

# Cooperative Path Planning of UAVs & UGVs for a Persistent Surveillance Task in Urban Environments

Yu Wu<sup>ID</sup>, Shaobo Wu, and Xinting Hu<sup>ID</sup>

**Abstract**—There have been many applications of drones in urban environments, such as delivery, rescue, and surveillance. In a persistent surveillance task, the drones sometimes cannot complete it independently when some regions are required to be covered on the ground. For this purpose, unmanned aerial vehicles and unmanned ground vehicles (UAVs & UGVs) system is introduced to perform such a task in this article, and the goal is to generate the circular paths for the drones and the UGVs, respectively, to minimize their travel time of realizing a complete coverage. First, the cooperative path planning problem of UAVs & UGVs is formulated into a large-scale 0-1 optimization problem, in which the on-off states of the discrete points are to be optimized. Second, a hybrid algorithm integrating the estimation of distribution algorithm (EDA) and the genetic algorithm (GA) algorithm is proposed to solve the problem. The advantages of EDA and GA in the global and local search are fully taken considering the demands in different phases of the iterative process. A simple sweep-based approach is employed to determine the optimal sequence of passing the open points. Then, an online local adjustment strategy is also applied to address the changes of the requirements on covering the ground area. Simulation results demonstrate that the UAVs & UGVs system can enhance the efficiency of the task. The hybrid EDA-GA algorithm can greatly improve the performance of EDA and GA in terms of the quality and the stability of solutions. The online adjustment strategy is effective to maintain a complete coverage while minimizing the impact on the circular paths.

**Index Terms**—Drones, cooperative path planning, hybrid EDA-GA algorithm, persistent surveillance, UGV, urban environments.

## I. INTRODUCTION

**S**MALL unmanned aerial vehicles (UAVs, usually called drones) have received a lot of attention in recent years and have been applied in people's daily life and public services. On the one hand, the advantages of easy operation and low

Manuscript received June 27, 2020; revised August 21, 2020 and September 23, 2020; accepted October 7, 2020. Date of publication October 13, 2020; date of current version March 5, 2021. This work was supported in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2020jcyj-msxmX0602; in part by the Fundamental Research Funds for the Central Universities under Project 2020 CDJ-LHZZ-066; and in part by the China Scholarship Council under Project 201906055030. (*Corresponding author: Yu Wu*)

Yu Wu is with the College of Aerospace Engineering, Chongqing University, Chongqing 400044, China, and also with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: cqwyuyu@cqu.edu.cn).

Shaobo Wu is with the Automation School, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: wushaobo@bupt.edu.cn).

Xinting Hu is with the School of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China (e-mail: 2018031027@cauc.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.3030240

price attract more people to use drones for work and entertainment. On the other hand, the policy of opening the low-altitude airspace encourages more applications of drones in urban environments [1]. Among the various applications, delivery, rescue, target tracking, and surveillance are the most popular ones, and the drone is required to complete the task by flying along the specified path [2]. Therefore, the path planning problem becomes the key issue to realize the autonomous flight of the drone [3]. The path planning problems in the above tasks can be classified into two groups according to their characteristics. In the first group, the drone is required to reach the specified target, such as in the delivery [4] or the target tracking task [5]. An optimal path from the start point to the destination is needed. The second type of tasks is less common, in which a complete coverage of a specified area is expected, and the goal is usually to minimize the flying distance, flying time, or fuel consumptions of drone [6], [7]. Aerial photography, data collection, surveillance, and patrol belong to this group of tasks.

In the existing studies, the surveillance task is usually performed by a single or multiple UAVs, and the unmanned ground vehicle (UGV) is rarely involved in such a task. In those studies, the shortest path of finishing a complete coverage of the specific area is usually focused on, but the corresponding circular path is often ignored [8]. Even if the circular path is considered in some studies, there is no or only a few obstacles distributed in the area [9], which is quite different from the situation in urban environments. Besides, the vision field and the shooting resolution of the drone are affected by its flight height, and the blocking of buildings should be considered when calculating the actual coverage area. As for the ground area to be covered, there are different requirements on the shooting resolution for subareas, and some subareas are even prohibited to be visited designedly [10]. However, the ground area is sometimes required to be covered with high accuracy or is hard to be covered (due to the blocking of buildings), such as in the task of precise observation or photographing on the ground objects. In those tasks, the shooting resolution of camera on UAVs can hardly meet the requirements, and the introduction of UGVs is a good way to solve the problem. Furthermore, UGVs can cooperate with UAVs to share the work, thus improving the efficiency of performing the task. In this article, the cooperative path planning problem of the unmanned aerial vehicles and unmanned ground vehicles (UAVs & UGVs) system in a persistent surveillance task is focused on, and the goal is to find two circular paths for UAVs and UGVs, respectively,

which can minimize their travel time of realizing a complete coverage of the ground area.

The main contributions of this work can be summarized as follows.

- 1) A UAVs & UGVs system is introduced to complete the persistent surveillance task in urban environments. In the established cooperative path planning model, various factors affecting the vision field and the shooting resolution of drone are considered to reflect the real requirements on the task. The problem is finally formulated into a large-scale 0-1 optimization problem, including the outer and inner loops.
- 2) A hybrid algorithm integrating the estimation of distribution algorithm (EDA) and genetic algorithm (GA) (named as the EDA-GA-based algorithm for convenience) is developed to solve the cooperative path planning problem. In the outer loop, the on-off states of the discrete points for the travels of drones and UGVs are optimized by the hybrid EDA-GA algorithm. In the inner loop, the sequences of passing those open points are determined by a simple sweep-based approach to generate the circular paths for the drones and the UGVs, respectively, which can realize a complete coverage.
- 3) An online adjustment strategy on the circular paths is designed considering the changes of demands on the coverage area. A local modification is imposed on the points of the circular path, which are influenced by the increase or decrease of the inaccessible subareas. The path points are added or deleted to maintain a complete coverage while having the minimal impact on the existing circular paths.

## II. RELATED WORKS

There have been many studies concentrating on the path planning problems of drones for different civil applications, such as the sowing, fertilization, and pollination in agriculture [11], the inspection around the high building [12], and the rescue in the fire fighting [13]. However, as the short flight endurance and the relatively low shooting resolution of drone may fail to satisfy the demand in many tasks, it is not always an effective way to deploy a team of drones to complete the complicated tasks. Therefore, UGVs are introduced to cooperate with UAVs, such as in exploration and mapping task [14], target localization [15], and environmental protection [16].

The UAVs & UGVs system is a typical kind of multiagent system with heterogeneous vehicles, and similar systems are composed by unmanned surface vehicles (USVs), unmanned underwater vehicles (UUVs), and unmanned aerial-aquatic vehicles (UAAVs), such as UAVs & USVs system [17], UAVs & UUVs system [18], UAV & USV & UUV system [19], and UAAV & UUV system [20]. In those systems, the heterogeneous vehicles can take their respective advantages and compensate for each other. In a UAVs & UGVs system, UGV can locate or detect the objects on the ground with higher accuracy and make up the weakness of UAVs. Besides, UGVs have a larger payload capability and a powerful computational ability, and they can also charge for the UAVs. As for the

UAVs, they can fly with high velocity and have a wide field of vision, which is beneficial to guide the UGVs to avoid obstacles [21]. In [22] and [23], a UAV equipped with a camera serves as the eye in the airspace and collects the ground images. Then, after some processing, such as denoising and correction, the ground map can be constructed to assist the UGV in planning its path to the destination. A hybrid path planning algorithm is proposed in [23]. GA works as the outer loop to conduct the global path planning, and the rolling optimization is applied in the inner loop to optimize the results obtained from GA locally. Considering the different dynamics characteristics of UAV and UGV, an improved artificial physics approach is developed in [24] to coordinate UAVs and UGVs to reach the same target. In the proposed approach, random forces are introduced to make the UAVs and UGVs avoid the local optimum and head for the target. The UAVs & UGVs system also can be applied in a wilderness search and rescue task [25]. The UAVs and UGVs carry out an aerial search and a ground search, respectively, and the targets detected by UAVs must be identified by a UGV again to ensure the accuracy. Isoprobability curves are used to search the target by both UAVs and UGVs, and the new approach allows UAVs to cover all possible target motions while also avoiding the search redundancies considering the paths of UGVs.

In other cases, UGVs serve as moving platforms, and they can both charge for the UAVs and carry the UAVs to voyage longer distance [26]. Under this work mode, there are usually multiple places to be visited in a task, and those places cannot be reached directly by the UGVs. First, UAVs are carried by the UGVs to arrival at some areas near the places and then fly to the destination to finish the “last mile” of the visit task [27]. According to the above descriptions, the process can be formulated into a traveling salesman problem (TSP), and many variations of the basic TSP are studied in the existing literatures. A package delivery task is completed by a UAV & UGV system in [28], and the packages must reach the customers located at different places. The path of UGV is restricted to the road network, and a two-stage strategy is proposed to optimize both the paths of UAV and UGV. An improved ant colony optimization (ACO) algorithm is used to obtain the optimal path of UGV that can make the UAV visit more targets, and the GA algorithm is then applied to optimize the path of UAV. Similarly, a UAV & UGV system is introduced to explore a planetary surface, and the goal is to visit a number of target points while minimizing the travel distance in [29]. In this task, UGV is regarded as a moving charging station for the UAV, and the UAV is responsible for reaching the target points. A cooperative exploration routing algorithm is proposed to solve the problem. The charging stops where the UGV can carry the UAV is optimized first, and the path of UAV is calculated by solving the TSP. UGVs also can serve as mobile charging stations in a persistent surveillance task of a partitioned environment, and the paths of UAVs and UGVs and the charging schedule of UAVs need to be optimized to minimize the time interval between two consecutive visits to regions [30]. Several UAVs & UGV teams are uniformly distributed over a cyclic path to execute the surveillance task of a partitioned region, and the UAVs are being recharged by the

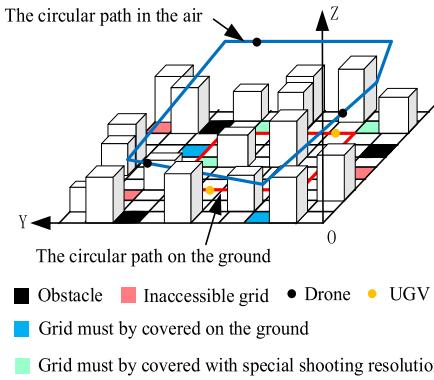


Fig. 1. Scenario of performing the persistent surveillance task by the UAVs & UGVs system.

UGV while they are transported from the current partition to the next one. With this strategy, the large-scale optimization problem is simplified into optimizing a supercycle of partitions for a single team composed by several UAVs and one UGV.

In most cases, a team of UAVs is qualified for the persistent surveillance task in urban environments [31]. However, when there are special requirements of the shooting resolution on the ground area, the UAVs may not be able to fulfill the task, and the introduction of UGVs can address this problem and improve the efficiency of performing the task at the same time. To the best of our knowledge, there are few publications reporting the studies on cooperative path planning of persistent surveillance task performed by the UAVs & UGVs system.

### III. COOPERATIVE PATH PLANNING MODEL FOR THE UAVS & UGVs SYSTEM IN THE PERSISTENT SURVEILLANCE TASK

To facilitate the operation of air traffic management for drones, the low-altitude airspace in urban environments is divided into a number of cubes with the same size, like the operation of traffic management in civil aviation [32]. The vertexes of the cubes are the possible path points which the drone can fly across. Accordingly, the ground area is modeled as a structure composed by many 2-D grids, which are the projections of corresponding cubes on the horizontal plane. The UGV can only pass the vertexes of the 2-D grids on the ground. Under the proposed operation framework, the contents of establishing the cooperative path planning model are depicted.

#### A. Description of the Persistent Surveillance Task and the Travel Rules of Vehicles

In urban environments, there are many buildings distributed in the low-altitude airspace, and the space occupied by the buildings is denoted by the cubes. Drones must avoid those buildings when they are flying, and the UGVs also need to keep certain distance away from the buildings. Besides, there are some obstacles on the ground, and the UGVs must avoid them. To perform the persistent surveillance task of the whole ground area, two circular paths must be obtained for the travel of drones and UGVs, respectively, as shown in Fig. 1.

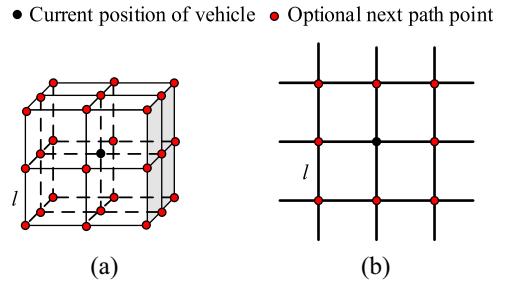


Fig. 2. Travel rules for the (a) drone and (b) UGV.

In Fig. 1, the coordinate system OXYZ is established to describe the locations of grids, buildings, drones, and UGVs. The cubes denote the buildings with different heights, and there are four special types of 2-D grids.

- 1) Inaccessible grids that are forbidden to be covered by the drone or the UGV.
- 2) Grids must be covered with special requirements on the shooting resolution, which means the drone must reduce its flight height to achieve a valid coverage.
- 3) Grids must be covered on the ground, and they only can be covered by the UGVs.
- 4) Grids which are occupied by obstacles, and the UGV must bypass them.

All the 2-D grids must be covered to realize a complete coverage except for the inaccessible grids and the grids occupied by the buildings and obstacles. There are several drones and UGVs located at the circular paths in the air and on the ground, respectively, with the same interval, which can balance the workload of each vehicle and ensure the normal operation of the persistent surveillance task.

The travel rules for the drone and the UGV are defined as follows. The drone located at one vertex can choose the vertex of a connected edge, a face diagonal, or a body diagonal in one neighboring cube as the next path point, and it is assumed to fly along the straight line between two neighboring path points. As for the UGV, it can select the vertex of a connected edge or a face diagonal in one neighboring 2-D grid as the next path point. The above travel rules are shown in Fig. 2.

In Fig. 2,  $l$  is the length of an edge both for a cube and a 2-D grid. According to the above descriptions, the equations expressing the motion of the drone and the UGV are presented, respectively, in

$$\begin{cases} x_D^i(n+1) = x_D^i(n) + d_{Dx}^i(n), & x_D^i(n) \in \{0, l, 2l, \dots, X_{\max}\} \\ y_D^i(n+1) = y_D^i(n) + d_{Dy}^i(n), & y_D^i(n) \in \{0, l, 2l, \dots, Y_{\max}\} \\ z_D^i(n+1) = z_D^i(n) + d_{Dz}^i(n), & z_D^i(n) \in \{l, 2l, \dots, Z_{\max}\} \\ \text{s.t. } \left| d_{Dx}^i(n) \right|, \left| d_{Dy}^i(n) \right|, \left| d_{Dz}^i(n) \right| \in \{-l, 0, l\} \\ \left| d_{Dx}^i(n) \right| + \left| d_{Dy}^i(n) \right| + \left| d_{Dz}^i(n) \right| \neq 0 \end{cases} \quad (1)$$

$$\begin{cases} x_G^j(m+1) = x_G^j(m) + d_{Gx}^j(m), & x_G^j(m) \in \{0, l, 2l, \dots, X_{\max}\} \\ y_G^j(m+1) = y_G^j(m) + d_{Gy}^j(m), & y_G^j(m) \in \{0, l, 2l, \dots, Y_{\max}\} \\ \text{s.t. } \left| d_{Gx}^j(m) \right|, \left| d_{Gy}^j(m) \right| \in \{-l, 0, l\} \\ \left| d_{Gx}^j(m) \right| + \left| d_{Gy}^j(m) \right| \neq 0 \end{cases} \quad (2)$$

where  $(x_D^i(n), y_D^i(n), z_D^i(n))$  is the position of drone No.  $i$  at the  $n$ th path point, and  $(d_{Dx}^i(n), d_{Dy}^i(n), d_{Dz}^i(n))$  is the step length along the axes OX, OY, and OZ, respectively. Besides, the drone cannot stay at the current path point, which is expressed in the second constraint of (1). The variables defined for the UGV in (2) have the similar meanings with those in (1).  $X_{\max}$ ,  $Y_{\max}$ , and  $Z_{\max}$  are the boundaries of the defined low-altitude airspace. With the above settings, the complete set of the cubes, 2-D grids, and the points in the defined low-altitude airspace can be coded as two 3-D arrays and one 2-D array, as shown in

$$\left\{ \begin{array}{l} G3(gx, gy, gz) \left( gx = 1, 2, \dots, \frac{X_{\max}}{l} \right. \\ \quad \left. gy = 1, 2, \dots, \frac{Y_{\max}}{l}; gz = 1, 2, \dots, \frac{Z_{\max}}{l} \right) \\ G2(gx, gy) \left( gx = 1, 2, \dots, \frac{X_{\max}}{l} \right. \\ \quad \left. gy = 1, 2, \dots, \frac{Y_{\max}}{l} \right) \\ P(px, py, pz) \left( px = 1, 2, \dots, \frac{X_{\max}+l}{l}; \right. \\ \quad \left. py = 1, 2, \dots, \frac{Y_{\max}+l}{l}; pz = 1, 2, \dots, \frac{Z_{\max}+l}{l} \right) \end{array} \right. \quad (3)$$

where the arrays  $G3$  and  $G2$  record the cubes and the 2-D grids, respectively, and the position information of the points is recorded by the array  $P$ . When formulating the cooperative path planning model into the mathematical form, the elements in the arrays  $G3$ ,  $G2$ , and  $P$  will be assigned specific values to denote their different states.

### B. Model of the Coverage Range for the Drone and the UGV

In this study, the camera is assumed to be equipped in the center of the drone toward the ground, and the covered ground area is centrosymmetric with respect to the projection of the drone on the ground. The above assumptions have been applied in many existing studies, such as [8] and [19]. With the above assumptions, the coverage range of drone is affected by the flight height and can be denoted as the following equation:

$$\left\{ \begin{array}{l} cx_{Dl}^i(n) = \max\{0, x_D^i(n) - z_D^i(n)\} \\ cx_{Dr}^i(n) = \min\{x_D^i(n) + z_D^i(n), X_{\max}\} \\ cy_{Dl}^i(n) = \max\{0, y_D^i(n) - z_D^i(n)\} \\ cy_{Dr}^i(n) = \min\{y_D^i(n) + z_D^i(n), Y_{\max}\} \end{array} \right. \quad (4)$$

where the intervals  $[cx_{Dl}^i(n), cx_{Dr}^i(n)]$  and  $[cy_{Dl}^i(n), cy_{Dr}^i(n)]$  are the boundaries of the coverage area along the axes OX and OY, respectively. It can be learned from (4) that the coverage range is a rectangle, and the area of the rectangle is expanded with the increasing flight height of the drone. Taking the cases of  $z_D^i(n) = l$  and  $z_D^i(n) = 2l$  as examples, the corresponding covering range of the drone is shown in Fig. 3.

In Fig. 3(a), the building cannot be located in one of the four covered 2-D grids, or the drone will collide with it. In Fig. 3(b), when calculating the valid covering range, the grids occupied by the building should be excluded. Besides, two more situations must be taken into account. First, the covering range may be blocked by the nearby buildings, and those 2-D grids are also regarded as the invalid coverage areas. Second, for the 2-D grids that have the special requirements on the shooting resolution, they are still treated as the uncovered ones

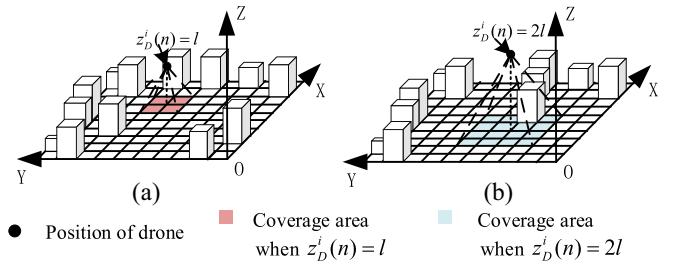


Fig. 3. Covering range of the drone when the flight height is  $l$  in (a) and  $2l$  in (b).

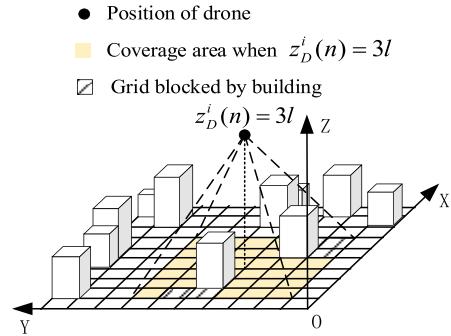


Fig. 4. Examples of the valid covering 2-D grids when  $z_D^i(n) = 3l$ .

when the flight height of drone is greater than the required value. Fig. 4 is given to show that the covering range is blocked by the buildings.

In Fig. 4, the shaded 2-D grids are the ones blocked by the buildings from the current position of the drone, and they are viewed as the uncovered area.

For the UGV No.  $j$  located at the point  $(x_G^j(m), y_G^j(m))$ , its coverage range can be expressed in

$$\left\{ \begin{array}{l} cx_{Gl}^j(m) = \max\{0, x_G^j(m)\} \\ cx_{Gr}^j(m) = \min\{x_G^j(m), X_{\max}\} \\ cy_{Gl}^j(m) = \max\{0, y_G^j(m)\} \\ cy_{Gr}^j(m) = \min\{y_G^j(m), Y_{\max}\} \end{array} \right. \quad (5)$$

where the intervals  $[cx_{Gl}^j(m), cx_{Gr}^j(m)]$  and  $[cy_{Gl}^j(m), cy_{Gr}^j(m)]$  are the boundaries of the coverage area along the axes OX and OY, respectively. The UGV can cover four neighboring 2-D grids, which has the same coverage range of drone when the flight height is  $l$ .

With the above descriptions, two functions can be introduced. The first one is  $f_{\text{cov}}$ , in which the input is the position of vehicle and the output is the corresponding covered 2-D grids. On the contrary, the function  $f_{\text{po}}$  is used to calculate the set of points which can cover the specific 2-D grid. The two functions are useful when checking whether a complete coverage can be realized.

### C. Mathematical Formulation of the Cooperative Path Planning Problem

As the low-altitude airspace in urban environments is described by a number of cubes, the cooperative path planning

problem for the persistent surveillance task becomes a discrete optimization problem. The 3-D array  $P$  is regarded as the optimization variables, and the values of the elements  $P(px, py, pz)$  are determined by

$$P(px, py, pz) = \begin{cases} -1, & G3(gx, gy, gz) \in B, G2(gx, gy) \in B^+ \\ & \text{or } f_{\text{cov}}(px, py, pz) \cap I \neq \emptyset \\ 0 \text{ or } 1, & \text{others} \end{cases} \quad (6)$$

where  $gx = \{px - 1, px\}$ ,  $gy = \{py - 1, py\}$ , and  $gz = \{pz - 1, pz\}$  denote the codes of the cubes or the 2-D grid next to  $P(px, py, pz)$ .  $B$  is the set of cubes that are occupied by the buildings,  $B^+$  is the set of 2-D grids that are occupied by the projections of the buildings in the XOY plane and the obstacles on the ground, and  $I$  is the set of 2-D grids that is forbidden to be covered (called as the inaccessible 2-D grids for convenience).

In (6),  $P(px, py, pz) = -1$  denotes that the point is located at the building, the obstacle, or the inaccessible 2-D grid. In this case, the point is always forbidden to be passed, and it is defined as the inactive point. The other points are defined as the active points, and their on-off states are expressed by “1” and “0” and need to be optimized. When the point is closed, the vehicle (drone or UGV) need not pass it during the travel while the vehicle must pass all the open points to realize a complete coverage.

Besides, another 2-D array  $S_{\text{ideal}}(sx, sy) (sx = 1, 2, \dots, (X_{\max}/l); sy = 1, 2, \dots, (Y_{\max}/l))$  is introduced to record the ideal state of the 2-D grids when a complete coverage is realized, the value of  $S_{\text{ideal}}(sx, sy)$  is defined in

$$S_{\text{ideal}}(sx, sy) = \begin{cases} 0, & S_{\text{ideal}}(sx, sy) \in (B^+ \cup I) \\ 1, & \text{others.} \end{cases} \quad (7)$$

Then, to realize a complete coverage, the following constraint must be satisfied:

$$\{(sx, sy) | S_{\text{ideal}}(sx, sy) = 1\} = \cup_{P(px, py, pz)=1} f_{\text{cov}}(px, py, pz) \quad (8)$$

where the symbol  $\cup$  is the union of 2-D grids that all the open points can cover. The goal of the persistent surveillance task is to minimize the time of finishing a complete coverage, which can be expressed by

$$J = \max \left\{ \frac{D_D}{N_D \cdot v_D}, \frac{D_G}{N_G \cdot v_G} \right\} \quad (9)$$

where  $D_D$  and  $D_G$  are the lengths of the circular paths for drones and UGVs, respectively,  $N_D$  and  $N_G$  are the number of drones and UGVs, and  $v_D$  and  $v_G$  are their travel velocities. The  $N_D$  drones or  $N_G$  UGVs are distributed evenly along the circular path in the air or on the ground, and they only travel a part of the circular path in every cycle. A greater number of drones and UGVs will result in shorter travel distance of them in each cycle, and the number of drones and UGVs is decided by considering the current available vehicles and the economic benefit comprehensively. The optimization index in (9) is the longer time of traveling the circular path between the drones and the UGVs. In fact, only the 2-D grids that must be covered on the ground are within the task of UGV, and the other 2-D grids all can be covered by the drones' flying. With

the index in (9), the UGVs can share the work of the drones by covering more 2-D grids in the premise of finishing their own work, thus reducing the time of performing a complete coverage for the UAVs & UGVs system.

#### IV. HYBRID EDA-GA-BASED ALGORITHM SOLVING THE CIRCULAR PATHS FOR THE DRONES AND THE UGVS

The cooperative path planning problem has been formulated into a 0-1 optimization problem, and two nested subproblems are contained. In the outer loop, the on-off states of the active points need to be optimized, but it is not enough to calculate the index value in (9) because the combination of the on-off states of the active points cannot result in the circular path. In the inner loop, the sequences of passing those open points in the air or on the ground must be determined for the drones and the UGVs, respectively, and the paths that satisfy the travel rules of the drones and the UGVs also need to be generated to form the circular path. As there are a great number of active points distributed in the low-altitude airspace, it is a large-scale 0-1 optimization problem with complicated constraints.

The exact methods and the heuristic methods are widely applied to solve such problems, but it is usually difficult to obtain the optimal solution. In [33] and [34], some studies are conducted to reduce the number of optimization variables and constraints, thus accelerating the computational speed of the exact methods. However, only the suboptimal solution can be guaranteed theoretically. Compared to the exact methods, the solutions obtained by the heuristic methods are close to the optimal solution and cost less computation time. Many heuristic methods are developed for solving the continuous optimization problems, which are not suitable for this problem. Besides, as the scale of the problem is large, it is difficult for the heuristic methods that begin with only one initial solution to search for the excellent solutions from a large number of feasible solutions. Swarm-based heuristic methods are begun with a number of initial solutions and are fit for solving this problem. EDA and GA are two typical swarm-based methods that are widely used to solve both the continuous and the discrete optimization problem. They have their own advantages and also suffer from drawbacks. In this study, a hybrid EDA-GA-based algorithm is proposed to complement the two algorithms, and some improvements are also made according to the characteristic of the established cooperative path planning model.

##### A. Determination of the Combination of Open Points by EDA and GA

In EDA, the solutions are updated based on the statistical learning theory, and a probabilistic model is established to describe the distribution information of solutions. New solutions are generated by adopting the random sampling on the established probabilistic model. Based on the basic flow of EDA, some improvements are made when initializing and updating the solutions to ensure that the combination of open points can always satisfy the constraint of realizing a complete coverage. Assuming that the number of initial solutions and the maximum iteration time are  $N_s$  and  $T_{\max}$ , respectively, the detailed steps of running EDA are listed as follows.

*Step 1:* Initialize  $N_s$  solutions.

It is hard to realize a complete coverage when the on-off states of the active points are randomly selected from 0 and 1. To obtain a feasible solution [the solution satisfying the constraint in (8) is called as a feasible solution], all the active points are assumed to be open first. Then, one open point is selected at a time, and it will be checked whether the combination of the remaining open points can still realize a complete coverage if the selected open point is turned to be closed. If it is true, the selected open point will be closed, and the above operation will be repeated by selecting another open point randomly. Otherwise, a feasible solution is found, which is the combination of current open points. To sum up, a feasible solution is obtained by closing the open point one by one until a complete coverage cannot be realized, and  $N_s$  initial solutions are all generated in this way. The index value of the solution is calculated according to (9). Then, set the current iteration time  $t = 1$  and the iterative process begins.

*Step 2:* Select  $N_b$  solutions with smaller index values to form the elite group.

*Step 3:* Establish a probabilistic model to describe the distribution information of the solutions in the elite group.

In this problem, the 0-1 distribution is introduced to denote the opening probability for each active point, and a 3-D probability array  $\tilde{P}$  that has the same size of  $P$  is defined. For the points which satisfy  $P(px, py, pz) = -1$ ,  $\tilde{P}(px, py, pz)$  to express that the point is always closed. The value of the other element in  $\tilde{P}$  can be determined by

$$\tilde{P}(px, py, pz) = \frac{\sum_{i=1}^{N_b} P(px, py, pz)}{N_b}. \quad (10)$$

In (10), the opening probability of each point is calculated, and new solutions can be generated relying on the probability array  $\tilde{P}$ .

*Step 4:* Generate the new solutions with random sampling.

As there is no guarantee that the new solutions are feasible, the following operation is adopted to repair an infeasible solution into a feasible one. The 2-D grids that have not been covered by the combination of open points corresponding to the infeasible solution are recorded first. For each 2-D grid, the function  $f_{po}$  is used to calculate the set of points which can cover it, from which one point is randomly selected and is turned to be open. After all the uncovered 2-D grids have carried out the above process, the combination of the existing open points can realize a complete coverage.

*Step 5:* Calculate the index value of the new solution using (6) and check whether  $t = T_{\max}$  is satisfied. If the answer is yes, output the best combination of open points. Otherwise,  $t = t + 1$  and repeat step 2.

GA also can be applied to solve this 0-1 optimization problem. Compared to the original form of GA, the *Selection* operator is not included because all the initial solutions are feasible ones. It will lose the diversity of solutions as half the initial solutions are abandoned when executing the *Selection* operator. Although the *Crossover* and *Mutation* operators are imposed, they do not show a great effect on the diversity of solution as they only modify the solution at the microlevel, which is not so obvious for a large-scale optimization problem.

The procedures of GA when solving the proposed problem are presented as follows.

*Step 1:* Initialize  $N_s$  solutions and make them feasible according to the same approach in step 1 of EDA. Set the current iteration time  $t = 1$  and the iterative process begins.

*Step 2:* Select  $N_c$  points randomly and treat them as the operating positions to execute the *Crossover* operator. A pool is defined to hold the  $N_s$  solutions, and the best and the worst solutions are picked out to exchange their values at the operating positions.  $N_s$  new solutions are generated.

*Step 3:* For each solution,  $N_m$  points are randomly selected and treat them as the operating positions to execute the *Mutation* operator with the probability  $p_m$ .  $N_s$  new solutions are generated.

*Step 4:* Repair the infeasible solutions into feasible ones using the approach in step 4 of EDA.

*Step 5:* Calculate the index value of the new solution using (9) and check whether  $t = T_{\max}$  is satisfied. If the answer is yes, output the best combination of open points. Otherwise,  $t = t + 1$  and repeat step 2.

From the above descriptions, EDA and GA share the similar framework but their principles are different. The group information is utilized in EDA and the evolutionary pattern of solutions is at the macrolevel. However, GA makes the solutions evolve by changing the information in some individual positions, which is at the microlevel. To sum up, the EDA is with a strong capability in the global search, and the GA has the advantage in the local search. The combination of EDA and GA is promising to further enhance their performances and obtain a more optimal combination of the on-off states of the active points.

## B. Improvement for EDA and GA: Hybrid EDA-GA Algorithm

For an excellent heuristic optimization algorithm, it should be capable of conducting an efficient global search in the early phase of the iterative process to narrow down the space where the optimal solution is located at. In the later phase of the iterative process, a strong ability in the local search is required to determine the location of the optimal solution [35]. Inspired by the advantages of EDA and GA, the hybrid EDA-GA algorithm is developed to further improve the solution quality when determining the on-off states of the active points. There have been many modified versions of EDA and GA, but they mainly focus on changing the probabilistic model in EDA [36] and introducing new operators to GA [37]. Those modifications can improve the performance of EDA and GA, respectively, in some specific problem but fail to make the best of the two algorithms considering their advantages. In this study, the modifications on EDA and GA have been described in Section IV-A based on the characteristics of the established cooperative path planning model, which can enhance the performance of the two algorithms at a microlevel. The hybridization between EDA and GA is performed in view of the following three drawbacks exposed in the two algorithms.

- 1) The excellent solutions in the previous iteration are not reserved. After the new solutions are generated by the two algorithms, respectively, those solutions are reserved

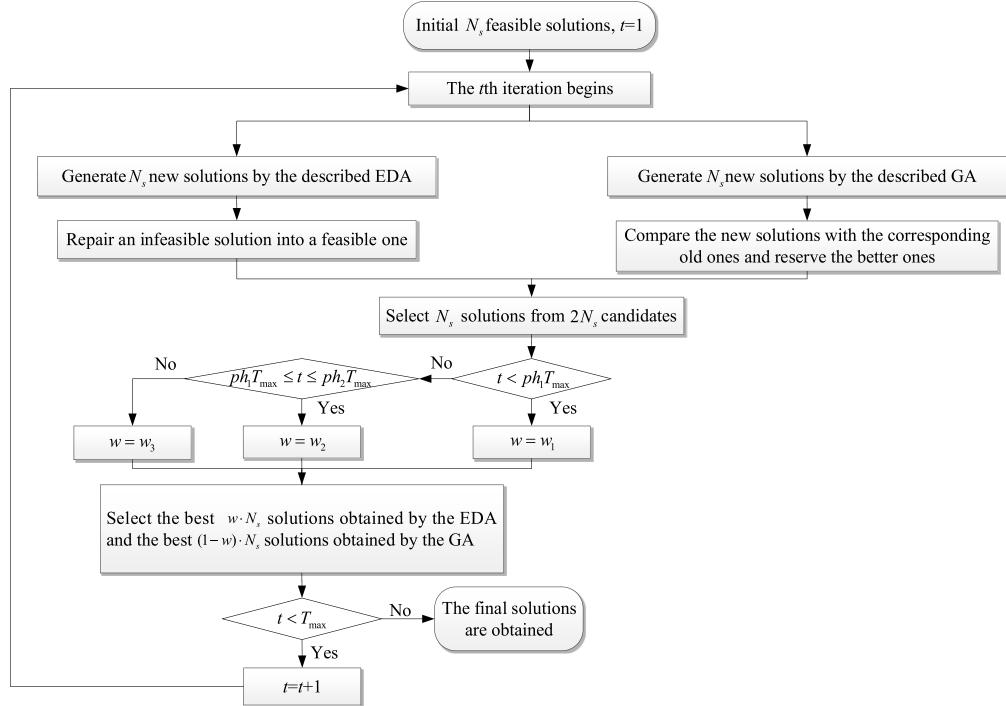


Fig. 5. Procedures of running the hybrid EDA-GA algorithm.

directly and take the places of the old ones without comparing their qualities. As there is no guarantee that the new solutions are better than the old ones, the excellent solutions in the previous iteration may be lost in this way.

- 2) The same operation is contained in the two algorithms. As the EDA and GA are hybridized, the repeat of the same operation in one iteration is not so effective in accelerating the convergence rate but adding the extra computation time. For example, the operation of repairing the new solution is applied both in EDA and GA and should be remained only in one algorithm after the hybridization.
- 3) The searchability in the global and local levels should be adaptively adjusted in different phases of the iterative process.

In the early phases of the iterative process, the global search is the emphasis, and the solutions obtained by EDA should be paid more attention. As the iterative process goes on, the local search that is mainly conducted by GA needs to be valued. With this mechanism, the performance of the hybrid EDA-GA algorithm can be enhanced in different phases of the iterative process.

With the above guidelines, the flow of the hybrid EDA-GA algorithm is presented in Fig. 5.

In each iteration, EDA and GA begin parallelly with the same solutions obtained from the previous iteration. Then, they follow their own steps shown in Section IV. A to generate the new solutions. Note that in GA, the repair operation on the new solutions is not performed. Instead, the new solution is compared with the corresponding old one, and the better one will be reserved. As all the initial solutions are feasible, the

TABLE I  
COMPUTATIONAL COMPLEXITY OF THE HYBRID EDA-GA ALGORITHM

Step	Computational complexity
Initialize the solutions	$N_s \times N_a$
Calculate the fitness value in Eq. (8)	$N_s$
Rank the solutions (in EDA branch)	$N_s$
Establish a probabilistic model (in EDA branch)	$N_b \times N_a$
Generate the new solutions (in EDA branch)	$N_s$
Execute the crossover operator (in GA branch)	$\frac{N_s}{2} \times N_c$
Execute the mutation operator (in GA branch)	$N_s \times N_m$
Compare the new solution with the corresponding old one (in GA branch)	$N_s$
Rank and select the solutions to continue the iterative process	$2N_s$

reserved solutions are always feasible ones. In this way, the excellent solutions in the previous iteration cannot be missed, and the repair operation will only be performed in EDA. There are  $2 \cdot N_s$  new solutions after one time of iteration in EDA and GA, and only  $N_s$  solutions will be reserved.  $w$  is a parameter to show the relative importance of EDA and GA in different phases of the iterative process. In the early phase of the iterative process ( $t < ph_1 T_{\max}$ ), more excellent solutions obtained by the EDA will be reserved to take the advantage of EDA in global search, and  $w$  should be set as a larger value. While in the later phase ( $t > ph_2 T_{\max}$ ), the influence of GA grows by giving emphasis on the local search, and  $w$  is set to be a smaller value.

Besides, the computational complexity of the hybrid EDA-GA algorithm, i.e., the computation times in each iteration, is presented in Table I.

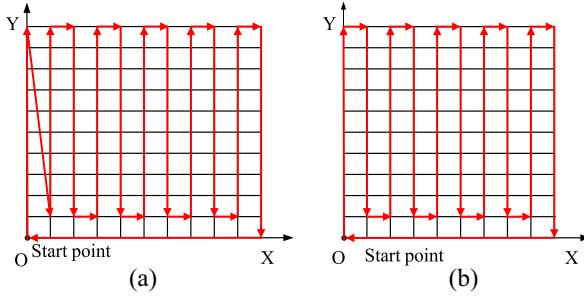


Fig. 6. Two cases of obtaining the circular path. (a) Number of points on each side is odd. (b) Number of points on each side is even.

In Table I,  $N_a$  is the number of active points defined in the discrete urban environments, and the overall computational complexity of the hybrid EDA-GA algorithm after  $T_{\max}$  iterations is  $N_s \times N_a + T_{\max}(6N_s + N_b \times N_a + (1/2)N_s \times N_c + N_s \times N_m)$ . In the hybrid EDA-GA algorithm, the operators in GA and EDA are both used, and new operators are also developed. Therefore, the computation load of the hybrid EDA-GA algorithm is heavier in each iteration. In real practices, the three algorithms are all run offline, and the technology of parallel computing can be applied to further reduce the computation time.

### C. Simple Sweep-Based Approach to Determine the Circular Paths for the Drones and the UGVs

The on-off states of the active points are determined by the hybrid EDA-GA algorithm while the index value for a combination of open points is calculated by obtaining the circular paths that are composed by those opening points and conform to the travel rules of the drone and UGV. Although they can be formulated into two TSPs, it will take a lot of computing time to solve the large-scale TSP using the heuristic algorithms, such as GA, ACO algorithm, and simulated annealing (SA) algorithm. Inspired by [9], a simple sweep approach is designed to obtain the circular paths for the drones and the UGVs, respectively. Fig. 6 is provided to explain the approach in the XOY plane.

In Fig. 6, for a square area, the results are different and depend on whether the number of points on each side is odd or even, and the drones need to change their flight height to pass every open point in the air. As an example, the pseudocode of forming a circular path for the drones when the number of points on each side of the square area is odd is provided in Algorithm 1.

The array circular path of drones (CPDs) is used to record the sequence of traveling the open points in the air. The drone flies across the points in CPD one by one, and the circular path in the air is obtained. The circular path for the UGVs also can be obtained in the same way, and the array circular path of ground vehicles (CPG) is defined to record the sequence of traveling the open points on the ground. Note that the drone may not be able to fly between the two neighboring points in the array CPD directly due to the blocking of buildings or obstacles and the constraint of the travel rules on the vehicles. A\* algorithm is used to calculate the travel paths between

---

### Algorithm 1 Obtain the Circular Path in the Air

---

```

1:  $d = 0$ 
2: for  $i = 1:(X_{\max}/l + 1)$ 
3:   if  $i == 1$  then
4:     for  $j = 1:(Y_{\max}/l + 1)$ 
5:       for  $k = 1:Z_{\max}/l$ 
6:         if  $P(i, j, k) == 1$  then
7:            $CPD(d + 1) = (i, j, k)'; d = d + 1$ 
8:         end
9:       end
10:    end
11:   else
12:     if  $mod(i, 2) == 0$  then
13:       for  $j = 2:(Y_{\max}/l + 1)$ 
14:         for  $k = 1:Z_{\max}/l$ 
15:           if  $P(i, j, k) == 1$  then
16:              $CPD(d + 1) = (i, j, k)'; d = d + 1$ 
17:           end
18:         end
19:       end
20:     else
21:       for  $j = (Y_{\max}/l + 1): - 1:2$ 
22:         for  $k = 1:Z_{\max}/l$ 
23:           if  $P(i, j, k) == 1$  then
24:              $CPD(d + 1) = (i, j, k)'; d = d + 1$ 
25:           end
26:         end
27:       end
28:     end
29:   end
30: end
31: for  $i = (X_{\max}/l + 1): - 1:1$ 
32:   for  $k = 1:Z_{\max}/l$ 
33:     if  $P(i, j, k) == 1$  then
34:        $CPD(d + 1) = (i, j, k)'; d = d + 1$ 
35:     end
36:   end
37: end

```

---

### Algorithm 2 Generate All the Path Points for Drones

---

```

1:  $d0 = 1$ 
2: while  $d0 \leq d - 1$ 
3:   if the point  $CPD(d0)$  has not been visited then
4:     for  $dd = (d0 + 1):d$ 
5:       if the point  $CPD(dd)$  has not visited then
6:         Calculate the travel path from the points
7:          $CPD(d0)$  to  $CPD(dd)$  using A* algorithm
8:         Add those path points to the array  $PPD$ .
9:       break
10:      end
11:    end
12:    $d0 = d0 + 1$ ;
13: end

```

---

two neighboring open points in the circular paths. Taking the drones' flying as an example again, the algorithm of generating all the path points of the circular path is shown in Algorithm 2.

The array path point of drones (PPDs) contains all the path points of the circular path for the drones, and the corresponding array for the UGVs is defined as path point of ground vehicles (PPG). In Algorithm 2, when the travel path from the open points  $CPD(d0)$  to  $CPD(dd)$  contains the open point  $CPD(dr)$ , ( $dr = dd+1, dd+2, \dots, d$ ), the open point  $CPD(dr)$  need not be passed again in the following travel. Besides,

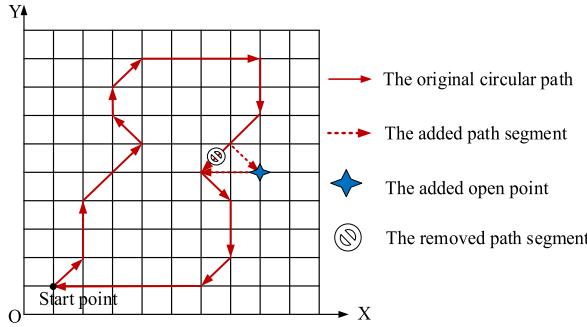


Fig. 7. Strategy of covering the extra 2-D grids by adding the open points.

some closed points may be contained when calculating the travel path between two open points. As those closed points also must be passed during the travel, they are turned into the open points. Therefore, when checking whether a complete coverage can be realized by the combination of the open points, it is the path points in the arrays PPD and PPG that are actually being checked. In this way, the drones and the UGVs can travel fewer repeated path points, thus reducing the travel time of realizing a complete coverage.

## V. STRATEGY OF ONLINE ADJUSTMENT FOR THE CIRCULAR PATHS

The state of the ground area will not always stay the same. Some regions may become inaccessible at a certain time, and some regions are newly required to be covered. To maintain the normal operation of the persistent surveillance task, an effective strategy is developed to adjust the travel paths online in some local regions which are influenced by the changes of states on the ground.

### A. Strategy of Covering the Extra 2-D Grids

In this case, some 2-D grids that were inaccessible previously need to be covered now. Assuming that the array recording the code of the extra 2-D grids is  $E$ , the number of the extra 2-D grids is  $ex$ , the additional open points covering the extra 2-D grids can be solved by using the function  $f_{po}$ . To reduce the influence of the additional open points on the existing circular path, the open point only can be added between two neighboring path points in the existing circular path, as shown in Fig. 7.

As there may be more than one active point that can cover an extra 2-D grid, the one which lead to the smallest index value is preferred. Followed by the idea, the algorithm of covering the extra 2-D grids is designed in Algorithm 3.

In line 2,  $cp_i$  is the set of points which can cover the 2-D grid  $E(i)$ . After running Algorithm 3, a final check is needed on the points in the set  $AP$  one by one to confirm whether they really contribute to covering the extra 2-D grid. If there is no additional 2-D grid covered by adding the point, this point will be removed from the set  $AP$ , and the remaining points in the set  $AP$  can still cover the extra 2-D grids. Then, every point in the set  $AP$  will be inserted into the corresponding position of the circular path. With the proposed approach, the extra 2-D grids can be covered by adding some open points

---

### Algorithm 3 Cover the Extra 2-D Grids

---

```

1: for  $i = 1:ex$ 
2:    $cp_i = f_{po}(E(i))$ 
3:   for  $j = 1:length(cp_i)$ 
4:     Record the set  $AD_{ij}$  composed by the two points that
       can minimize the index value in Eq. (8).
5:   The index value corresponding to  $cp_i(j)$  and  $AD_{ij}$  is
       recorded in an array  $VA_i(j)$ .
6: end
7: end

```

---

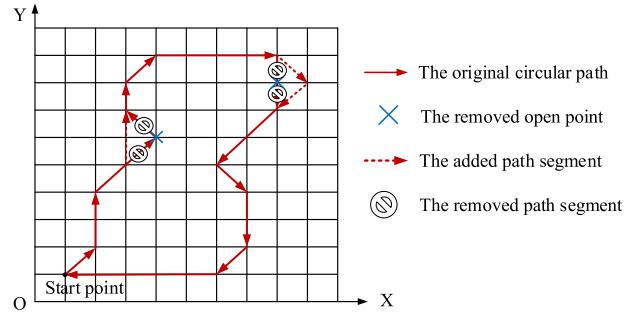


Fig. 8. Strategy of avoiding the extra inaccessible 2-D grids by removing the open points.

locally to the existing circular path considering both reducing the influence on the existing circular path and optimizing the index value.

### B. Strategy of Addressing the Extra Inaccessible 2-D Grids

Opposite to the situation in Section V-A, some 2-D grids which could be covered by traveling the circular paths may become the inaccessible grids at a certain time. The circular path must be changed to avoid covering those inaccessible grids. By using the function  $f_{po}$  again, the path points in the circular paths that cover the extra inaccessible 2-D grids can be obtained, and those points must be removed from the arrays PPD and PPG and are regarded as the inactive points. The remaining points in the arrays PPD and PPG will compose the new circular paths, and the corresponding new arrays are denoted as  $\widetilde{PPD}'$  and  $\widetilde{PPG}'$ , respectively, as shown in Fig. 8. The corresponding algorithm is presented in Algorithm 4.

The arrays  $\widetilde{PPD}$  and  $\widetilde{PPG}$  are the new circular paths containing all the path points for the drones and the UGVs that can avoid the extra inaccessible 2-D grids. Note that after removing the points from the arrays PPD and PPG, it is probable that some 2-D grids cannot be covered by the points in the arrays  $\widetilde{PPD}$  and  $\widetilde{PPG}$ , so Algorithm 3 is run again to supplement some points and realize a complete coverage.

Sometimes, the situation will become more complicated as the increase and reduction of inaccessible 2-D grids are happened simultaneously, and a comprehensive adjustment is needed. Algorithm 4 is also fit for addressing the complicated situations. The open points that can cover the inaccessible 2-D grids should be removed from the circular paths. Those points are treated as the inactive points and cannot be passed in the future travel of drones and UGVs. Then, Algorithm 3 is performed to guarantee the extra 2-D grids being covered, and the added open points are contributory to realize a complete coverage.

**Algorithm 4** Avoid the Extra Inaccessible 2-D Grids

```

1: for  $i = 2:length(PPD')$ 
2:   if the drone can fly directly from  $PPD'(i - 1)$  to
       $PPD'(i)$  then
3:     Add the points  $PPD'(i - 1)$  and  $PPD'(i)$  to
       the array  $\widetilde{PPD}$ 
4:   else
5:     Generate the path points from  $PPD'(i - 1)$  to
        $PPD'(i)$  using A* algorithm
6:     Add the path points to the array  $\widetilde{PPD}$ .
7:   end
8: end
9: The above steps also will be imposed on  $PPG'$  to form the array
 $\widetilde{PPG}$ .
10: if the combination of points in  $\widetilde{PPD}$  and  $\widetilde{PPG}$  fail to realize a
    complete coverage then
11: Algorithm 3 is performed to add the points to  $\widetilde{PPD}$  and  $\widetilde{PPG}$ 
12: The new arrays are denoted as  $\overline{PPD}$  and  $\overline{PPG}$  respectively.
13: end

```

TABLE II  
SETTING OF PARAMETERS IN SIMULATIONS

Parameter	Value	Parameter	Value
$l$	30 m	$v_G$	10 m/s
$X_{max}$	1200 m	$N_s$	100
$Y_{max}$	1200 m	$N_b$	50
$Z_{max}$	90 m	$N_c$	1000
$N_D$	3	$N_m$	1000
$N_G$	2	$p_m$	0.3
$v_D$	10 m/s	$T_{max}$	100

To sum up, the proposed online adjustment strategy for the circular paths can cope with different situations when the requirements of the persistent surveillance task are changed. The approach is simple and effective and has a small influence on the existing circular paths. The efficiency of the persistent surveillance task is also considered.

## VI. SIMULATION STUDIES

To investigate the rationality of the established cooperative path planning model and the superiority of the hybrid EDA-GA-based algorithm in the persistent surveillance task performed by the UAVs & UGVs system, three groups of simulations are conducted. In the first simulation, three drones and two UGVs are assigned to perform the persistent surveillance task cooperatively, and the results are compared to the case when the drones and the UGVs perform the task independently. The results of cooperative path planning with different number of drones and UGVs are also presented and discussed. In the second scenario, the EDA-based algorithm and the GA-based algorithm are also applied to solve the same problem, respectively, and the comparisons among the three algorithms are made. The online adjustment strategy on the circular paths is verified in the last group of simulations. In all of the above simulations, the distributions of buildings and obstacles in urban environments are the same. Some 2-D grids are set to have special requirements on the shooting resolution, and some 2-D grids are inaccessible, as shown in Fig. 9.

Besides, the values of parameters in the simulations are presented in Table II.

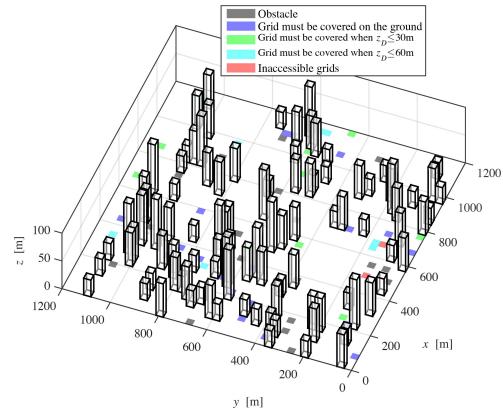


Fig. 9. Settings of urban environments and the persistent surveillance task in simulations.

In Table II, the boundaries of urban environments are set considering the flight endurance of drone, and the length of an edge in a cube is set considering that the drone could not make turns too frequently. Assume that there are three drones and two UGVs available currently, and their velocities are given when they are under typical work modes. The parameters in the hybrid EDA-GA algorithm are determined considering the scale of the problem. Actually, the values of the above parameters can be adjusted when facing different cases and employing different types of drones and UGVs. When applying the hybrid EDA-GA algorithm, the percentages of reserving the solutions obtained by EDA and GA in each iteration will vary in different phases of the iterative process, and the values of  $w$  are set according to

$$w = \begin{cases} 0.7, & t < T_{max}/3 \\ 0.5, & T_{max}/3 \leq t \leq 2 \cdot T_{max}/3 \\ 0.3, & t > 2 \cdot T_{max}/3. \end{cases} \quad (11)$$

In (11), the whole iterative process is divided into three phases evenly. In the first phase, the solutions obtained by EDA are paid more attention, and  $w$  is set to 0.7 to enhance the global search ability. In the middle of the iterative process, the solutions obtained from EDA and GA are equally treated, so  $w$  is set to 0.5. In the last phase, local search is more important, and  $w$  is set to 0.3 to focus more on the solutions obtained by GA.

The simulations are conducted under the environment of MATLAB (version R2016b). All the results are obtained by running the programs on a desktop with Intel Xeon CPU E5-1630 3.70 GHz.

### A. Results of Path Planning for the UAVs & UGVs System in the Persistent Surveillance Task

The circular paths of the drones and the UGVs obtained by the hybrid EDA-GA-based algorithm are shown in Fig. 10.

In general, the travel paths of the drones and UGVs follow the pattern in Fig. 6(a), and the vehicles must take a detour when facing the buildings and obstacles. In Fig. 10(a), the drones fly at the height of 90 m most of the time to expand the vision field. They also change the flight height during the travel to cover the 2-D grids, which have special requirements on the shooting resolution. The UGVs can pass all the 2-D

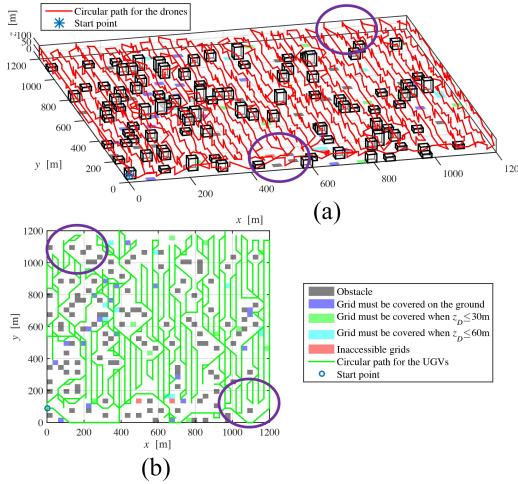


Fig. 10. Circular paths of the UAVs & UGVs system. (a) In the air and (b) on the ground.

grids which must be covered on the ground, and the inaccessible 2-D grids are also avoided. Besides, the UGVs travel more distance in the circular path to help the drones cover the 2-D grids, thus balancing the workload in the air and on the ground and improving the efficiency of performing the persistent surveillance task. To make a comparison, the operation mode that the drones and the UGVs work independently is applied. The UGVs are only responsible for the 2-D grids that must be covered on the ground, and the rest 2-D grids are allocated to the drones. By running the hybrid EDA-GA-based algorithm again, the circular paths are presented in Fig. 11, and the results corresponding to the two operation modes are summarized in Table III.

In Figs. 10 and 11, the areas circled by purple are the differences of the two figures for examples. Compared to the results in Fig. 10, the drones fly more distance in the independent mode, and the UGVs travel less distance, which is consistent with the common sense according to their workloads. In the independent mode, the time gap of traveling the circular path for the drones and the UGVs becomes greater, which leads to a larger index value. In the cooperative mode, the elapsed time of taking a round of the circular path for the drones and the UGVs is close, which means that the workload of the persistent surveillance task can be made an equal distribution approximately. The above results demonstrate that the established cooperative path planning model is rational for the persistent surveillance task considering the different requirements of the 2-D grids on the ground, and the cooperation between the drones and the UGVs can enhance the efficiency of performing the persistent surveillance task.

To take a further consideration of the circular path shown in Fig. 11(b), the UGVs travel the redundant distance as the goal is only to pass the 2-D grids that must be covered on the ground. It indicates that the established cooperative path planning model is not so effective to deal with the subproblem, and the reason can be explained as follows. In the subproblem, only 20 2-D grids need to be covered by the UGVs, which is not a large-scale optimization problem. The subproblem can be solved by obtaining the combination of points that can cover

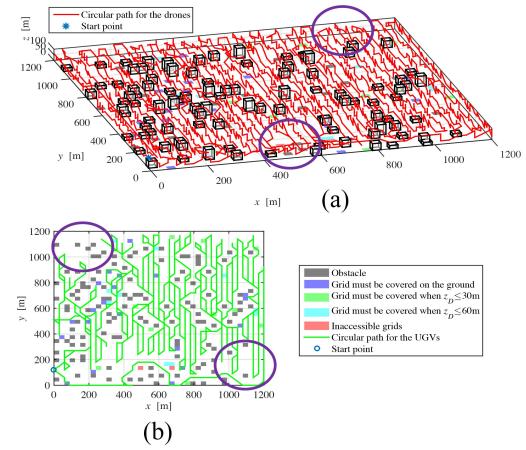


Fig. 11. Circular paths of the drones and the UGVs when they work independently. (a) In the air and (b) on the ground.

TABLE III  
INFORMATION OF THE CIRCULAR PATHS AND THE INDEX  
VALUES FOR THE TWO OPERATION MODES

Drones and UGVs	Length of circular path		Travel time of each vehicle		Index value
	Air	Ground	Air	Ground	
Cooperative mode	74733.8m	50269.3m	2491.1s	2513.4s	2513.4s
Independent mode	90216.1m	41505.6m	3007.2s	2075.2	3007.2s

TABLE IV  
RESULTS OF PATH PLANNING WITH DIFFERENT NUMBERS OF DRONES  
AND UGVs

Drone	UGV	Length of circular path		Travel time of each vehicle		Index value
		Air	Ground	Air	Ground	
3	2	74733.8m	50269.3m	2491.1s	2513.4s	2513.4s
5	2	114292.8m	45954.1m	2285.9s	2297.7s	2297.7s
7	2	133594.0m	44668.0m	1908.5s	2233.4s	2233.4s
3	4	58260.6m	60705.3m	1942.0s	1517.6s	1942.0s
5	4	68625.2m	54902.1m	1372.5s	1372.5s	1372.5s
7	4	81383.7m	46776.5m	1162.6s	1169.4s	1169.4s

the 20 2-D grids using the function  $f_{cov}$ , and the combination of points can be further optimized. There is no need to consider the on-off states of all the active points on the ground, which unnecessarily increases the number of variables to be optimized and the difficulty of searching the optimal solution. The above analysis in turn shows that the established cooperative path planning model is suitable to reflect the characteristic of the persistent surveillance task.

Note that the types of drones and UGVs are not specified in this study. In real practices, the circular paths can be generated by the proposed method after the types, the number, and the velocities of drones and UGVs are given, and the travel time of drones and UGVs in every cycle can be further reduced by employing more vehicles. However, the number of drones and UGVs are restricted by the current available vehicles and the comprehensive consideration on the economic benefit. Next, the results of path planning with different number of drones and UGVs are presented in Table IV to show a more complete perspective of the proposed method.

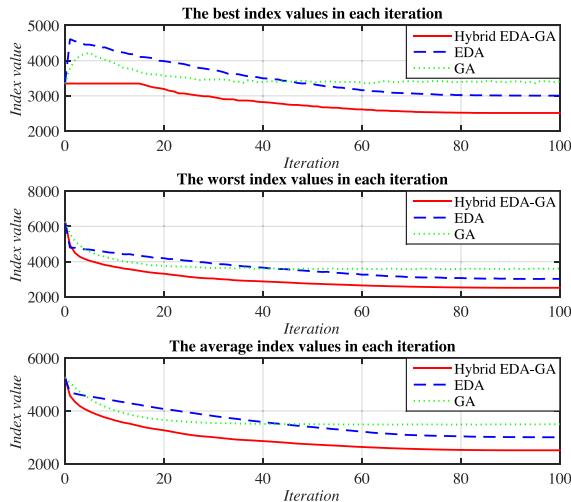


Fig. 12. Best, worst, and average index values in each iteration for the three algorithms.

In general, the length of circular path changes with the increasing number of drones or UGVs both in the air and on the ground, and the travel time of vehicles and the index value are all reduced. It demonstrates that in the persistent surveillance task, each vehicle can travel shorter time in every cycle by increasing the number of vehicles. When the number of UGVs is fixed (2 or 4), the length of circular path in the air increases with a greater number of drones while the length of circular path on the ground decreases. Similarly, when the number of drones is fixed (3, 5, or 7), the length of circular path in the air and on the ground decreases and increases, respectively, with a greater number of UGVs. The above analysis shows that the workload of drones and UGVs can be adjusted adaptively with different numbers of vehicles, which can always balance the workload between the drones and UGVs. Besides, the total workload of the UAVs & UGVs system can be reduced with a greater number of drones and UGVs in general, which results in a shorter time of finishing a complete coverage. To sum up, the number of drones and UGVs will have an influence on the efficiency of the cooperative persistent surveillance task.

#### B. Comparison Among the Hybrid EDA-GA Algorithm, EDA, and GA

To verify the superiority of the developed hybrid EDA-GA algorithm in solving this problem, EDA and GA are both applied to generate the circular paths with the same settings in Section VI-A, and three drones and two UGVs are still employed. To make a fair comparison, the three algorithms begin with the same initial solutions, and the convergence curves are presented in Fig. 12.

In Fig. 12, the hybrid EDA-GA algorithm performs the best in all of the three subgraphs. EDA has a better performance than GA because the solutions are evolved at the macrolevel, and all the optimization variables can be changed with certain probabilities, which is more suitable to solve the large-scale optimization problem. In GA, the changes imposing on the optimization variables are local, and only a small part of them

TABLE V  
RESULTS OF THE THREE ALGORITHMS

Algorithm	Length of path (D)	Length of path (G)	Best	Worst	Average
EDA	89707.1m	60117.3m	3005.9	3018.6	3006.3
GA	102260.3m	60200.4m	3408.7	3618.0	3502.1
EDA-GA	74733.8m	50269.3m	2513.4	2513.4	2513.4

are selected to execute the crossover and the mutation operations with certain probabilities. In a large-scale optimization problem, local variations may not be so effective to lead to the improvement of the solutions without the guide from the excellent solutions.

The advantages of EDA and GA are integrated into the hybrid EDA-GA algorithm, and they are paid different attention over the iterative process based on the preference on the global and local search. In the first few iterations, the best index values of the EDA and the GA are increased compared to the initial solutions due to the fact that there is no mechanism in EDA or GA to reserve the excellent solutions, and the new solutions are not guaranteed to be better than the old ones. In the hybrid EDA-GA algorithm, the excellent solutions are reserved in the branch of GA, so the best index value stay the same in the first dozen of iterations. It is beneficial for accelerating the convergence rate and resulting in a better solution. Besides, the results of the three algorithms after 100 times of iterations are listed in Table V.

In Table V, the letters "D" and "G" denote the drones and the UGVs, respectively, and the length of paths refer to that of the circular paths corresponding to the best solutions of the three algorithms. The final solutions obtained by GA show the biggest difference, which demonstrates that the evolution process in GA is mainly imposed on the individual rather than on the swarm, and the stability of solutions is not so good. The situation is improved in EDA as the solutions are evolved integrally. The index values of many solutions are close to the best value. In the hybrid EDA-GA algorithm, all the final solutions in the swarm converge to the best value, which shows its strong search ability at the macrolevel.

To sum up, the proposed hybrid EDA-GA algorithm is superior to EDA and GA in terms of the accuracy and stability of the solution as it incorporates their advantages and shows the powerful global and local search ability in the right phase of the iterative process.

#### C. Results of Online Adjustment for the Circular Paths

In this simulation, the locations of inaccessible grids are set to be changed when the UAVs & UGVs system is performing the persistent surveillance task. The modified circular paths for the drones and the UGVs are shown in Figs. 13 and 14, respectively, using the proposed online adjustment strategy.

It can be seen from Figs. 13 and 14 that the original circular paths are changed only in a few local regions (near the inaccessible 2-D grids, which are circled by green), and the vehicles can avoid the new inaccessible grids and cover the old inaccessible grids (they must be covered now) successfully while still realizing a complete coverage conforming to

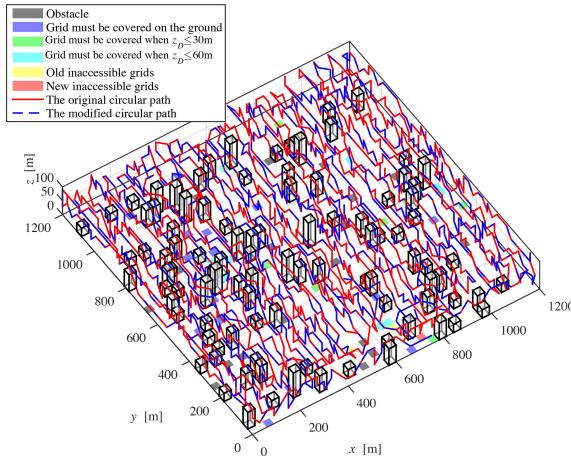


Fig. 13. Circular paths in the air before and after the online adjustment.

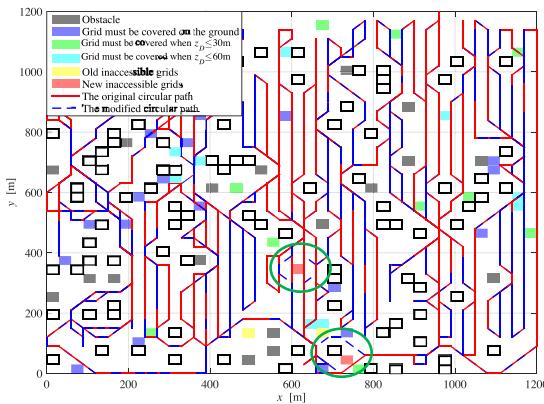


Fig. 14. Circular paths on the ground before and after the online adjustment.

TABLE VI  
COMPARISON BETWEEN THE ORIGINAL AND THE  
MODIFIED CIRCULAR PATHS

Circular path	Length of path (D)	Length of path (G)	Index value
Original	74733.8m	50269.3m	2513.4
Modified	74483.8m	50506.3m	2525.3

the new requirements. To further compare the original circular paths with the modified ones, the important information is summarized in Table VI.

With the proposed online adjustment strategy, the length of the circular path in the air decreases and the length of the circular path on the ground has a slight increase. Besides, the index value only has a small growth. It can be learned from the established model of the coverage range for the drone and the UGV that the change of inaccessible 2-D grids has a greater influence on the circular path of the drones than that of the UGVs because there are more points in the air whose on-off states are related to the accessibility of the 2-D grids. To sum up, the online adjustment is effective in satisfying the new requirements of the persistent surveillance task and only has a small influence on the existing circular paths by carrying out the modification in the local regions.

## VII. CONCLUSION

The cooperative path planning problem of the UAVs & UGVs system for the persistent surveillance task in urban environments is studied in this article. Literature investigation shows that the complete coverage path planning problem in the persistent surveillance task is rarely considered, especially when it is performed by heterogeneous vehicles. UGVs are introduced to cooperate with drones when there are special requirements on the shooting resolution, or the ground area is blocked by the buildings, thus improving the efficiency of performing the task.

First, the low-altitude airspace is modeled as a number of cubes with the same size. Under this framework, the travel rules for the drone and the UGV are defined, and the models of coverage range for the drone and the UGV considering the different flight heights and the blocking of buildings are built. The problem is formulated into a large-scale 0-1 optimization problem in which the on-off states of active points need to be optimized. The goal is to find the circular paths for the UAVs & UGVs system that can minimize the travel time of realizing a complete coverage.

To solve the problem, a hybrid EDA-GA algorithm is proposed integrating the advantages of both algorithms in the global and local search. The simple sweep-based approach is used to determine the circular paths for the drones and the UGVs. Considering the change of inaccessible 2-D grids during the task, the local adjustment is conducted to remove the open points covering the inaccessible 2-D grids and add new points to cover the extra 2-D grids.

In the simulation studies, the rationality and superiority of the hybrid EDA-GA-based cooperative path planning method are verified. The UAVs & UGVs system can balance the workload among the vehicles, and a greater number of drones and UGVs can further reduce the index value. The hybrid EDA-GA algorithm performs much better than EDA and GA in terms of obtaining the solution with higher quality and stability. When the state of the ground area changes, the local adjustment strategy is effective in modifying the circular paths while minimizing the influence on the circular paths and the index value. In the future, some possible applications, such as the campus patrol, the search and rescue in disaster, and the surveillance in the wildlife refuge, can be considered based on the established model and the proposed cooperative path planning algorithm.

## REFERENCES

- [1] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.
- [2] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 198–212, May 2018.
- [3] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, and D. Wu, "Offline and online search: UAV multiobjective path planning under dynamic urban environment," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 546–558, Apr. 2018.
- [4] H. Hu, Y. Wu, J. Xu, and Q. Sun, "Cuckoo search-based method for trajectory planning of quadrotor in an urban environment," *Proc. Inst. Mech. Eng. G, J. Aerosp. Eng.*, vol. 233, no. 12, pp. 4571–4582, 2019.

- [5] C. Hu, Z. Zhang, N. Yang, H. S. Shin, and A. Tsourdos, "Fuzzy multiobjective cooperative surveillance of multiple UAVs based on distributed predictive control for unknown ground moving target in urban environment," *Aerospace Sci. Technol.*, vol. 84, pp. 329–338, Jan. 2019.
- [6] E. Arribas, V. Mancuso, and V. Cholvi, "Coverage optimization with a dynamic network of drone relays," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2278–2298, Oct. 2020, doi: [10.1109/TMC.2019.292735](https://doi.org/10.1109/TMC.2019.292735).
- [7] S. Goudarzi, N. Kama, M. H. Anisi, S. Zeadally, and S. Mumtaz, "Data collection using unmanned aerial vehicles for Internet of Things platforms," *Comput. Electr. Eng.*, vol. 75, pp. 1–15, May 2019.
- [8] J. Xie, L. R. G. Carrillo, and L. Jin, "An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 67–72, Jan. 2019.
- [9] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation," *J. Intell. Robot. Syst.*, vol. 97, no. 1, pp. 125–139, 2020.
- [10] L. Geng, Y. F. Zhang, J. J. Wang, J. Y. Fuh, and S. H. Teo, "Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms," in *Proc. 10th IEEE Int. Conf. Control Autom. (ICCA)*, Hangzhou, China, Jun. 2013, pp. 828–833.
- [11] C. Song *et al.*, "Design and test of centrifugal disc type sowing device for unmanned helicopter," *Int. J. Agric. Biol. Eng.*, vol. 11, no. 2, pp. 55–61, 2018.
- [12] H. Freimuth and M. König, "Planning and executing construction inspections with unmanned aerial vehicles," *Autom. Const.*, vol. 96, pp. 540–553, Dec. 2018.
- [13] K. Harikumar, J. Senthilnath, and S. Sundaram, "Multi-UAV oxyrrhis marina-inspired search and dynamic formation control for forest fire-fighting," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 863–873, Apr. 2019.
- [14] H. Qin *et al.*, "Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1339–1350, Feb. 2019.
- [15] S. Minaelian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.
- [16] S. Zhang, H. Wang, S. He, C. Zhang, and J. Liu, "An autonomous air-ground cooperative field surveillance system with quadrotor UAV and unmanned ATV robots," in *Proc. IEEE 8th Annu. Int. Conf. CYBER Technol. Autom. Control Intell. Syst. (CYBER)*, Tianjin, China, Jul. 2018, pp. 1527–1532.
- [17] T. Yang, Z. Jiang, R. Sun, N. Cheng, and H. Feng, "Maritime search and rescue based on group mobile computing for unmanned aerial vehicles and unmanned surface vehicles," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7700–7708, Dec. 2020.
- [18] P. B. Sujit and S. Saripalli, "An empirical evaluation of co-ordination strategies for an AUV and UAV," *J. Intell. Robot. Syst.*, vol. 70, nos. 1–4, pp. 373–384, 2013.
- [19] Y. Wu, K. H. Low, and C. Lv, "Cooperative path planning for heterogeneous unmanned vehicles in a search-and-track mission aiming at an underwater target," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6782–6787, Jun. 2020.
- [20] Y. Wu, "Coordinated path planning for an unmanned aerial-aquatic vehicle (UAAV) and an autonomous underwater vehicle (AUV) in an underwater target strike mission," *Ocean Eng.*, vol. 182, pp. 162–173, Jun. 2019.
- [21] J. Chen, X. Zhang, B. Xin, and H. Fang, "Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 959–972, Apr. 2016.
- [22] A. Lakas, B. Belkhouche, O. Benkraouda, A. Shuaib, and H. J. Alasmawi, "A framework for a cooperative UAV-UGV system for path discovery and planning," in *Proc. IEEE Int. Conf. Innovat. Inf. Technol. (IIT)*, Al Ain, UAE, Nov. 2018, pp. 42–46.
- [23] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016.
- [24] Q. Luo and H. Duan, "An improved artificial physics approach to multiple UAVs/UGVs heterogeneous coordination," *Sci. China Technol. Sci.*, vol. 56, no. 10, pp. 2473–2479, 2013.
- [25] Z. Kashino, G. Nejat, and B. Benhabib, "Aerial wilderness search and rescue with ground support," *J. Intell. Robot. Syst.*, vol. 99, pp. 147–163, Nov. 2019.
- [26] P. Tokekár, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016.
- [27] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, "Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10483–10495, Dec. 2019.
- [28] M. Chen, Y. Chen, Z. Chen, and Y. Yang, "Path planning of UAV-UGV heterogeneous robot system in road network," in *Proc. Int. Conf. Intell. Robot. Appl.*, Aug. 2019, pp. 497–507.
- [29] F. Ropero, P. Muñoz, and M. D. R.-Moreno, "TERRA: A path planning algorithm for cooperative UGV-UAV exploration," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 260–272, Feb. 2019.
- [30] S. Seyedi, Y. Yazıcıoğlu, and D. Akşaray, "Persistent surveillance with energy-constrained UAVs and mobile charging stations," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 193–198, 2019.
- [31] J. Li, Y. Xiong, J. She, and M. Wu, "A path planning method for sweep coverage with multiple UAVs," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8967–8978, Sep. 2020.
- [32] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.
- [33] Y. Kim, D. W. Gu, and I. Postlethwaite, "Real-time optimal mission scheduling and flight path selection," *IEEE Trans. Autom. Control*, vol. 52, no. 6, pp. 1119–1123, Jun. 2007.
- [34] B. Alidaee, H. Wang, and F. Landram, "A note on integer programming formulations of the real-time optimal scheduling and flight path selection of UAVs," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 839–843, Jul. 2009.
- [35] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.
- [36] Y. Sun, G. G. Yen, and Z. Yi, "Improved regularity model-based EDA for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 662–678, Oct. 2018.
- [37] M. Xu, G. Feng, Y. Ren, and X. Zhang, "On cloud storage optimization of blockchain with a clustering-based genetic algorithm," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8547–8558, Sep. 2020.