Article

Solving the Multi-Functional Heterogeneous UAV Cooperative Mission Planning Problem Using Multi-Swarm Fruit Fly Optimization Algorithm

Rubin Luo ¹, Hongxing Zheng ² and Jifeng Guo ^{2,*}¹ Institute of Aerospace Systems Engineering, Beijing 100076, China; armybin@163.com² School of Astronautics, Harbin Institute of Technology, Harbin 150001, China; zhenghongxing_hit@aliyun.com

* Correspondence: guojifeng@hit.edu.cn

Received: 7 August 2020; Accepted: 1 September 2020; Published: 4 September 2020

Abstract: The complexity of unmanned aerial vehicle (UAV) missions is increasing with the rapid development of UAV technology. Multiple UAVs usually cooperate in the form of teams to improve the efficiency of mission execution. The UAVs are equipped with multiple sensors with complementary functions to adapt to the complex mission constraints. Reasonable task assignment, task scheduling, and UAV trajectory planning are the prerequisites for efficient cooperation of multi-functional heterogeneous UAVs. In this paper, a multi-swarm fruit fly optimization algorithm (MFOA) with dual strategy switching is proposed to solve the multi-functional heterogeneous UAV cooperative mission planning problem with the criterion of simultaneously minimizing the makespan and the total mission time. First, the multi-swarm mechanism is introduced to enhance the global search capability of the fruit fly optimization algorithm. Second, in the smell-based search phase, the local search strategies and large-scale search strategies are designed to drive multiple fruit fly swarms, and the dual strategy switching method is presented. Third, in the vision-based search stage, the greedy selection strategy is adopted. Finally, numerical simulation experiments are designed. The simulation results show that the MFOA algorithm is more effective and stable for solving the multi-functional heterogeneous UAV cooperative mission planning problem compared with other algorithms.

Keywords: multi-functional heterogeneous UAVs; cooperative mission planning; multi-swarm fruit fly optimization algorithm

1. Introduction

The application of unmanned aerial vehicles (UAV) in military and civil fields is growing rapidly, such as for pesticide spraying, forest fire prevention, material delivery, reconnaissance, etc. Low cost and high safety are the two main driving forces for the rapid development of unmanned aerial vehicles. The unmanned aerial vehicle control method is very flexible [1,2], specifically including the following: (i) human operator remote control; (ii) pre-programmed or uploaded mission plan via communication device; (iii) UAV autonomous operation. The improvement of the autonomous level of unmanned aerial vehicles can effectively reduce the operator's burden and the probability of mistakes, while improving the efficiency and quality of mission completion. Moreover, for complex tasks such as large-scale information gathering, regional reconnaissance, and search and rescue, the cooperation of multiple UAVs in the form of a team is usually required. The multi-UAV cooperative mission planning problem (MCMPP) plays an essential role in the autonomous operation of multiple UAVs, which is a non-deterministic polynomial time (NP)-hard combinatorial optimization problem. The MCMPP involves rationally assigning the UAVs and scheduling the tasks on the premise of meeting the

constraints of the task sequence, spatial location, energy supply, etc. to achieve the optimization of the established mission planning objectives.

Task assignment is an essential sub-problem of the MCMPP, which was extensively studied in robotics, UAVs, operations research, etc. From the modeling perspective, the task assignment model is very relevant to classic combinatorial optimization problems, such as vehicle routing problem (VRP) [3], job-shop scheduling problem (JSP) [4], traveling salesman problem [5], team orienteering problem [6], etc. Classic combinatorial optimization problems or variant problems can be transformed into mathematical models for missions such as cooperative reconnaissance, data collection, search and rescue, etc. Faigl [7] formulated the data collection problem as a variant of the combinatorial optimization problems, i.e., the price-collecting traveling salesman problem (PCTSP) and the orienteering problem (OP). An unsupervised learning method called growing self-organizing array was proposed, which only takes tens of milliseconds to solve the information collection problem with a large number of sensor nodes. Yu et al. [8] proposed the correlated orienteering problem for modeling persistent monitoring tasks with time-invariant spatial correlations using autonomous mobile robots.

From the perspective of the solution method, exact and heuristic methods are practical solutions. Exact methods (e.g., branch-and-bound [9], branch-and-price [10,11], etc.) are optimal, but the computational complexity is proportional to the scale of the problem. Therefore, exact methods are only suitable for solving the small-sized problems. In order to reduce the difficulty of solving problems caused by the scale of the problem, heuristic methods (e.g., genetic algorithms (GA) [12–15], particle swarm optimization [16], differential evolution algorithm [17], etc.) have always been an essential solution to the MCMPP and combinatorial optimization problem. Ramirez-Atencia et al. [18] presented a new multi-objective genetic algorithm for solving complex mission planning problems involving a team of UAVs and a set of ground control stations. A new odor concentration calculation function was designed in conjunction with the constraint satisfiability problem, which accelerates the problem's convergence rate. Chen et al. [19] presented a modified multi-objective symbiotic organism search algorithm for solving the reconnaissance task assignment problem for multiple UAVs with different sensor capacities. Chen et al. [20] presented a modified two-part wolf pack search algorithm to solve the multi-UAV cooperative task assignment problem.

Path planning is another critical sub-problem of the MCMPP. Due to its broad application in various fields (such as data collection, forest fire prevention, and military), the path planning problem was studied extensively. Path planning algorithms can be mainly divided into the following categories: graph-based search method [21], sampling-based method [22,23], and heuristics method [24]. The Dubins method is a classic graph-based search method, first introduced by Dubins [25] and later inspiring several studies. Chitsaz et al. [26] proposed the Dubins airplane model, an extension of the Dubins method, which aims to find the UAV's optimal trajectory in three-dimensional space. Isaiah et al. [27] presented two motion planning algorithms for solving the Dubins travelling salesperson problem. Owen et al. [28] presented a system architecture for implementing the Dubins airplane model for a fixed-wing UAV. Macharet et al. [29] combined a metaheuristic algorithm with the Dubins method to solve the robot path planning problem.

Some research on the MCMPP problem integrated task assignment and path planning to design a unified system framework and generated a flyable path for the UAV while reasonably allocating team resources. Rasmussen et al. [30] proposed a tree search algorithm for solving the CMTAP, formulated in the form of a decision tree. Edison et al. [31] considered that multiple UAVs must perform reconnaissance, attack, and verify tasks on multiple mission targets. Task assignment and path planning are integrated and solved by genetic algorithm. Deng et al. [32] proposed a modified genetic algorithm with multiple gene types for solving the CMTAP. In the mission scenario, UAVs have different operational capabilities and kinematic constraints and carry limited resources. Weinan et al. [33] presented a coupled and distributed planning method based on the distributed genetic algorithm, using a fixed-wing UAV model with six degrees of freedom to verify the method.

In this paper, the multi-functional heterogeneous UAV cooperative mission planning problem (MFHCMPP) is proposed, as an extension of the MCMPP. In the MFHCMPP, each UAV has multiple functions with specific levels. Each task cannot be executed unless the UAV has the functions required by the task, and the functional level is not less than the standard functional level. In addition to the constraints of functions and functional levels, timing constraints exist between tasks, i.e., some tasks cannot be executed before their predecessors are completed. The UAV heterogeneity is mainly reflected in the difference in function, caused by different sensors or loads loaded by the UAVs. It is worth noting that the tasks are associated with some specific locations; thus, the task assignment of UAVs, the order of task execution, and the heading angle of the UAV approaching a task location will affect the length of the UAV flight path. The MFHCMPP problem proposed in this paper mainly solves three problems: task assignment, task scheduling, and path planning.

To solve the MFHCMPP problem, we propose the multi-swarm fruit fly optimization algorithm (MFOA). As an emerging swarm intelligence optimization algorithm, fruit fly optimization (FOA), in recent years, became one of the hottest research algorithms. The mechanism of FOA stems from a simulation of the process of searching for food sources in the fruit fly population. Based on the principles of olfactory search and visual search in the foraging process of fruit flies, the fruit fly optimization algorithm is mainly divided into two stages, namely, the smell-based search stage and the vision-based search stage. The two-stage continuous iterative optimization achieves the approach of the fruit fly population to the food source, before finally obtaining an optimal or satisfactory solution. Xiangyin et al. [34] proposed an improved fruit fly optimization (FOA) to solve the unmanned aerial vehicle (UAV) path planning problem, and the mutation adaptation mechanism was adopted to enhance the search ability of FOA and prevent it from falling into a local optimum. Xiaolong et al. [35] proposed a knowledge-based fruit fly optimization algorithm for solving the multi-skill resource-constrained project scheduling problem. A knowledge-based search procedure was introduced to enhance exploration. The knowledge of a fruit fly comes from the experience of the evolution of better fruit fly individuals. Zheng et al. [36] presented a novel fruit fly optimization algorithm to solve the semiconductor final testing scheduling problem. A collaborative search process was designed, including improved priority operation crossover and improved multi-point protection crossover, on the basis of two famous flexible job shop scheduling structures. The local search ability of FOA is reliable, but it is easy to cause a loss of population diversity, leading to premature convergence of the algorithm. Especially for problems with a large search space and many locally optimal solutions, the fruit fly optimization algorithm struggles to obtain a satisfactory solution.

The main contributions of this article are reflected in the following aspects:

- The basic problem definition of multifunctional heterogeneous UAV collaborative mission planning is given. The Dubins airplane model is introduced in the mission planning framework to generate the three-dimensional flight trajectory of each UAV.
- In order to improve the global search ability of the fruit fly optimization algorithm, multiple swarm mechanisms are introduced. In addition, two smell-based search strategies are designed to prevent the algorithm from prematurely falling into a local optimum. A two-strategy switching method is designed, and each fruit fly swarm is adaptively assigned the next iteration's smell-based search strategy according to the results of each iteration.
- The application of the fruit fly optimization algorithm in the field of UAV mission planning expands its application field.

The remainder of this paper is organized as follows: Section 2 formulates the MFHCMPP using a mixed integer linear programming model, and the Dubins airplane model is introduced. The basic FOA is introduced in Section 3, and then the MFOA for solving the MFHCMPP is proposed in Section 4. The results of numerical tests and comparisons are provided in Section 5. Finally, the conclusions of this paper and future work are given in Section 6.

2. Problem Formulation

2.1. Task and UAV

Suppose that a mission contains a series of diverse tasks to be performed by the UAVs, such as photographing a building, data gathering from sensor network, or monitoring fixed positions. The mission is defined as a T -sized set $\mathbf{T} = \{T_1, T_2, \dots, T_T\}$, where each task $T_t (t = 1, 2, \dots, T)$ involved in the mission scenario is asked to be performed only once. In this paper, the duration of each task performed by the UAV is ignored, and the task is deemed completed when the UAV arrives at the task location. In addition, a precedence constraint exists between some tasks, which is set by the human operator. Specifically, some tasks cannot be executed unless the predecessor tasks are performed. Let T'_t be the predecessor task of the task T_t , such that the execution time of T_t should be later than the execution time of T'_t , that is, $d_{T_t} > d_{T'_t}$. Let $\mathbf{L} = \{L_1, L_2, \dots, L_T\}$ be a set of T task positions, where $L_t = (x_t, y_t, z_t) \in \mathbb{R}^3$ stands for the task position of task T_t . The termination condition is defined as that the task position is visited by the UAV performing the task.

Let $\mathbf{U} = \{U_1, U_2, \dots, U_W\}$ be the set of heterogeneous UAVs involved in the mission scenario. Each UAV has several functions with certain levels. The UAV heterogeneity is mainly reflected in the functions or functional level, related to the sensors or loads equipped on the UAVs. For example, UAV U_1 is equipped with an infrared sensor and a high-resolution vision sensor, and UAV U_2 is equipped with synthetic aperture radars and a low-resolution vision sensor. It is obvious that the UAV U_1 has the functions of detecting infrared information, target recognition, and photographing, but it does not have all-weather remote sensing functions. UAV U_2 has all-weather remote sensing and target recognition functions, but the visual sensor's resolution limits the target recognition functional level of UAV. It may not satisfy the requirement of specific tasks, such as the recognition of small-size targets or complex background targets. In summary, the premise for a task to be executed is that the UAV meets the task's function and corresponding level requirements.

To make it clearer, suppose that W heterogeneous UAVs have N functions and each function is divided into K levels. Let $U_w \Rightarrow F_n^k$ represent the function of UAV $U_w \in \mathbf{U}$ and the corresponding functional level, $1 \leq n \leq N, 1 \leq k \leq K, 1 \leq w \leq W$. Let $T_t \Rightarrow SF_n^k$ denote the function and corresponding functional level of the UAV required to perform each task $T_t \in \mathbf{T}$. In other words, the task can be performed unless the UAV has the function n , and the functional level is no less than k , where n is the type of function and k is the corresponding functional level.

Examples of functional constraints are presented in Figure 1. Taking T_4 as an example, it requires that the UAV performing the task has function 4, and the corresponding level of the function is not less than 3. Thus, U_1 and U_2 cannot be assigned to perform T_4 , since U_1 and U_2 have no function 4. In addition, U_4 has function 4, but it also cannot perform T_4 , because the function level corresponding to function 4 is less than 3.

Figure 1. Examples of functional constraints.

2.2. UAV Trajectory Generation Model

In this paper, the Dubins airplane model [26] is introduced to generate the optimal flight path satisfying the kinematic constraints of the UAV in three-dimensional space. The spatial state of UAVs can be expressed by the following configuration:

$$q = (p, \theta, \psi), \quad (1)$$

where $p = (x, y, z)$ is the position of the UAV in three-dimensional space, $p \in \mathbb{R}^3$, ψ is the heading angle of the UAV, and θ is the flight-path angle. The UAV's kinematic equations are expressed as follows:

$$\begin{cases} \dot{x} = v_u \cdot \cos \psi \cdot \cos \theta \\ \dot{y} = v_u \cdot \sin \psi \cdot \cos \theta \\ \dot{z} = v_u \cdot \sin \theta \\ \dot{\psi} = c_\psi \cdot v_u \cdot \rho_{\min}^{-1} \end{cases}, \quad (2)$$

where v_u is the constant flight speed of the UAV, $\rho_{\min}$ is the minimum turning radius for the UAV, and c_ψ is the steering control command such that $c_\psi \in [-1, 1]$.

The Dubins airplane model was introduced as a trajectory planning model, as an extension of the Dubins car model. The Dubins car model is used for minimum-length trajectory planning in the two-dimensional plane from the start configuration $(x_{start}, y_{start}, \psi_{start})$ to the final configuration $(x_{final}, y_{final}, \psi_{final})$. The minimum-length trajectory generated by the Dubins car model is represented by six forms, such that the Dubins car set is defined by

$$D = \{LSL, LSR, LRL, RSL, RSR, RLR\}, \quad (3)$$

where L is an anticlockwise arc of a circle with radius $\rho_{\min}$, R is a clockwise arc of a circle with radius $\rho_{\min}$, and S is a straight line segment. Taking RSL and RLR as examples, RSL represents a trajectory composed by a clockwise arc followed by a straight line segment followed by an anticlockwise arc, while RLR represents a trajectory composed by a clockwise arc followed by an anticlockwise arc followed by a clockwise arc.

Based on the Dubins car model, the Dubins airplane model extends the motion to the vertical plane to generate a minimum-length trajectory in three-dimensional space. The Dubins airplane model divides the space into three cases: low altitude, medium altitude, and high altitude. For low altitude, the trajectory can be obtained by adjusting the flight angle of the UAV on the basis of the trajectory generated by the Dubins car model. For high altitude, a certain number of additional helix segments need to be added to the start or end arcs to give the UAV enough distance to adjust the altitude. For the minimum altitude case, the minimum-length trajectory can be obtained without adding several helix segments, as an incomplete helix segment is enough. More specific implementation details be found in references [26,28]. The minimum length trajectory forms generated by the Dubins airplane model are given as follows:

$$D^+ = \{RLSR, RLSTL, LRSR, LRSL, RSLR, RSRL, LSLR, LSRL\} \quad (4)$$

It is worth noting that, in order to achieve the minimum-length trajectory, the optimal heading angle ψ^* and optimal flight-path angle θ^* of the UAV flying between various targets need to be solved. The optimal flight-path angle θ^* can be obtained by solving the Dubins airplane model. As the input of the Dubins airplane model, the optimal heading angle cannot be optimized directly by the model. We use the method of uniform sampling to discretize the heading angle for approximately solving the optimal heading angle, such that the heading angle sample set is denoted as

$$\mathbf{H} = \{\psi_i : \psi_i = 2\pi i / N_\psi, i = 0, 1, \dots, N_\psi - 1\}, \quad (5)$$

where N_ψ is a positive integer, representing the number of samples. The approximate optimal heading angle is selected from the set of heading angle samples.

2.3. Mixed Integer Linear Programming Model

In this paper, we formulate the MFHCMPP using a mixed integer linear programming model. The objective is to optimize the cumulative flight distance of all UAVs for mission execution and the makespan. The mathematical model is as follows:

$$J = \alpha \cdot \sum_{t=1}^T \sum_{i=1}^W \sum_{j=1}^{N_\psi} X_{i,T_t}^{\psi_j} \cdot L_{i,T_t}^{\psi_j} / v_u^i + \beta \cdot \max \sum_{t=1}^T \sum_{j=1}^{N_\psi} X_{i,T_t}^{\psi_j} \cdot L_{i,T_t}^{\psi_j} / v_u^i, \quad (6)$$

$$f_{T'_t} \leq f_{T_t}; \forall t \in \mathbf{T}, \quad (7)$$

$$y_{T_t,i} = 1; t \in \mathbf{T}, \exists i \in \mathbf{U}, \quad (8)$$

$$\sum_{i=1}^W X_{i,T_t}^{\psi_j} \cdot F_n^k \geq Y_{T_t,n,k} \cdot SF_n^k; \forall t \in \mathbf{T}, \quad (9)$$

$$\sum_{t=1}^T y_{T_t,i,d} \leq 1; \forall i \in \mathbf{U}, \exists t \in \mathbf{T}, d. \quad (10)$$

The cost function is shown in Equation (6). The objective is to minimize the makespan and the total mission time at the same time. In Equation (6), α and β are positive real values and their sum is 1; they are used to weigh the importance of the makespan and total mission time. In this paper, α is set to 0.6 and β is set to 0.4. $X_{i,T_t}^{\psi_j}$ is a binary decision variable. If U_i performs T_t with heading angle ψ_j , $X_{i,T_t}^{\psi_j} = 1$; otherwise, $X_{i,T_t}^{\psi_j} = 0$. $L_{i,T_t}^{\psi_j}$ represents the trajectory length of the UAV i approaching the location of task T_t with the heading angle ψ_j . v_u^i is the flight speed of U_i . The constraint in Equation (7) guarantees the precedence constraints of the task, where $f_{T'_t}$ is the performing time of T'_t and f_{T_t} is the performing time of T_t . The constraint in Equation (8) ensures that each task can be handled only once, $y_{T_t,i}$ is a binary decision variable. If T_t is performed by U_i , $y_{T_t,i} = 1$; otherwise, $y_{T_t,i} = 0$. The constraint in Equation (9) guarantees that the UAV handling the task needs to meet the standard functional requirements of the task, where $Y_{t,n,k}$ represents that T_t can be executed by the UAV, with function n and a corresponding functional level no less than k . The constraint in Equation (10) guarantees that each UAV can handle at most one task at a time, where $y_{T_t,u,d}$ is a binary decision variable. If T_t is being performed by U_i at time d , $y_{T_t,u,d} = 1$; otherwise, $y_{T_t,u,d} = 0$.

3. Fruit Fly Optimization Algorithm

The fruit fly optimization algorithm is an emerging swarm intelligent optimization algorithm, which is based on the bionics principle of fruit fly foraging behavior. The excellent foraging behavior of a fruit fly mainly depends on its keen sense of smell and vision. The fruit fly can smell food sources far away and estimate the approximate location. The fruit fly can judge the odor concentration in the air and fly in the best possible direction of the food source. After approaching the food source, the fruit fly can discover other fruit flies' locations or find the food source directly using vision.

The fruit fly optimization algorithm uses the principles of olfaction and visual search in the predation process of fruit flies to design a smell-based and vision-based search operation. The search for the solution is achieved mainly by alternating the smell-based and vision-based search operation. The process of the original fruit fly optimization algorithm is as follows:

- Step 1: Initialization. The swarm center of the fruit flies, the size of the fruit fly swarm, and the maximum number of iterations are initialized.
- Step 2: Smell-based search operation.

- Step 2.1: Several fruit fly individuals are randomly generated near the center of the fruit fly swarm. The number of individuals is the size of the fruit fly swarm.
- Step 2.2: The individuals of the fruit fly swarm are evaluated, the corresponding odor concentration value is calculated, and the odor concentrations are ranked.
- Step 3: Vision-based search operation.
- Step 3.1: The individual with the best odor concentration is selected.
- Step 3.2: The center of the fruit fly swarm is updated.
- Step 4: If the termination condition is reached, the current optimal solution is taken as the final solution. Otherwise, the process returns to step 2.

4. Multi-Swarm Fruit Fly Optimization Algorithm

4.1. Procedure of MFOA

The local search ability of FOA is outstanding, but its global search ability is relatively weak. Thus, it is easy to fall into a local optimum, leading to premature convergence. In the process of iterative optimization, the swarm of FOA updates the center according to the location of the optimal individual of the fruit flies. Once the swarm center's location is determined, all the fruit flies in the swarm will fly toward the center of the swarm. This process leads to diversity loss, resulting in the FOA falling into a local optimal solution.

Two improved mechanisms are introduced to maintain the diversity and improve the global search ability of the algorithm: the multi-swarm mechanism and the dual search strategy mechanism. In the MFOA, the multi-swarm mechanism is used to determine multiple swarm centers, which effectively avoids the individual fruit flies gathering in one center, protecting the algorithm from falling into a local optimum. The dual search strategy mechanism includes a large-scale search strategy and a local search strategy, applied in the smell-based stage. The fruit flies have a larger fly radius with the large-scale search strategy, and the corresponding swarm focuses on the global search. For swarms with local search strategies, the fly radius of the fruit flies is small, focusing on the optimization of the current optimal solution. In each round of iterative optimization, the fruit fly swarms are divided into two groups according to a specific rule. Some of the swarms perform the large-scale search strategy, while the other swarms perform the local search strategy.

The flowchart of MFOA for solving the MFHCMPP is shown in Figure 2. It can be seen that the MFOA mainly consists of four phases, i.e., initialization, smell-based search, vision-based search, and dual search strategy switching. Firstly, the number of fruit fly swarms NS , the size of fruit fly swarm NP , and the maximum number of iterations $Maxgen$ are initialized. Then, based on the number of fruit fly swarms, NS fruit fly individuals are generated as the centers of the swarms. Each individual fruit fly represents a feasible solution to this problem. In the smell-based phase, some fruit fly swarms execute the local search strategy, while the remaining fruit fly swarms execute the large-scale search strategy. It is worth noting that all fruit fly swarms are set to perform the local search strategy in the first iteration. The search strategy of each fruit fly swarm in the subsequent iterations is determined by the dual strategy switching process. For the vision-based search stage, the new centers of the swarms are generated using the greedy selection strategy. In other words, during each iteration, the fruit fly individual with the optimal odor concentration in a swarm is designated as the new center of the swarm. Finally, the appropriate search strategy is assigned to the next iteration of the swarm. The iterative optimization process is repeated until the termination condition is reached.

Figure 2. The flowchart of the multi-swarm fruit fly optimization algorithm (MFOA).

4.2. Encoding

Each fruit fly individual represents a feasible solution for the MFHCMPP. Thus, the fruit fly individual encoding is the precondition to solve the problem. The fruit fly individual encoding is designed to be composed of six lists, i.e., task sequencing list (TS), predecessor task list (PT), task function requirements list (TF), heterogeneous UAV allocation list (UA), heterogeneous UAV function list (UF), and heading angle list (HA). The encoding example of a fruit fly individual is shown in Figure 3. The TS list represents the task execution order. The PT list indicates the predecessor task of the corresponding tasks in the TS list (e.g., in Figure 3, the predecessor task of task T_4 is task T_1 , while task T_1 has no predecessor tasks, which is represented by *Null* in the PT list). The TF list represents the functional constraint of the task (e.g., in Figure 3, task 5 cannot be performed unless the UAV has function 3 and the function level is not less than the standard level 2). The UA list denotes the heterogeneous UAV assignment for each task. The HA determines the terminal heading angle when the UAV approaches the task position. The mathematical description of the lists is given in Equations (11)–(16).

$$TS = [ts(1), ts(2), \dots, ts(T)], \quad (11)$$

$$PT = [pt(1), pt(2), \dots, pt(T)], \quad (12)$$

$$TF = [tf(1), tf(2), \dots, tf(T)], \quad (13)$$

$$UA = [ua(1), ua(2), \dots, ua(T)], \quad (14)$$

$$UF = [uf(1), uf(2), \dots, uf(T)], \quad (15)$$

$$HA = [ha(1), ha(2), \dots, ha(T)]. \quad (16)$$

TS	T_1	T_4	T_2	T_6	T_5	T_3
PT	Null	T_1	Null	T_4	T_2	Null
TF	SE_1^3	SE_4^3	SE_3^1	SE_3^3	SE_2^2	SE_4^2
UA	U_1	U_3	U_3	U_2	U_1	U_4
UF	$E_1^4 E_2^4$	$E_3^2 E_4^3$	$E_3^2 E_4^3$	$E_2^3 E_3^4$	$E_1^4 E_2^4$	$E_1^3 E_4^2$
HA	ψ_1	ψ_4	ψ_2	ψ_5	ψ_7	ψ_8

Figure 3. Encoding example of a fruit fly individual.

To be specific, $ts(i)$ ($i = 1, 2, \dots, T$) represents the i -th task to be assigned, $pt(i)$ ($i = 1, 2, \dots, T$) denotes the predecessor task of $ts(i)$, $tf(i)$ ($i = 1, 2, \dots, T$) indicates the standard functional requirements of the i -th task, $ua(i)$ ($i = 1, 2, \dots, T$) implies the UAV assigned to the i -th task, $uf(i)$ ($i = 1, 2, \dots, T$) represents the functions of $ua(i)$, and $ha(i)$ ($i = 1, 2, \dots, T$) indicates the terminal heading angle when the $ua(i)$ approaches the i -th task's location.

4.3. Initialization

The number of fruit fly swarms is NS , and the number of fruit flies in each swarm is NP . Thus, $NS \times NP$ fruit fly individuals should be generated. Each fruit fly individual is randomly generated.

Specifically, the TS list is generated by randomly selecting a task from the set of tasks **T** and inserted randomly into the TS list without violating the task priority constraints. The selection and insertion process is repeated until all tasks are inserted into the TS list. The PT list and the TF list are generated according to the task priority constraints and the task functional constraint, respectively. The UA list is generated by randomly allocating a UAV for each task, with the premise that the function of the UAV has to meet the task functional constraints. The UF list is generated according to the function of UAV and the corresponding function level. The HA list is generated by randomly selecting an approach heading angle from the heading angle sample set **H** for the UAV.

4.4. Smell-Based Search

The smell-based search stage, as the core of FOA, greatly affects the efficiency of problem solving and the quality of the solution. In the smell-based search stage of this paper, we drive the fruit fly swarm through the dual search strategy mechanism, i.e., incorporating a large-scale search strategy and a local search strategy. The purpose is to maintain the diversity of the swarm while ensuring the convergence rate of the problem.

4.4.1. Local Search Strategy

The local search strategy focuses on further optimizing the advantage of fruit fly individuals. The operation operators do not change the position of individual fruit flies significantly, because this would cause the fruit flies to be too far away from the current position and reduce the search efficiency. An example of the local search strategy is shown in Figure 4. The specific operators are given as follows:

- **Task Exchange:** The task exchange operation adopts the neighborhood exchange method to exchange the columns of two adjacent tasks without priority constraints (e.g., in Figure 3, the element of the PT list corresponding to the task T_2 is "Null", which means that T_2 has no predecessor task. The element of the PT list corresponding to the task T_6 is " T_4 "; thus, T_4 is the predecessor task of the task T_6 . Task T_2 and task T_6 have no priority constraints; thus, the task exchange operator can be used for operation. The columns of tasks T_1 and T_4 cannot be exchanged because the predecessor task of T_4 is task T_1).
- **UAV Reassign:** The UAV reassign operation randomly selects a task and reassigns it to a new UAV that meets the task function constraints (e.g., in Figure 3, task T_5 requires that the UAV performing the task has the function 5 and the standard level is not less than 2. Therefore, the UAV reassign operator traverses the UF list and select another UAV that meets the task function constraints for reassignment).
- **Heading Angle Switch:** The heading angle switch operation randomly selects an approach heading angle for resampling according to N_ψ .

Figure 4. Example of the local search strategy.

4.4.2. Large-Scale Search Strategy

The large-scale search strategy focuses on keeping the fruit fly individuals with poor performance away from the current position. An example of the large-scale search strategy is shown in Figure 5. The specific operators are given as follows:

- **Interval task reordering:** The interval task reordering operator randomly selects the starting point and the ending point in the TS list. The number of tasks included between the start point and the end point is N_{inter} , where N_{inter} is even and $4 \leq N_{inter} \leq T - 4$. Then, $N_{inter}/2$ columns are taken out from the fruit fly individuals and reinserted randomly without violating the task priority constraints.
- **Multi-UAV Reassign:** The multi-UAV reassign operation randomly selects $N_{reassign}$ tasks and reassigns them to new UAVs that meet the task function constraints.
- **Multi-Heading Angle Switch:** The heading angle switch operation randomly selects N_{switch} approach heading angles for resampling according to N_ψ .

Figure 5. Example of large-scale search strategy.

4.5. Vision-Based Search

In the vision-based search stage, each fruit fly swarm evaluates the fruit fly individuals to obtain each fruit fly individual's odor concentration value. In order to update the central position of each fruit fly swarm, a greedy selection strategy is adopted. The individual's position with the optimal odor concentration in each fruit fly swarm is updated to the new center of the fruit fly swarm. It is worth noting that, if the fruit fly individual's position with the optimal odor concentration is the same as the center of the current fruit fly swarm, the position of the center of the fruit fly swarm is not updated.

The odor concentration value of each fruit fly individual is determined by the makespan and the total mission time. Thus, to calculate the odor concentration value, the task commands for each UAV should be generated. The process of UAV task command generation is illustrated in Figure 6. The fruit fly individual is split according to the UAV identifier (ID). Taking U_2 as an example, its task command is to execute T_5 and T_6 in sequence. The heading angle of U_2 is designated as ψ_7 when it approaches the location of T_5 . Then, U_2 leaves the location of T_5 with the heading angle ψ_7 and approaches the location of T_6 with the heading angle ψ_5 . According to the task commands of each UAV and the Dubins airplane model, the flight time of each UAV can be calculated. Furthermore, the odor concentration value of the fruit fly individual, i.e., the makespan and the total mission time, can be determined.

Figure 6. Process of UAV task command generation.

Next, taking a swarm of multiple fruit fly swarms as an example to illustrate the greedy selection strategy, once the odor concentration values of all individuals in the fruit fly swarm are calculated, the greedy selection strategy compares the odor concentration values of all fruit flies and selects the fruit fly with the best odor concentration value as the new swarm center. In the next iteration of the process, the center of the fruit fly swarm executes a smell-based search NP times to generate

NP new fruit fly individuals. A more detailed explanation of the vision-based search can be found in Reference [37].

4.6. Dual Strategy Switching

The dual strategy switching process is an important process that controls the smell-based search strategy of all fruit fly swarms. The local search strategy and the large-scale search strategy are assigned to NS fruit fly swarms. The process of the dual strategy switching method assigns the search strategy to each fruit fly swarm as follows:

- Step 1: The odor concentration value of each fruit fly center F_i is calculated, where $1 \leq i \leq NS$.
- Step 2: The average odor concentration value of all the fruit fly swarm centers F_{avg} is calculated, where $i = 1$.
- Step 3: The odor concentration value F_i is compared with the average odor concentration value F_{avg} of all fruit fly swarm centers. If F_i is larger than the average odor concentration value F_{avg} , then the local search strategy is assigned to the fruit fly swarm; otherwise, the large-scale search strategy is assigned.
- Step 4: Let $i = i + 1$. If $i = NS + 1$, it means that the search strategy of all fruit fly swarms is assigned, and the dual strategy switching process is completed. Otherwise, to the process returns to step 3.

The average odor concentration value of all the fruit fly swarm centers can be calculated as follows:

$$F_{avg} = \sum_{i=1}^{NS} F_i / NS, \quad (17)$$

where F_{avg} is the average odor concentration of the center of all fruit fly swarms, and F_i represents the center odor concentration of the fruit fly swarm i ($1 \leq i \leq NS$).

5. Results

5.1. Sample Run

In order to test the feasibility of the solution, a simple example experiment scene is given. The example scene is generated in three-dimensional (3D) space and contains three multi-functional heterogeneous UAVs and nine tasks. As shown in Table 1, the UAVs in the scene differ in terms of function type, function level, and mobility.

Table 1. Parameters of heterogeneous unmanned aerial vehicles (UAVs).

Parameters	No. UAV		
	U_1	U_2	U_3
Base Location	[1000, 1000]	[1000, 1000]	[1000, 1000]
Base Height	100	100	100
Function	$[F_2^3, F_3^2]$	$[F_1^2, F_3^3]$	$[F_1^3, F_2^2]$
Airspeed (m/s)	50	40	60
Maximum Bank Angle	$\pi/4$	$\pi/4$	$\pi/4$
Maximum Flight-Path angle	$\pi/6$	$\pi/6$	$\pi/6$

Table 2 shows the task settings of task 1 to task 4, and Table 3 shows the task settings of task 5 to task 9. The task settings are composed of four parts: task location, task height, predecessors, and functional requirements. Taking task 3 as an example, its space configuration is composed of task location and task height, and the UAV can access it at any flight path angle and heading angle. Its predecessor is task 1, which means that the UAV must access it after task 1 is completed.

Table 2. Parameters of tasks 1–4.

Parameters	No. Task			
	T_1	T_2	T_3	T_4
Task Location	[3700, 3400]	[2500, 2500]	[2500, 4000]	[3700, 4400]
Task Height	600	1100	1100	1900
Predecessor Task	T_3	Null	Null	Null
Function Requirement	F_1^2	F_2^2	F_1^1	F_1^2

Table 3. Parameters of tasks 5–9.

Parameters	No. Task				
	T_5	T_6	T_7	T_8	T_9
Location	[4700, 3400]	[3800, 1500]	[4500, 1200]	[2500, 1500]	[1800, 1800]
Height	2100	500	1800	200	1100
Predecessor Task	T_9	Null	Null	Null	Null
Function Requirement	F_3^2	F_2^2	F_1^2	F_3^1	F_2^1

The MFOA parameters are set to $NS = 30$, $NP = 20$, and $N_\psi = 100$. In order to verify the generation of three-dimensional trajectories, the three-dimensional trajectories of multifunctional heterogeneous UAV performing tasks are shown in Figure 7. The blue line represents the three-dimensional trajectory of UAV 1's mission. After starting from the base, it performs task 9, task 4, and task 7, before returning to the base. The yellow line represents the three-dimensional trajectory of UAV 2's mission. After starting from the base, it performs task 3, task 1, and task 8, before returning to the base. The red line represents the three-dimensional trajectory of UAV 3's mission. After it departs from the base, it executes task 9, task 4, and task 7, before returning to the base. The cumulative mission completion time for all UAVs is 772.26 s. The mission completion times for UAV 1, UAV 2, and UAV 3 are 260.75 s, 284.65 s, and 226.86 s, respectively. It can be seen from Figure 7 that all tasks are assigned to multifunctional UAVs that meet the task functional requirements.

Figure 7. The three-dimensional trajectories of multi-UAVs.

Regarding the timing of tasks, the predecessor of task 1 is task 3, and both tasks are executed by UAV 2. The timing constraints are satisfied. As shown in Table 3, the predecessor of task 5 in the initial task setting is task 9. Task 9 is executed by UAV 3 at 66.55 s, and task 5 is executed by UAV 1 at 139.28 s. Therefore, the timing constraints between task 5 and task 9 are also satisfied.

The MFOA algorithm was verified in this section for 3D trajectory generation problems, timing constraint processing, and function satisfaction allocation. Next, the performance of the MFOA algorithm is evaluated.

5.2. MFOA Performance Analysis

5.2.1. Experimental Set-Up

In order to verify the performance of the proposed MFOA algorithm, nine datasets with increasing complexity were designed. Table 4 shows the characteristics of the nine datasets, covering the number of tasks, the number of UAVs, the upper limit of function type, the upper limit of corresponding function level, the number of tasks with timing constraints, and the number of UAV functions. Taking dataset 1 as an example, it contained nine missions and three multifunctional heterogeneous UAVs. The number of functions of each multifunctional heterogeneous UAV was 2, the upper limit of the function type was 3, and the upper limit of the corresponding function level was 2. The functional requirements of the task were generated according to the functions of the UAV to ensure that all tasks could be executed.

Table 4. The characteristics of the datasets. ID—identifier.

Dataset ID	Tasks	UAVs	Number of UAV Functions	Function Types Upper Limit	Function Levels Upper Limit	Predecessor Tasks
1	9	3	2	3	2	3
2	12	3	2	3	3	5
3	16	4	2	4	3	4
4	25	6	3	4	3	7
5	28	6	3	5	4	9
6	35	8	3	5	4	12
7	50	12	4	6	5	13
8	72	14	4	6	5	15
9	85	16	4	6	6	15

On this basis, the datasets were divided into a small-scale task set, medium-scale task set, and large-scale task set. Among them, datasets 1–3 belonged to the small-scale task set, datasets 4–6 belonged to the medium-scale task set, and datasets 7–9 belonged to the large-scale task set. The coordinates of the task locations were randomly generated within the area of 8000×8000 , and the height was randomly selected within the vertical range of $[100, 5000]$. The maximum bank angle and maximum flight-path angle were uniformly set to $\pi/4$ and $\pi/6$ degrees. The airspeed of the UAVs was uniformly set to 40–70 ms/s.

To test the performance of MFOA in different scale mission scenarios, the proposed MOFA algorithm was compared with the random search algorithm (RSM) and FOA algorithm in a specific small-scale mission scenario, medium-scale mission scenario, and large-scale mission scenario. The random search algorithm is usually used as a benchmark method for comparison to prove the effectiveness of new algorithms. It uses a random strategy to update the individuals in the population and selects the best individual as the starting point for the next iteration. In Reference [32], random search algorithm was used to compare the performance of algorithms. The scale of the RSM solution was consistent with FOA and MFOA. Initially, $NS \times NP$ candidate solutions were generated by random selection. In each iteration, RSM kept the candidate solution with the optimal objective value, and the remaining $NS \times NP - 1$ candidate solutions were generated by the random selection strategy. For FOA, $NS \times NP$ initial solutions were randomly generated based on the random selection strategy. All fruit fly individuals were updated based on the only fruit fly center. The smell-based search stage for FOA adopted the local search strategy like the MFOA, and the visual search stage adopted the greedy selection strategy.

5.2.2. Small-Scale Mission Scenario Test

As shown in Table 4, dataset 1 was selected as the test scenario to verify the performance of the MFOA algorithm in the small-scale mission scenario. The operating parameters of the RSM, FOA, and MFOA algorithms are shown in Table 5.

Table 5. The operating parameters of the random search algorithm (RSM), FOA, and MFOA for the small-scale mission scenario test.

RSM			FOA			MFOA			
<i>NP</i>	<i>N_φ</i>	<i>Maxgen</i>	<i>NP</i>	<i>N_φ</i>	<i>Maxgen</i>	<i>NS</i>	<i>NP</i>	<i>N_φ</i>	<i>Maxgen</i>
300	100	400	300	100	400	50	6	100	400

The comparison of convergence efficiency for the RSM, FOA, and MFOA in the small-scale mission scenario is shown in Figure 8. The convergence efficiency of the algorithms is described by $E(J_k/J_0)$, where J_0 represents the optimal odor concentration in the initial population, and J_k represents the optimal odor concentration in the population of the k -th generation. As shown in Figure 8, MFOA had the fastest convergence rate and RSM had the slowest convergence rate in the small-scale mission scenario. The convergence performance of MFOA was slightly better than that of FOA.

Figure 8. Comparison of convergence efficiency of RSM, FOA, and MFOA in the small-scale mission scenario (dataset 1).

The objective value distribution of the RSM, FOA, and MFOA for the small-scale mission scenario is shown in Figure 9. The RSM algorithm had the highest concentration of target values and the lowest edge value, indicating that the RSM algorithm had low optimization efficiency and fell prematurely into a local optimum. The FOA algorithm had a faster convergence speed and stronger optimization performance when processing the small-scale mission scenario, but the ability to jump out of the local optimal value was weaker. Compared with the other two algorithms, the objective value of MFOA was concentrated in a lower value, indicating that the optimization effect was better when processing the small-scale task scenario than the other two algorithms.

Figure 9. Comparison of objective value distribution of RSM, FOA, and MFOA in the small-scale mission scenario (dataset 1).

5.2.3. Medium-Scale Mission Scenario Test

As shown in Table 4, dataset 4 was selected as the test scenario to verify the performance of the MFOA algorithm in the medium-scale mission scenario. The operating parameters of the RSM, FOA, and MFOA are shown in Table 6.

Table 6. The operating parameters of the RSM, FOA, and MFOA for the medium-scale mission scenario test.

Dataset ID	RSM			FOA			MFOA			
	NP	N_φ	Maxgen	NP	N_φ	Maxgen	NS	NP	N_φ	Maxgen
1	800	100	400	800	100	400	80	10	100	400

The comparison of convergence efficiency of the RSM, FOA, and MFOA in the medium-scale mission scenario is shown in Figure 10. The convergence performance of RSM was poor, and it was prematurely caught in a local optimum. In terms of optimizing the processing of large-scale task scenarios, the convergence performance of the FOA algorithm was weaker than when processing the small-scale mission scenario. The MFOA algorithm had better convergence performance when dealing with mid-scale problems, and, after 1000 iterations, the algorithm did not fully converge and still had a certain optimization space.

Figure 10. Comparison of convergence efficiency of RSM, FOA, and MFOA algorithms in the medium-scale mission scenario (dataset 4).

The objective value distribution of the RSM, FOA, and MFOA for the medium-scale mission scenario is shown in Figure 11. The lower edge value of RSM and FOA was higher, indicating that the optimization effect of the two algorithms when processing the medium-scale task scenario was weak. The median value of the target value of the MFOA algorithm was still far away from the lower edge value, indicating that the algorithm did not fall into a local optimum.

Figure 11. Comparison of objective value distribution of the RSM, FOA, and MFOA in the medium-scale mission scenario (dataset 4).

5.2.4. Large-Scale Mission Scenario Test

As shown in Table 4, dataset 7 was selected as the test scenario to verify the performance of the MFOA algorithm in a large-scale mission scenario. The operating parameters of the RSM, FOA, and MFOA are shown in Table 7.

Table 7. The operating parameters of the RSM, FOA, and MFOA for the large-scale mission scenario test.

Dataset ID	RSM			FOA			MFOA			
	NP	N_{φ}	Maxgen	NP	N_{φ}	Maxgen	NS	NP	N_{φ}	Maxgen
1	2000	100	400	2000	100	400	100	20	100	400

The comparison of convergence efficiency of the RSM, FOA, and MFOA in the large-scale mission scenario is shown in Figure 12. The objective value distribution of the RMS, FOA, and MFOA for the large-scale mission scenario is shown in Figure 13. Due to the increased task complexity of the large-scale mission scenario, the convergence efficiency of RSM and FOA was low. When the algorithm converged, the lower edge of the objective value distribution was high, indicating that the optimization effect was poor. In the large-scale task scenario, the MFOA algorithm had obvious advantages over the other two algorithms, and the algorithm optimization performance was better. It shows that the MFOA algorithm is more suitable to deal with complex optimization problems with a large-scale search space.

5.3. Monte Carlo Stability Analysis

In order to test the stability of the MFOA algorithm, the stability of the MFOA algorithm was tested by Monte Carlo simulation and compared with RSM and FOA. The test dataset is shown in Table 4, which contains the small-scale mission scenario set, medium-scale mission scenario set, and large-scale mission scenario set. The FOA, RSM, and MFOA were run 1000 times for each mission scenario in the dataset shown in Table 4. In the Monte Carlo simulation, the algorithm parameter settings for the small-scale mission scenario, medium-scale mission scenario, and large-scale mission scenario are shown in Tables 5–7, respectively.

The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the small-scale mission scenario represented by datasets 1–3 are shown in Figure 14. In Monte Carlo simulations based on datasets 1–3, the average objective value of MFOA was smaller than that of RSM and FOA, indicating that the MFOA is stable in small-scale mission scenarios. The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the medium-scale mission scenario represented by datasets 4–6 are shown in Figure 15.

Figure 12. Comparison of convergence efficiency of the RSM, FOA, and MFOA in the large-scale mission scenario (dataset 7).

Figure 13. Comparison of objective value distribution of the RSM, FOA, and MFOA in the large-scale mission scenario (dataset 7).

Figure 14. The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the small-scale mission scenario (datasets 1–3).

Figure 15. The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the medium-scale mission scenario (datasets 4–6).

Similar to the small-scale scenario, the optimization performance of MFOA in the medium-scale mission scenario was superior to RSM and FOA. The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the large-scale mission scenario represented by datasets 7–9 are shown in Figure 16. From the analysis of Monte Carlo simulation results, it can be concluded that, due to the increase in the number of tasks, the number of drones, the heterogeneity of UAVs, and the timing correlations, the convergence efficiency of RSM and FOA was significantly reduced compared to the medium- and small-scale scenarios. Simulation results show that RSM and FOA are not suitable for large-scale mission scenarios. The optimization results of the MFOA algorithm in the large-scale task scenario still indicated good performance. It shows that the optimized performance of MFOA can be stable in small-, medium-, and large-scale mission scenarios.

Figure 16. The results of the Monte Carlo simulation of RSM, FOA, and MFOA in the large-scale mission scenario (datasets 7–9).

6. Conclusions

In this paper, a multi-swarm fruit fly optimization algorithm with dual strategy switching (MFOA) was proposed to solve the multi-functional heterogeneous UAV cooperative mission planning problem with the criterion of simultaneously minimizing the makespan and the total mission time. The multi-functional heterogeneous UAV cooperative mission planning problem was considered to include three sub-problems of task assignment, task scheduling, and three-dimensional trajectory planning. Thus, a multi-list encoding architecture was proposed to represent the candidate solutions. The multi-swarm mechanism was introduced to enhance the global search performance of the basic fruit fly optimization algorithm. Meanwhile, in the smell-based search stage, a dual strategy collaborative search method was proposed to balance global and local search performance. In the vision-search stage, the decoding method of UAV task commands was given, and the greedy selection strategy

was applied to update the center of the fruit fly swarm. A dual strategy adaptive switching method was given for adaptively assigning a multi-swarm search strategy in the smell-based search stage. A sample run was used to verify the effectiveness of the MFOA algorithm to deal with timing constraints, functional constraints, and the three-dimensional trajectory planning problem. Next, the performance of three algorithms (RSM, FOA, and MFOA) was compared in small-scale, medium-scale, and large-scale mission scenarios. The simulation results proved the superiority of MFOA in the three scenarios. Finally, Monte Carlo simulation experiments were carried out, and the experimental results showed that MFOA maintained its stability in small-, medium-, and large-scale mission scenarios.

Regarding further work in the future, we will consider the following aspects:

- More practical factors will be introduced in the mission planning problem, such as fuel constraints, UAV cost constraints, target value, and so on.
- Uncertain factors will be considered in the mission planning problem, such as the uncertainty of the target location or the uncertainty of the UAV's flying speed.
- The fruit fly optimization algorithm will be further optimized to improve its performance.
- The complexity of the fruit fly optimization algorithm will be reduced, and new optimization mechanisms will be explored to test its performance in real-time mission planning problems.

Author Contributions: Conceptualization, R.L. and J.G.; methodology, R.L. and H.Z.; software, R.L.; data curation, R.L.; writing—review and editing, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Fund of China (No.61973101) and the Aeronautical Science Foundation of China (No.20180577005).

Acknowledgments: The authors acknowledge the support from the ISAP Laboratory in Harbin Institute of Technology, Harbin, China.

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

References

1. Jianqing, L.; Changsheng, G.; Chaoyong, L.; Wuxing, J. A survey on moving mass control technology. *Aerosp. Sci. Technol.* **2018**, *82*, 594–606.
2. Miao, Y.; Deqing, H.; Wei, H. Robust adaptive iterative learning control for discrete-time nonlinear systems with both parametric and nonparametric uncertainties. *Int. J. Adapt. Control. Signal. Process.* **2016**, *30*, 972–985.
3. Tas, D.; Dellaert, N.; Woensel, T.V.; Kok, T.D. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Comput. Oper. Res.* **2013**, *40*, 214–224. [[CrossRef](#)]
4. Oddi, A.; Rasconi, R.; Cesta, A.; Smith, S.F. Solving job shop scheduling with setup times through constraint-based iterative sampling: An experimental analysis. *Ann. Math. Artif. Intell.* **2011**, *62*, 371–402. [[CrossRef](#)]
5. Garone, E.; Naldi, R.; Casavola, A. Traveling salesman problem for a class of carrier-vehicle systems. *J. Guid. Control. Dynam.* **2011**, *34*, 1272–1276. [[CrossRef](#)]
6. Chao, I.M.; Golden, B.L.; Isil, E.A. The team orienteering problem. *Eur. J. Oper. Res.* **1996**, *88*, 464–474. [[CrossRef](#)]
7. Faigl, J. Data collection path planning with spatially correlated measurements using growing self-organizing array. *Appl. Soft Comput.* **2018**, *75*. [[CrossRef](#)]
8. Yu, J.; Schwager, M.; Rus, D. Correlated orienteering problem and its application to persistent monitoring tasks. *IEEE Trans. Robot.* **2016**, *32*, 1106–1118. [[CrossRef](#)]
9. Clausen, J. Branch and bound algorithms—principles and examples. *Computer* **1999**, *22*, 658–663.
10. Barnhart, C.; Johnson, E.L.; Nemhauser, G.L.; Savelsbergh, M.W.P.; Vance, P.H. Branch-and-Price: Column generation for solving huge integer programs. *Oper. Res.* **1998**, *46*, 316–329. [[CrossRef](#)]
11. Feillet, D. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR* **2010**, *8*, 407–424. [[CrossRef](#)]
12. Shima, T.; Rasmussen, S.J.; Sparks, A.G.; Passino, K.M. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **2006**, *33*, 3252–3269. [[CrossRef](#)]

13. Eun, Y.; Bang, H. Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithm. *J. Aircr.* **2009**, *46*, 338–343. [\[CrossRef\]](#)
14. 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.* **2017**, *310*, 339–350. [\[CrossRef\]](#)
15. Ramirez-Atencia, C.; Camacho, D. Extending QGroundControl for Automated Mission Planning of UAVs. *Sensors* **2018**, *18*, 2339. [\[CrossRef\]](#)
16. Zhu, Z.; Tang, B.; Yuan, J. Multirobot task allocation based on an improved particle swarm optimization approach. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1–22. [\[CrossRef\]](#)
17. Zhang, X.; Duan, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. *Appl. Soft Comput.* **2015**, *26*, 270–284. [\[CrossRef\]](#)
18. Ramirez-Atencia, C.; Bello-Ortiz, G.; R-Moreno, M.D.; Camacho, D. Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms. *Soft Comput.* **2016**, *21*, 1–18. [\[CrossRef\]](#)
19. 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. [\[CrossRef\]](#)
20. 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. [\[CrossRef\]](#)
21. Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A*. *J. ACM* **1985**, *32*, 505–536. [\[CrossRef\]](#)
22. Lavalle, S.M.; Kuffner, J.J. Randomized kinodynamic planning. In Proceedings of the IEEE International Conference on Robotics & Automation (IEEE 1999), Detroit, MI, USA, 10–15 May 1999.
23. Kuffner, J.J.; Lavalle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings, (IEEE 2002), San Francisco, CA, USA, 24–28 April 2000.
24. Gemeinder, M.; Gerke, M. GA-based path planning for mobile robot systems employing an active search algorithm. *Appl. Soft Comput.* **2003**, *3*, 149–158. [\[CrossRef\]](#)
25. Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal position. *Am. J. Math.* **1957**, *79*, 497–516. [\[CrossRef\]](#)
26. Chitsaz, H.; LaValle, S.M. Time-optimal paths for a dubins airplane. In Proceedings of the 46th IEEE CDC, New Orleans, LA, USA, 12–14 December 2007; pp. 2379–2384.
27. Isaiah, P.; Shima, T. Motion planning algorithms for the dubins travelling salesperson problem. *Automatica* **2015**, *53*, 247–255. [\[CrossRef\]](#)
28. Owen, M.; Beard, R.W.; McLain, T.W. Implementing dubins airplane paths on fixed-wing UAVs. In *Handbook of Unmanned Aerial Vehicles*; Springer: Dordrecht, The Netherlands, 2015; pp. 1677–1701.
29. Macharet, D.G.; Neto, A.A.; Neto, V.F.D.C.; Campos, M.F.M. Nonholonomic path planning optimization for Dubins' vehicles. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011.
30. Rasmussen, S.J.; Shima, T. Tree search algorithm for assigning cooperating UAVs to multiple tasks. *Int. J. Robust. Nonlinear Control.* **2008**, *18*, 135–153. [\[CrossRef\]](#)
31. Edison, E.; Shima, T. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **2011**, *38*, 340–356. [\[CrossRef\]](#)
32. Deng, Q.; Yu, J.; Wang, N. Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes. *Chin. J. Aeronaut.* **2013**, *26*, 1238–1250. [\[CrossRef\]](#)
33. Weinan, W.; Xiaogang, W.; Naigang, C. Fast and coupled solution for cooperative mission planning of multiple heterogeneous unmanned aerial vehicles. *Aerosp. Sci. Technol.* **2018**, *79*, 131–144.
34. Xiangyin, Z.; Xingyang, L.; Songmin, J.; Xiuzhi, L. A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning. *Appl. Soft Comput.* **2018**, *70*, 371–388.
35. Xiaolong, Z.; Ling, W.; Huanyu, Z. A knowledge-based fruit fly optimization algorithm for multi-skill resource-constrained project scheduling problem. *Soft Comput.* **2015**, *21*, 1–12.

36. Zheng, X.L.; Wang, L.; Wang, S.Y. A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. *Knowl. Based Syst.* **2014**, *57*, 95–103. [[CrossRef](#)]
37. Pan, W.T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [[CrossRef](#)]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).