Route coordination of UAV fleet to track a ground moving target in search and lock (SAL) task over urban airspace

Yu Wu, Kin Huat Low

Abstract—**Drone has become more and more popular in various civil applications due to the open of the low-altitude airspace and its easy operation. Unlike the common search and track task for the target, the new search and lock (SAL) task is focused on in this paper. In the SAL task, multiple drones first try to detect the moving ground target cooperatively. Then they must lock the target by covering all the surrounding area of it at a low flight altitude, which indicates that the target can be watched clearly in all directions from then on. The SAL task can be applied in the panorama shot for the moving ground target. First, the low-altitude urban airspace is discretized into cubes, based on which the flight rules of drone are defined. The field of view (FOV) of drone is modeled considering the flight altitude and the block of buildings. For the cooperation among multiple drones, the constraints on the type of waypoint, the communication distance and the collision avoidance are all included. The goal in the search phase is to cover more area which have not been visited recently to increase the probability of detecting the target, and it is expected to lock the target as soon as possible in the lock phase. A new swarm-based imitative learning optimization (SBILO) algorithm is proposed to determine the waypoint of drone in the search phase considering the characteristic of the established SAL model. To have a quick response to the escape behavior of the target in the lock phase, the waypoint of drone is generated in a distributed way to cover more surrounding area of the target and lock it gradually. The case of losing the target in the FOV of all drones is also addressed by covering more possible places where the target may appear. Simulation results demonstrate that the SAL task can be performed efficiently by the drones with the flight routes obtained by the proposed SBILO algorithm and the distributed asynchronous decision-make (DADM) approach.**

Index Terms—**drone; low-altitude airspace; search and lock; urban environments; swarm-based imitative learning optimization algorithm; distributed asynchronous decision-making approach**

Yu Wu is with the College of Aerospace Engineering, Chongqing University, Chongqing 400044, China, and also with the Air Traffic Management Research Institute (ATMRI), Nanyang Technological University, Singapore 637460, Singapore (e-mail: cquwuyu@cqu.edu.cn).

Kin Huat Low is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore, and also with the Principal Investigator of UAS Programme, ATMRI, Nanyang Technological University, Singapore 639798, Singapore (e-mail: mkhlow@ntu.edu.sg).

I. Introduction

Last decade witnesses the growth and prosperity of unmanned aerial vehicle (UAV), or known as drone. UAVs have increasingly been used in many fields, both in military and civil applications. The drone delivery can avoid the crowded ground traffic and realize a more accurate service [1], and the emergency can be discovered by the drone in time and activate the rescue [2]. In a search and track task, the drone has a wider view angle to cover more area and watch the object on the ground after installed a camera [3] [4]. In the above tasks, the capacity of a single drone is limited due to the endurance of battery and payload [5], and the cooperation among multiple drones can enhance the efficiency and reliability of performing the task.

There are some differences between the delivery (or rescue) and the search (or track) task. To be specific, the destination in a delivery or rescue task is fixed, and the drone only needs to fly to the known point. However, the ground target in the search and track task is usually moving with incomplete information, which adds the difficulty of success [6] [7]. Besides, the area coverage problem is often involved to increase the probability of finding the ground target [8]. To track the movement of the ground target, the real-time position of it is evaluated to improve the track performance [9] [10].

In some situations, the moving target is required to be surrounded and watched from all directions at all times. In this case, the tracking of the ground target is not sufficient to satisfy this demand [11]. To ensure the target is effectively locked, a group of coordinated drones according to a specific pattern would need to be deployed to surround and follow the moving target. For example, when the panorama shot for a moving target is conducted, multiple drones must surround it to record the complete information [12]. Besides, the strategy for the drone coordination should be updated due to the target lost caused by the blocking of building.

To this end, the flight route coordination problem for the search and lock (SAL) task towards the moving ground target is studied in this paper. First, the drones fly with the aim of covering more ground area, which can increase the probability of searching the unknown target. After one drone has detected the target within its field of view (FOV), the current position of the target will be broadcasted to other drones, and they will come over to lock the target cooperatively. The term ‘lock’ means the target is surrounded by the drones at a low flight altitude, and its future position is always within the FOV of drones. Due to the block of buildings and the escape behavior of the target, the target may be without the FOV of all drones at a time again, which causes the re-search process and adds the difficulty of locking the target.

In this study, the low-altitude urban airspace is divided into cubes with the same size borrowing the idea of 'AirMatrix' [13] considering the requirement of air traffic management for drone operations [14], and the ground area is also gridded to match the structure of 'AirMatrix'. Under this framework, the mathematical model is established to formulate the SAL task into the standard optimization problems, and flight route optimization algorithms, which contains the search, lock and re-search process, are developed. The main contributions of this work are concluded as follows.

1) A new cooperative SAL model for the moving ground target is established. In this model, the waypoints of drones are treated as the optimization variables, and the distance between the drones are restricted considering the possible collision and the maximum communication distance. The cooperation among the drones is realized by selecting the same type of waypoint. In general, the drones are required to cover more 2D grids, but the details are different considering the characteristic of the search, lock and re-search process.

2) A new swarm-based imitative learning optimization (SBILO) algorithm is proposed to determine the waypoint of drone in the search phase. In the SBILO algorithm, the idea of clustering is utilized, which matches well with the different types of waypoints in the model. Besides, the three learning behaviors, i.e., learn from teacher, classmate and oneself, which form a complete learning process, are integrated.

3) A distributed asynchronous decision-make (DADM) approach is designed to lock the target. The drones are classified into two groups by whether the target is within their FOVs. The target is also regarded as an intelligent agent which always tries to escape the track of drones. The strategy of dealing with the case that the target is out of the FOV of all drones is developed utilizing the historical position to make the target detected again as soon as possible.

II. Related Works

There are typical applications of the search task in the real situation, and the goal is usually to maximize the coverage degree and target detection probability [15]. For example, multiple UAVs are deployed to search for the missing tourists in Ref. [16]. The location of tourist is estimated according to the changes of topographic features, weather conditions, and time, and a hybrid evolutionary algorithm which consists of the main part and subordinate part is proposed to optimize the path of each UAV. The specific migration and mutation operators are used to evolve a population of main solutions in the main part, and a problem-specific heuristic and tabu search method is combined with the subordinate part. In Ref. [17], the coverage search mission in a river region is considered, and the goal is to generate the path of UAV which can maximize the cumulative probability of finding a stationary target within the required time. The prior likelihood distribution of area importance is approximated by the Gaussian mixture model, and the positive/negative greedy method is proposed to expand or contract waypoints to meet the terminal time constraint. Sometimes, the target is required to be successfully detected within specific angles, and this problem is solved in Ref. [18]. A target-feature-information-based disintegration method is designed to divide the search space into a set of cubes, and a Kuhn-Munkres-algorithm-based path planning method using real-time probability map is proposed for UAVs to traverse the cubes.

In the target tracking problem, many efforts have been made to establish the reasonable model to predict the movement of the target, and the action of UAV must be determined quickly to satisfy the real-time requirement [19]. In Ref. [20], the models of the sensor coverage area, line of sight and collision avoidance are depicted, and a modified grey wolf optimization (GWO) algorithm which combines with the Gaussian estimation of distribution (GED) strategy is proposed. Gauss probability model is used to estimate the distribution of the selected superior individuals and change the weighted mean to adjust the search directions. A two-step strategy, i.e., the coarse-to-fine deep scheme is proposed in Ref. [21] to deal with the aspect ratio variation in the UAV tracking problem. An initial estimate for the target is first produced by the coarse-tracker, and then a sequence of actions is learned to fine-tune the four boundaries of the bounding box. The two trackers are designed to have different action spaces and operating target, and they are trained jointly by sharing the perception network with an end-to-end reinforcement learning architecture. To guide the behavior of UAV target tracking, an improved deep deterministic policy gradient algorithm is presented in Ref. [22]. The reward function is based on the line of sight and artificial potential field, and multiple UAVs are controlled by the same policy network to perform tasks in each episode. Besides, the long short-term memory networks are used to improve the approximation accuracy and the efficiency of data utilization.

In other works, the search and track task are both considered [23] [24], and some situations which are similar to the search and track scenario, i.e., the moving target is involved in the path planning problem, are also attracted attentions [25] [26]. The search and track mission for an underwater target is addressed in Ref. [27], and this mission is performed by a UAV, an unmanned surface vehicle (USV) and an autonomous underwater vehicle (AUV). Strategies based on random simulation experiments and asynchronous planning are developed in the search and track phase respectively, and the paths of vehicles are generated by an improved particle swarm optimization algorithm with a centralized or a distributed mode. In Ref. [28], an Oxyrrhis Marina-inspired search and dynamic formation control framework for UAVs is established to search and neutralize a dynamic target (forest fire) in an uncertain environment. In the target identification stage, each UAV choose one of the three actions, i.e., Levy flight, Brownian search, and directionally driven Brownian, and the selection is based on the available sensor information about the possible fire location. In the mitigation stage, the UAVs fly in a dynamic formation to quench the fire using water. UAV and human can collaborate to search for the escaped criminals, and the aim is to minimize the expected time of capture. Considering that the criminals will attempt to avoid detection and capture, a hybrid evolutionary algorithm including three evolutionary operators, i.e., comprehensive learning, variable mutation, and local search, is proposed in Ref. [29] to explore the solution space efficiently.

In most of the above studies, the UAVs are thought to be able to communicate with each other, which ignores the constraint on the maximum communication distance [30]. Moreover, the goal of target detection should not always be maximizing the coverage area [31], and the time interval of visiting certain area also should be valued. Sometimes, tracking the ground target cannot satisfy the requirement of surrounding the target, and the possibility of losing the target in the tracking process is not considered [32].

III. Mathematical Model of the Search and Lock Task Towards the Moving Ground Target

The SAL task is performed in the low-altitude urban airspace. With the gridded airspace structure, the models for the flight rules and FOV of a single drone are defined. Then the cooperation mechanism among the drones, such as the collision avoidance, communication distance and waypoint selection, are established. The motion of the ground target is also described considering its different behaviors in the search and lock phases. At last, the specific constraints and the goals in the SAL task are formulated respectively.

A. Models for a Single Drone Flying and Multiple Drones' Cooperation

The drone is assumed to located at one vertex of a grid in AirMatrix, and the possible waypoints for the drone in the next time step is expressed in Fig. 1.

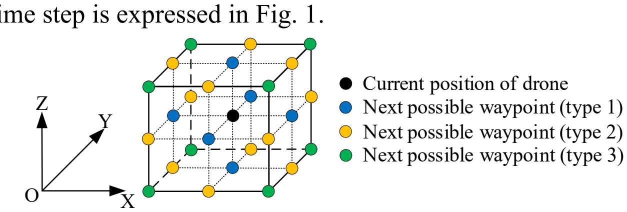

Fig. 1. The flight rules for a single drone

In Fig. 1, the global coordinate system OXYZ is established to describe the locations of grid and drone, and there are eight neighboring grids for a drone. The drone can choose any one neighboring vertex as the next waypoint. Assume that the drone flies with a constant velocity, the mathematical model of the flight rules is expressed as

$$\begin{cases} x_i(t+1) = x_i(t) + C_{xi}(t)l \\ y_i(t+1) = y_i(t) + C_{yi}(t)l, \quad i = 1,2,\dots,N_D \\ z_i(t+1) = z_i(t) + C_{zi}(t)l \end{cases} \tag{1}$$

where $(x_i(t), y_i(t), z_i(t))$ is the position of drone i at the tth time step, N_D is the number of drones, and l is the side length of the cube. $C_{xi}(t)$, $C_{yi}(t)$ and $C_{zi}(t)$ are the control variables along axes OX, OY and OZ respectively, which can be selected from the set U={-1, 0, 1}. Note that the concept of time step is introduced to denote the time information of each waypoint in Eq. (1), which is more convenient to record the state of drone, and the accurate time information of each waypoint also can be calculated after the velocity of drone is given. Besides, the next possible waypoints are classified into three types according to their distance to the current position of drone, and the definition on the type of waypoint is defined as

$$P_j = |C_{xi}(t)| + |C_{yi}(t)| + |C_{zi}(t)|, j = 1,2,\dots,27 \tag{2}$$

There are 27 alternative waypoints in the next time step, and four types of waypoints are defined, i.e., type 0, 1, 2 and 3. Note that $P_j = 0$ means the drone is hovering in the current position. In the low-altitude urban airspace, certain grids are occupied by the buildings, which make some vertexes of grids unreachable. Assume that the boundaries of the considered low-altitude urban airspace along each direction are x_{min}, x_{max}, y_{min}, y_{max}, z_{min} and z_{max}, the grid and the vertex of grid can be coded as $G(g_x, g_y, g_z)(g_x = 1,2,\dots,\frac{x_{max}-x_{min}}{l}; g_y = 1,2,\dots,\frac{y_{max}-y_{min}}{l}; g_z = 1,2,\dots,\frac{z_{max}-z_{min}}{l})$ and $W(w_x, w_y, w_z)(w_x = 1,2,\dots,\frac{x_{max}-x_{min}}{l}+1; w_y = 1,2,\dots,\frac{y_{max}-y_{min}}{l}+1; w_z = 1,2,\dots,\frac{z_{max}-z_{min}}{l}+1)$. When a grid $G(a,b,c)$ is occupied the building, the corresponding vertexes $W(e,f,g)(e = a, a+1; f = b, b+1; g = c, c+1)$ is unreachable to ensure the flight safety of drone.

Assume that the camera is equipped in the center of the drone, the covered area is centrosymmetric with respect to the projection of the drone on the ground [33] [34]. The 2D grid on the ground is coded as $F(f_x, f_y)(f_x = 1,2,\dots,\frac{x_{max}-x_{min}}{l}; f_y = 1,2,\dots,\frac{y_{max}-y_{min}}{l})$, and the union of the covered 2D grids $F(n_x, n_y)(n_x = x_l,\dots,x_r; n_y = y_l,\dots,y_r$ can be denoted in Eq. (3) when the drone is located at the point (x_o, y_o, z_o). Fig. 2 is also given to show the coverage area when the drone is located at different altitudes.

$$\begin{cases} X_l = \max\{1, \frac{x_o - z_o}{l}\} \\ X_r = \min\{\frac{x_o + z_o}{l}, \frac{x_{max}-x_{min}}{l}\} \\ Y_l = \max\{1, \frac{x_o - z_o}{l}\} \\ Y_r = \min\{\frac{y_o + z_o}{l}, \frac{y_{max}-y_{min}}{l}\} \end{cases} \tag{3}$$

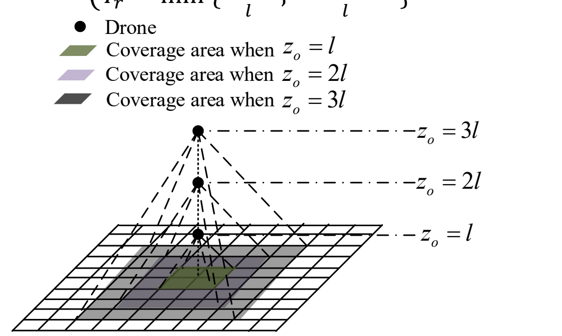

Fig. 2. Coverage area when the drone is located at different flight altitudes

In Eq. (3), x_l, x_r, y_l and y_r represent the codes of 2D grids on the boundaries of drone's FOV. Note that certain 2D grids will be blocked the building considering the FOV of drone, which will reduce the effective coverage area. It happens in cases b) and c) of Fig. 2, and two examples are provided in Fig. 3.

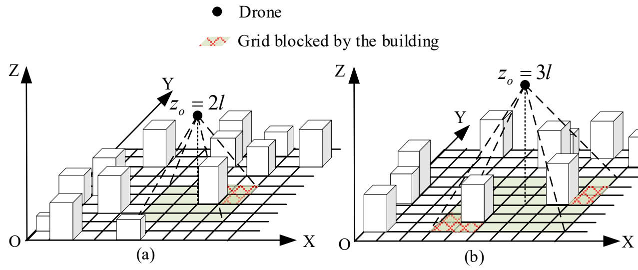

Fig. 3. FOV of a drone blocked by the building

The deployment of multiple drones performing the SAL task can increase the probability of searching and locking the target. As the flight velocity of drone is low and, all the drones are regarded as flying with the same velocity in this study. The drones mustn't collide with each other to ensure the flight safety. For two drones i_1 and i_2 $(i_1, i_2 \epsilon \{1,2,\dots,N_D\}; i_1 \neq i_2)$, the collision can be avoided when Eqs. (4) and (5) are satisfied simultaneously.

$$\left|x_{i_1}(t)-x_{i_2}(t)\right|+\left|y_{i_1}(t)-y_{i_2}(t)\right|+\left|z_{i_1}(t)-z_{i_2}(t)\right| \neq 0 \tag{4}$$

$$\left|\frac{x_{i_1}(t)+x_{i_1}(t+1)}{2}-\frac{x_{i_2}(t)+x_{i_2}(t+1)}{2}\right|+\left|\frac{y_{i_1}(t)+y_{i_1}(t+1)}{2}-\frac{y_{i_2}(t)+y_{i_2}(t+1)}{2}\right|+\left|\frac{z_{i_1}(t)+z_{i_1}(t+1)}{2}-\frac{z_{i_2}(t)+z_{i_2}(t+1)}{2}\right| \neq 0 \tag{5}$$

Eq. (4) denotes that the positions of drones cannot be the same in each time step. Besides, the line segments composed by the waypoints from two different drones in the two neighboring time steps cannot be crossed, or the collision will be happened at the intersection of the two line segments.

On the other hand, the drones cannot be too far away from each other, or the communication signal will be weakened. The drones can be treated as the vertices in a graph, and the communication between two drones is bidirectional, i.e., the drone can both send and receive the message with another one. When the distance between two drones is no greater than the maximum communication range (defined as d_{com}), they are linked by an edge. In this graph, the position of a drone can be known by other drones only when it is a connected graph.

As the drones are assumed to fly with the same velocity, their flying distance in each time step must be the same to coordinate their motions. In other words, the type of the next waypoint must be the same for all the drones, and this constraint is denoted in Eq. (6).

$$\begin{cases} |C_{xi}(t)|+|C_{yi}(t)|+|C_{zi}(t)|=0 (i=1,2,\dots,N_D) \\ or \\ |C_{xi_1}(t)|+|C_{yi_1}(t)|+|C_{zi_1}(t)|=|C_{xi_2}(t)|+|C_{yi_2}(t)|+|C_{zi_2}(t)| (i_1 \neq i_2) \end{cases} \tag{6}$$

Note that a drone can hover in the current position, and the 'type 0' waypoint is special can be seen as the same as any other type of waypoint.

In the SAL task, the drones are cooperative to cover the 2D grids, and some 2D grids may be covered by more than one drone at the same time step. Assume that the union of 2D grids covered by drone i at the position $(x_i(t), y_i(t), z_i(t))$ is $A_i(t)$, the sum of covered 2D grids at the time step t can be calculated by combining the results from all drones, as shown in Eq. (7).

$$A(t) = A_1(t) \cup A_2(t) \cup \dots A_i(t) \dots \cup A_{N_D}(t), i = 1,2,\dots,N_D \tag{7}$$

where $A(t)$ is the total covered 2D grids by all the drones at the time step t.

B. Motion Model of the Ground Target in Different Phases of the SAL Task

In this study, the ground target refers to the people who moves with a low velocity, such as the runner or the cyclist. The object moving with a greater velocity than the drone is not taken into account as it may be beyond the drone's ability to search and lock it. In a certain time step, the ground target is located in a 2D grid $F(o_x(t), o_y(t)) (o_x(t) = 1,2,\dots,\frac{x_{max}-x_{min}}{l}; o_y(t) = 1,2,\dots,\frac{y_{max}-y_{min}}{l})$, and the possible 2D grid which the ground target may be located in the next time step is shown in Fig. 4.

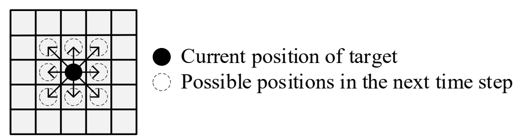

Fig. 4. The motion rules of the ground target

As the velocity of runner or cyclist is usually lower than the drone, the ground target is thought to change its located 2D grid in every two time steps, and the formula is presented in Eq. (8).

$$\begin{cases} o_x(t+1) = o_x(t), o_y(t+1) = o_y(t), & mod(t,2) = 0 \\ o_x(t+1) = o_x(t) + C_{tx}(t), o_y(t+1) = o_y(t) + C_{ty}(t), & mod(t,2) = 1 \end{cases} \tag{8}$$

where $C_{tx}(t), C_{ty}(t) \in \{-1,0,1\}$ and $mod(t,2)$ denotes the remainder that t is divided by 2.

In the search phase, the motion of target is random, and $C_{tx}(t)$ and $C_{ty}(t)$ can select a random number from the union {-1, 0, 1} if the 2D grid is not occupied by the build or beyond the boundary. While as the target has been aware of being detected in the lock phase, its motion is no longer random and always tends to escape being detected and locked by the drones, and a two-level cost function is set to describe the behavior of the target, as expressed in Eqs. (9) and (10).

$$(o_x(t+1), o_y(t+1)) = arg\left\{\max_{1 \le q \le S_p(t+1)} K_1(q)\right\} \tag{9}$$

$$(o_x(t+1), o_y(t+1)) = arg\left\{\max_{1 \le q \le S_p(t+1)} K_2(q)\right\} \tag{10}$$

where $S_p(t+1)$ is the number of feasible 2D grids that the target can locate in the next time step (the 2D grid which is occupied by the building or beyond the boundary is not feasible). $K_1(q)$ is the number of feasible 2D grids in time step t+2 if the qth feasible 2D grid is selected in time step t+1, and $K_2(q)$ is the distance between the drone (the average distance between the drones and the target) and the center of the qth feasible 2D grid for time step t+1. In general, after the target has been detected by the drones, it will tend to choose the 2D grid which can lead to the maximum number of feasible 2D grids in time step t+2. With more feasible 2D grids around the target, the probability of being locked will be reduced. However, the cost value in Eq. (9) may be the same for more than one feasible 2D grid, further judgement needs to be made to determine the location of target in the next time step. The target will prefer to go to the 2D grid which has the greatest distance from the drone to avoid being detected, as expressed in Eq. (10).

C. Constraints and the optimization goal in the SAL task

In the search phase, the drones should cover more 2D grids to increase the probability of detecting the target. On the other hand, the 2D grid which has not been visited by the drone for a long period should be paid more attention in case that the target appears in those grids. Having a comprehensive consideration of the above two aspects, the optimization goal in each time step of the search phase is expressed in Eq. (11).

$$J_1(t) = \max\{\sum_{c=1}^{\arg\{A(t)\}}(t - v_r)\} \tag{11}$$

where $\arg\{A(t)\}$ corresponds to the number of covered 2D grids by the drones, and v_r denotes the time step which the 2D grid *r* was visited last time. In Eq. (11), both the number and the visit time interval of the covered 2D grids are considered, and the goal is to maximize $J_1(t)$ in every time step of the search phase. The search phase will be ended when the target is within the FOV of one drone, as shown in Eq. (12).

$$(o_x(t), o_y(t)) \in A(t) \tag{12}$$

In the lock phase, the aim is to lock the target as soon as possible, as expressed in Eq. (13).

$$J_2(t) = \min\{t\} \tag{13}$$

The definition of locking the target is described in Fig. 5, and the nine 2D grids marked in Fig. 5 must be covered by the drones flying at the altitude *l*. The 'lock' action demonstrates that the target is watched by the drones from all directions, and it cannot be escaped any more in the future time steps.

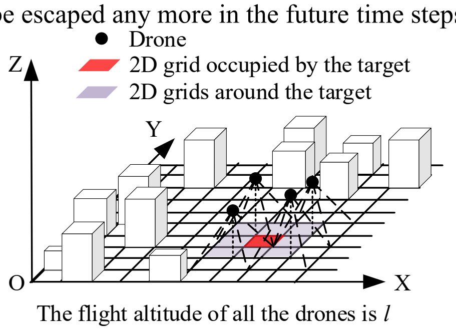

Fig. 5. The scene that the target is locked by drones

Besides, according to the model for a single drone's FOV, there must be at least four drones to realize the lock action if all the nearby eight 2D grids are feasible. When there is a building occupying one of the eight 2D grids around the target, the grid is also regarded as being covered because the target cannot go to that 2D grid in the next time step.

IV. Swarm-based Imitative Learning Optimization Algorithm for Target Searching

In this section, the waypoints of drones in the search phase are determined. It can be learned from the established model that the motion of the target has no influence on the waypoints of drones, and the drones need to maximize the value of Eq. (11) in each time step of search to increase the probability of detecting the target. Therefore, there is no dynamic factors affecting the results of the search phase, and the waypoints of drones can be determined in advance before performing the SAL task, i.e., an offline planning. Meta-heuristic algorithms are effective tools to deal with the optimization problem with complicated constraints and have been widely applied in many fields in recent years [35]. Brain storm optimization (BSO) and teach and learn-based optimization (TLBO) algorithms both belong to the meta-heuristic algorithms, and they summarize the behaviors of human in generating new idea and learning the knowledge, which is different from other algorithms inspired by the nature or the behaviors of animals. There are some similarities between the BSO and TLBO algorithms, which provides the possibility of combination. Besides, the idea of clustering is utilized in the BSO algorithm, which well matches the characteristic of the established SAL model that there are three types of waypoints. New algorithm combining the BSO and TLBO algorithm can make full use of their advantages to further improve the performance of algorithm.

A. Basic Principles of BSO and TLBO algorithms

BSO algorithm was first proposed by Shi in 2011 and was inspired by the human's behavior of generating the new ideas [36], i.e., good ideas are expected to be generated by discussion and borrowing from others' ideas. In the BSO algorithm, the individuals which are similar are classified into the same group, and further operations are made based on the clusters. In each iteration, one or two clusters are selected with a certain probability. When one cluster is selected, the mutation operator is conducted on one random individual in the cluster with certain probability. While when two clusters are selected, the crossover operator is conducted on two random individuals from different clusters with certain probability. Then only the better individual between the new and the corresponding old one can be remained. The above procedures are repeated in every iteration and will not be terminated until the maximum iteration times is reached. The best individual is outputted as the final solution.

TLBO algorithm was first proposed by Rao et al. in 2011, which imitates the students' learning process [37]. In each iteration, the best individual is regarded as the teacher, and other individuals are the students. For every student, the learning process is composed by two phases. First, the student learns from the gap between the teacher and the average level of all the students. In the second phase, the student learns from the gap between two randomly selected students. Note that in both phases, the updated student will compare with the corresponding old one, and only the better one can be reserved and enter the next iteration. The terminal condition in the TLBO algorithm is as the same as that in the BSO algorithm.

According to the above descriptions, the main characteristic of BSO and TLBO algorithms are summarized in TABLE I.

TABLE I
Comparison between BSO and TLBO algorithms

Operation	BSO	TLBO
Clustering mechanism	√	×
Mutation	√	×
Crossover (with the specified individual)	×	√
Crossover (with a random individual)	√	√
Imposed on every individual in each iteration	×	√
Imposed on the selected individuals in each iteration	√	×

In TABLE I, not all the operations are included in the BSO or TLBO algorithm. For a complete brainstorm or learning process, three parts should be involved, i.e., learning from the best individual, discussing with the companions and thinking by oneself. 'Learning from the best individual' has been included in the TLBO algorithm, which is equivalent to the crossover operation with the specified individual. 'Discussing with the companions' has the same content with the crossover operation with a specified individual and is contained both in the BSO and TLBO algorithms. As for 'thinking by oneself', it is equivalent to the mutation operation and appears only in the BSO algorithm. Based on the above information, a new SBILO algorithm integrating the idea of BSO and TLBO algorithms is developed, which can cover all the three parts in a complete learning process. Besides, the clustering mechanism in the BSO algorithm are executed in this problem to satisfy the constraint in Eq. (6) automatically. All the above considerations make the development of SBILO algorithm necessary and reasonable.

B. Operators and Procedures of the SBILO Algorithm for searching the target

Although the ideas in the BSO and TLBO algorithms are borrowed in the SBILO algorithm, the standard BSO and TLBO algorithms are both utilized in the continuous optimization problem initially. However, the values of control variables are selected from the union {-1, 0, 1} in the SAL task, so the operators in the SBILO algorithm need to be redefined to fit this discrete optimization problem. Assume that there are four drones performing the SAL task, the detail of each operator is described below.

1) Teaching operator

For each cluster, the individual should learn from the best one in this cluster, and this process is described in Fig. 6.

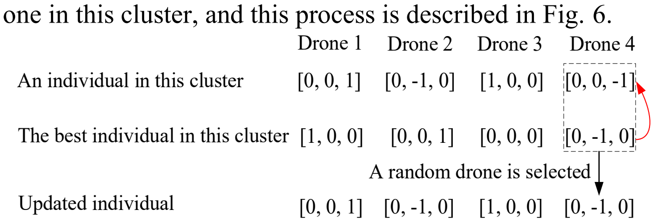

Fig. 6. Teaching operator: take type 1 waypoint as an example

In Fig. 6, a random drone is selected first, and the control variable of this drone is replaced by that from the best individual in the same cluster. Note the control variable [0, 0, 0] can belong to any type of waypoint, so it also joins the operation. It is probable that the updated individual is infeasible, i.e., the constraints in Eq. (4), Eq. (5) or d_{com} is violated. In this case, the value of $J_1(t)$ in Eq. (11) is set to 0 to denote the infeasibility of the individual.

2) Discussion operator

The discussion operator is conducted between two individuals, and the two individuals must be selected from the same cluster to keep the updated individuals meet the constraint in Eq. (6), which is different from the situation in the BSO algorithm. The detail is described in Fig. 7.

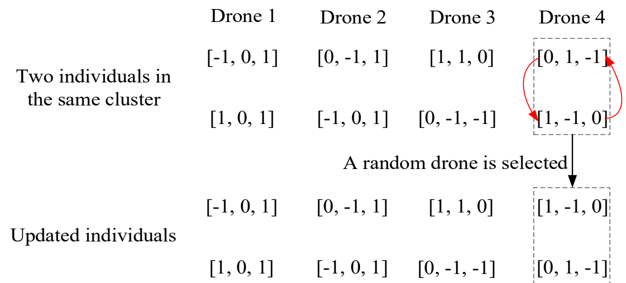

Fig. 7. Discussion operator: take type 2 waypoint as an example

A random drone is also selected first, and the control variable of the drone in the two individuals are exchanged. Only the better individual between the old and the updated one will be remained. If two individuals from different clusters join the discussion operator, the constraint in Eq. (6) will be violated.

3) Self-thinking operator

In the above two operators, the individual is updated by referring to the control variables from other individuals, in which the new control variable is not created. In the self-thinking operator, a random drone is selected, and the corresponding control variable is renewed by a random one with the same type, thus the individual is updated.

The three operators cover all the parts of a complete learning process, and the contents of these operators are designed considering the characteristic of the established SAL model.

The SBILO algorithm is utilized in each time step of the search phase to ensure that the value of $J_1(t)$ can always be maximized to increase the probability of detecting the target. In each iteration of the SBILO algorithm, the designed three operators are executed with a certain probability. The procedures of solving the target searching problem are summarized in Fig. 8.

In time step *t*, when utilizing the SBILO algorithm, it should be first checked whether a certain type of waypoint can be flown by all the drones, and only the 'public' type of waypoint is further considered to coordinate the motion of drones. Then *M* individuals belonging to the 'public' type of waypoint are initialized. In each iteration, for each 'public' type, a certain number of individuals are selected to execute the three operators with some probability, and it is repeated before the maximum iteration times (denoted as *Max_iter*) is reached. The best combination of the control variable for all the drones is obtained and applied to calculate the position of each drone, and the matrix recording the latest time step of each 2D grid being covered is also updated. If the target still has not been detected, the SBILO algorithm will be run again.

In the SBILO algorithm, the framework of BSO algorithm, i.e., the idea of clustering and the approach of selecting several individuals to update themselves, is utilized, and the operators in the BSO and TLBO algorithms are improved. From an overall perspective, the control variable of each drone is determined in a centralized way by the SBILO algorithm, i.e., all the control variables and constraints are considered simultaneously when determining the waypoints of drones in the next time step.

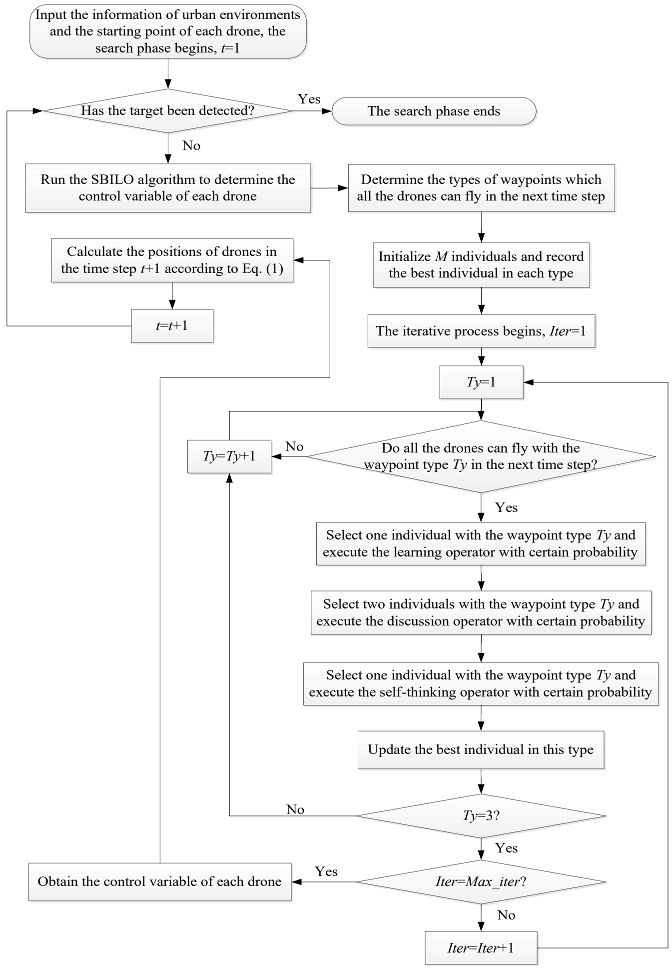

Fig. 8. The procedures of SBILO algorithm in the search phase

V. Distributed Asynchronous Decision-making Approach for Target Locking

After the target is detected by one of the drones, all the drones should fly cooperatively to lock the target. Different from the search phase, the waypoint of drone must be generated online to make a quick response to the motion of target in the lock phase. In this section, the general idea of the target locking algorithm is presented, and the drones are classified into different groups according to their status. Then the strategy of planning for each group of drones is developed, and the case of losing the target is also dealt with to contain all the possibilities.

A. Description of the Distributed Decision-making approach

The proposed SBILO algorithm is not suitable for determining the waypoint of drone in the lock phase as it will cost much time to optimize the control variables in the iterative process, which cannot meet the requirement of online computation. Moreover, as the roles of drones are different in the lock phase, they have different priorities when making a decision. Therefore, a distributed decision-making approach is designed to divide the problem into a series of sub-problems.

First, the drones are classified into two groups. The drones which have detected the target make up the pioneer group, and other drones compose the follower group automatically. Note that, the drones in the two groups are changed dynamically in each time step depending on whether the target is within the FOV of the drone. The drones in the pioneer group have the priority to determine their waypoints first because they are more important and are responsible for keeping the target within the FOV. Besides, they must try to transmit the current position of the target to the drones in the follower group to make them detect the target earlier. The drones in the pioneer group make decisions one by one, and the waypoints must be with the same type to coordinate the motion of drones. For the drones in the follower group, their waypoints must follow the type of those in the pioneer group, and they also make decisions one after another. The drones in the follower group know the current position of the target from the drones in the pioneer group and try to make the target within the FOV as soon as possible.

B. Strategy of Drone in Each Group When Determining the Waypoint in the Next Time Step

As the whole target locking problem has been decomposed into a certain number of subproblems, the strategy of drone when determining the next waypoint must be developed in a distributed way. The drones in the two groups fly for different purposes, and their strategies are described respectively.

1) Strategy of drone in the pioneer group

First, the case that only one drone having detected the target is considered. The primary goal of this drone is to cover more 2D grids surrounding the current position of target (as shown in Fig. 5) to increase the probability that the target is still within its FOV in the next time step. An example is given in Fig. 9.

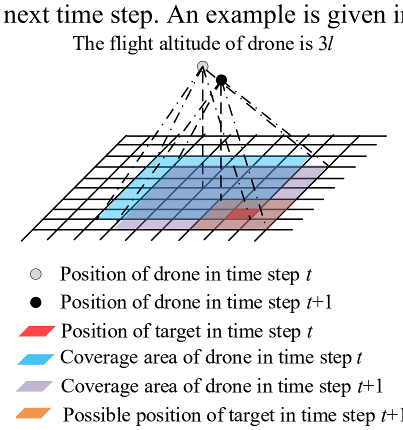

Fig. 9. The scene that the drone determines the waypoint in the next time step

In Fig. 9, the target has been detected by the drone in time step t. To keep the target still within its FOV, the drone should try the best to cover all the possible positions of the target in time step t+1. Note that if the possible position is out of boundary or occupied by the building, this position is also regarded as being covered as the target will not appear in those positions. Especially, when there are eight possible positions being covered, the position of target in time step t+1 still can be known even if it is not within the FOV of drone because the target must be in the remaining 2D grid that the drone does not cover. There is a probability that more than one position of waypoint can cover the same number of 2D grids in orange in Fig. 9, and the waypoint which is the nearest to the current position of target is chosen. Therefore, a two-level index is designed for the only drone having detected the target when determining the next waypoint.

When the target is within the FOV of more than one drone, the situation becomes more complicated as the drones must coordinate their motion, i.e., they must fly with the same type of waypoint, avoid the collision with each other and enable the communication. Assume that N_{det} drones have detected the target in time step *t*, the array G_p with N_{det} elements are introduced to record the serial of drones in the pioneer group. Besides, an array *E* with three elements is introduced to record whether the type of waypoint is feasible all drones. *E*(*j*)=0 (*j*=1,2,3) means the type of waypoint is feasible for all drones, and *E*(*j*)=1 shows the opposite meaning. The pseudocode of determining the next waypoints for N_{det} drones is presented below.

Algorithm Approach of determining the next waypoints of drones in the pioneer group

A matrix Q with two rows and three columns is defined to record the value of the two-level index for a certain type of waypoint. Q is initialized with zero elements.
While *sum*(Q)==0
  For *j*=1:3
    If *E*(*j*)==0
      For *u*=1:N_{det}
        If *u*==1
          Determine the waypoint of drone $G_p(u)$ with the approach in Fig. 9 considering the constraint on the type of waypoint.
        Else
          The waypoint of drone $G_p(u)$ is determined by combining the result of covering 2D grids from the former drones
        End
      End
    End
    Update Q (: *j*)
  End
  If *sum*(Q)==0
    Generate a random order of drones in G_p
  Else
    Choose the best waypoints of drones according to the data in Q
End

In general, each type of waypoint which all the drones in the pioneer group can fly with is tried to calculate the value of the two-level index. The meaning of the two-level index is as the same as that in the case when only one drone has detected the target. Note that, when calculating the number of covered 2D grids for the possible positions of the target in time step *t*+1, the contributions from the former drones are also counted. In the best circumstance, all the possible positions of target in time step *t*+1 have been covered by the former drones, and the waypoint which is the nearest to the current position of target is chosen. The former drones refer to the drones in the pioneer group which have determined their waypoints in the next time step. When determining the waypoint for the drone $G_p(u)$, it should be checked that all the following constraints are satisfied, or the waypoint is infeasible.

① The drone must not be out of boundary or collide with the building in the next time step.

② The type of waypoint must be the same with that of the former drones.

③ The drone should be free from collision with all the former drones.

④ The drone should be able to communicate with at least one former drone to keep the normal communication.

If the waypoint cannot satisfy the above four constraints simultaneously, it is infeasible and will not be chosen. In the worst case, when there is no feasible waypoint for a drone, the order of drones in G_p will be randomly generated, and the above process is repeated.

2) Strategy of drone in the follower group

After all the drones in the pioneer group have determined their waypoints in the next time step, the drones in the follower group will do the same thing one after another. Similarly, an array G_F having $N_D - N_{det}$ elements are defined to record the drones in the follower group. To detect the target as soon as possible, for each drone, the waypoint which is the nearest to the current position of target will be chosen. The type of waypoint must be the same with that of the drones in the pioneer group, and the constraints of ①, ③ and ④ also must be met. Also, if no feasible waypoint exists for a drone, the order of drones in G_F will be rearranged to repeat the above procedures for each drone.

By classifying the drones into two groups, the drones determine their next waypoints in a distributed way asynchronously, and the drone which determines the waypoint earlier can deal with fewer constraints. With the DADM approach, the model established in section III. D is solved by dividing it into small-scale subproblems. The drones in 他和 pioneer group keep tracking the target and wait for the drones in the follower group to make the target within their FOVs. Besides, it can be learned from the DADM approach that the drones in the pioneer group which determine their waypoints earlier mainly contribute to covering more 2D grids surrounding the current position of target, and the later drones can get closer to the target by descending the flight altitude, which can accelerate the process of locking the target.

C. Re-search Process Considering the Case of Losing the Target

Sometimes, the influence of buildings on the FOV of drone and the reachable waypoint makes the target which is within the FOV of drone in time step *t* missed in the next time step. It is more likely to occur when there is only one drone having detected the target. An example is presented in Fig. 10 to explain the situation.

In Fig. 10, the target is near the building, and the FOV of drone is blocked by the building. In time step *t*+1, there is not such a waypoint for the drone which can cover all the possible positions of the target. Therefore, the target may be lost in time step t+1 and must be searched again.

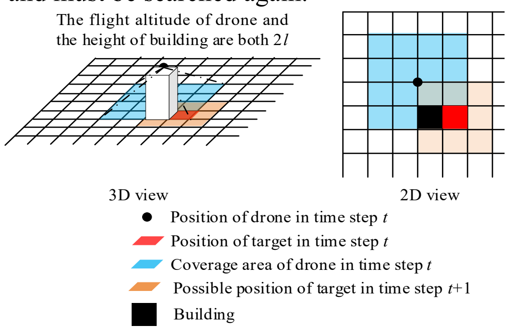

Fig. 10. The scene that the target is lost in the next time step

The SBILO algorithm proposed in section IV is not suitable to solve the target re-searching problem duo to the following two reasons. First, the SBILO algorithm is applied to the case that there is no prior information about the target, which is different from the target re-searching problem that the position of target in the recent time steps are known. Second, the computational load of SBILO algorithm is heavy and is not competent for online planning. The idea of distributed decision-making is utilized again to design the strategy of finding the target.

First, the model of predicting the position of target is established to restrict the search range. Assume that the last known position of target is $(o_x(t_l), o_y(t_l))$ in time step t_l, $\Delta t = t + 1 - t_l$ is defined to record the number of passed time steps. The possible position of target in time step t+1 can be calculated by Eq. (14).

$$\begin{cases} D_x = \max\{1, o_x(t_l) - \Delta t\} \\ U_x = \min\{o_x(t_l) + \Delta t, \frac{x_{max} - x_{min}}{l}\} \\ D_y = \max\{1, o_y(t_l) + \Delta t\} \\ U_y = \max(o_y(t_l) + \Delta t, \frac{y_{max} - y_{min}}{l}) \end{cases} \tag{14}$$

When D_x, U_x, D_y and U_y are the lower and upper bound of serial number of 2D grid. The possible position of target in time step t+1 when Δt=1,2,3 is shown in Fig. 11.

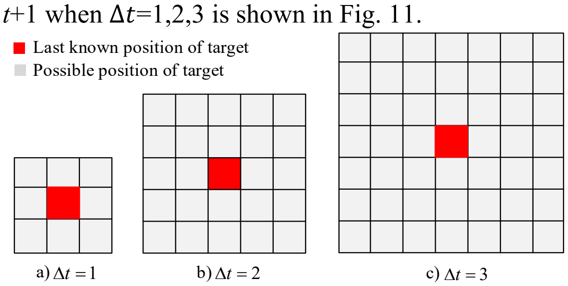

Fig. 11. The possible position of target in time step t+1 when $\Delta t = 1,2,3$

With a greater value of Δt, the number of possible positions increases, and the drones should cover those 2D grids as much as possible to increase the probability of detecting the target in time step t+1. The procedures of determining the waypoint for each drone is shown below.

Step 1: Determine the waypoint of the drone which detected the target in time step t_l to maximize the number of covered possible positions, and waypoint must satisfy the constraint ①.

Step 2: Generate a random order for the remaining drones to determine their next waypoint.

Step 3: For each drone, the waypoint is determined by integrating the contributions from the former drones, and the constraints ①, ②, ③ and ④ must be met. If there is no feasible waypoint for the drone, turn to *step* 2. After all the drones have determined their waypoints in the next time step, the process ends.

The drone which detected the target in time step t_l has a priority because it is the nearest one to the target and is most likely to find the target. More freedom should be given to the drone when determining its next waypoint, i.e., only the constraint ① needs to be met. The other drones all try their best to cover more possible positions and detect the target as soon as possible. In the worst case, if the target is lost for a long time, the possible positions of target may be almost the whole area, and the target re-searching problem will be transformed into the problem solved in section IV. However, the SBILO algorithm still cannot be applied due to its long computational time.

VI. Simulation and Discussions

A series of simulations are conducted to investigate the rationality of the established SAL model and the superiority of the SBILO algorithm and the DADM approach. The case of four drones performing the SAL task is taken as an example in each group as it is the minimum number of drones to realize the lock action without considering the influence of the buildings. First, the search phase is concerned, and different number of drones are utilized to solve the problem. Discussion is made to show the influence of the number of drones on the success rate of search. Then, to demonstrate the advantages of the SBILO algorithm, three other algorithms, i.e., BSO, TLBO and genetic algorithm (GA), are also applied to solve the same problem, and comparisons are made. The results of target locking are shown in section VI. C, and the case of losing the target is also included. In all the above simulations, the same urban environments setting is used, and the buildings are randomly distributed. The side length of the cube is set as l=30 m in AirMatrix, and the range of the low-altitude urban airspace along axes OX, OY and OZ is set as [0, 3000] m, [0, 3000] m and [0, 90] m respectively. All the results are obtained by running the programs on a desktop with Intel(R) Xeon(R) CPU E5-1630 3.70 GHz.

A. Results of Target Searching with Different Number of Drones

First, four drones are assigned to perform the SAL task, and they initially park in different places on the ground. The maximum communication distance among the drones is set as $d_{com} = 1000\ m$, and the SBILO algorithm is applied to obtain the flight routes of drones. In the SBILO algorithm, the number of individuals is set as M=100, and the maximum iteration times (Max_iter) is 50. With the above settings, the flight routes of four drones are shown in Fig. 12 and Fig. 13.

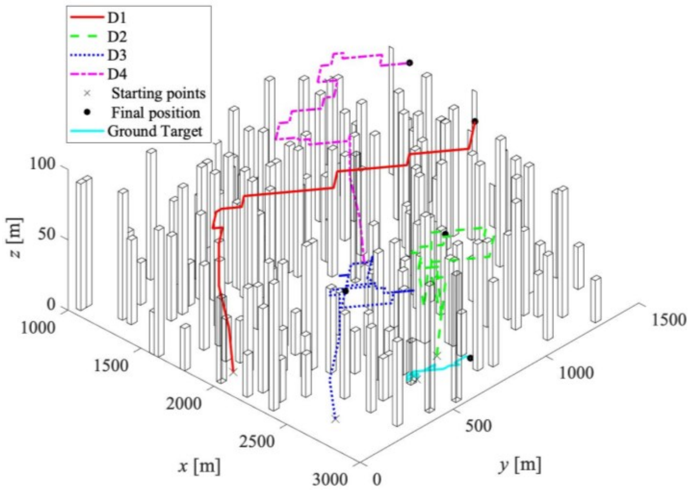

Fig. 12. Flight routes of four drones in the search phase

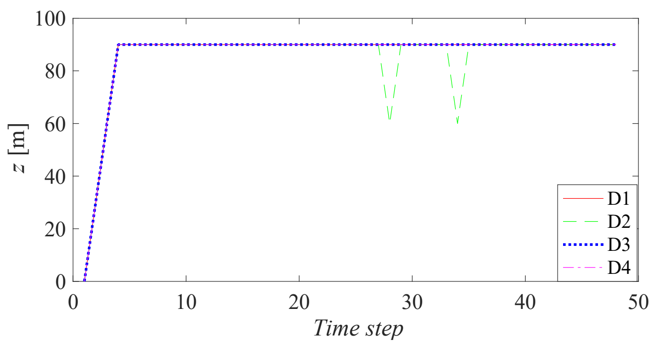

Fig. 13. Flight altitude of drones in the search phase

In Fig. 12, the motion of the ground target is random and cannot be known in advance. The four drones fly cooperatively and stay at the altitude of 90 m most of the time to maximize the index in Eq. (11) and increase the probability of detecting the target. After 48 time steps, the target is within the FOV of drone No. 2, and the distance among the drones in each time step of the search phase is presented in Fig. 14.

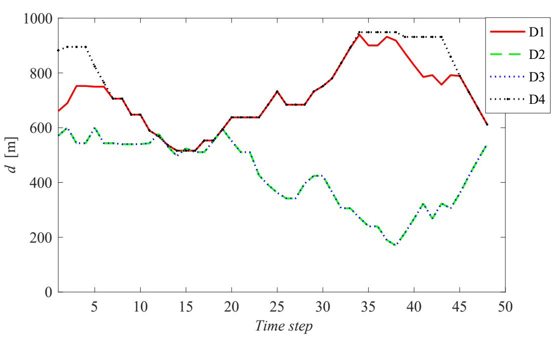

Fig. 14. The minimum distance between the drone and others in each time step of the search phase

Take the red curve in Fig. 14 as an example, it records the minimum distance between the drone No. 1 and other drones (drone No. 2, No. 3 and No. 4) in each time step. In the search phase, the minimum distance is always between 0 and 1000 m for each drone, which indicates that the drone can communicate with at least another drone in each time step, and the drones do not collide with each other. The normal communication and flight safety of drones are ensured. Besides, the index value corresponding to the coverage area of drones in each time step is shown in Fig. 15.

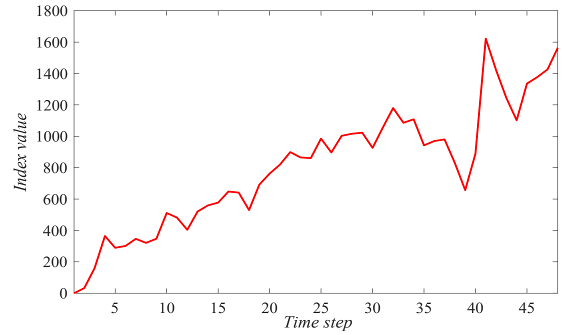

Fig. 15. Index value optimized by the SBILO algorithm in each time step of the search phase

As the drones are initially located on the ground, no 2D grids can be covered in the first time step. With the increasing flight altitude, more 2D grids can be covered according to the established FOV model of drone. As the target searching process continues, the number of covered 2D grids in not the only concern, and the time interval of visiting each 2D grid is also considered to reduce the repeated visit of some 2D grids in the continuous time steps. In general, the index value increases with a greater number of time steps but sometimes decreases to avoid the frequent visit of some 2D grids.

To further explore the influence of the number of drones on the result of searching the target, 4, 5, 6 and 7 drones are employed to perform the target searching task respectively. As there are random factors in the SBILO algorithm and the motion of target, 50 independent runs are conducted for different number of drones. The results are summarized in TABLE II.

TABLE II
RESULTS IN THE SEARCH PHASE WITH DIFFERENT NUMBER OF DRONES

Number of drones	Success rate	Maximum	Minimum	Average	Std
4	54%	468	18	145.7	146.6
5	60%	428	5	131.2	124.2
6	74%	347	8	121.6	123.5
7	90%	341	7	92.8	94.2

In TABLE II, the target searching task is regarded as a success if the target is detected by one drone within 500 time steps, and '500 time steps' is set considering the maximum flight time of drone. The terms 'maximum', 'minimum' and 'average' denote the elapsed time steps when detecting the target among the 50 independent runs, and 'std' means the standard deviation. With a greater number of drones, the success rate increases because more drones can cover a larger number of 2D grids in each time step, which improves the success rate of detecting the target. Besides, more drones also mean taking fewer time steps to finish the search phase, and it is also more reliable with a smaller standard deviation. However, as the motion of target is random, it is possible that the target can be detected earlier with fewer number of drones, as presented in the data of the fourth column in TABLE II. In real applications, although a greater of drones can improve the search efficiency with a high probability, the number of drones performing the target searching task should be determined after a comprehensive consideration of the cost and the revenue. Besides, the initial positions of drones are also important factors affecting the results, which can be calculated carefully by combining the information on the distribution of buildings in urban environments.

B. Comparison Among Several Algorithms in the Search Phase

To verify the advantages of the SBILO algorithm in searching the target, similar algorithms are introduced to make a comparison. As the basis of SBILO algorithm, BSO and TLBO algorithms are used to see the improvement. GA is also fit for solving the discrete optimization problem, and the operators in GA are similar to those in the SBILO algorithm. Therefore, the above three algorithms are selected. In the BSO algorithm, the clustering mechanism is not executed as the individuals have been classified automatically by the type of waypoint, and the teaching operator in the SBILO algorithm is not included. As the standard TLBO algorithm is used to solve the continuous optimization problem, the operators must be modified to make it suitable to solve this problem. Compared to the SBILO algorithm, the self-thinking operator is not contained, and every individual is updated in each iteration. As the motion of target has an influence on the result of target searching, only the index value in each time step is concerned, and a greater index value means a higher probability of detecting the target, which can reflect the performance of algorithm. In all the four algorithms, four drones are employed with the same initial positions, and M=100 and Max_iter=50. 50 independent runs are also conducted for every algorithm to get the statistical property, and 500 time steps are proceeded in each run. With the above conditions, the results of the four algorithms are shown in Fig. 16.

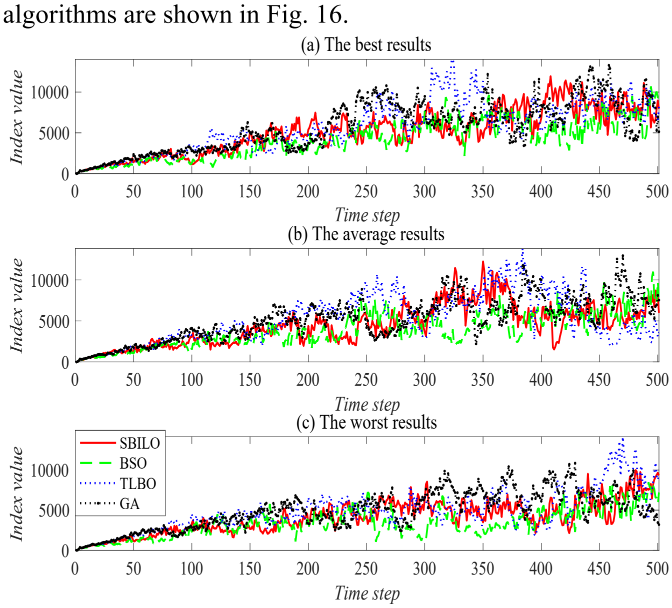

Fig. 16. Index value in each time step with four different algorithms

The best result for each algorithm corresponds to the solution which can result in the maximum sum of the index value in the 500 time steps, and the term ‘worst result’ has the similar meaning. The term ‘average result’ corresponds to the solution which ranks in the 25th place with regards to the sum of the index value in the 500 time steps. In Fig. 16, the index values are approximately the same in the first 100 time steps for the four algorithms. While, with the increasing number of steps, it is difficult to evaluate which algorithm is superior, and the reason can be explained as follows. In the one-step optimization problem, it is intuitive to compare the results from different algorithms. However, in the optimization problem with multiple steps, it is probable that one algorithm has a better performance than others in some steps but is worse in other steps, which makes the evaluation difficult. Besides, the index value in one step is affected by the previous steps in this problem, so it is also unreasonable to compare the result in each time step respectively. To have a comprehensive judgement, the average index value in the 500 time steps is chosen as the criterion, and the statistical results of the four algorithms for the computational time and the average index value are presented in TABLE III and TABLE IV respectively.

TABLE III
STATISTICAL RESULTS OF THE FOUR ALGORITHMS (THE AVERAGE INDEX VALUE)

Algorithm	Maximum	Minimum	Mean	Std
SBILO	5713.5	4857.4	5304.9	291.1
BSO	4388.1	3348.7	3829.6	278.8
TLBO	5341.3	4125.5	4617.4	322.4
GA	5487.2	4897.1	4549.5	253.4

TABLE IV
STATISTICAL RESULTS OF THE FOUR ALGORITHMS (THE COMPUTATIONAL TIME)

Algorithm	Maximum (s)	Minimum (s)	Mean (s)	Std (s)
SBILO	846.3	763.2	806.3	27.12
BSO	578.1	512.7	537.3	20.62
TLBO	17483.6	16996.7	17252.9	176.24
GA	12540.9	11850.1	12241	217.91

In TABLE III, the SBILO algorithm has the best performance in 50 independent runs, which demonstrates the advantages of the designed operators. The standard deviations of the four algorithms do not show a big difference, and all the four algorithms have stable performances. In the SBILO algorithm, the three operators form a complete learning process, i.e., learning from teacher and other classmates and self-learning, which is superior to the other three algorithms because only a part of the three operators are included in them. A complete learning process indicates that there are more ways for an individual to update itself, which is more likely to generate a better solution and enhances the exploration ability of algorithm. In TABLE IV, the computational time of the TLBO and GA algorithms are far more than the SBILO and BSO algorithm because in each iteration, every individual is updated in the TLBO algorithm, which will cost much time. In GA, with the selection operator, only half of individuals need to be updated, which can save the time compared to the TLBO algorithm. While, for the SBILO and BSO algorithm, only a few individuals are selected to update themselves in each iteration, so the computational time can be greatly reduced. As the teaching operator is not contained in the BSO algorithm, fewer computation time is needed. To sum up the results in TABLE III and TABLE IV, the SBILO algorithm can obtain the best result among the four algorithms within a relatively short computational time, thus showing the high efficiency of the algorithm in solving this target searching problem.

C. Results of Target Locking Considering the Case of Losing the Target

Following the results of target searching shown in Fig. 12 and Fig. 13 in section VI. A, the target locking is conducted immediately using the proposed DADM approach. Note that, the maximum communication distance is set as d_{com}=1300 m in the lock phase, which is slightly greater than that in the search phase. This is because when the distance between two drones is near the threshold at the end of the search phase, it is possible that there is no feasible waypoint for a drone in the next time step. Unlike the way of determining the waypoint of drone in the search phase, the drones calculate their waypoints one by one asynchronously in the lock phase. The drone which determines its waypoint last must satisfy more constraints. Therefore, the constraint on the maximum communication distance is looser in the lock phase. Under such conditions, the flight routes of the four drones and the distance among the drones in the lock phase are shown from Fig. 17 to Fig. 19 respectively.

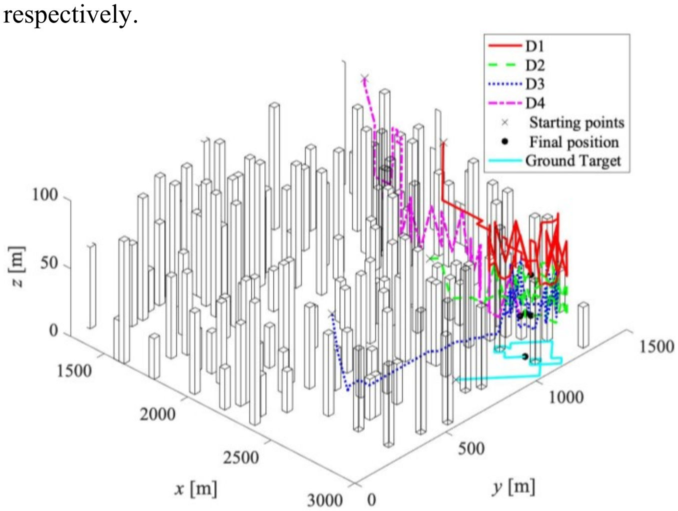

Fig. 17. Flight routes of four drones in the lock phase

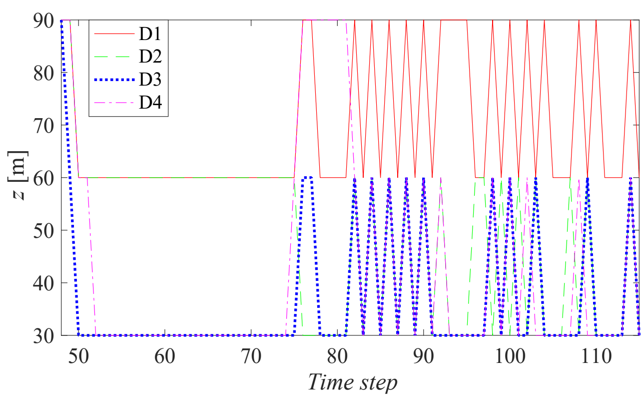

Fig. 18. Flight altitude of drones in the lock phase

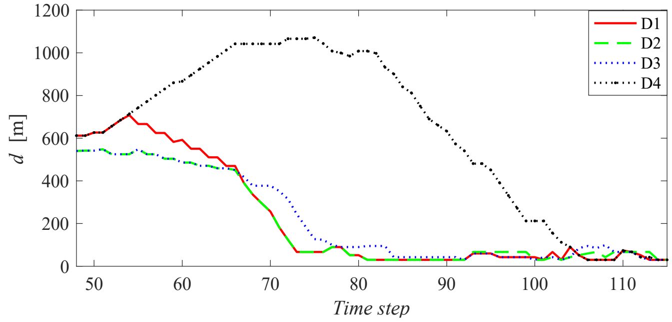

Fig. 19. The minimum distance between the drone and others in each time step of the lock phase

In Fig. 17 and Fig. 18, as the drone No. 2 detects the target first, it will broadcast the current position of target to the other three drone to call them to come to the target. During the lock phase, the drones can always communicate with each other to keep a smooth exchange of information, as shown in Fig. 19. To provide more information on whether the target is within the FOV of a drone in each time step of the lock phase, Fig. 20 is presented.

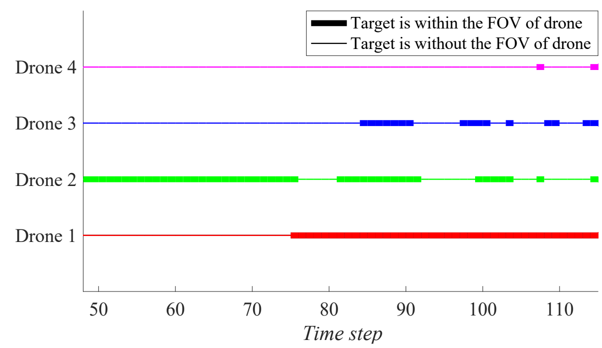

Fig. 20. The state of drone in each time step of the lock phase

It can be learned from Fig. 20 that only the drone No. 2 detects the target in the early time of the lock phase, and it keeps tracking the target and waits for the other drones. Then the target falls into the FOV of drones No. 1 and No. 3 gradually, and the drone No. 4 is the last one to detect the target. Note that, the drone may not always keep the target within its FOV, and sometimes the target is lost due to the block of buildings. With the position information of the target from the other drones, the target can return to its FOV within several time steps, and the case that the target is without the FOV of all the drones does not appear. Finally, drones No. 2, No. 3 and No. 4 descend to the altitude of 30 m, and the drone No. 1 stays at the altitude of 60 m (see Fig. 18). With the help of the buildings and the contributions of drones No. 2, No. 3 and No. 4, the nine 2D grids surrounding the target are all covered, and the SAL task is finished.

To further validate the effectiveness of the proposed DADM approach in dealing with the case of losing the target, another simulation with the random motion of target in the search phase is conducted, and the results of the SAL task are shown from Fig. 21 to Fig. 24.

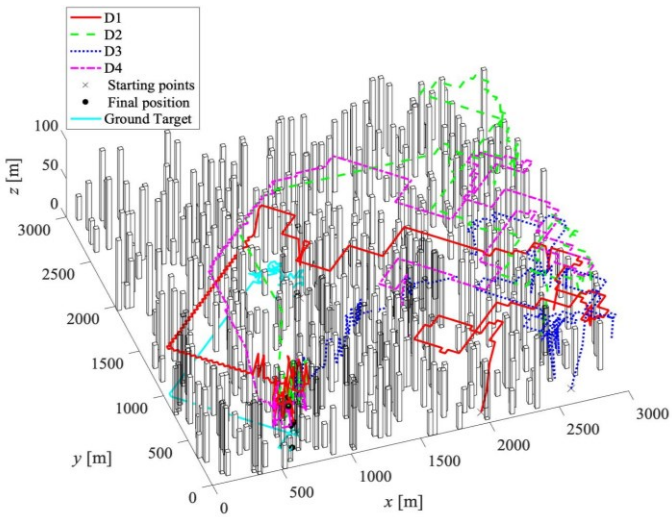

Fig. 21. Flight routes of four drones in the SAL task

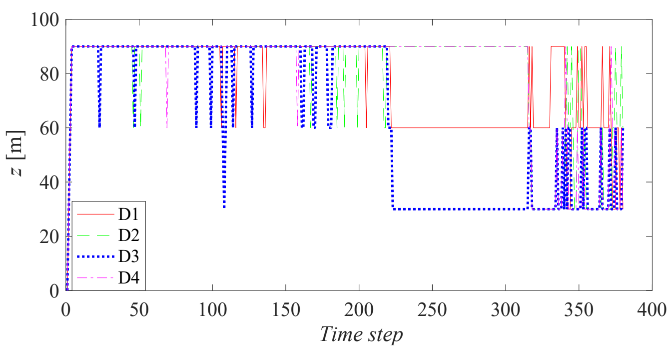

Fig. 22. Flight altitude of four drones in the SAL task

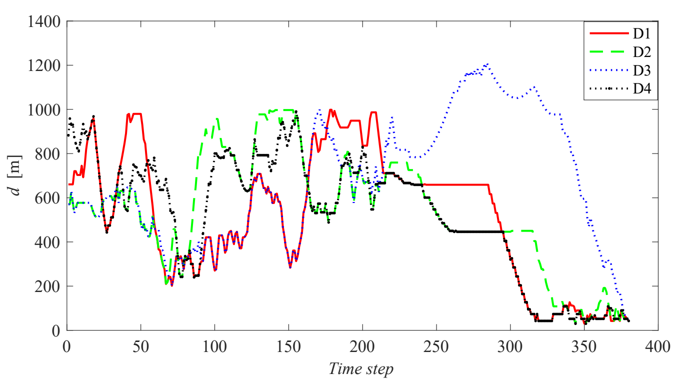

Fig. 23. The minimum distance between the drone and others in each time step of the SAL task

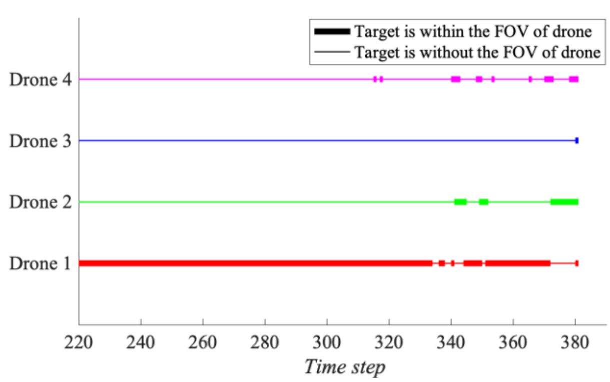

Fig. 24. The state of drone in each time step of the SAL task

According to the above results, the four drones can search and lock the target while satisfying the constraints on the communication distance and flight safety. In Fig. 24, the target is without the FOV of all the drones in two time periods, i.e., from time step 334 to 336 and 338 to 340, and the drones can quickly re-search the target within two time steps by covering more 2D grids that the target may appear. With the proposed DADM approach, the lost target can be detected again even though the detailed information on the motion of target is not predicted.

Next, the influence of different number of drones on the result of the lock phase are discussed. Based on the results of search phase in section VI. A, the lock phase is continued for the circumstance that the target has been successfully detected, and the statistical result is presented in TABLE V and TABLE VI.

TABLE V
STATISTICAL RESULT ON THE NUMBER OF TIME STEPS WHEN THE LOCK PHASE IS ENDED

Number of drones	Success rate	Maximum	Minimum	Average	Std
4	36.4%	380	115	210.3	124.5
5	50%	335	80	185.7	107.7
6	66.7%	445	57	163.3	129.9
7	100%	456	71	208.7	131.1

TABLE VI
STATISTICAL RESULTS ON THE COMPUTATIONAL TIME IN THE LOCK PHASE

Number of drones	Maximum (s)	Minimum (s)	Mean (s)	Std (s)
4	1.359	0.746	0.973	0.269
5	1.147	0.844	0.982	0.120
6	1.339	0.736	0.978	0.222
7	2.439	0.602	1.255	0.485

Note that, when calculating the success rate in TABLE V, the denominator is not 50 (the number of independent runs) but the number of circumstances having detected the target successfully, and the SAL task is successful when it is finished within 500 time steps also considering the battery capacity of drone. The success rate of SAL task increases when the number of drones is greater, and it reaches 100% when the number of drones is seven. However, a greater number of drones does not necessarily guarantee fewer time steps of flying for drones because the result is closely related to the position of drones and the motion of target. Therefore, the case with seven drones does not show its advantage over others in TABLE V, and it is even inferior to the result with six drones in general. As for the computational time, it costs about 1 s to obtain the result of target locking in all the cases, and the computational time of calculate the waypoint in each time step is even fewer (for example, when it takes 50 time steps to finish the lock phase, the computational time for each time step is approximately 20 ms). Compared to the SBILO algorithm, the computational time of the DADM approach is greatly reduced, which is competent for online computation and is suitable for solving the target locking problem.

VII. Conclusion

The SAL task for the moving ground target in the low-altitude urban airspace is a focus of the present work. Multiple cooperative drones are deployed to perform the task. Literature investigation shows that the search and track problem has been paid much attention, but the target locking task is rarely considered. Besides, the detected target may be lost due to the block of buildings, and this case is not well addressed in the existing studies.

First, the concept of 'AirMatrix' is utilized to discretize the low-altitude urban airspace. Under this framework, the models of flight rules and FOV for a single drone are established, and the cooperation among the drones are realized by selecting the same type of waypoint, considering the communication distance and avoiding the collision. The random motion of the ground target is also described in the discrete environment. In the search phase, the drones cover more 2D grids which have not been visited recently to increase the probability of detecting the target. While, in the lock phase, the drones are required to lock the target as soon as possible.

An SBILO algorithm which integrating the advantages of the BSO and TLBO algorithms is proposed to determine the waypoint of each drone in the search phase. In this multi-step optimization problem, three operators, i.e., teaching operator, discussion operator and self-thinking operator are designed to update some selected individuals in each iteration, and the waypoint of drone in the next time step is obtained when the maximum iteration times is reached. The above process will continue before the target is detected by any drone. In the lock phase, a DADM approach is developed to make a quick response to the current position of the target. The drones are classified into two groups, i.e., the pioneer group and the follower group, and they calculate the waypoint in the next time step one by one. For the drones in the pioneer group, they first try to cover the nine 2D grids surrounding the target to make the target always within the FOV of drone. Then they will get closer to the target to realize the lock operation. As the drones in the follower group have not detected the target yet, they will try to reduce the distance to the target and turn into a member of the pioneer group. When the target is without the FOV of all the drones, the drones will cover more 2D grids which the target may stay to increase the probability of detecting the target again.

In the simulation studies, the SAL task is performed by different number of drones with different algorithms considering the random factors in the algorithms and the motion of target. The results demonstrate that the success rate of the SAL task increases with a greater number of drones, but the efficiency of performing the task is not necessarily improved due to the random motion of target. The SBILO algorithm performs better than the BSO, TLBO and GA in the search phase. With the DADM approach, the target can be locked by the drones, and the lost target can be re-searched within short time steps.

In the future, the SAL task can be popularized in more real applications, and the drones can combine with other vehicles (unmanned ground vehicle and USV) and man to further improve the efficiency, such as searching for the escaped criminals and the missing people. Besides, other types of tasks (e.g. delivery, rescue and surveillance) can be considered together to analyze the utilization of the low-altitude urban airspace [38].

Acknowledgment

This research work is supported by the Chongqing Research Program of Basic Research and Frontier Technology with the grant numbers of cstc2020jcyjmsxmX0602, Fundamental Research Funds for the Central Universities with the project reference number of 2020CDJ-LHZZ-066, China Scholarship Council with the project reference number of 201906055030.

This collaborative research is also supported by the Ministry of Education (MOE, Singapore) Tier-1 project research grant (Project ID: 2018-T1-002-124) and the UAS Program on "Urban Aerial Transport Traffic Management and Systems" in the ATMRI, NTU, Singapore.

Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not reflect the views of any of the abovementioned funding agents.

References

[1] Wang, D., Hu, P., Du, J., Zhou, P., Deng, T., & Hu, M. (2019). Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. *IEEE Internet of Things Journal*, *6*(6), 10483-10495.

[2] Klaine, P. V., Nadas, J. P., Souza, R. D., & Imran, M. A. (2018). Distributed drone base station positioning for emergency cellular networks using reinforcement learning. *Cognitive computation*, 10(5), 790-804.

[3] Dong, F., You, K., & Zhang, J. (2019). Flight Control for UAV Loitering Over a Ground Target with Unknown Maneuver. *IEEE Transactions on Control Systems Technology*, 28(6), 2461-2473.

[4] Liu, Y., Wang, Q., Hu, H., & He, Y. (2018). A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11), 2362-2372.

[5] Yaacoub, J. P., Noura, H., Salman, O., & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet of Things*, 11, 100218.

[6] Moon, J., Papaioannou, S., Laoudias, C., Kolios, P., & Kim, S. (2021). Deep Reinforcement Learning Multi-UAV Trajectory Control for Target Tracking. *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2021.3073973.

[7] Dong, F., You, K., & Zhang, J. (2019). Flight control for UAV loitering over a ground target with unknown maneuver. *IEEE Transactions on Control Systems Technology*, 28(6), 2461-2473.

[8] Gramajo, G., & Shankar, P. (2017). An efficient energy constraint based UAV path planning for search and coverage. *International Journal of Aerospace Engineering*, 2017, doi: 10.1155/2017/8085623.

[9] Li, J. M., Chen, C. W., & Cheng, T. H. (2020). Motion Prediction and Robust Tracking of a Dynamic and Temporarily-Occluded Target by an Unmanned Aerial Vehicle. *IEEE Transactions on Control Systems Technology*, doi: 10.1109/TCST.2020.3012619.

[10] Yang, B., Cao, X., Yuen, C., & Qian, L. (2020). Offloading Optimization in Edge Computing for Deep Learning Enabled Target Tracking by Internet-of-UAVs. *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2020.3016694.

[11] Liao, S. L., Zhu, R. M., Wu, N. Q., Shaikh, T. A., Sharaf, M., & Mostafa, A. M. (2020). Path planning for moving target tracking by fixed-wing UAV. *Defence Technology*, 16(4), 811-824.

[12] Hildmann, H., & Kovacs, E. (2019). Using unmanned aerial vehicles (uavs) as mobile sensing platforms (msps) for disaster response, civil security and public safety. *Drones*, 3(3), 59.

[13] Mohamed Salleh, M. F. B., Wanchao, C., Wang, Z., Huang, S., Tan, D. Y., Huang, T., & Low, K. H. (2018). Preliminary concept of adaptive

urban airspace management for unmanned aircraft operations. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace* (p. 2260).

[14] Tan, Q., Wang, Z., Ong, Y. S., & Low, K. H. (2019, June). Evolutionary optimization-based mission planning for UAS traffic management (UTM). In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 952-958). IEEE.

[15] Sun, P., & Boukerche, A. (2018). Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network. *Computer Networks*, 146, 217-231.

[16] Du, Y. C., Zhang, M. X., Ling, H. F., & Zheng, Y. J. (2019). Evolutionary planning of multi-UAV search for missing tourists. *IEEE Access*, 7, 73480-73492.

[17] Yao, P., Xie, Z., & Ren, P. (2017). Optimal UAV route planning for coverage search of stationary target in river. *IEEE Transactions on Control Systems Technology*, 27(2), 822-829.

[18] Zheng, X., & Ma, C. (2021). An intelligent target detection method of UAV swarms based on improved KM algorithm. *Chinese Journal of Aeronautics*, 34(2), 539-553.

[19] Zhao, Y., Wang, X., Wang, C., Cong, Y., & Shen, L. (2019). Systemic design of distributed multi-UAV cooperative decision-making for multi-target tracking. *Autonomous Agents and Multi-Agent Systems*, 33(1), 132-158.

[20] Wang, X., Zhao, H., Han, T., Zhou, H., & Li, C. (2019). A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem. *Applied Soft Computing*, 78, 240-260.

[21] Zhang, W., Song, K., Rong, X., & Li, Y. (2018). Coarse-to-fine uav target tracking with deep reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 16(4), 1522-1530.

[22] Li, B., & Wu, Y. (2020). Path planning for UAV ground target tracking via deep reinforcement learning. *IEEE Access*, 8, 29064-29074.

[23] Wang, T., Qin, R., Chen, Y., Snoussi, H., & Choi, C. (2019). A reinforcement learning approach for UAV target searching and tracking. *Multimedia Tools and Applications*, 78(4), 4347-4364.

[24] Khan, M., Heurtefeux, K., Mohamed, A., Harras, K. A., & Hassan, M. M. (2017). Mobile target coverage and tracking on drone-be-gone UAV cyber-physical testbed. *IEEE Systems Journal*, 12(4), 3485-3496.

[25] Gu, J., Su, T., Wang, Q., Du, X., & Guizani, M. (2018). Multiple moving targets surveillance based on a cooperative network for multi-UAV. *IEEE Communications Magazine*, 56(4), 82-89.

[26] Hu, C., Zhang, Z., Yang, N., Shin, H. S., & Tsourdos, A. (2019). Fuzzy multiobjective cooperative surveillance of multiple UAVs based on distributed predictive control for unknown ground moving target in urban environment. *Aerospace Science and Technology*, 84, 329-338.

[27] Wu, Y., Low, K. H., & Lv, C. (2020). Cooperative Path Planning for Heterogeneous Unmanned Vehicles in a Search-and-Track Mission Aiming at an Underwater Target. *IEEE Transactions on Vehicular Technology*, 69(6), 6782-6787.

[28] Harikumar, K., Senthilnath, J., & Sundaram, S. (2018). Multi-UAV oxyrrhis marina-inspired search and dynamic formation control for forest firefighting. *IEEE Transactions on Automation Science and Engineering*, 16(2), 863-873.

[29] Zheng, Y. J., Du, Y. C., Ling, H. F., Sheng, W. G., & Chen, S. Y. (2019). Evolutionary collaborative human-UAV search for escaped criminals. *IEEE Transactions on Evolutionary Computation*, 24(2), 217-231.

[30] Chen, Q. Y., Lu, Y. F., Jia, G. W., Li, Y., Zhu, B. J., & Lin, J. C. (2018). Path planning for UAVs formation reconfiguration based on Dubins trajectory. *Journal of Central South University*, 25(11), 2664-2676.

[31] Dai, R., Fotedar, S., Radmanesh, M., & Kumar, M. (2018). Quality-aware UAV coverage and path planning in geometrically complex environments. *Ad Hoc Networks*, 73, 95-105.

[32] Wang, X., Zhao, H., Han, T., Zhou, H., & Li, C. (2019). A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem. *Applied Soft Computing*, 78, 240-260.

[33] Zorbas, D., Pugliese, L. D. P., Razafindralambo, T., & Guerriero, F. (2016). Optimal drone placement and cost-efficient target coverage. *Journal of Network and Computer Applications*, 75, 16-31.

[34] Xie, J., Carrillo, L. R. G., & Jin, L. (2018). An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems. *IEEE control systems letters*, 3(1), 67-72.

[35] Wu, Y. (2021). A survey on population-based meta-heuristic algorithms for motion planning of aircraft. *Swarm and Evolutionary Computation*, 62, 100844.

[36] Shi, Y. (2011, June). Brain storm optimization algorithm. In *International conference in swarm intelligence* (pp. 303-309). Springer, Berlin, Heidelberg.

[37] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.

[38] Wu, Y., Low, K. H., Pang, B., & Tan, Q. (2021). Swarm-based 4D Path Planning for Drone Operations in Urban Environments. *IEEE Transactions on Vehicular Technology*, Early Access.