a UAV team. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, USA, 5–8 August 2002.
- Shima, T.; Rasmussen, J. *UAV Cooperative Decision and Control: Challenges and Practical Approaches*; SIAM: Philadelphia, PA, USA, 2009.
- Shima, T.; Schumacher, C. Assigning cooperating UAVs to simultaneous tasks on consecutive targets using genetic algorithms. *J. Oper. Res. Soc.* **2009**, *60*, 973–982. [[CrossRef](#)]
- Choi, H.; Kim, Y.; Kim, H. Genetic Algorithm Based Decentralized Task Assignment for Multiple Unmanned Aerial Vehicles in Dynamic Environments. *Int. J. Aeronaut. Space Sci.* **2011**, *12*, 163–174. [[CrossRef](#)]
- Edison, E.; Shima, T. Integrated Task Assignment and Path Optimization for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms. *Comput. Oper. Res.* **2011**, *38*, 34–356. [[CrossRef](#)]
- Shaferman, V.; Shima, T. Unmanned Aerial Vehicles Cooperative Tracking of Moving Ground Target in Urban Environments. *J. Guid. Control Dyn.* **2008**, *31*, 1360–1371. [[CrossRef](#)]
- Oh, G.; Kim, Y.; Ahn, J.; Choi, H.-L. Market-Based Distributed Task Assignment of Multiple Unmanned Aerial Vehicles for Cooperative Timing Mission. *J. Aircr.* **2017**, *54*, 1–13. [[CrossRef](#)]
- Larranaga, P.; Kuijpers, C.M.H.; Murga, R.H. Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artif. Intell. Rev.* **1999**, *13*, 129–170. [[CrossRef](#)]
- Dubins, L.E. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [[CrossRef](#)]
- Savla, K.; Frazzoli, E.; Bullo, F. Traveling Salesperson Problems for the Dubins Vehicle. *IEEE Trans. Autom. Control* **2008**, *53*, 1378–1391. [[CrossRef](#)]
- Zhang, X.; Chen, J.; Xin, B.; Peng, Z. A memetic algorithm for path planning of curvature-constrained uavs performing surveillance of multiple ground targets. *Chin. J. Aeronaut.* **2014**, *27*, 622–633. [[CrossRef](#)]
- Wang, Z.; Liu, L.; Long, T.; Wen, Y. Multi-UAV Reconnaissance Task Allocation for Heterogeneous Targets Using an Opposition-Based Genetic Algorithm with Double-Chromosome Encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350. [[CrossRef](#)]
- Karabulut, K.; Tasgetiren, M.F. A variable iterated greedy algorithm for the traveling salesman problem with time windows. *Inform. Sci.* **2014**, *279*, 383–395. [[CrossRef](#)]

14. Nunes, E.; Manner, M.; Mitiche, H.; Gini, M. A taxonomy for task allocation problems with temporal and ordering constraints. *Rob. Auto. Syst.* **2017**, *90*, 55–70. [[CrossRef](#)]
15. Lanillos, P.; Yañez-Zuluaga, J.; Ruz, J.J.; Besada-Portas, E. A bayesian approach for constrained multi-agent minimum time search in uncertain dynamic domains. In Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2013), Amsterdam, The Netherlands, 6–10 July 2013; pp. 391–398.
16. Darrah, M.A.; Niland, W.; Stolarik, B. Multiple UAV Task Allocation for an Electronic Warfare Mission Comparing Genetic Algorithms and Simulated Annealing. 2006; preprint.
17. Ryan, J.L.; Bailey, T.G.; Moore, J.T.; Carlton, W.B. Reactive tabu search in unmanned aerial reconnaissance simulations. In Proceedings of the 30th conference on Winter simulation, Washington, DC, USA, 13–16 December 1998.
18. Shima, T.; Rasmussen, S.; Sparks, A.; Passino, K. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **2006**, *33*, 3252–3269. [[CrossRef](#)]
19. Wang, S.; Ali, S.; Yue, T.; Liaaen, M. Integrating Weight Assignment Strategies with NSGA-II for Supporting User Preference Multiobjective Optimization. *IEEE Trans. Evolut. Comput.* **2018**, *22*, 378–393. [[CrossRef](#)]
20. Cai, Y. Artificial Fish School Algorithm Applied in a Combinatorial Optimization Problem. *Int. J. Intell. Syst. Appl.* **2010**, *2*, 37–43. [[CrossRef](#)]
21. Perez-Carabaza, S.; Besada-Portas, E.; Lopez-Orozco, J.A.; Jesus, M. Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Appl. Soft Comput.* **2018**, *62*, 789–806. [[CrossRef](#)]
22. Geng, N.; Gong, D.W.; Zhang, Y. PSO-Based Robot Path Planning for Multisurvivor Rescue in Limited Survival Time. *Math. Probl. Eng.* **2014**, *2014*, 187370. [[CrossRef](#)]
23. Gong, D.-W.; Zhang, Y.; Qi, C.-L. Localising odour source using multi-robot and anemotaxis-based particle swarm optimisation. *IET Contr. Theor. Appl.* **2012**, *6*, 1661–1670. [[CrossRef](#)]
24. Geng, N.; Gong, D.; Zhang, Y. Robot path planning in an environment with many terrains based on interval multi-objective PSO. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, 20–23 June 2013.
25. Zhang, J.; Gong, D.; Zhang, Y. A niching PSO-based multi-robot cooperation method for localizing odor sources. *Neurocomputing* **2014**, *123*, 308–317. [[CrossRef](#)]
26. Cheng, M.-Y.; Prayogo, D. Symbiotic Organisms Search: A new meta-heuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [[CrossRef](#)]
27. Tran, D.H.; Cheng, M.Y.; Prayogo, D. A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem. *Knowledge-Based Syst.* **2016**, *94*, 132–145. [[CrossRef](#)]
28. Panda, A.; Pani, S. A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problem. *Appl. Soft Comput.* **2016**, *46*, 344–360. [[CrossRef](#)]
29. Korsah, G.A.; Kannan, B.; Browning, B.; Stentz, A.; Dias, M.B. xBots: An approach to generating and executing optimal multi-robot plans with cross-schedule dependencies. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012.
30. Lanillos, P.; Besada-Portas, E.; Lopez-Orozco, J.A.; de la Cruz, J.M. Minimum time search in uncertain dynamic domains with complex sensorial platforms. *Sensors* **2014**, *14*, 14131–14179. [[CrossRef](#)] [[PubMed](#)]
31. Tejani, G.G.; Savsani, V.J.; Patel, V.K. Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *J. Comput. Design Eng.* **2016**, *3*, 226–249. [[CrossRef](#)]
32. Van Den Berg, E.; Friedlander, M.P. Probing the Pareto Frontier for Basis Pursuit Solutions. *Siam J. Sci. Comput.* **2008**, *31*, 890–912. [[CrossRef](#)]
33. Ding, S.; Chen, C.; Xin, B. A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches. *Appl. Soft Comput.* **2018**, *63*, 249–267. [[CrossRef](#)]

