An Integrated Mission Planning Framework for Sensor Allocation and Path Planning of Heterogeneous Multi-UAV Systems

Hongxing Zheng 1,* and Jinpeng Yuan 2

Abstract: Mission planning is the guidance for a UAV team to perform missions, which plays the most critical role in military and civil applications. For complex tasks, it requires heterogeneous cooperative multi-UAVs to satisfy several mission requirements. Meanwhile, airborne sensor allocation and path planning are the critical components of heterogeneous multi-UAVs system mission planning problems, which affect the mission profit to a large extent. This paper establishes the mathematical model for the integrated sensor allocation and path planning problem to maximize the total task profit and minimize travel costs, simultaneously. We present an integrated mission planning framework based on a two-level adaptive variable neighborhood search algorithm to address the coupled problem. The first-level is devoted to planning a reasonable airborne sensor allocation plan, and the second-level aims to optimize the path of the heterogeneous multi-UAVs system. To improve the mission planning framework’s efficiency, an adaptive mechanism is presented to guide the search direction intelligently during the iterative process. Simulation results show that the effectiveness of the proposed framework. Compared to the conventional methods, the better performance of planning results is achieved.

Keywords: heterogeneous multi-UAVs system; airborne sensor allocation; path planning; mission planning; two-level adaptive variable neighborhood search

1. Introduction

Low cost, safety, robust endurance, and autonomous operation are essential advantages of unmanned aerial vehicles (UAVs), leading to widespread application in many fields. With the rapid development of UAVs, sensors, and communications technology, multiple heterogeneous UAVs perform complex tasks collaboratively has become an essential application paradigm. Due to its broad application in various practical, real-world missions (e.g., intelligence gathering, reconnaissance, and combat intelligence), the importance of multi-UAVs system mission planning is increasingly highlighted. In recent years, the mission planning of multi-UAV systems based on specific issues has been extensively studied. The mission planning for heterogeneous multi-UAVs systems, as an essential component of the conventional mission planning problem, focus on solving the integrated task assignment and path planning problem for heterogeneous multi-UAVs systems. Specifically, it aims to find an optimal solution to assign a reasonable mission route to each heterogeneous UAV to satisfy complex mission requirements and generate a corresponding flyable collision-free path.

Task assignment is an essential subproblem in mission planning, which belongs to the combinatorial optimization problem in operations research. For the multi-UAVs task assignment problem, it generally refers to assigning tasks to be performed for a multi-UAVs system, while meeting the task requirements, optimizing the given objectives, such as mission completion deadline, total fuel consumption, etc. The heterogeneous multi-UAVs system improves task execution efficiency, but it also increases the task assignment
problem’s complexity. Due to the task assignment’s natural complexity, it belongs to a type of NP-hard combinatorial optimization problem, which makes the deterministic method unable to solve the problem in polynomial time [1]. Therefore, many heuristics methods have been studied for solving the task assignment problem to generate a suboptimal solution within the accepted time, such as self-organizing map neural network [2], genetic algorithm [3,4], ant colony algorithm [5], and particle swarm optimization [6].

Path planning is another essential subproblem in mission planning. Generally, path planning is to generate collision-free paths connecting the start and end positions in space. As a fundamental problem, path planning has been widely studied in many fields, such as robots, robotic arms, games, and UAVs. Path planning methods are mainly classified in terms of heuristic-based [7], learning-based [8–10], sampling-based [11], geometry-based [12], etc. Note that solving mission planning problems usually requires evaluating many solutions to obtain an optimized mission plan. Therefore, the path planning method incorporates into the mission planning framework is required to generate the optimal collision-free path as fast as possible. Generally, heuristic-based incorporating geometry-based applications are suitable for generating collision-free and flyable paths in the mission planning framework, which depends on its fast computing speed.

Task assignment and path planning problems are usually integrated into a unified framework, depending on the coupling of the two problems. Edison et al. [13] addressed the mission planning problem that multi-UAVs cooperatively execute a group of successive tasks with strict priority on each ground target. The mission planning issue is formulated as an integrated task assignment and path planning problem. Meanwhile, a mission planning framework based on a genetic algorithm is developed to solve the problem. Deng et al. [14] presented an improved genetic algorithm with multi-type genes to solve the integrated task assignment and path planning problem. Further, a mirror representation of vehicles is developed to deal with the limited resource constraint. Wang et al. [15] presented a reconnaissance task assignment model to optimize UAVs’ task sequence by the opposition-based genetic algorithm using double-chromosomes encoding and multiple mutations. Meanwhile, heterogeneous targets classified according to features of target geometry and sensor’s field of view are addressed. Chen et al. [16] presented a systematical framework based on the modified two-part wolf pack search algorithm. In addition, the parameter and time-sensitive uncertainties are considered in the extended task assignment problem, which is formulated by a robust optimization method. Jia et al. [17] studied the mission planning problem with stochastic velocities and time windows for a heterogeneous multi-UAVs system, which is solved by a modified genetic algorithm. Further, a path extension mechanism based on the dubins method is proposed to meet the tasks’ priority constraints. Mufalli et al. [18] proposed a column generation algorithm to deal with integrated sensor selection and route planning for UAVs. Deng et al. [19] proposed a novel large-area-coverage planning framework that collectively optimizes the paths for aerial and ground vehicles under the background of multiple UAVs and a ground vehicle cooperative large area coverage mission. Causa et al. [20] presented an algorithm for solving multi-UAV path planning in scenarios with heterogeneous Global Navigation Satellite Systems (GNSS) coverage. Further, Causa et al. [21] tackled strategic path planning for a multi-UAV routing problem in low altitude urban environment. A multi-step strategy was presented, which included an automated definition of GNSS-challenging volumes based on a georeferenced three-dimensional environment model, derivation of candidate obstacle-free paths between waypoints, waypoint assignment, and definition of time-tagged trajectories for all UAVs. Kim et al. [22] address the drone-aided delivery and pickup route planning problem under the background of healthcare services, a preprocessing algorithm, a Partition method, and a Lagrangian Relaxation (LR) method are developed as solution approaches to improve the computational performance. Lim et al. [23] presented a reliable parcel delivery schedule using drones, which considered the battery consumption rate as a function of payload in the operational planning optimization. Meanwhile, a
variable preprocessing algorithm and primal and dual bound generation methods are proposed to improve the computational time.

Notably, most of the current research on multi-UAVs mission planning focuses on the integrated task assignment and path planning problem, which has led to a series of variants with additional attributes, such as multiple ground control stations, heterogeneous UAVs team, time window, task priority, heterogeneous target, etc. The sensors are assumed to be fixed on the UAV and cannot be replaced; thus, the sensor allocation problem is ignored. However, interchangeable sensor configuration and multiple sensors coordination are the essential features of the modern UAV platform. In general, little research considers sensor allocation, task assignment, and collision-free trajectory generation problems, simultaneously. In order to meet the application requirements of the real world, sensor allocation, task assignment, and collision-free path generation problems are integrated into this paper.

From the optimization algorithm’s perspective, to improve the quality of the solution and the optimized efficiency, many heuristic algorithms are presented to solve the multi-UAVs mission planning problems. Ramirez-Atencia et al. [24] presented a multi-objective genetic algorithm for solving the complex mission planning problem involving a team of UAVs and a group of ground control stations. The constraint satisfaction problem is merged into the mission planning framework to improve search efficiency. Then, Ramirez-Atencia et al. [25] proposed a weighted random generator incorporated into the mission planning framework to further improve the convergence rate. Wu et al. [26] presented a coupled and distributed mission planning framework for solving the integrated task assignment and path planning problem. The distributed genetic algorithm implements the mission planning framework to obtain a fast and feasible solution. Luo et al. [27] presented the multi-functions heterogeneous UAVs cooperative mission planning model, which restricts the heterogeneous UAV performed the task from the function level and function type.

Most of the mission planning framework is implemented by population-based optimization algorithms, such as genetic algorithm, particle swarm optimization, fruit fly optimization algorithm. These population-based optimization methods must repair many infeasible solutions in the iterative process [28], and the parameters are usually difficult to set. The variable neighborhood search (VNS) is an effective metaheuristic—several applications [29–33] in operations research have shown very competitive results. The heterogeneous multi-UAVs mission planning is an NP-hard combinatorial optimization problem with a large search space, challenging to be solved effectively by conventional methods. The variable neighborhood search algorithm optimizes the solution through neighborhood structure, which has high flexibility and is suitable for solving complex problems.

Our aim through this research is to explore an efficient mission planning framework for the heterogeneous multi-UAV systems to effectively deal with the integrated sensor allocation and path planning problem. Furthermore, we expect to develop an effective heuristic algorithm to improve the performance of the mission planning framework for large-sized problems. The main contributions and innovations of this paper are as follows:

(1) We build an integrated sensor allocation and path planning model, which is associated with three subproblems: Sensor allocation, task assignment, and collision-free path generation, simultaneously. Meanwhile, the model has a certain practical significance for cooperative reconnaissance, sensor information collection, and continuous monitoring missions in the real world, etc.

(2) Depending on the natural complexity of the problem, an integrated mission planning framework based on a novel two-level adaptive variable neighborhood search algorithm is proposed. The first-level is devoted to planning a reasonable airborne sensor allocation plan, and the second-level aims to optimize the collision-free path of the heterogeneous multi-UAVs system.

(3) The size of the integrated sensor allocation and path planning problem is determined by many factors: The number of heterogeneous UAVs, tasks, and sensors, etc. Due to there are many factors affecting the problem’s size, the size of the problem is usually
large. In order to improve the optimization efficiency of the mission planning framework for large-sized problems. An adaptive mechanism is proposed to guide the search direction intelligently. Detailed simulations are given to demonstrate the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. Problem formulation is given in Section 2. A brief introduction of the conventional Variable Neighborhood Search Algorithm is presented in Section 3. The integrated mission planning framework adopted in this paper is proposed in Section 4. Our method is comparatively simulated in Section 5. The Limitation analysis is shown in Section 6. In the end, conclusions are given in Section 7.

2. Problem Formulation

In this section, the integrated sensor allocation and path planning problem for a heterogeneous multi-UAVs system is presented. The problem is considered for mission scenarios involving a group of tasks that need to be performed by a heterogeneous multi-UAVs system in an environment with obstacles. Meanwhile, a group of interchangeable sensors is available at the base. Due to the task’s profit is related to the sensors loaded by the UAV performing the task. Therefore, the optimal objective of the problem is to maximize the total profit of the tasks and minimize the total path length of the heterogeneous multi-UAVs system. The integrated airborne sensor allocation and path planning problem is formulated as a mixed integer linear programming model, which is based on the outstanding work of Warrier [34] and Mufalli et al. [18].

2.1. Parameter Definitions

Let $\mathcal{T} = \{0, \ldots, N\}$ be the set of task locations required to be visited by a heterogeneous multi-UAVs system and 0 represents the base location, and each task $i \in \mathcal{T}$ must be performed only once. Let $\mathcal{U} = \{1, \ldots, M\}$ be the M-sized set of heterogeneous UAVs involving in the mission scenario, the heterogeneity of the UAVs is reflected in the upper load limit and kinematics characteristics. Let $\mathcal{T}_h$ be the maximum travel distance of UAV $h \in \mathcal{U}$. Note that the sensor loaded by the UAV $h$ will reduce the maximum travel distance. The reduced travel distance is proportional to the weight of the sensor. Let $\varsigma_h$ represents the maximum loading weight of UAV $h$. Meanwhile, each UAV may be loaded with multiple sensors. Let $\tau_h$ be the maximum number of sensors that can be loaded by the UAV $h$. Let $\mathcal{S} = \{1, \ldots, Z\}$ be the Z-sized set of available sensors at the base and $Q_s$ be the quantity of sensor $s \in \mathcal{S}$ available at base. As mentioned above, the weight of the sensor will reduce the maximum travel distance of the UAVs. Let $w_s$ be the weight of sensor $s$ and $C_s$ denotes the reduced travel distance when sensor $s$ is loaded. Note that the task profit is no-static, which is associated with the sensors loaded by the UAV performing the task. Let $R_{js}$ be the task profit when the UAV is loaded with sensor $s$ to perform the task $j$.

In this problem, the UAVs are required to fly over the tasks’ locations to complete them. For the sake of simplicity, we assume that (1) the UAVs flight can be maintained at a given altitude; (2) The interference of wind is ignored; (3) The flying altitude of UAVs is different, so the problem of collision between UAVs is ignored; (4) The airspeed of each UAV is constant.

The kinematics model of UAVs is expressed by the following formula:

\begin{align*}
\dot{x} &= v_u \cos \psi \\
\dot{y} &= v_u \sin \psi \\
\dot{\psi} &= c / \rho_{\text{min}}
\end{align*}

where $x$ and $y$ are the Cartesian coordinates of UAVs. $v_u$ is the cruise velocity. $\psi$ is the heading angle. $c$ is the steering command, $-1 \leq c \leq 1$. $\rho_{\text{min}}$ is the minimum turning radius. Let $X_{\text{obs}}$ represents the obstacle region in the mission scenario. Suppose that $\Sigma_h = \{\sigma_1, \ldots, \sigma_k\}$ represents the permutation of the locations labels, that need to be visited by the UAV $h$, and $\pi^h_\Sigma$ be the collision-free path to visit the tasks’ locations associated with
\( \Sigma_h \). Let \( L(.) \) is the collision-free path length calculation function, the collision-free path length between \( i \in T \) and \( j \in T \) can be represented by \( L(i,j) \).

2.2. Combinatorial Optimization Problem

The integrated sensor allocation and path planning model is expressed by the following optimization problem:

Maximize \[ M \sum_{h=1}^{N} \sum_{s=1}^{Z} R_{js} z_{jsh} - \lambda \sum_{h=1}^{M} \sum_{i,j \neq i}^{N} L(i,j) y_{ijh} \] (4)

Subject to :

\[ \sum_{s=1}^{Z} f_{hs} \leq \tau_h \quad \forall h \in U \] (5)

\[ \sum_{h=1}^{M} f_{hs} \leq Q_s \quad \forall s \in S \] (6)

\[ \sum_{j=0, j \neq i}^{N} y_{ijh} \leq 1 \quad \forall i \in T, h \in U \] (7)

\[ \sum_{j=0, j \neq i}^{N} y_{ijh} - \sum_{j=0, j \neq i}^{N} y_{jih} = 0 \quad \forall h \in U \] (8)

\[ \sum_{s=1}^{S} f_{hs} w_s \leq \varsigma_h \quad \forall h \in U \] (9)

\[ \sum_{j=0, j \neq i}^{N} \sum_{i=0}^{N} L(i,j)y_{ijh} \leq T_h - \sum_{s=1}^{S} C_s f_{hs} \quad \forall h \in U \] (10)

\[ \sum_{i=0}^{N} y_{ih} = 1 \quad \forall h \in U \] (11)

\[ \sum_{i=0}^{N} y_{0ih} = 1 \quad \forall h \in U \] (12)

\[ \pi_h \cap X_{obs} = \emptyset \quad \forall h \in U \] (13)

\[ \rho_h \geq \rho_{min} \quad \forall h \in U \] (14)

\[ f_{hs}, y_{ijh}, z_{jsh} \in \{0,1\} \] (15)

where \( z_{jsh}, y_{ijh}, \) and \( f_{hs} \) are binary decision variables. \( z_{jsh} \) equals one if UAV \( h \) is equipped with sensor \( s \) to perform task \( j \), and \( y_{ijh} \) equals one if UAV \( h \) travels from the task location of task \( i \) to the task location of task \( j \). \( f_{hs} \) equals one if UAV \( h \) is equipped with sensor \( s \). Note that \( \lambda \) is a weight factor, which is used to balance the optimization objective. The first term of the objective function (1) aims to maximize the total task profit, while the second term attempts to minimize travel costs for the heterogeneous multi-UAVs system.

Constraint (5) ensures that the number of sensors equipped with UAV does not exceed the maximum number of sensors that UAV can be equipped with. Constraint (6) guarantees that the total number of any sensors equipped by the UAV does not exceed the number of sensors available at the base. Constraint (7) and (8) ensure that each task is only performed once. Constraint (9) makes sure that the weight of the sensor loaded on the UAV does not exceed the maximum load of the UAV.

Constraint (10) guarantees that the travel distance of the UAV does not exceed its maximum travel distance. Constraint (11) and (12) ensure that all UAVs begin and end their path at the base. Constraint (13) makes sure that the path of the UAVs will not collide with obstacles. Constraint (14) is the kinematic constraint of UAV, which ensures that the
turning radius of UAV is not less than its minimum turning radius. Constraint (15) is the binary constraint.

3. Brief Introduction of the Conventional Variable Neighborhood Search Algorithm

As an efficient meta-heuristic algorithm, VNS has been successfully applied in many fields. Unlike the most meta-heuristic algorithm, the VNS uses the neighborhood structure to search the solution space alternately. The concept diagram of the conventional VNS is shown in Figure 1. The first shaking neighborhood structure will first change the current solution, and then the newly generated solution will be systematically searched by the local search heuristic. If the current solution is not improved, the next shaking neighborhood structure will change it. If the current solution is improved, it will turn to the first shaking neighborhood structure.

Figure 1. Conventional VNS concept diagram.

The VNS is mainly composed of two essential procedures: (1) Shaking procedure, (2) local search. The shaking procedure is used to change the current optimal solution by shaking neighborhood structures to prevent the solution from falling into the local optimum. The local search algorithm will change the solution systematically by several local search neighborhood structures. The procedure of conventional VNS is shown in Algorithm 1.

Algorithm 1. The Procedure of the Conventional VNS

1: Let $s$ be the initial solution, $N_k$ be the shaking neighborhood structures, $k = 1, \ldots, k_{\text{max}}$ and $N_l$ be the local search neighborhood structures, $l = 1, \ldots, l_{\text{max}}$. Set $k = 1$.
2: Repeat
3: \hspace{1em} [Shaking procedure]
4: \hspace{2em} $s' = \text{Shaking}(s, N_k)$ \hspace{1em} / Shaking procedure
5: \hspace{2em} [Local Search]
6: for $l = 1$ to $l_{\text{max}}$ do \hspace{1em} / Local search
7: \hspace{3em} $s'' = \text{LocalSearch}(s', N_l)$ \hspace{1em} //perform local search on $s'$ with neighborhood structure $N_l$
8: \hspace{3em} if $f(s'') > f(s')$ \hspace{1em} //checking if $s'$ is improved
9: \hspace{3em} $s' = s''$
10: \hspace{2em} $l = 1$ //turn to the first neighborhood structure for local search
11: else
12: \hspace{2em} $l = l + 1$ //turn to the next neighborhood structure for local search
13: end for
14: [Accept decision]
15: if $f(s') > f(s)$ \hspace{1em} //checking if $s$ is improved
16: \hspace{2em} $s = s'$
17: $k = 1$ //turn to the first neighborhood structure for shaking
18: else
19: \hspace{2em} $k = k + 1$ //turn to the next neighborhood structure for shaking
20: end if
21: Until the upper limit $T$ of time or the maximum number of iterations $Z_{\text{max}}$ is reached
4. Integrated Mission Planning Framework Base on the Two-Level Adaptive Variable Neighborhood Search Algorithm

4.1. Overview of the Integrated Mission Planning Framework

Depending on the problem’s complexity, we propose an integrated mission planning framework base on the two-level adaptive variable neighborhood search algorithm. The first-level focuses on allocating sensors for the heterogeneous multi-UAVs system to maximize the total task profit. The second-level aims to generate the sequence of tasks assigned to each vehicle within the heterogeneous multi-UAV system to minimize travel costs. Collision-free path definition between a couple of consecutive tasks as detailed in Section 4.5. The structure of the mission planning framework based on the adaptive two-level variable neighborhood search algorithm is shown in Algorithm 2 in pseudo-code.

Algorithm 2. The Structure of the Mission Planning Framework Based on the Adaptive Two-Level Variable Neighborhood Search Algorithm

1: Set the shaking neighborhoods structures of the first-level $N^1_k$ with $k = 1, \ldots, k_{\text{max}}$ and the shaking neighborhoods structures of the second-level $N^2_d$ with $d = 1, \ldots, d_{\text{max}}$. Set the local search neighborhood structures of the first-level $N^1_l$ with $l = 1, \ldots, l_{\text{max}}$, the local search neighborhood structures of the second-level $N^2_w$ with $w = 1, \ldots, w_{\text{max}}$. Let $i = 0$, $l = 1$ and $d = 1$.

2: \textit{[Initialization phase]}
3: \hspace{1em} $s = \text{GenInitialSolution}()$ \hspace{0.5em} // generate an initial solution

4: \textbf{Repeat}
5: \hspace{1em} \{First-Level Adaptive Shaking: Sensor allocation\} \hspace{0.5em} //first-level
6: \hspace{2em} $k = \text{AdaptiveShaking}(N^1_k)$ \hspace{0.5em} //Adaptive select the shaking neighborhood structure, the UAVs, and sensors for shaking
7: \hspace{2em} $s_1 = \text{FirstLevelShaking}(s, N^1_k)$ \hspace{0.5em} //generate the shaking solution according to shaking neighborhoods structure $N^1_k$.
8: \hspace{1em} \{First-Level Local Search Phase: Sensor allocation\}
9: \hspace{2em} \textbf{while} $l < l_{\text{max}}$
10: \hspace{3em} $s_2 = \text{FirstLevelLocalSearch}(s_1, N^1_l)$ \hspace{0.5em} //perform local search on $s_1$ according to local search neighborhoods structure $N^1_l$.
11: \hspace{2em} \textbf{end while}
12: \{Second-Level Shaking Phase: Route Planning\}
13: \hspace{2em} $s_3 = \text{SecondLevelShaking}(s_2, N^2_d)$ \hspace{0.5em} //generate the shaking solution of the second-level base on the shaking neighborhoods structure $N^2_d$.
14: \hspace{1em} \{Second-Level Local Search Phase: Route Planning\}
15: \hspace{2em} \textbf{while} $w < w_{\text{max}}$
16: \hspace{3em} $s_4 = \text{SecondLevelLocalSearch}(s_3, N^2_w)$ \hspace{0.5em} //perform local search on $s_3$ according to local search neighborhoods structure $N^2_w$.
17: \hspace{2em} \textbf{if} $f(s_4) > f(s_2)$
18: \hspace{3em} $s_2 = s_4$
19: \hspace{3em} $d = 1$
20: \hspace{2em} \textbf{else}
21: \hspace{3em} $d = d + 1$
22: \hspace{2em} \textbf{end if}
23: \hspace{2em} \textbf{end while}
24: \hspace{1em} \{Acceptance decision\}
25: \hspace{2em} \textbf{if} $f(s_2) > f(s)$ or $s_2$ is accepted
26: \hspace{3em} $s = s_2$
27: \hspace{3em} $l = 1$
28: \hspace{2em} \textbf{else}
29: \hspace{3em} \textbf{end if}
30: \hspace{2em} \textbf{end if}
31: \hspace{2em} \textbf{end while}
32: \hspace{1em} \textbf{end while}
33: \textbf{Until} the upper limit $T$ of time or a given number $\text{Iter}$ of iterations without improvement is reached

The adaptive two-level variable neighborhood search algorithm starts with an initial solution $s$. The initial solution is generated by the $\text{GenInitialSolution}()$ function, which is detailed in Section 4.2. Subsequently, the main procedure is repeated until a given stopping criterion is met. In the main procedure, the shaking neighborhood struc-
ture, the UAVs, and the sensors involving in the shaking procedure of the first-level is adaptively selected based on the $AdaptiveShaking(N_s^1)$ procedure to guide the search direction. Then, given the current solution $s$, the $FirstLevelShaking$ procedure is applied, which generates a new solution $s_1$ in the $k$-th shaking neighborhood structure for the first-level $N_s^1$. The $AdaptiveShaking(N_s^1)$ and $FirstLevelShaking$ procedures are detailed in Sections 4.3.1 and 4.3.2. In the next step, the $FirstLevelLocalSearch(s_1, N_s^1)$ procedure is performed and the solution $s_2$ is generated, which is detailed in Section 4.3.3. The second-level variable neighborhood search procedure is embedded in the local search procedure of the first-level. The main components of the second-level variable neighborhood search also consist of a shaking procedure and a local search procedure, which are detailed in Section 4.4. Notably, the second-level attempts to optimize the target location visited sequence of each UAV and redistributed the target location responsible for the routes. In the process of adjusting the visit order of target locations, it is a critical procedure to calculate the collision-free trajectory length of UAVs, which is detailed in Section 4.5. The local search step of the first-level and the second-level variable neighborhood search algorithm jointly determine the local optimum solution $s_2$. If the solution $s_2$ improves the solution $s$ or the solution $s_2$ is accepted by a specify acceptance mechanism, then $s_2$ replaces $s$, the acceptance mechanism is detailed in Section 4.6.

4.2. Initial Solution

The initial solution for the adaptive two-level variable neighborhood search algorithm is conducted by a two-phase greedy heuristic. The first phase aims to generate the path for each UAV by the savings algorithm. The saving algorithm was proposed by Clarke and Wright, which is a classical algorithm in the field of logistics and transportation. The saving algorithm is mainly used to optimize the team’s driving distance in multivehicle transportation. Its principle is to continuously merge paths to shorten the total driving distance. For more details about the saving algorithm, please refer to the literature [35]. The second phase randomly assign sensors to the heterogeneous multi-UAV system.

4.3. Adaptive Shaking and Local Search of the First-Level

In the shaking procedure of the first-level, the heterogeneous UAVs and the sensors involving in the shaking procedure are determined by several adaptive rules. Then, the shaking solution of the first-level is generated based on the predefined shaking neighborhood structures. The adaptive mechanism adopted in the first-level shaking procedure will guide the following search direction to improve the algorithm’s convergence efficiency.

4.3.1. Shaking Neighborhood Structures of the First-Level

The neighborhood structures are directly related to the search range of the algorithm, which will greatly affect the algorithm’s convergence and the optimization quality of the solution. In the first-level shaking procedure, the sensors loaded by the heterogeneous multi-UAVs system will be optimized to maximize the total task profit. Let $S_K = \{s_1, \ldots, s_M\}$ be the sensors set loaded by the heterogeneous multi-UAVs system in the current solution. Let $s_k$ represents the sensors set loaded by UAV $k \in K$, $s_k \in S_K$. The total number of sensors loaded by the UAV $k$ is $|s_k|$ and the $i$-th sensor loaded by the UAV $k$ is denoted by $e_i^k$. Thus, the sensors set loaded by the UAV $k$ can be expressed by $s_k = \{e_1^k, e_2^k, \ldots, e_{|s_k|}^k\}$. The remaining sensors in the base are uniformly denoted as a set $Z = \{z_1, \ldots, z_N\}$. The shaking neighborhood structures are shown in Table 1. The shaking neighborhood operators adopted in the first-level is as follows:

1. Sensor cyclic exchange operator: The sensor cyclic exchange operator will exchange the sensors currently loaded by the heterogeneous multi-UAVs system cyclically. An example of a sensor cyclic exchange operator is shown in Figure 2. Two critical parameters are involved in the sensor cyclic exchange operator: The number of heterogeneous UAVs involved in the shaking procedure $NU$ and the maximum number of sensors to be exchanged $NS$. 
Thus, we develop several adaptive selection methods to determine the heterogeneous UAVs. The shaking neighborhood structures.

### Table 1. The shaking neighborhood structures.

| ID | Operator                           | NU (N^Z_k) | NS (N^Z_k) | ID | Operator                           | NU (N^Z_k) | NS (N^Z_k) |
|----|------------------------------------|------------|------------|----|------------------------------------|------------|------------|
| 1  | Unbalanced exchange operator       | 1          | 1          | 8  | Sensor cyclic exchange             | 3          | 1          |
| 2  | Unbalanced exchange operator       | 1          | 2          | 9  | Sensor cyclic exchange             | 3          | 2          |
| 3  | Unbalanced exchange operator       | 2          | 1          | 10 | Sensor cyclic exchange             | 3          | 3          |
| 5  | Sensor cyclic exchange             | 2          | 1          | 11 | Sensor cyclic exchange             | 4          | 1          |
| 6  | Sensor cyclic exchange             | 2          | 2          | 12 | Sensor cyclic exchange             | 4          | 2          |
| 7  | Sensor cyclic exchange             | 2          | 3          | 13 | Sensor cyclic exchange             | 4          | 3          |

For each heterogeneous UAV $k$ involved in the shaking procedure, a subset of sensors is selected from $s_k \in S_k$. Then, the sensors will be inserted to another heterogeneous UAV $k + 1$ involving in the shaking procedure. Note that, the constraints (5), (6), (9), and (10) cannot be violated in this neighborhood structure.

2. **Unbalanced exchange operator:** For the heterogeneous UAV $k \in K$, the unbalanced delete & insert operator will delete up to $N^Z_k$ sensors from the sensor set $s_k \in S_k$. Then, up to $N^Z_k$ sensors will be selected from the sensor set $Z$ and inserted in the sensor set $s_k$. Notice that the number of sensors deleted from the sensor set $s_k$ maybe not equal to the number of sensors inserted. Similarly, the constraints (5), (6), (9) and (10) cannot be violated in this neighborhood structure. An example of an unbalanced exchange operator is shown in Figure 3.

### Figure 3. An example of an unbalanced exchange operator.

**4.3.2. Adaptive Mechanism**

Selecting suitable heterogeneous UAVs and sensors to participate in the shaking procedure can improve the algorithm’s efficiency in searching for high-quality solutions. Thus, we develop several adaptive selection methods to determine the heterogeneous UAVs...
and sensors involving the first-level shaking procedure. The selection method includes the
selection method of heterogeneous UAV and the selection method of sensors.

Heterogeneous UAV selection method. The heterogeneous UAV selection method
selects $NU$ heterogeneous UAVs that be involved in the sensor cyclic exchange operator.
The selection method is performed in two steps: (1) Select the first heterogeneous UAV to
participate in the shaking procedure through the following three selection mechanisms. (2)
Randomly select the remaining $NU-1$ heterogeneous UAVs. The three methods are used to
select the first heterogeneous UAV is expressed as follow:

1. Random: The heterogeneous UAVs in the current solution have the same probability
to be selected.
2. Largest remaining load: The selection probability of a heterogeneous UAV is propor-
tional to its remaining load. The goal is to optimize the sensor configuration for the
UAV with the largest remaining load.
3. Maximum number of visit targets: The probability of the heterogeneous UAV being
selected is proportional to the number of visit targets. The goal is to optimize the
sensor configuration of the heterogeneous UAV with the maximum number of visit
targets.

Sensor selection method. After the first heterogeneous UAV to be involved in the
sensor cyclic exchange operator is selected. The sensors of the heterogeneous UAV to be
involved in the sensor cyclic exchange operator will be determined. The three methods are
used to select the sensors loaded by the selected heterogeneous UAV is expressed as follow:

1. Random: All the sensors involved loaded by the selected heterogeneous UAV have
the same probability of being selected.
2. Total task profit: For the selected UAV, the selection probability of the sensor is
inversely proportional to the total task profit. The goal is to assign sensors with higher
total task profit to the selected UAV.
3. Travel distance reduction: For the selected UAV, the selection probability of the sensor
is proportional to the travel distance reduction. The goal is to remove the sensor
with higher travel distance reduction to increase the maximum travel distance of the
selected UAV.

The selection method will significantly affect the efficiency of the search process.
In the first-level shaking procedure, we design an adaptive mechanism to realize the
selection method’s intelligent decision. The roulette wheel selection procedure is adopted
to determine the heterogeneous UAV and sensor selection methods. In the algorithm’s
initial stage, each selection method’s selected probability is set to the same value. With
the progress of the search process, the selected probability of each selection method is
gradually updated. Specifically, each method’s selected probability is uniformly updated
after $\gamma$ iterations of the two-level mission planning framework based on a scoring system.
The scoring system has the following statistical rules: (1) A score of nine will be added to a
selection method if the overall optimal solution is obtained after applying the method. (2)
A score of three will be added to a selection method if the current solution is improved. (3)
A score of one will be added to a selection method if the new solution is worse than the
current solution, but the solution is accepted base on the acceptance criterion. Let $\eta$ be the
total number of the selection method and $\delta_i$ be the current score of the method $i$, $1 \leq i \leq \eta$.
Notice that each selection methods $i = 1, \ldots, \eta$ is associated with a weight coefficient $\omega_i$
and the probability of the method $i$ being selected is $\omega_i / \sum_{j=1}^{\eta} \omega_j$. Suppose that as of the last
score update, the method $i$ has been selected $a_i$ times, then the new score of the method $i$ is
updated by $\omega_i(1 - \rho) + \rho(\delta_i / a_i)$, where $\rho \in [0, 1]$ is a system parameter for controlling the
trend of score update.
4.3.3. Local Search Neighborhood Structures of the First-Level

The solution generated by the shaking procedure is then improved by the local search step. The local search is implemented by a set of neighborhood structures. Five neighborhood structures involving in the local search step is defined as follow:

1. Insertion operator: The operator will randomly select a sensor from the sensor set $Z$ and insert it into the sensor set $s_k$ loaded on a UAV $k \in K$.
2. Exchange operator: The operator will randomly exchange a sensor $e_i^k \in s_k$ with a sensor $e_j^l \in s_l$, where $1 \leq i \leq |s_k|$, $1 \leq j \leq |s_l|$, $s_k, s_l \in S$ and $k \neq l$.
3. 2-exchange operator: Similar to the exchange operator, the 2-exchange will randomly exchange two sensors in $s_k$ with two sensors in $s_l$, where $s_k, s_l \in S$ and $k \neq l$.
4. 1-deletion-1-insertion operator: The operator will delete a sensor from $s_k$ and insert another sensor from $Z$ to $s_k$.
5. 2-deletion-2-insertion operator: The operator will delete two sensors from $s_k$ and add them to $Z$. Then, two sensors from $Z$ is inserted to $s_k$.

4.4. Shaking and Local Search of the Second-Level

The second-level of the adaptive variable neighborhood search aims to minimize travel costs for the heterogeneous multi-UAVs system. As shown in Algorithm 2, the second-level variable neighborhood search is embedded in the first-level’s local search step to improve the solution generated by the first-level. The shaking and local search neighborhood structures are given in this section.

Shaking procedure. Similar to the first-level perturbation procedure, the second-level perturbation procedure is to prevent the solution from falling into the local optimum. The shaking neighborhood structures are designed as follow:

1. Intraroute 3-opt: The interroute 3-opt operator is an extension of the 2-opt operator, which is used to change the visit sequence of the nodes in a route. As shown in Figure 4, eight forms of the intraroute 3-opt operator are given, and one of them will be randomly selected.
2. Interroute 2-exchange: The interroute 2-exchange operator randomly exchanges two pairs of nodes in two different routes.

![Figure 4. Eight forms of the intraroute 3-opt operator.](image)

The solution generated by the second-level’s shaking produce is then input to the second-level’s local search. The local search aims to find an improved solution by changing the visit sequence of nodes in the same route and exchanging the visit nodes between different routes. Four local search neighborhoods structures are given as follow:

1. Intraroute 2-opt: The intraroute 2-opt operator will reverse the nodes visit sequence of a sub-route.
2. Intraroute or-opt: The intraroute or-opt operator relocates several consecutive nodes in a route. An example of intraroute 2-opt and intraroute or-opt are shown in Figure 5.
3. Interroute exchange: The interroute exchange operator randomly exchanges two nodes in the same route.
4. Interroute relocated: The interroute relocated randomly delete a node from a route and insert it to another route. An example of interroute exchange and interroute relocated is shown in Figure 6.

![Figure 6](image_url)

**Figure 6.** An example of interroute exchange and interroute relocated.

4.5. Collision-Free Path Planning

The optimization objective is closely related to the flight path length of the heterogeneous UAV system. To calculate the path length of a heterogeneous multi-UAV system, we must first obtain the collision-free path of each UAV in space. Note that the collision-free path needs to satisfy both kinematic and collision-free constraints of heterogeneous UAVs. However, a large number of solutions are generated in the two-level mission planning framework. In order to improve the optimization efficiency, the constraints that the UAV needs to satisfy are relaxed during the iteration process. That is, in the iterative process, the collision-free route length of the UAV to visit each node is evaluated, while the kinematic constraints of the heterogeneous UAV system. To calculate the path length of a heterogeneous multi-UAV system, we must first obtain the collision-free path of each UAV in space. Note that the collision-free path planning process is separated into multiple phases, which are presented as follows:

1. Map preprocessing. In this stage, the continuous space is discretized based on the target location and the fixed point of the obstacle. Note that this procedure is only evaluated once when the objective function value is calculated for the first time. The visibility map is adopted in this stage to discretize the space efficiently.

2. Route assessment. The route assessment stage is used to calculate the objective value of the solutions in the iterative process. In this stage, the A* algorithm is adopted to generate the shortest polyline path connecting the UAV’s initial location and target location sequence.
3. Path smoothing. As a postprocessing stage, this stage is used to optimize the path of the final solution. The polyline path of the UAV mentioned above is smoothed to generate a collision-free trajectory satisfying the kinematic constraints of the UAV.

In order to reduce the calculation cost, the collision-free path between any two locations is precalculated and stored in the form of distance matrix $D_M = [d_0^0, d_1^0, \ldots, d_N^0; d_0^1, d_1^1, \ldots, d_N^1; \ldots; d_0^N, d_1^N, \ldots, d_N^N]_{(N+1) \times (N+1)}$. The second-level adaptive variable neighborhood search algorithm obtains the collision-free path length between any two task locations by querying the distance matrix.

4.6. Acceptance Decision

Acceptance Decision. The acceptance decision process is executed after each round of the second-level local search procedure. The new solution generated by the second-level local search procedure $s_2$ will be compared with the current solution $s$. Once the new solution $s_2$ is accepted, the solution $s_2$ will replace the solution $s$ become the current solution. In order to improve the diversity of solutions, the acceptance decision process incorporates the solution acceptance mechanism of the simulated annealing algorithm. Specifically, the improved solution will always be accepted by the algorithm, that is $f(s_2) > f(s)$. Notice that non-improving solution will also be accepted with a certain probability $e^{-(f(s)-f(s_2))/\theta}$, where $\theta$ is the temperature parameter and $f(.)$ is the objective value of the objective function. The initial value of $\theta$ is set to $\theta_0$, which decreases with each iteration of the algorithm with a coefficient $\theta^{-}$. If the solution cannot be improved after $v$ iterations, then $\theta$ will be reset to zero.

5. Simulation and Discussion

In this section, the proposed integrated mission planning framework for heterogeneous multi-UAV systems was implemented. To verify the feasibility of the mission planning framework, we designed a simple mission scenario and optimized it through the mission planning framework. Meanwhile, we analyzed the solution generated by the mission planning framework to ensure that the constraints were not violated. To test the performance of the mission planning framework, the algorithm is compared with other algorithms. The simulation hardware environment is Intel Core (TM) i7-6500u CPU @ 2.50 GHz notebook computer, and the programming environment is visual studio 2015.

5.1. Feasibility Verification
5.1.1. Test Scenario and Parameter Settings

This section gives a specific test scenario to verify the feasibility of the mission planning framework’s solution. The scenario involves four heterogeneous UAVs and three types of sensors. The attributes of the sensors and UAVs are shown in Tables 2 and 3, respectively. In this scenario, four obstacles are set up to test whether the heterogeneous Multi-UAVs system violates the collision-free and kinematic constraints. In this scenario, 31 tasks are given, and the sensors—task benefit matrix is shown in Table 4. The maximum number of iterations for the two-level adaptive variable neighborhood search algorithm was set to 30,000 and terminated when no improvement was found in 1000 iterations. The parameter $\theta_0$ is related to the convergence effect of the algorithm. A larger value can expand the search space of the solution, but cause a slower convergence rate. Further, the coefficient $\theta^{-}$ and the number of no improving iterations $v$ also directly impact the convergence effect. Through numerical experiments, we set $\theta_0 = 18.7$, $\theta^{-} = 0.05%$, and $v = 180$. Further, the weight factor $\lambda$ is set to 0.1. The mission area is set to a square area of $2000 \times 2000$ m.$^2$. 
Table 2. Sensors attributes.

| Sensor | Quantity | Weight | Travel Distance Reduction (m) |
|--------|----------|--------|-----------------------------|
| S1     | 3        | 125    | 2500                        |
| S2     | 6        | 100    | 2000                        |
| S3     | 4        | 75     | 1500                        |

Table 3. UAVs attributes.

| UAV    | Load Limit | Travel Distance (m) | Cruising Speed (m/s) | Minimum Turning Radius (m) | Sensor Capacity |
|--------|------------|---------------------|----------------------|-----------------------------|----------------|
| U1     | 200        | 11,000              | 20                   | 45                          | 2              |
| U2     | 240        | 12,000              | 20                   | 45                          | 2              |
| U3     | 175        | 10,000              | 20                   | 45                          | 2              |
| U4     | 220        | 13,000              | 20                   | 45                          | 2              |

Table 4. Sensor-Task Benefit Matrix.

|        | T1-T3 | T4-T6 | T7-T9 | T10-T12 | T13-T15 | T16-T18 | T19-T21 | T22-T25 | T26-T28 | T29-T31 |
|--------|-------|-------|-------|---------|---------|---------|---------|---------|---------|---------|
| S1     | 72    | 24    | 46    | 35      | 12      | 83      | 42      | 52      | 23      | 42      |
| S2     | 98    | 92    | 88    | 62      | 45      | 67      | 57      | 84      | 46      | 54      |
| S3     | 44    | 75    | 37    | 27      | 29      | 23      | 44      | 45      | 57      | 36      |

5.1.2. Feasibility Analysis

The result of sensor allocation is shown in Table 5. From the calculation results, it can be found that all constraints have not been violated. The task execution paths of the heterogeneous multi-UAVs system are shown in Figure 7. Notice that the collision-free constraints of the UAVs are respected. Meanwhile, the paths of the heterogeneous multi-UAVs system satisfy the constraints of kinematics. Therefore, the feasibility of the solution generated by the mission planning framework is verified.

Table 5. Result of sensor allocation.

| Sensors #1 | Sensors #2 | Load Limit | Sensors Weight | Maximum Travel Distance | Task Profit | Travel Distance |
|------------|------------|------------|----------------|-------------------------|-------------|----------------|
| UAV1       | S2         | S3         | 200            | 175                     | 7500        | 764            | 39209.98   |
| UAV2       | S1         | S2         | 240            | 225                     | 7500        | 1129           | 4425.97    |
| UAV3       | S2         | S3         | 175            | 175                     | 6500        | 944            | 39556.3    |
| UAV4       | S2         | S2         | 220            | 200                     | 9000        | 1054           | 2882.15    |

Figure 7. The task execution paths of the heterogeneous unmanned system.
5.2. Numerical Comparisons

5.2.1. Experimental Conditions

To verify the integrated mission planning framework’s performance based on the two-level adaptive variable neighborhood search algorithm. We developed nine mission scenarios as evaluation benchmarks. The summary of the test case is given in Table 6. The locations of the tasks were randomly distributed in the mission area. The cruise speed of heterogeneous UAVs is uniformly set to 20 m/s, and the minimum turning radius is randomly selected from 30 m to 45 m. Meanwhile, the probability that a task has benefited from each sensor is set evenly. The task benefit value is set based on the distribution, benefit range 0~50 with the probability 0.1, 50~100 with the probability 0.2, 100~150 with the probability 0.3, and 150~200 with the probability 0.4. The upper load limit of the first UAV in each test case is set to 150, and the subsequent UAVs are increased by 30. The first type of sensor in each test case is set to 50, and the subsequent type of UAVs are increased by 25.

Table 6. Summary of the test case.

| Case ID | Task Quantity | UAV Quantity | Sensor Type | Sensor Quantity | UAV Sensor Capacity | Mission Area m² |
|---------|---------------|--------------|-------------|-----------------|---------------------|-----------------|
| 1       | 20            | 3            | 6           | 24              | 3                   | 2000 × 2000     |
| 2       | 30            | 4            | 6           | 40              | 3                   | 2000 × 2000     |
| 3       | 40            | 5            | 6           | 60              | 4                   | 2000 × 2000     |
| 4       | 50            | 4            | 8           | 32              | 3                   | 4000 × 4000     |
| 5       | 60            | 5            | 8           | 50              | 3                   | 4000 × 4000     |
| 6       | 70            | 6            | 8           | 64              | 4                   | 4000 × 4000     |
| 7       | 80            | 6            | 12          | 48              | 4                   | 6000 × 6000     |
| 8       | 90            | 7            | 12          | 70              | 5                   | 6000 × 6000     |
| 9       | 100           | 8            | 12          | 96              | 6                   | 6000 × 6000     |

The case ID from 1~3 is considered the benchmark of small-sized problems, 4~6 is considered the benchmark of medium-sized problems, and 7~9 is considered the benchmark of large-sized problems. The value of maximum unloads travel distance for each UAV is set based on the number of UAVs in the mission scenario. For the small-sized problems, the maximum unload travel distance for each UAV is randomly selected from 10,000~15,000 m, 15,000~25,000 m for the medium-sized problems, and 25,000~35,000 m for the large-sized problems. The weight factor is set to 0.5 for the small-sized problems, 0.2 for the medium-sized problems, and 0.1 for the large-sized problems.

5.2.2. Monte Carlo Study

To test the mission planning framework’s performance in the small-sized problems, medium-sized problems, and large-sized problems. We have developed another two variable neighborhood search algorithms for comparative analysis. They are the two-level variable neighborhood search algorithm without the adaptive mechanism (TLVNS) and the two-level variable neighborhood search algorithm with a random selection strategy (TLVNS_RSS).

For the TLVNS, compared to TLAVNS, its adaptive mechanism is removed, and the shaking neighborhood structures: ID 1, 5, 6, 9, 12 in Table 1 is performed sequentially, as did in conventional VNS. For the TLVNS_RSS, the shaking neighborhood structures are randomly selected from ID 1 ~ 13 in Table 1. The number of the neighborhood is set to 5. In order to compare the algorithms effectively, Monte Carlo simulation is introduced. The TLVNS, TLVNS_RSS, and TLAVNS algorithms are run 2000 times, respectively, for each test case in Table 6.

The distribution of solutions for the small-sized problem is shown in Figure 8a–c, corresponding to the case-1, case-2, and case 3 in Table 6. In the Monte Carlo simulations for the small-sized problems, the proposed TLAVNS can obtain better results than the
TLVNS_RSS and TLVNS. However, the distribution of the results is similar, especially between TLAVNS and TLVNS. The distribution of solutions for the medium-sized problem is shown in Figure 8d–f, corresponding to the case-4, case-5, and case 6 in Table 6. In the Monte Carlo simulations for the medium-sized problems, compared with small problems, the TLAVNS algorithm has better performance in medium problems than the other two algorithms. Meanwhile, the solution space of the TLVNS_RSS and TLVNS becomes smaller because they cannot find the optimization direction accurately. The distribution of solutions for the large-sized problem is shown in Figure 8g–i, corresponding to the case-7, case-8, and case 9 in Table 6. In the Monte Carlo simulations for the large-sized problems, the results show that our proposed TLAVNS can effectively deal with large-sized problems, and the adaptive mechanism can effectively guide the search direction. The average calculation time of the TLVNS_RSS algorithm in all test cases is about 132.2 s. The average calculation time of the TLVNS algorithm in all test cases is about 121.6 s. And the average calculation time of the TLAVNS algorithm in all test cases is about 169.3 s. From the perspective of calculation cost, the average calculation time of the TLAVNS algorithm is slightly higher than the other two algorithms. However, the increase in computation time is not obvious. Compared with the improvement of optimization performance, the increase in computation time can be tolerated. In general, our integrated mission planning framework based on the adaptive two-level variable neighborhood search algorithm can effectively deal with integrated sensor allocation and path planning problems. Furthermore, the adaptive mechanism can effectively guide the solution's search direction, which makes the mission planning framework have good performance in dealing with medium and large-scale problems.

Figure 8. (a–c) The distribution of solutions for the small sized problem; (d–f) The distribution of solutions for the medium sized problem; (g–i) The distribution of solutions for the large sized problem.

6. Limitation Analysis
The integrated sensor assignment and path planning model proposed in this paper may have some limitations in the real world applications. These limitations may lead to the failure of mission planning. The main application limitations are as follows:

1. Uncertain factors in the process of task execution may lead to the failure of the plan, such as wind interference, weather impact on the performance of the sensor, etc.

2. In order to simplify the complexity of the problem, we assume that UAVs fly at different altitudes to avoid collision between UAVs. However, this may not meet the requirements of some real-world tasks.

7. Conclusions
6. Limitation Analysis

The integrated sensor assignment and path planning model proposed in this paper may have some limitations in the real world applications. These limitations may lead to the failure of mission planning. The main application limitations are as follows:

(1) Uncertain factors in the process of task execution may lead to the failure of the plan, such as wind interference, weather impact on the performance of the sensor, etc.

(2) In order to simplify the complexity of the problem, we assume that UAVs fly at different altitudes to avoid collision between UAVs. However, this may not meet the requirements of some real-world tasks.

7. Conclusions

This paper focused on the integrated sensor allocation and path planning problem for a heterogeneous multi-UAVs system with interchangeable sensors and no-static mission profit. We presented an integrated mission planning framework based on a two-level adaptive variable neighborhood search algorithm to address the mission planning problem. The first-level is devoted to solving the sensor allocation problem. The second-level is embedded in the first-level framework, and the second-level is used to solve the path planning problem. The adaptive mechanism is introduced into the mission planning framework, which effectively improves the performance for large-sized problems. Simulation results showed our mission planning framework can generate good-quality solutions for large-sized mission planning problems. In order to deal with the above problems, we will focus on environmental uncertainty and collision avoidance in future research. Meanwhile, we will optimize the mission planning framework to improve its operational efficiency.

Author Contributions: Conceptualization, H.Z. and J.Y.; Methodology, H.Z. and J.Y.; Software, H.Z.; Data curation, H.Z.; Writing—review and editing, H.Z. and J.Y. All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

1. 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]

2. Zhu, A.; Yang, S.X. A neural network approach to dynamic task assignment of multirobots. *IEEE Trans. Neural Netw.* **2006**, *17*, 1278–1287.

3. 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]

4. Eun, Y.; Bang, H. Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithm. *J. Aircr.* **2009**, *46*, 338–343. [CrossRef]

5. Zhen, Z.; Xing, D.; Gao, C. Cooperative search-attack mission planning for multiUAV based on intelligent self-organized algorithm. *Aerosp. Sci. Technol.* **2018**, *76*, 402–411. [CrossRef]

6. Zhu, B.; Chen, L.; Zhou, Y.; Zheng, D.; Pan, H. An improved a* algorithm for the industrial robot path planning with high success rate and short length. *Rob. Autom. Syst.* **2018**, *106*, 26–37. [CrossRef]

7. Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegwart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Australia, 21–25 May 2018; pp. 5113–5120.

8. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 24–28 September 2017; pp. 31–36.

9. Yuan, C.R.; Liu, G.F.; Zhang, W.Q.; Pan, X.L. An efficient RRT cache method in dynamic environments for path planning. *Rob. Autom. Syst.* **2020**, *131*, 103594. [CrossRef]

10. Mufalli, F.; Batta, R.; Nagi, R. Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Aerosp. Sci. Technol.* **2013**, *110*, 106507. [CrossRef]

11. 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. Aeronom.* **2016**, *23*, 1238–1250. [CrossRef]

12. 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]

13. Mufalli, F.; Batta, R.; Nagi, R. Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Comput. Oper. Res.* **2012**, *39*, 2787–2799. [CrossRef]

14. Kim, S.; Lim, G.; Cho, J.; Côté, M. Drone-Aided Healthcare Services for Patients with Chronic Diseases in Rural Areas. *J. Intel1. Rob. Syst. Theor. Appl.* **2017**, *88*, 163–180. [CrossRef]

15. Lim, G.T.; Kim, S. Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates. *J. Intell. Rob. Syst. Theor. Appl.* **2020**, *97*, 471–487. [CrossRef]

16. Ramirez-Atencia, C.; Bello-Orgaz, G.; R-Moreno, M.D.; Camacho, D. Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms. *Soft Comput.* **2016**, *21*, 4883–4900. [CrossRef]

17. Ramirez-Atencia, C.; Ser, D.J.; Camacho, D. Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning. *Swarm Evolut. Comput.* **2019**, *44*, 480–495. [CrossRef]

18. Wu, W.; Wang, X.; Cui, N. Fast and coupled solution for cooperative mission planning of multiple heterogeneous unmanned aerial vehicles. *Aerosp. Sci. Technol.* **2018**, *79*, 131–144. [CrossRef]

19. Luo, R.; Zheng, H.; Guo, J. Solving the Multi-Functional Heterogeneous UAV Cooperative Mission Planning Problem Using Multi-Swarm Fruit Fly Optimization Algorithm. *Sensors* **2020**, *20*, 5026. [CrossRef] [PubMed]
28. Deng, Q.; Yu, J.; Mei, Y. Deadlock-free consecutive task assignment of multiple heterogeneous unmanned aerial vehicles. *J. Aircr.*, 2014, 51, 596–605. [CrossRef]
29. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* 1997, 24, 1097–1100. [CrossRef]
30. Wei, L.; Zhang, Z.; Lim, A. An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *IEEE Comput. Intell. Mag.* 2014, 9, 18–30. [CrossRef]
31. Kuo, Y.; Wang, C.C. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Syst. Appl.* 2012, 39, 6949–6954. [CrossRef]
32. Polacek, M.; Hartl, R.F.; Doerner, K.; Reimann, M. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *J. Heuristics* 2004, 10, 613–627. [CrossRef]
33. Hemmelmayr, V.C.; Doerner, K.F.; Hartl, R.F. A variable neighborhood search heuristic for periodic routing problems. *Eur. J. Oper. Res.* 2009, 195, 791–802. [CrossRef]
34. Warrier, A. A Decision Support Tool for the Inventory Allocation and Vehicle Routing Problem. Ph.D. Thesis, State University of New York at Buffalo, Buffalo, NY, USA, 2001.
35. Clarke, G.; Wright, J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 1964, 12, 568–581. [CrossRef]