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 space.
- 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

Mobile sensing is the process of using the sensors of a mobile device to acquire data from the environment. Traditional mobile sensing uses traffic tools like bikes [**?**] or cars [**?**] to realize mobility.

Nowadays, drones are popular in mobile sensing applications since they could capture different dimensions of signals in the environment that are beyond our sensing capability, such as aerial photo or video-graphy [?], 3D wireless signal survey [?], inspection in inaccessible places where people rarely venture [?]. However, drones are constrained by its limited battery life which require an efficient measurement approach to overcome the limitations.

B. Trajectory planning in mobile sensing

In conventional mobile sensing system, genetic algorithm [?], [?] and particle swarm optimization [?] are always proposed for real-time path planning which could find an optimal or near-optimal robot path in both complex static and dynamic environments. It is now embedded into UAVs which could ensure partial minimality between two nodes.

On the other hand, large scale mobile sensing could be formulated as a ordinary Travelling Salesman Problem (TSP) or a special TSP problem [?], [?]. Also, some heuristic algorithms like Life Long A* [?] are developed to find shortest paths for path planning which repeatedly finds shortest paths from a given start vertex to a given goal vertex while the edge costs of a graph change or vertices are added or deleted. For conventional trajectory planning in mobile sensing, the first step is to find measurement locations and the second step is to find a path connect them. In this paper, we will give a different approach that reverses order of path finding and locations selecting, and we will give an algorithm based on graph theory which could be optimal if considering flight time only.

III. System Model

In this section, we establish a three-dimensional (3D) network model that characterizes the ordinary mobile sensing scene for drone. Then, we analyse correlation between OLs and relationship between 3D and 2D network model. Afterwards, we formulate optimal trajectory problem as a constraint set coverage problem and we consider two special scenes respectively in this paper. In the first scene, we consider flight time only and formulate problem as a minimum dominating path problem. In the second scene, we consider total time consuming and formulate problem as a combination of minimum dominating path problem and constraint minimum dominating set problem. We will make further discussion in next subsection. Finally, we define variables that would be used to mathematical proof next section.

A. Network establishment

Dividing 3D space into cuboids: We assume sensing object is largely same in fixed area for every position in 3D space. So 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 4), which is denoted by the 3-tuple (longitude, latitude, and altitude), i.e.,

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

where x_i, y_i, z_i are 3D coordinates of OL_i.

3D network of OLs: OLs form a 3D network graph $G = (V; E)$, where V denotes the set of vertices and E represents the edges connecting neighboring vertices. Specifically, the OL inside each cuboid i is considered as a vertex in G, and an edge (i, j) exists if cuboid i is the same as cuboid j in two coordinates and adjacent to cuboid j on the third dimension. Therefore, OL-network forms a three-dimensional grid which has fine topology structure.

COL: Obviously, if we take all of OLs in the network into consideration, it would be too time consuming. Therefore, we select critical observation locations (COL) from OLs to cover whole OL network while find trajectory to cover COLs.

B. Time consuming

For general mobile sensing, total time consuming consists of flight time and measurement time which depends on both COLs and trajectory. We denote $V_C \subseteq V$ as the set of COLs and v_{C_i} as $i-th$ vertex in V_C. And we also denote $V_P \subseteq V$ as the trajectory of UAV which forms a path in G and we have $V_C \subseteq V_P$ since trajectory should contain vertexes in V_C.

Measurement time: Measurement time T_M is time consuming in mobile sensing. From empirical view, we could assume that measurement time is the same for all OLs. So measurement time is proportion to the number of COLs we select. The function is written as

$$T_M = t_M |V_C|,$$

where t_M is the measurement time for each OL.

Flight time: Flight time T_F is time consuming in drones' flight. Since we formulate 3D space into a 3D grid, we use Hamiltonian distance to characterize distance between OLs. So flight time is proportion to the length of trajectory. The function is written as

$$T_F = t_F |V_P|,$$

where t_F is the flight time for unit length of coordinate system.

Therefore total time consuming T is

$$T = T_M + T_F.$$

C. Correlation between OLs

To characterize general mobile sensing, we assume adjacent OLs have correlation. To characterize different adjacency, we consider following two typical scenes in mobile sensing.

Star adjacency: In this scene, we assume OLs have star adjacent neighborhood relationship so the sum of three coordinates difference is at most 1 and the coverage set of an OL is the union of its vertex adjacent neighbors and itself. Specifically, an OL has 2 neighbors in every dimension and the total size of coverage set is 7.

Cubic adjacency: In this scene, we assume OLs have cubic adjacent neighborhood relationship so the max of three coordinates difference is at most 1 and the coverage set of an OL is OLs in a cube whose center is the target. Specifically, an OL has 8 neighbors in each plane and the total size of coverage set is 27.

D. Problem formulation

Given a 3D space, we first establish a 3D OL network $G = (V; E)$ which forms a 3D grid. Each OL in grid has a coverage set. Due to drones' limited battery life, we want to complete flight in the shortest time. Hence, in order to minimize total time consuming, we select COLs from OLs to cover OL-network while find a OL-path(trajectory) covers COLs. We formulate this problem as a constraint set coverage problem in 3D grid.

Simplification from adjacency: We have discussed about different adjacencies in the last subsection and in this subsection we can use them to simplify the problem.

1) *Star adjacency*: Under start adjacency, we often consider two dimensions only because two distant OLs in a line is usually unpredictable, i.e., OLs in different heights. Then, we could divide 3D grid graph into multiple 2D grids and consider set coverage problem in each grid.
2) *Cubic adjacency*: Under cubic adjacency, if a drone flight over a plane, then adjacent planes will also be covered. Then, we could also simplify 3D grid graph into multiple 2D grids.

Therefore, instead of 3D grid, we formulate the problem as a constraint set coverage problem in 2D grid.

Competition between flight time and consideration time: In order to minimize total time consuming, we want to minimize both flight time and consideration time. However, the minimality of flight time contradicts with the minimality of flight time. Therefore, since in most cases flight time would account for majority part of total time consuming, we give priority to flight time.

Two special cases of time consuming: Due to last topic, we would give priority to flight time in this paper. Therefore, we will consider following two scenes.

1) *Consider flight time only*: In this scene, we assume measurement time negligible and consider flight time only. As we show in, flight time is proportion to length of trajectory. Therefore, we could select every OL in trajectory as COL. We find the shortest OL-path in grid which could cover the whole OL-network and select OLs in path as COLs. We formulate this problem as a minimum dominating path problem in grid.
2) *Consider total time consuming*: In this scene, we consider both measurement time and flight time. Since we give priority to flight time, we find the shortest OL-path first and select least OLs as COLs for measurement in OL-path. We could formulate this problem as a combination of a minimum dominating path problem and a constraint minimum dominating set problem in grid.

Therefore, for two scenes above, we give definitions to them respectively.

Problem 1 (Consider flight time only)**.** *Given a 3D measurement space, we divide it into several 2D grid. For each grid $G = L_{m,n} = (V; E)$ where $V = \{v_0, v_1, ..., v_{|V|-1}\}$ is vertex set of G, assuming the drone's trajectory time consuming is only relevant to flight time, let $C(v_i)$ be the coverage set of vertex v_i in V, since $V_C = V_P$, we want to find the minimum path $V_P \subseteq V$ which contains COLs in V and covers all OLs in grid.*

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

Problem 2 (Consider total time consuming)**.** *Given a 3D measurement space, we divide it into several 2D grid. For each grid $G = L_{m,n} = (V; E)$ where $V = \{v_0, v_1, ..., v_{|V|-1}\}$ is vertex set of G, assuming the drone's trajectory time consuming is the sum of flight time and measurement time, let $C(v_i)$ be the coverage set of vertex v_i in V, we want to find the minimum path $V_P \subseteq V$ which covers all OLs in grid and select COLs from V_P as dominating set $V_C \subseteq V_P$.*

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

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

IV. Lower bound and construction for two cases

A. Definitions and notations

For proof in the following subsections, we define some variables in this subsection.

First, we give specific definitions of dominating set, connected dominating set and dominating path.

- Dominating Set: A dominating set for graph $G = (V, E)$ is a subset $D \subset V$ such that every vertex not in D has a neighbor on D.
- Connected Dominating Set: 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.
- Dominating path: 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. L consists of **dominating vertexes** which cover vertexes in G and **connecting vertexes** which connect dominating vertexes. A connecting vertex v is a vertex whose coverage set is covered by other vertexes while a dominating vertex is on the contrary.

Then, we give notations of variables.

- $G = (V, E)$: 3D OL network graph.
- $L_{m,n}$: Grid graph with m rows and n columns.
- G_i^c: Leftmost $i-th$ column in grid G.
- G_i^r: Topmost $i-th$ row in grid G.

- $G^c_{i,j}$: Columns between G^c_i and G^c_j.
- $G^r_{i,j}$: Rows between G^r_i and G^r_j.
- $v_{i,j}$: Vertex of intersection of G^r_i and G^c_j 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.
- $L^c_i = L \cap G^c_i$: Intersection between L and G^c_i.
- $L^r_i = L \cap G^r_i$: Intersection between L and G^r_i.
- $L^c_{i,j}(G) = L \cap G^c_{i,j}$: Intersection between L and $G^c_{i,j}$.
- $L^r_{i,j}(G) = L \cap G^r_{i,j}$: Intersection between L and $G^r_{i,j}$.
- $D(L)$: Dominating vertexes in L.
- $C(L)$: Connecting vertexes in L.
- $D^c_i(L)$: Dominating vertex in L^c_i.
- $D^r_i(L)$: Dominating vertex in L^r_i.
- $D^c_{i,j}(L)$: Dominating vertexes in $L^c_{i,j}(G)$.
- $D^r_{i,j}(L)$: Dominating vertexes in $L^r_{i,j}(G)$.
- $C^c_{i,j}(L)$: Connecting vertexes that connect vertexes in $D^c_{i,j}(L)$.
- $C^r_{i,j}(L)$: Connecting vertexes that connect vertexes in $D^r_{i,j}(G)$.
- $C^{ccon}_i(L)$: Connecting vertexes that connect vertexes between $D^c_{1,i}(L)$ and $D^c_{i+1,n}(L)$.
- $\gamma(G)$: 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. Consider flight time only

In this subsection, we will give the lower bound and construction of minimum dominating path of $L_{m,n}$ which is the shortest path in grid when consider flight time only.

Lemma 1. *Let $n > 3$, $m > 0$ be integers, and L is a dominating path in $G = L_{m,n}$. Then $|L^c_{n-2,n}(G)| \geq m$. Further, if $3 \nmid m$, then $|L^c_{n-2,n}(G)| \geq m + 1$.*

Proof: We follow proof in [**?**]. Since dominating path is a special case of connected dominating set, the conclusion is also applicable to our result.

■

In lemma 1, we know that for L in $G = L_{m,n}$, $|L^c_{n-2,n}(G)| \geq m$. Therefore, we consider the three periodicity of dominating path. Specifically, we could extend L in $L_{m,n}$ to $L_{m,n+3}$.

Since vertexes in L consists of dominating vertexes and connecting vertexes, we split dominating path in $L_{m,n+3}$ into 5 parts: $L = D^c_{1,n}(L) \cup D^c_{n+1,n+3}(L) \cup C^c_{1,n}(L) \cup C^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)$.

Lemma 2. *Let $n \geq 2$ is integer and L is minimum dominating path of $G = L_{m,n+3}$. We denote $G^* = L_{m,n}$ and L^* is a minimum dominating path of G^*. Then there is at least one condition that $|D^c_{1,n}(L)| \geq |D(L^*)|$. In other words, none of vertexes in $G^c_{1,n}$ is covered by $D^c_{n+1,n+3}(L)$.*

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

Consider there are k continuous vertexes v_k in G^c_n dominated by $D^c_{n+1}(L)$.

If $k \geq 2$, since these k vertexes are not dominated by $D^c_{1,n}(L)$, the corresponding k vertexes $v_{k'}$ which are in the same row with v_k in G^c_{n-1} can not belong to $D^c_{1,n}(L)$. Therefore, k vertexes $v_{k''}$ in G^c_{n-2} should belong to $D^c_{1,n}(L)$ 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 vertexes in G^c_{n-2} to replace them so as to shorten L. Then, we could use $v_{k''}$ to construct. As shown in .

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

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

In the second case, when D^c_{n+1} come from D^c_{n+2}, we have the following three cases. When $|v_k| > 3$, L would need more vertexes in G^c_{n+3} to dominate vertexes in G^c_{n+2}. And we could use similar construct like the first case. When $|v_k| < 3$, L will need more connecting vertexes which could also use the same construct. When $|v_k| = 3$ and vertex in v_k do not reach G^r_m, then vertex below $v_{k'}$ must belong to L. So vertexes 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^c_{n+1} do not come from D^c_{n+2}, L would step into $G^c_{n+1,n+3}$ in the first row, move down to $v_{m-1,n+1}$ and use vertexes in G^c_{n+3} to dominate remain vertexes. Since $v_{1,n+1} \in L$, v_k starts from G^r_3. 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 vertexes between G^r_{n+1} and G^r_{n+2}. 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^r_{n-1} to dominate vertexes in G^r_n and it will bring more vertexes 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, $|D^c_{1,n}(L)| \geq |D(L_{m,n})|$.

■

Lemma 3. *Given L as the minimum dominating path of $G = L_{m,n+3}$, then $|D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq m$. Further, if $3 \nmid m$, then $|D^c_{n+1,n+3}(L) \cup C^{ccon}_n(L)| \geq m + 1$.*

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 vertexes like Fig which can be transformed by extending one vertex to dominate all vertexes 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.

■

Theorem 4. *Let $m \geq 2$ and $n \geq 2$ as integers. We assume $G = L_{m,n+3}$, $G^* = L_{m,n}$. Then, $\gamma_l(G) \geq \gamma_l(G^*) + m$. 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 vertexes 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 vertexes 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 vertexes 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 vertexes 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 vertexes 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 vertexes in dominating path. Specifically, if there are two 2-period extends, the dominated rows will decrease more than gain in the dominating vertexes. 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 $m'+1$ vertexes 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 vertexes 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 vertexes from $C^c_{1,3}$ to $C^c_{4,n}$ if necessary without add more vertexes 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.

■

From theorem 4, we could also know that when $n = 1$, $\gamma_l(G) \geq \gamma_l(G^*) + m$.

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 vertexes 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 vertexes.

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 vertexes in G^r_n to dominate uncovered vertexes 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 vertexes 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 vertexes 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 vertexes in G^r_n, $v_{1,n-1}$ should be dominated by G^r_{n-2}. Therefore, there are $2m$ vertexes 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.

■

Proposition 1.

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) $\gamma_l(L_{3,n})=n$

Proof: Since connected dominating set is also dominating path when $m\leq 3$ or $n\leq 3$, we have the same result as [?]. ■

We denote $a=\lfloor\frac{m}{3}\rfloor$, $b=\lfloor\frac{n}{3}\rfloor$. And we will show six different cases for minimum dominating path for $L_{m,n}$ and give its proof.

Proposition 2. *Assume* $m\geq 4$, $n\geq 4$. *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*

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

.

Proof: When $m\equiv 0 \pmod 3$ or $m\equiv 2 \pmod 3$, standard extend row could reach m_3 $extend$ and $(m+1)_3$ $extend$ respectively which is better than other structure.

When $n=1$, $\gamma_l L_{m,1}=m-2$. L contains $m-2-\lceil\frac{m}{3}\rceil$ connecting vertexes which can be used to in standard extend rows. As shown in . Further, when $m\equiv 2 \pmod 3$ and $m\geq 11$, the structure in Fig. could bring a m_3 $extend$.

Then, when $n>1$, due to theorem 4, we could use standard extend rows to achieve optimal results.

Therefore, when $m\equiv 0 \pmod 3$, $n\equiv 1 \pmod 3$,

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

When $m\equiv 2 \pmod 3$, $n\equiv 1 \pmod 3$, if $a\leq 2$,

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

if $a>2$,

$$\gamma_l(L_{m,n}) = 3ab+3a+3b-1$$

■

Proposition 3. *Assume* $m\geq 4$, $n\geq 4$. *When* $m\equiv 0 \pmod 3$, $n\equiv 0 \pmod 3$, *assume* $m\leq n$, *we have* $\gamma_l(L_{m,n})=3ab+2a-2$. *When* $m\equiv 0 \pmod 3$, $n\equiv 2 \pmod 3$, *we have*

$$\gamma_l(L_{m,n}) = \begin{cases} min(3ab+4a-2, 3ab+3a+2b-1) & b\leq 2 \\ min(3ab+4a-2, 3ab+3a+2b-2) & otherwise \end{cases}$$

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

$$\gamma_l(L_{m,n}) = \begin{cases} 3ab+4a+3b+1 & a\leq 2 \\ 3ab+4a+3b & otherwise \end{cases}$$

. Moreover, the results have a same structure.

Proof: Since we use similar proof methods for 3 cases, we only prove $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$. We will use reduction in a.

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.

■

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

$$\gamma_l(L_{m,n}) = \begin{cases} 3ab + 3a + 3b - 3 & a + b \leq 4 \\ 3ab + 2a + 2b + 1 & otherwise \end{cases}$$

.

Proof: 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 $L_{4,4}$.

■

C. 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 vertexes in last subsection, we only need to add all dominating vertexes 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 5.

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 6. *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 7. *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 8. *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*

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

.

V. 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 a point at boundary as start point. Then, we use row-first
2) *Greedy COL First*: We select the minimum dominating set in grid first and select a point at boundary as start point. Then, we use select next point which could maximum $\frac{coverageset}{distance}$.

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 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 disparity of time consuming decreases as measurement time increases and when measurement time is 5 seconds it becomes 0.

Figure 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 than twice the area of two other algorithms. And as measurement time increases, coverage areas become closer.

C. Results of simulations

We simulate grids of different lengths and widths in this subsection.

Figure show that

Figure show that

D.

VII. Conclusions

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^r_1 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^r_1 into V_P;
  else
    put G^r_2 into V_P;
  end if
else if $m \equiv 0 \pmod 3$ **then**
  put G^r_2, G^r_5 ... , G^r_{3a-1} into V_P;
  put connecting vertexes in G^c_1 and G^c_n 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^r_2, G^r_5, G^r_8 except vertexes in G^c_1 into V_P;
    put $G^r_1 0$, $G^r_1 3$..., G^r_{3a+1} into V_P;
    put connecting vertexes in G^c_1 and G^c_n into V_P;
  else
    put G^r_2 into V_P;
    put G^r_4, ..., G^r_{3a+1} into V_P;
    put connecting vertexes in G^c_1 and G^c_n 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^c_2, G^c_4, G^c_7 except vertexes in G^r_1 into V_P;
    put G^c_9, $G^c_1 2$, ..., G^c_{3b} into V_P;
    put connecting vertexes in G^r_1 and G^r_m 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 vertexes 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;