Optimal Trajectory Planning of Drones for 3D Mobile Sensing

Anonymous

***Abstract*—Mobile sensing is usually limited in 3D space, as there are many inaccessible places where people rarely venture. Unmanned aerial vehicle (UAV), commonly known as drone, has greatly extends the scope of mobile sensing in 3D space, and pushed forward a variety of 3D mobile sensing applications, such as aerial photo- or video-graphy, 3D wireless signal survey, air quality monitoring. However, the limited battery life of drones has largely restricted the wide adoption of these applications; meanwhile, the flight between two locations consumes more of the drone's battery power than hovering over one location. To maximally expand the sensing scope of the drone, in this paper, we study the trajectory planning problem for optimizing its flight route in 3D space, given its limited battery life. Specifically, we divide the target 3D space into a network of observation locations formed by multiple 2D grids, formulate the minimum dominating path problem in each 2D grid to find the optimal trajectory that has the maximal coverage in 3D space, and then select necessary critical observation locations along the trajectory for the drone to hover and perform the measurement. Experimental results show that the proposed algorithm outperforms other approaches XXX.**

I. Introduction

Unmanned aerial vehicle (UAV), commonly known as drone, is an aircraft without a human pilot aboard, which is commonly used in measurement and sampling. Compared to manned aircraft, drones are more suitable for data collections and mobile sensing applications that capture different dimensions of signals in the environment that are beyond our sensing capability, such as aerial photography, 3D wireless signal survey, air quality index (AQI) measurement.

However, civilian drones are still not popular these days. Furthermore, a lot of drone companies were broken down. It could be a quite confusing problem if you have never come into attach with a drone. If you've actually tried using them, you could find that civilian drones do not really apply to daily life due to:

- Low battery available time.
- Great noise during flight.
- Wing rock and more battery drain caused by poor carrying capacity.

Therefore, in order to make more use of existing drones, we must consider the following problem: **How to complete flight in the shortest possible time? In other words, how to find the optimal trajectory? Furthermore, in the three-dimensional space?**

Similar to traditional sensor networks and mobile base station, we consider data collection in mobile environment. So total time consumption consists of two parts: **flight time** and **measurement time**. While we also have the following difference:

- We consider optimal trajectory in three-dimensional s-pace.
- We use the routing algorithm based on graph theory apart from traditional greedy algorithms.

In this paper, we consider optimal trajectory in three-dimensional mobile sensing. We divide three-dimensional space into a network of observation locations (OLs), and generate trajectory in two steps: select critical observation locations (COLs) from OLs to cover measurement space while find a OL-path(trajectory) that covers COLs. We formulate the problem as a constraint set coverage problem in graph theory. Specifically, we consider the following two special cases:

1) *Consider flight time only*: Under this condition, we assume measurement time negligible and consider flight time only. We choose the shortest OL-path in OL-network to minimize flight time and select all OLs in OL-path as COLs. Therefore, we could formulate problem as a minimum dominating path. In this paper, we solved this problem in grid which could extend to three-dimensional space.
2) *Consider total time consuming*: Under this condition, we consider total time consuming which is the sum of flight time and measurement time. Since flight time usually take up most of time consuming, we find the shortest OL-path first. Then in order to minimize measurement time, we select least OLs in OL-path to cover OL-network. Also, we solve this problem in grid.

Because of algorithms we use is based on graph theory, We could get size of COL and OL-path in O(1) time while select COLs and draw trajectory in O(n) time. We use drones to verify our simulation. We find out that the flight time we use is less than ordinary approach.

II. Related Work

A. Drones in 3D mobile sensing

Conventional mobile sensing systems rely on the mobile devices or ground-vehicles to perform the environmental sensing in a 2D plane, e.g., the mobile data analysis using bikes [2] or cars [1].

Drones become more and more popular in 3D mobile sensing applications since they could capture different dimensions of signals in the 3D environment that are usually beyond our sensing capability. Aerial photo or video-graphy applications [6] facilitate the collection of images, and the analysis over the collected data. For example, it is feasible to locate anomalies in image data, and link particular image data to an address of the property where the anomaly is detected. DroneSense is a system for 3D wireless signal survey [10], i.e., measuring wireless signals in the 3D space, which provides us with an efficient method to quickly analyze wireless coverage and test their wireless propagation models. SensorFly is designed for indoor emergency response or inspections in inaccessible places where people cannot reach [7], which forms aerial sensor network platform that can adapt to node and network disruptions in harsh environments.

However, these 3D mobile sensing applications of drones are constrained by the limited battery life, which motivates a more efficient measurement approach to better design the trajectory.

B. Trajectory planning problem

To address the trajectory planning problem, the genetic algorithm [5], [11] and particle swarm optimization [3] are always proposed for real-time path planning, which could find an optimal or near-optimal path for robotics in both complicated static and dynamic environments. These approaches have been applied on the UAV platform which could ensure partial minimality between two nodes.

Meanwhile, the trajectory planning for a large scale mobile sensing scenario is usually formulated as an ordinary Travelling Salesman Problem (TSP) or a special TSP problem [9], [8]. Existing solutions take a two-step approach: (1) the first step is to find a minimum vertex cover in the underlying network graph (i.e., a number of measurement locations that cover the given sensing area); and (2) the second step is to find the shortest path connecting these vertices (locations), along which the mobile device can traverse to complete the mobile sensing task in space.

In this paper, we take a different approach by finding the optimal path first and then selecting the measurement locations, as the flight consumes more battery than hovering for drone operations.

III. System Model and Problem Formulation

In this section, we establish a multi-layer 3D network model that are formed by multiple 2D networks for mobile sensing in the 3D space. In the first case when the flight time is dominant, we formulate the trajectory planning problem as a minimum dominating path problem. In the second case, we consider both the flight time and hovering time of the drone, and formulate the problem as a combination of the minimum dominating path problem and the constrained minimum dominating set problem.

A. 3D network model

Dividing the 3D space into cuboids: We divide a 3D space into cuboids with a meters long, b meters wide and h meters high. We define the center point of cuboid i as its observation location (OL) (as shown in Figure 1), which is denoted by the 3-tuple (longitude, latitude, and altitude), i.e.,

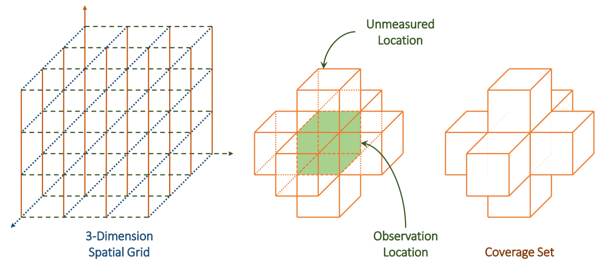

Fig. 1: The divided cuboids of a 3D space; a center OL in the cuboid in the green color; and the OLs and their cuboids in the coverage set of the center OL.

$$OL_i = (x_i, y_i, z_i),$$

where x_i, y_i, z_i are 3D coordinates of OL i. Note that we call the OL at the center of cuboid i as OL i.

3D network of observation locations (OLs): OLs form a 3D network graph $\mathcal{G} = (\mathcal{V}; \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ represents the edges connecting neighboring vertices. Specifically, the OL inside each cuboid i is considered as a vertex in $\mathcal{G}$, and an edge (i, j) exists, if cuboid i is adjacent to cuboid j (i.e., they are the same in two coordinates and adjacent to each other in the third dimension coordinate). As a result, the OLs forms a 3D grid in the space.

The coverage set of OL i: We define the coverage set of OL i as the set of OLs including those located in the cuboids of the cuboid i, plus OL i itself, as shown in Figure 1.

Critical OL (COL): Given limited battery life, it is impossible for a drone to traverse all OLs in the space. Hence, we need to select a number of critical observation locations (COL) from OLs where the drone can perform the measurement.

Correlation between OLs: We assume that the sensed data at two OLs in the same 3D space may have certain correlation. Intuitively, the more distant two OLs are located, the less correlation they may have. That is, two adjacent OLs have the strongest correlation in their sensed data; the sensed data at two adjacent OLs are almost the same if the distance between them is small. Hence, if the drone has sensed at one OL, it can skip sensing at its adjacent OLs.

Multiple layers of 2D grids: The 3D grid can be divided into multiple layers of 2D grids at different heights. If the drone has sensed at OLs in one layer of 2D grid, it can skip sensing at OLs in its adjacent layers of 2D grids (i.e, the one upper layer and the one lower layer). Let $G = L_{m,n} = (V; E)$ denote a 2D grid with m rows and n columns.

B. Time consumed

The time consumed by the 3D mobile sensing of drones consists of two parts: (1) the flight time, and (2) the measurement time.

Let $V_C \subseteq V$ denote the set of COLs, and v_{C_i} is $i-th$ vertex in V_C. Let $V_P \subseteq V$ denote the set of vertices in the drone's trajectory which forms a path in $\mathcal{G}$. We have $V_C \subseteq V_P$ since the trajectory should contain the COLs, i.e., vertices in V_C.

Flight time: the flight time T_F is time consumed by the drone's flight. In the formed 3D grid, we use Hamiltonian distance to characterize the distance between OLs. So teh flight time is proportion to the length of the trajectory, which is written as $T_F = t_F|V_P|$, where t_F is the flight time for a unit length in the coordinate system of the 3D grid.

Measurement time: the measurement time T_M is time consumed by the drone for hovering and measurement. For the same sensing task (e.g., WiFi signal survey), we could assume that measurement time is the same for all OLs. So the total measurement time is proportion to the number of COLs, which can be written as $T_M = t_M|V_C|$, where t_M is the measurement time at each OL.

Therefore, the total time T consumed by the drone is

$$T = T_F + T_M.$$

C. Problem formulation

We formulate the following two problems based on two observations: (1) the flight mode consumes more battery per unit time than the hovering mode for a drone; (2) the flight time much more than the measurement time in the trajectory for most sensing applications.

1) Case 1: Considering flight time only: In this case, we assume the flight time is dominant, and the measurement time at each OL can be negligible. Then, we formulate the problem as a minimum dominating path problem in multi-layer 2D grids.

Problem 1 (Trajectory planning by flight time only). *Given a 3D grid $\mathcal{G}$ divided into multi-layer 2D grids, for each 2D grid $G = L_{m,n} = (V; E)$ where $V = \{v_0, v_1, ..., v_{|V|-1}\}$, assume the time required for completing a drone's trajectory is only relevant to the drone's flight time, and let $C(v_i)$ be the coverage set of vertex v_i in V.*

Since $V_C = V_P$, we seek to find the minimum dominating path $V_P \subseteq V$ which contains COLs in V and covers all OLs in the 3D grid.

$$\begin{aligned} &\textit{minimize} && |V_P| \\ &\textit{subject to} && \bigcup_{v \in V_P} C(v) = V, \quad V_P \textit{ is a path.} \end{aligned}$$

2) Case 2: Considering both flight time and measurement time: In this case, we assume the measurement time at each OL is not negligible. As the flight mode consumes more power per unit time, we first search for the shortest path that has the maximum coverage, and then select the least number of OLs along the path as COLs for measurement. We formulate this problem as a combination of a minimum dominating path problem and a constrained minimum dominating set problem in multi-layer 2D grids.

Problem 2 (Trajectory planning by both flight time and measurement time). *Given a 3D grid $\mathcal{G}$ divided into multi-layer 2D grids, for each grid $G = L_{m,n} = (V; E)$ where $V = \{v_0, v_1, ..., v_{|V|-1}\}$, assume the time required for completing a drone's trajectory is dependent on the drone's flight time and measurement time, and let $C(v_i)$ be the coverage set of vertex v_i in V.*

We seek to find the minimum dominating path $V_P \subseteq V$ which covers all OLs in the 3D grid, and then select a set V_C of COLs from V_P as the dominating set.

$$\begin{aligned} &\textit{minimize} && |V_P|, \ |V_C| \\ &\textit{subject to} && \bigcup_{v \in V_P} C(v) = V, \quad \bigcup_{v \in V_C} C(v) = V, \\ & && V_P \textit{ is a path}, \quad V_C \subseteq V_P. \end{aligned}$$

Therefore, we will discuss these two problems in the next section and give corresponding certifications.

IV. Construction of the minimum dominating path in 2D grids

In this section, we first extend the well-known concept of the dominating set to define the dominating path in a graph. Then, we address Problem 1 by proposing an optimal algorithm to find the the minimum dominating path in the 2D grid. We can concatenate the discovered paths (trajectories) in 2D grids to construct the one in the 3D grid.

A. From the dominating set to dominating path

Our definition of dominating path: Recall that a dominating set for the graph $G = (V, E)$ is a subset $D \subset V$ such that every vertex not in D has a neighbor on D. A connected dominating set D for graph $G = (V, E)$ is a special dominating set such that any vertex in D can reach any other node in D by a path that stays entirely within D. Similarly, we have the following definition.

Definition 1. A dominating path L for graph $G = (V, E)$ is path $L \subset V$ such that every vertex not in L has a neighbor on L which is a special connected dominating set. In a dominating path L,

- there exists a dominating set D_L of G, and vertices in D_L are called *dominating vertices* of path L;
- and the other vertices in $L \setminus D_l$ are called *connecting vertices* of path L.

Start from the trivial cases: The goal is to find the minimum dominating path in a 2D grid $L_{m,n}$, where m, n are positive integers.

- *Step 1: trivial cases*. When $m \leq 3$ and $n \leq 3$, it is easy to find the minimum dominating path;
- *Step 2: adding more columns to trivial cases*. When $m \leq 3$ and $3 < n \leq 6$, we seek to construct the minimum dominating path in the 2D grid $L_{m,n}$ by extending the the minimum dominating path in the 2D grid $L_{m,(n-3)}$ that has been found in Step 1;
- *Step 3: adding more rows to trivial cases and transposing the grid*. When $3 < m \leq 6$ and $n \leq 3$, we first transpose the grid—swap the rows and columns of the grid, and construct the minimum dominating path in the 2D grid $L_{n,m}$ by following Step 2;

Any bigger-size grid $L_{m,n}$ when $3 < m \leq 6$ and $3 < n \leq 6$ can be expended by adding three more rows and/or three more columns from one of the grids in trivial cases. The The minimum dominating path in the bigger-size grid $L_{m,n}$ can be also found by repeating above Steps 2 and 3.

Next, we show the correlation between the smaller grid $L_{m,n}$ and the expanded grid $L_{m,(n+3)}$; note that the latter is expanded by adding three columns to the former.

We give notations of variables.

- G_i^c: Leftmost $i-th$ column in grid G.
- G_i^r: Topmost $i-th$ row in grid G.
- $G_{i,j}^c$: Columns between G_i^c and G_j^c.
- $G_{i,j}^r$: Rows between G_i^r and G_j^r.
- $v_{i,j}$: Vertex of intersection of G_i^r and G_j^c in grid.
- $N[y] = \{v, y \in V : yv \in E\} \cup \{y\}$: Coverage set of vertex y.
- $N[S] = \bigcup_{v \in S} N[v]$: Coverage set of vertex set $S \subset V$.
- L: Vertex set of dominating path.
- $L_i^c = L \cap G_i^c$: Intersection between L and G_i^c.
- $L_i^r = L \cap G_i^r$: Intersection between L and G_i^r.
- $L_{i,j}^c(G) = L \cap G_{i,j}^c$: Intersection between L and $G_{i,j}^c$.
- $L_{i,j}^r(G) = L \cap G_{i,j}^r$: Intersection between L and $G_{i,j}^r$.
- $D(L)$: Dominating vertices in L.
- $C(L)$: Connecting vertices in L.
- $D_i^c(L)$: Dominating vertex in L_i^c.
- $D_i^r(L)$: Dominating vertex in L_i^r.
- $D_{i,j}^c(L)$: Dominating vertices in $L_{i,j}^c(G)$.
- $D_{i,j}^r(L)$: Dominating vertices in $L_{i,j}^r(G)$.
- $C_{i,j}^c(L)$: Connecting vertices that connect vertices in $D_{i,j}^c(L)$.
- $C_{i,j}^r(L)$: Connecting vertices that connect vertices in $D_{i,j}^r(G)$.
- $C_i^{ccon}(L)$: Connecting vertices that connect vertices between $D_{1,i}^c(L)$ and $D_{i+1,n}^c(L)$.
- $\gamma(G)$: The size of minimum dominating set of G.
- $\gamma_c(G)$: The size of minimum connected dominating set of G.
- $\gamma_l(G)$: The size of minimum dominating path of G.

B. The size of the minimum dominating path in the expanded grid $L_{m,n+3}$

1) Decomposing the minimum dominating path in $L_{m,n+3}$: In the following lemmas and theorems, we derive the lower bound for the number of vertices needed to add into the minimum dominating path of $L_{m,n}$ so as to find the minimum dominating path in the expanded grid $L_{m,n+3}$.

Since vertices in L consists of dominating vertices and connecting vertices, we split dominating path in $L_{m,n+3}$ into 5 parts: $L = D_{1,n}^c(L) \cup D_{n+1,n+3}^c(L) \cup C_{1,n}^c(L) \cup C_{n+1,n+3}^c(L) \cup C_n^{ccon}(L)$.

Lemma 2. *Let L^* denote minimum dominating path of $L_{m,n}$, and L denote a minimum dominating path of $L_{m,(n+3)}$. Then there exists at least one minimum dominating path L in $L_{m,(n+3)}$, such that the number of dominating vertices of path L is no less than the number of dominating vertices of path L^*. That is, $|D_{1,n}^c(L)| \geq |D(L^*)|$.*

Proof: We have $|D_{1,n}^c(L)| \geq |D(L_{m,n})|$ if all vertices in $G_{1,n}^c$ is dominated by $D_{1,n}^c(L)$ since $G_{1,n}^c = G^*$. Therefore, if $|D_{1,n}^c(L)| < |D(L^*)|$, some vertices in G_n^c must be dominated by $D_{n+1}^c(L)$ and do not have neighbors in $D_{1,n}^c(L)$.

Consider there are k continuous vertices v_k in G_n^c dominated by $D_{n+1}^c(L)$.

If $k \geq 2$, since these k vertices are not dominated by $D_{1,n}^c(L)$, the corresponding k vertices $v_{k'}$ which are in the same row with v_k in G_{n-1}^c can not belong to $D_{1,n}^c(L)$. Therefore, k vertices $v_{k''}$ in G_{n-2}^c should belong to $D_{1,n}^c(L)$ to dominate $v_{k'}$ because only two endpoints in $v_{k'}$ could be dominated by its top and bottom vertex instead, but their right neighbors could not belong to L which makes L irregular. So we could use the corresponding vertices in G_{n-2}^c to replace them so as to shorten L. Then, we could use $v_{k''}$ to construct. As shown in .

Since $L_{n+1,n+3}^c$ may have multiple connected components, L may step into $G_{n+1,n+3}^c$ and then move out $G_{n+1,n+3}^c$ or just move to the end point.

In the first case, since L may move out from $G_{n+1,n+3}^c$, we could construct as Fig and add connecting vertex to corresponding position.

In the second case, when D_{n+1}^c come from D_{n+2}^c, we have the following three cases. When $|v_k| > 3$, L would need more vertices in G_{n+3}^c to dominate vertices in G_{n+2}^c. And we could use similar construct like the first case. When $|v_k| < 3$, L will need more connecting vertices which could also use the same construct. When $|v_k| = 3$ and vertex in v_k do not reach G_m^r, then vertex below $v_{k'}$ must belong to L. So vertices in Fig is a dominating path for $L_{6,n+3}$ partial but can not reach the minimum so that L could not be the minimum dominating path because the form of minimum dominating path for has same start pointing as L.

When D_{n+1}^c do not come from D_{n+2}^c, L would step into $G_{n+1,n+3}^c$ in the first row, move down to $v_{m-1,n+1}$ and use vertices in G_{n+3}^c to dominate remain vertices. Since $v_{1,n+1} \in L$, v_k starts from G_3^r. Therefore, we could use similar construct before in Fig to replace L to another dominating path L^* where $|L^*| = |L|$

If $k = 1$, then the vertex must lay in boundary otherwise it will need extra connecting vertices between G_{n+1}^r and G_{n+2}^r. Therefore, we assume $v_{1,n+1} \in L$. Then $v_{1,n}, v_{1,n-1} \notin L$ and one of $v_{1,n-2}$ and $v_{2,n-1}$ must belong to L to dominate $v_{1,n-1}$. If $v_{1,n-2} \in L$, L will turn to G_{n-1}^r to dominate vertices in G_n^r and it will bring more vertices than the following condition. If $v_{2,n-1} \in L$, we will have L like Fig (like $L_{4,11}$). This case could only exist once. We transform $L_{m,n+3}$ symmetrical. Then, $|D_{1,n}^c(L)| \geq |D(L_{m,n})|$.

■

Given L as the minimum dominating path of the grid $L_{m,n+3}$, let $D_{n+1,n+3}^c(L)$ denote the set of dominating vertices of path L from the $n+1$-th column to the $n+3$-th column of grid $L_{m,n+3}$; and let $C_n^{ccon}(L)$ denote the set of connecting vertices of path L between the n-th column and the $n+1$-th column. Then, we have the following lemma.

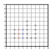

Fig. 2: The viewing distribution of a hot message originated from Beijing.

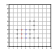

Fig. 3: The change of modularity towards Information Increment.

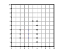

Fig. 4: Performance of UDM on WM and Gowalla datasets.

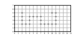

Fig. 5: Performance of UDM on WM and Gowalla datasets.

Lemma 3. *The number of dominating vertices of path L from the $n+1$-th column to the $n+3$-th column of grid $L_{m,n+3}$, plus the number of connecting vertices of path L between the n-th column and the $n+1$-th column is no less than m. That is $|D^c_{n+1,n+3}(L)| + |C^{ccon}_n(L)| \geq m$. Further, if $3 \nmid m$, then $|D^c_{n+1,n+3}(L)| + |C^{ccon}_n(L)| \geq m+1$.*

Proof: Since G^c_{n+1} might be dominated by $D^c_{1,n}(L)$, we consider the coverage problem of $G^c_{n+2,n+3}$ only.

Before formal proof, we will prove that expect for one single case, $G^c_{n+2,n+3}$ is dominated by rows. Specifically, every row in $G^c_{n+2,n+3}$ is dominated by one connected component in $L^c_{n+1,n+3}$.

If G^r_i in $G^c_{n+2,n+3}$ is dominated by two connected components in $L^c_{n+2,n+3}$, then we assume $v_{i,n+2}$ is dominated by a component above and $v_{i,n+3}$ is dominated by the other component beneath.

Therefore, there are two different scenarios. Under the first scenery, G^r_i is dominated by two end vertices like Fig which can be transformed by extending one vertex to dominate all vertices dominated by two components. Under the second scenery, G^r_i is dominated by one end vertex and one intermediate vertex. This is the unique case that could not be replaced. But we could take them as one part since the union of two components follows the result.

Then, we prove the lemma by induction. Obviously, when $m = 1$, $|D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq 2$, when $m = 2$, $|D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq 3$ and when $m = 3, |D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq 3$.

Now assume the result holds for $m = k$. When $m = k+1$, if there is only one connecting component in $L^c_{n+1,n+3}$, $|\gamma_c(G^c_{n+2,n+3})| \geq m$. Adding 1 connecting vertex in G^c_{n+1}, $|D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq m+1$. If there are multiple connecting components in $L^c_{n+1,n+3}$, we assume $G^c_{n+2,n+3}$ is dominated by rows. If a rows and b rows are dominated by two connected components L_a and L_b respectively. If $3 \mid m$, then at most two of a and b could be divided by 3 so that $|(D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)) \cap (L_a \cup L_b)| \geq a+b$. If $3 \nmid m$, then at most one of a and b could be divided by 3 so that $|(D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)) \cap (L_a \cup L_b)| \geq a+b+1$. Therefore, multiple connected components can finally reduce to one component which also holds the result. ∎

2) How many vertices are needed for finding the minimum dominating path in $L_{m,n+3}$: Based on the results of Lemma 2 and Lemma 3, we decompose a minimum dominating path in $L_{m,n+3}$ into two parts: (1) the left part from 1st column to the n-th column of $L_{m,n+3}$; and (2) the right part from $n+1$-th column to the $n+3$-th column of $L_{m,n+3}$.

Since the left part is a grid $L_{m,n}$, the following theorem tells the relationship between the minimum dominating paths of $L_{m,n+3}$ and $L_{m,n}$.

Theorem 4. *The minimum dominating paths of $L_{m,n+3}$ contains at least m more vertices than that of $L_{m,n}$. That is $\gamma_l(G) \geq \gamma_l(G^*) + m$. Further, when $3 \nmid m$, $\gamma_l(G) \geq \gamma_l(G^*) + m + 1$.*

Proof: In Lemma 2 and Lemma 3, we know that there exist at least one minimum dominating path L in $G = L_{m,n+3}$, $|D^c_{n+1,n+3}(L)| \cup |C^{ccon}_n(L)| \cup |D^c_{1,n}(L)|$ could fulfill additive part in result. Therefore, if the result is false, $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)|$ in G must be less than $C(L^*)$ of a minimum dominating path L^* in G^*.

Connectivity on the boundary: Because connectivity depends on structure of G^r_n and there are only two start vertices in L, we have structures in G^r_n like Fig. We will consider different structures of L in G^r_n.

If there are only one connecting vertex $v_{i,n}$ in connected component, we have following two cases. In the first case, like Fig , we have $v_{i-1,n+1}$, $v_{i-1,n+2}$, $v_{i+1,n+1}$, $v_{i+1,n+2}$, $v_{i,n+2} \in L$. In this case, although $v_{i-1,n+1}, v_{i+1,n+1} \in C^{ccon}_n$ which decreases $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)|$, but $|C^{ccon}_n(L) \cup D^c_{n+1,n+3}(L)| = 5$ which still use 4 vertices addition to dominate 3 rows. In the second case, we have $v_{i-1,n+1}$, $v_{i-1,n+2}$, $v_{i-1,n+3}$, $v_{i+1,n+1}$, $v_{i+1,n+2}$, $v_{i+1,n+3}$, $v_{i,n+3} \in L$ where $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)| = |C(G^*)|$.

If there are two connecting vertices in connected component, when we consider them separately, result is the same as one connecting vertex. If we consider them together, the size of L in G^r_n must be 4 as shown in Fig . We assume two vertices are $v_{i,n}$ and $v_{i+1,n}$. Also, we have two cases. In the first case, we have $v_{i-1,n+1}$, $v_{i-1,n+2}$, $v_{i-1,n+3}$, $v_{i,n+3}$, $v_{i+1,n+3}$, $v_{i+2,n+1}$, $v_{i+2,n+2}$, $v_{i+2,n+3} \in L$, also $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)| = |C(L^*)|$. In the second case, $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)|$ decreases 2, $|C^{conn}_n|$ add 2 and $|D^c_{n+1,n+3}|$ add 4 which use only 4 vertices addition to dominate 4 rows. It is less than additive part in result. However, this kind of structure could exist when n is small since to reach that structure, G^r_{i-2} and G^r_{i+3} must be dominated by other components and this destroy the 3 periodic structure which will add more vertices in dominating path. Specifically, if there are two 2-period extends, the dominated rows will decrease more than gain in the dominating vertices. And if there are two 3-period extends, it could be replaced by similar structure which move two start points to G^r_{n+3}. Therefore, there are only two possibilities that the structure could exist. In the first possibility, $m \equiv 2 \pmod 3$ and $m \geq 11$, as shown in Fig , we have the structure. Consider the structure in $L_{m',n'}$.

We assume the structure first appear when $n' = b$. Since $|L^c_{n'-2,n'}| = m'$, minimum dominating path L' in $n' = b-3$ must hold the same structure which drop $m'+1$ vertices in origin L'. And this structure will hold till $n' \leq 3$. Except for $n' = 1$ which is out of range, $n' = 2$ or $n' = 3$ do not have the same structure which makes contradictions. In the second possibility, $m' \equiv 1 \pmod 3$. However, we could simplify this case, since when $n' \equiv 0 \pmod 3$ or $n' \equiv 2 \pmod 3$, we could change m' and n'. Therefore, we consider $m' \equiv 1 \pmod 3$ and $n' \equiv 1 \pmod 3$ only. Then, we could simplify to $L_{m',n'}$ to $L_{m',1}$ with the same approach like the first possibility which is also out of range.

Connectivity in the middle: In this case, we split $L^{*c}_{1,n}$ on middle and connect two connected components in $L^c_{n+1,n+3}$ and generate L. We assume the origin endpoints in L^* lay in G^{*c}_n, otherwise we need more connecting vertices between $L^c_{1,n}$ and $L^c_{n+1,n+3}$. If $C^c_{1,n}(L)|\cup|C^c_{n+1,n+3}(L)$ could be less than $|C(G^*)|$, we consider $L^c_{1,3}$. In Lemma 1, $|L^c_{1,3}| \geq m$ and if $3 \nmid m$, $|L^c_{1,3}| \geq m+1$. Therefore, we could move connecting vertices from $C^c_{1,3}$ to $C^c_{4,n}$ if necessary without add more vertices since there are no start point in G^{*r}_4 and we could reduce the extend from minimum dominating path in $L_{m,n-3}$ to L^*. Then, we could construct a minimum dominating path L^{**} from $L^c_{4,n+3}$ and $|L^{**}| < \gamma_l(G^*)$ and this makes contradiction.

Therefore, since $|C^c_{1,n}(L)| \cup |C^c_{n+1,n+3}(L)|$ is no less than $C(L^*)$, we could prove the result.

■

3) Meta structs for constructing the minimum dominating path in $L_{m,n+3}$: From Theorem 4, we could also know that when $n = 1$, $\gamma_l(G) \geq \gamma_l(G^*) + m$.

Struct 1: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m$.

Struct 2: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m+1$.

Struct 3: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m+2$.

Assume $G = L_{m,n+3}$ and $G^* = L_{m,n}$, we denote **standard extend row** as the minimum dominating path of $L_{3,3}$. It uses only 3 vertices to dominate 3 rows which is also scalable. And it is the only structure that reach $\gamma_l(G) = \gamma_l(G^*) + m$ when $3 \mid m$ while it is one of structures that reach $\gamma_l(G) = \gamma_l(G^*) + m + 1$ when $3 \mid m+1$ which satisfy additive part in Theorem 4. Besides, we could combine standard extend row with other structure while hold minimality of vertices.

Theorem 5. *We denote s_3 $extend$ as extending three columns right from $L_{m,n}$ to $L_{m,n+3}$ where $|L_{m,n+3}| - |L_{m,n}| = s$. Assume $G = L_{m,n}$, there are only finite structure for m_3 $extend$ and $(m+1)_3$ $extend$.*

Proof: Multiple connected components with two start points can transform into one connected components. Therefore, we could consider the coverage problem with one start point or from middle of path.

Extend from start point: We consider 4 different cases of $(m+1)_3$ $extend$ from start point in $L^c_{n+1,n+3}$. As shown in Fig .

For case (a), we need $v_{1,n}$ or $v_{m,n}$ to connect L. For case (b), we need $v_{1,n}$ to connect. For case (c) and (d), we need corresponding vertices in G^r_n to dominate uncovered vertices in G^r_{n+1}.

However, case(a), case (b) and case (d) can not guarantee minimality of L for further extends. Under case (b), L do not reach G^r_{n+3} which is still a $(m+2)_3$ $extend$. Under case(a), except for $v_{m,n}$, L need at least m+3 vertices in next 3-period so we could use case(b) replace case(a). Under case (d), except for standard extend row, to extend 3 columns further, L will need two more extra vertices to connect. Therefore, we have at most one case (b) or case (d) in last period.

Under case (c), we consider previous 3-period of L. Since G^r_{n+1} should be dominated by corresponding vertices in G^r_n, $v_{1,n-1}$ should be dominated by G^r_{n-2}. Therefore, there are $2m$ vertices in previous period. Therefore, we could have at most one case (c).

Extend from middle: Obviously, we could use standard extend row extend from middle.

Except for pure standard extend row, there is still one structure in Theorem 4. With standard extend row, when $n = 1$, it could bring m_3 $extend$, and when $n \equiv 2 \pmod 3$ and $n \geq 11$, it could bring $(m+1)_3$ $extend$. But they can only occur once.

Rotate direction: Since we may change the origin structure of L, we could rotate direction of extend direction by 90 degree. Therefore, when $m \equiv 0 \pmod 3$, we may use standard extend row on rotated structure which could bring a $(m+1)_3$ $extend$. But similarly, this method could only exist once for the same m.

Therefore, there are only several cases for m_3 $extend$ and $(m+1)_3$ $extend$.

m_3 $extend$:

- When $m \equiv 0 \pmod 3$, pure standard extend row.
- When $n = 1$, structure in Theorem 4.

$(m+1)_3$ $extend$:

- Once case (c).
- Once case (b) or (d).
- Once Rotate direction with standard extend row on rotated structure.
- When $n \equiv 2 \pmod 3$ and $n \geq 11$, structure in Theorem 4.

■

C. Optimal Trajectory Planning Algorithms

We have four cases to construct minimum dominating path L in grid $G = L_{m,n}$: (1) $m < 4$ and $m < n$; (2) $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$ or $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$; (3) $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$ or $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$ or $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$; and (4) $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$.

1) Case 1: Since connected dominating set is also dominating path when $m < 4$ and $m < n$, we have the same structure as shown in Figure of [4]. Therefore, we have

1) $\gamma_l(L_{1,1}) = \gamma_l(L_{1,2}) = 1$, and if $3 \leq n$, $\gamma_l(L_{1,n}) = n-2$
2) $\gamma_l(L_{2,2}) = \gamma_l(L_{2,3}) = 2$, and if $4 \leq n$, $\gamma_l(L_{2,n}) = n$
3) $\gamma_l(L_{3,n}) = n$

2) Case 2: Since minimum dominating path in $Lm, 1$ contains $m - 2 - \lceil \frac{m}{3} \rceil$ connecting vertices, standard extend row can be used to construct a $m_3 extend$ or $(m+1)_3 extend$. Further, from Theorem 5, we know that when $m \equiv 2 \pmod 3$ and $n = 1$, we could construct a $m_3 extend$. Therefore, when $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$ or $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$, we could use standard extend row to construct minimum dominating path, as shown in Figure . And when $m \equiv 2 \pmod 3$, we have structure shown in Figure When $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$, we have $\gamma_l(L_{m,n}) = 3ab + 3a - 2$. When $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$, we have

$$\gamma_l(L_{m,n}) = \begin{cases} 3ab + 3a + 3b & a \leq 2 \\ 3ab + 3a + 3b - 1 & otherwise \end{cases}$$

.

3) Case 3: When $a = 1$, $\gamma_l(L_{3,n}) = 3b + 2$. And the two structures is as shown in Fig . Since $n \equiv 2 \pmod 3$, we could use $(m + 1)_3 extend$ when $n \geq 11$. Assume we have proved the case when $a \leq k$, we consider the case when $a = k+1$. When $b \leq \lfloor \frac{k}{2} \rfloor$, $\gamma_l(L_{m,n}) = 3kb+3k+2b-2$ has structure (a), we could use only standard extend row to reach a $(n+1)_3$ $extend$ which reaches lower bound. When $b > \lfloor \frac{k}{2} \rfloor$, $\gamma_l(L_{m,n}) = 3kb+4k-2$ has structure (b), and we could only have a $(n+2)_3$ $extend$ from its structure. Since we could get $\gamma_l(L_{3k+3,3b+2}) = 3kb + 3k + 5b + 1$ which has structure (a). Since we do not have structure in 5 for minimum dominating path in both $L_{3k+3,3b+2}$ and $L_{3k,3b+5}$. We could only use standard extend row in structure (a) or (b) and the result is also in structure (a) or (b). Therefore, we could use standard extend rows in structure (b) to reach optimal solution in Theorem 4. We could use reduction to prove the result on the other two cases. When $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$, assume $m \leq n$, we use standard extend row to extend columns. Similarly, When $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$, assume $m \leq n$, we use standard extend row to extend column and when $a > 2$, we use structure in Fig. once.

4) Case 4: Consider the condition when $a = 1$. When $b = 1$, we have $\gamma_l(L_{4,4}) = 8$ with following structure. Therefore, we could use $(m + 1)_3$ $extend$ case (b) and case (c) respectively to get $\gamma_l(L_{4,7}) = 13$ and $\gamma_l(L_{4,10}) = 18$. Till now, $L_{4,10}$ is a m_3 $extend$ for $L_{1,10}$ using ... and reach the lower bound on Theorem 4. When $b \geq 4$, we could also get m_3 $extend$ on $L_{4,n}$ using standard extend row on the extend of $L_{4,10}$. Therefore, when $b \geq 3$, we have $\gamma_l(L_{4,3b+1}) = 6b$. When $a = 2$ and $b = 2$, we could use a $(m + 1)_3$ $extend$ on $L_{4,7}$ and get $\gamma_l(L_{7,7}) = 21$ When $a \geq 2$, we know that case (b), case (d) and structure in Fig could only exist once. Therefore, when $b \geq 3$, we have $\gamma_l(L_{3a+1,3b+1}) = 3ab + 3a + 3b - 3$ which is $a - 1$ times $(m + 2)_3$ $extend$ from $L_{4,3b+1}$. Therefore, when $a + b \leq 4$, we could use a $(m + 1)_3$ $extend$ for each add one on a or b on the basis of $L_{4,4}$.

D. *How to concatenate 2D trajectories to a 3D one*

V. Finding the COLs from the minimum dominating path

A. *Consider total time consuming*

From last subsection, we know how to find the minimum dominating path in $L_{m,n}$. Therefore, we just have to find the set of COLs V_C in dominating path L. However, since we have already discussed dominating vertices in last subsection, we only need to add all dominating vertices into V_C. In general, we only have 1 case for COL which is $standardextendrow$ and in this case, we just take the entire row into V_C and add

Therefore, for the 6 different cases, we have the following result.

Proposition 1.

1) $\gamma_l(L_{1,n}) = \lfloor \frac{n+2}{3} \rfloor$
2) $\gamma_l(L_{2,2}) = \gamma_l(L_{2,3}) = 2$, *and if* $4 \leq n$, $\gamma_l(L_{2,n}) = n$
3) $\gamma_l(L_{3,n}) = n$

Proposition 2. *Assume* $m \geq 4$, $n \geq 4$. *When* $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$, *we have* $|V_C| = 3ab + a$. *When* $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$, *we have*

$$|V_C| = \begin{cases} 3ab + a + 3b + 1 & a \leq 2 \\ 3ab + a + 3b + 2 & otherwise \end{cases}$$

.

Proposition 3. *Assume* $m \geq 4$, $n \geq 4$. *When* $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$, *we have* $|V_C| = 3ab$. *When* $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$, *we have*

$$|V_C| = \begin{cases} 3ab + 2a & b \leq 2, 3ab + 4a - 2 \leq 3ab + 3a + 2b - 1 \\ 3ab + 3a & b \leq 2, 3ab + 4a - 2 > 3ab + 3a + 2b - 1 \\ 3ab + 2a & b > 2, 3ab + 4a - 2 \leq 3ab + 3a + 2b - 2 \\ 3ab + 3a + 1 & otherwise \end{cases}$$

. *When* $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$, *assume* $m \leq n$, *we have*

$$|V_C| = \begin{cases} 3ab + 2a + 3b + 2 & a \leq 2 \\ 3ab + 2a + 3b + 3 & otherwise \end{cases}$$

. *Besides, the results have a same structure.*

Proposition 4. *Assume* $m \geq 4$, $n \geq 4$. *When* $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$, $a \geq 1$ *and* $b \geq 1$, *assume* $m \leq n$, *we have*

$$|V_C| = \begin{cases} 7 & a = 1, b = 1 \\ 10 & a = 1, b = 2 \\ 17 & a = 2, b = 2 \\ 3ab + 3a + b - 1 & otherwise \end{cases}$$

.

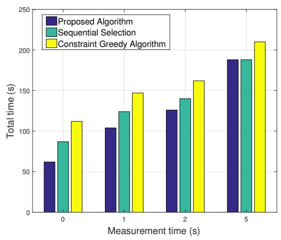

Fig. 6: Time consuming of three algorithms with different granularity of measurement time.

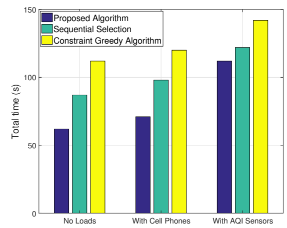

Fig. 7: Battery coverage of three algorithms with different granularity of measurement time.

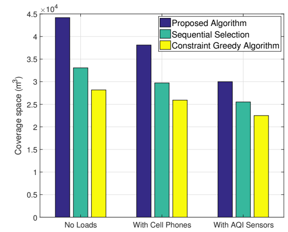

Fig. 8: Time consuming of three algorithms with different loads when considering flight time only.

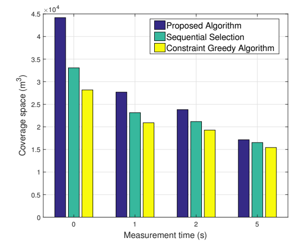

Fig. 9: Battery coverage of three algorithms with different loads when considering flight time only.

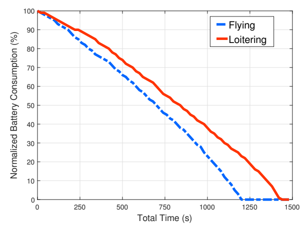

Fig. 10: Comparison of the Adaptive Monitoring Algorithm, Greedy Algorithm and Sequential Selection.

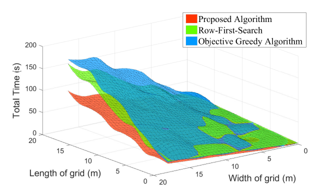

Fig. 11: Comparison of the Adaptive Monitoring Algorithm, Greedy Algorithm and Sequential Selection.

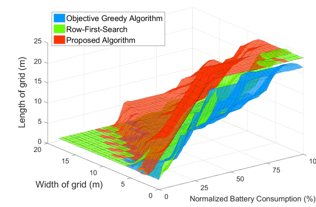

Fig. 12: Comparison of the Adaptive Monitoring Algorithm, Greedy Algorithm and Sequential Selection.

B. Optimal Trajectory Planning Algorithms

VI. Evaluation

A. Experiment Setup

Experimental preconditions: We use drones to verify our results. Since we divide 3D space into OL-network by cuboids which forms a 3D grid partitioned by height, we only consider algorithms in 2D grid.

Comparison algorithms: We consider two different algorithms for comparison. Since in our algorithm we choose the minimum dominating path in grid first and select COLs from path, we consider conventional algorithms as comparison which select COLs in grid first and choose a path to connect them.

1) *Row-first COL First*: We select the minimum dominating set in grid first and select an OL at boundary as start point. Then, we use row-first form a suboptimal trajectory by prior choosing OLs row by row (the longer side of the grid).
2) *Greedy COL First*: We select the minimum dominating set in grid first and select a point at boundary as start point. Then, we follow the greedy algorithm that

$$\min_{d(v,v')} \max_{N[y]-N[P]} v,$$

where $v, v' \in NG$, $P \subset NG$ is selected vertices of path, and v' is the last vertex in path.

Performance metrics: We use the following two metrics for evaluating algorithm performance.

- *Time Consuming*: The time consuming is defined as total time of flight and measurement of drone to cover the space under specific algorithm.
- *Battery Coverage*: The battery coverage is defined as maximum coverage area during entire battery life. Since algorithm is affected by input width and length, We fix the width of grid.

B. Results in specific open space

We choose a open space in university which has a size of 45 m length, 35 m width and 10m height, and it is divided into a 9*7*2 3D grid by 5*5*5 cuboids. We consider the 2D 9*7 grid in space. Since time consuming is the sum of flight time and measurement time, we consider measurement time of different granularity which is 0, 1, 2, 5 seconds for each COL.

Figure 6 show the time consuming in grid of different algorithms on different granularity of measurement time. We observe that our algorithm uses less time when consider flight time only. The gap between our algorithm and sequential algorithm decreases as measurement time increases and when measurement time is 5 seconds total time of two algorithms are equal.

Figure 7 show the battery coverage in grid of different algorithms on different granularity of measurement time. We fix the width of grid as n. Similar with time consuming, when consider flight time only, our algorithm could covers more area than other algorithms. And as measurement time increases, coverage areas of three algorithms as well as difference among three algorithms decrease and when measurement time is 5 seconds, coverage sets of three algorithms are nearly equal.

In a real scenario, drones might be used to delivery packages or other goods. Therefore, we also consider time consuming and battery coverage with different loads when considering flight time only. Results in Figure 8 and 9 show that as load increases, coverage set of three algorithms decreases and time consuming of three algorithms increases. Meanwhile, gap among three algorithms decrease as well.

C. Results of different measurement space

Figure 10 show the flight time in different measurement spaces with different width and length. We observe that our algorithm uses less time in every grid. And fixing width (or length), gap among algorithms increases as length (or width) increases.

Similarly, we have the same result when considering battery coverage in Figure 12. When fixing width (or length), our algorithm could cover more area than other algorithms.

We also observe that, when fixing width, as length increases, the ratio between our algorithm and row-first algorithm could converge to a periodic result.

VII. CONCLUSIONS

REFERENCES

[1] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath. Real-time air quality monitoring through mobile sensing in metropolitan areas. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 15. ACM, 2013.

[2] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):6, 2009.

[3] Y. Fu, M. Ding, and C. Zhou. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for uav. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(2):511–526, 2012.

[4] P. Hamburger, R. Vandell, and M. Walsh. Routing sets in the integer lattice. *Discrete Applied Mathematics*, 155(11):1384–1394, 2007.

[5] Y. Hu and S. X. Yang. A knowledge based genetic algorithm for path planning of a mobile robot. *Acta Electronica Sinica*, 34(5):4350–4355 Vol.5, 2006.

[6] D. L. Newman. Drone for collecting images and system for categorizing image data, 2014.

[7] A. Purohit, Z. Sun, F. Mokaya, and P. Zhang. Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications. In *International Conference on Information Processing in Sensor Networks*, pages 223–234, 2011.

[8] K. Savla, F. Bullo, and E. Frazzoli. On traveling salesperson problems for dubins ¡¯ vehicle: stochastic and dynamic environments. In *Decision and Control, 2005 and 2005 European Control Conference. Cdc-Ecc '05. IEEE Conference on*, pages 4530–4535, 2005.

[9] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis. Using mobile robots to harvest data from sensor fields. wirel commun. *Wireless Communications IEEE*, 16(1):22–28, 2009.

[10] E. Yu, X. Xiong, and X. Zhou. Automating 3d wireless measurements with drones. In *Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 65–72, 2016.

[11] S. C. Yun, V. Ganapathy, and L. O. Chong. Improved genetic algorithms based optimum path planning for mobile robot. In *International Conference on Control Automation Robotics & Vision*, pages 1565–1570, 2011.

Algorithm 1 Trajectory planning for UAV in grid when consider flight time only

Input: The graph of trajectory planning, $L_{m,n}$; The length of grid, m; The width of gird, n;
Output: The trajectory of two-dimensional coordinate of grid, V_P;

$a = \lfloor \frac{m}{3} \rfloor$, $b = \lfloor \frac{m}{3} \rfloor$, $ra \equiv m \pmod 3$, $rb \equiv n \pmod 3$;
if ($ra = 1$ and $rb = 0$) or ($ra = 1$ and $rb = 2$) or ($ra = 0$ and $rb = 2$ and $a > 2b$) or ($ra = 2$ and $rb = 0$ and $b \leq 2a$) or ($ra \equiv rb \pmod 3$ and $a > b$) **then**
    swap(m, n), swap(a, b), swap(ra, rb);
end if
if $m < 4$ or $n < 4$ **then**
    if $m > n$ **then**
        swap(m, n), swap(a, b), swap(ra, rb);
    end if
    if $m = 1$ **then**
        put G_1^r except $v_{1,1}$ into V_P;
        if $n = 1$, **then** put $v_{1,1}$ into V_P;
    else if $m = 2$ and $n = 3$ **then**
        put $v_{1,2}$, $v_{2,2}$ into V_P;
    else if (**then**$m = 2$)
        put G_1^r into V_P;
    else
        put G_2^r into V_P;
    end if
else if $m \equiv 0 \pmod 3$ **then**
    put G_2^r, G_5^r ... , G_{3a-1}^r into V_P;
    put connecting vertices in G_1^c and G_n^c into V_P;
else if $m \equiv 2 \pmod 3$ **then**
    if $a > 2$ **then**
        put $v_{2,1}$, $v_{3,1}$, $v_{6,2}$, $v_{7,2}$ into V_P;
        put G_2^r, G_5^r, G_8^r except vertices in G_1^c into V_P;
        put G_1^r0, G_1^r3..., G_{3a+1}^r into V_P;
        put connecting vertices in G_1^c and G_n^c into V_P;
    else
        put G_2^r into V_P;
        put G_4^r, ..., G_{3a+1}^r into V_P;
        put connecting vertices in G_1^c and G_n^c into V_P;
    end if
else
    if $a \leq 2$ and $b \leq 2$ **then**
        if $a \geq 1$ and $b \geq 1$, **then** put $v_{1,2}$, $v_{2,2}$, $v_{3,2}$, $v_{4,2}$, $v_{4,3}$, $v_{4,4}$, $v_{3,4}$, $v_{2,4}$ into V_P;
        if $a \geq 1$ and $b = 2$, **then** put $v_{2,5}$, $v_{2,6}$, $v_{2,7}$, $v_{3,7}$, $v_{4,7}$ into V_P;
        if $a = 2$, **then** put $v_{5,7}$, $v_{6,7}$, $v_{6,6}$, $v_{6,5}$, $v_{6,4}$, $v_{6,3}$, $v_{6,2}$, $v_{6,1}$ into V_P;
    else
        put $v_{1,2}$, $v_{2,5}$, $v_{2,6}$ into V_P;
        put G_2^c, G_4^c, G_7^c except vertices in G_1^r into V_P;
        put G_9^c, G_1^c2, ..., G_{3b}^c into V_P;
        put connecting vertices in G_1^r and G_m^r into V_P;
    end if
end if
        return V_P;

Algorithm 2 Trajectory planning for UAV in grid when consider total time consuming

Input: The graph of trajectory planning, $L_{m,n}$; The length of grid, m; The width of gird, n;

Output: The trajectory of two-dimensional coordinate of grid, V_P; The COL set of two-dimensional coordinate of grid, V_C;

do algorithm 1 and get V_P.
$a = \lfloor \frac{m}{3} \rfloor$, $b = \lfloor \frac{m}{3} \rfloor$, $ra \equiv m \pmod 3$, $rb \equiv n \pmod 3$;
if $m < 4$ or $n < 4$ **then**
    if $(m = 1$ and $n \leq 3)$ or $m \geq 2$ **then**
        $V_C = V_P$;
    else
        put $v_{1,2}$, $v_{1,5}$, ..., $v_{1,3b-1}$ into V_C;
        If $rb = 1$ or $rb = 2$, **then** put $v_{1,3b+1}$ into V_C
    end if
else if $m \equiv 0 \pmod 3$ **then**
    put G_2^r, G_5^r ... , G_{3a-1}^r into V_C;
else if $m \equiv 2 \pmod 3$ **then**
    if $a > 2$ **then**
        put $v_{2,1}$, $v_{3,1}$, $v_{6,2}$, $v_{7,2}$ into V_C;
        put G_2^r, G_5^r, G_8^r except vertices in G_1^c into V_C;
        put G_1^r0, G_1^r3..., G_{3a+1}^r into V_C;
    else
        put G_2^r into V_C;
        put G_4^r, ..., G_{3a+1}^r into V_C;
    end if
else
    if $a \leq 2$ and $b \leq 2$ **then**
        if $a \geq 1$ and $b \geq 1$, **then** put $v_{1,2}$, $v_{2,2}$, $v_{3,2}$, $v_{4,2}$, $v_{4,4}$, $v_{3,4}$, $v_{2,4}$ into V_C;
        if $a \geq 1$ and $b = 2$, **then** put $v_{2,5}$, $v_{2,6}$, $v_{2,7}$, $v_{4,7}$ into V_C and move $v_{3,4}$ out from V_C;
        if $a = 2$, **then** put $v_{6,7}$, $v_{6,6}$, $v_{6,5}$, $v_{6,4}$, $v_{6,3}$, $v_{6,2}$, $v_{6,1}$ into V_C;
    else
        put $v_{2,4}$, $v_{2,5}$, $v_{2,6}$, $v_{2,7}$ into V_C;
        put G_2^c into V_C;
        put G_4^c, G_7^c except for $G_{1,3}^r$ into V_C;
        put G_{10}^c, G_{13}^c, ..., G_{3b}^c into V_C;
    end if
end if
    return V_P, V_C;