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:

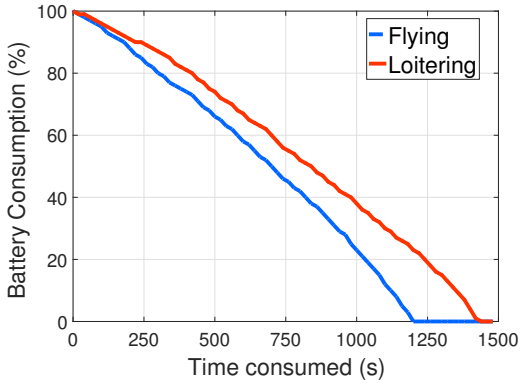


Fig. 1: Comparison of the Adaptive Monitoring Algorithm, Greedy Algorithm and Sequential Selection.

- 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

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.

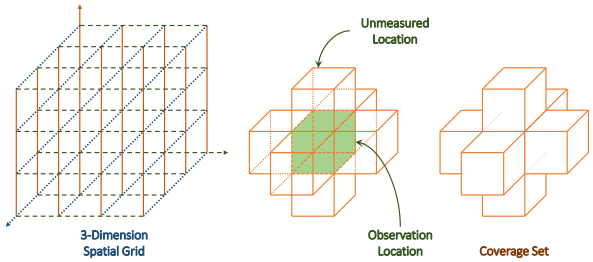


Fig. 2: The divided cuboids of a 3D space; a center OL in the cuboid in the green color; and the OLs and their cuboids in the coverage set of the center OL.

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 2), 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. 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 2.

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

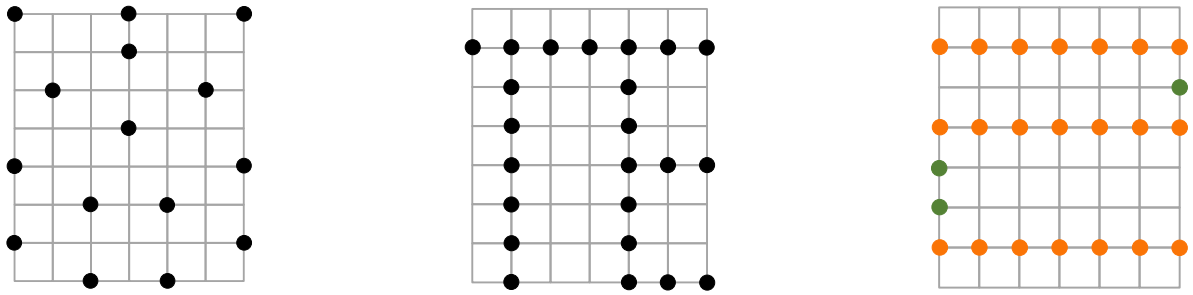
In this section, we first extend the well-known concept of the dominating set to define the dominating path in a graph. Then, we address Problem 1 by proposing an optimal algorithm to find the minimum dominating path in the 2D grid. We can concatenate the discovered paths (trajectories) in 2D grids to 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 P for graph $G = (V, E)$ is path $P \subset V$ such that every vertex not in P has a neighbor on P which is a special connected dominating set. In a dominating path P,

- there exists a dominating set D_P of G, and vertices in D_P are called *dominating vertices* of path P;



(a) A dominating set of $L_{8,7}$; (b) A connected dominating set of $L_{8,7}$; (c) A dominating path of $L_{8,7}$;

Fig. 3: A dominating set, connected dominating set and dominating path of $L_{8,7}$. In dominating path, orange vertices are dominating vertices and green vertices are connecting vertices.

- and the other vertices in $P \setminus D_P$ are called *connecting vertices* of path P.

We show difference among dominating set, connected dominating set and dominating path in Figure 3.

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 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 minimum dominating path in the bigger-size grid $L_{m,n}$ can be also found by repeating above Steps 2 and 3. A complete process from minimum dominating path in $L_{3,3}$ to minimum dominating path in $L_{6,6}$ is shown in Figure 4.

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.

- c_i: Leftmost $i-th$ column in grid G.
- r_i: Topmost $i-th$ row in grid G.
- $C_{i,j}$: Columns between c_i and c_j.
- $R_{i,j}$: Rows between r_i and r_j.
- $v_{i,j} = r_i \cap c_j$: Vertex of intersection of i-th row and j-th column in grid.
- $N[y]$: Coverage set of vertex y.
- $N[S]$: Coverage set of vertices in set $S \subset V$.
- $P_i^c = P \cap c_i$: Intersection between P and column c_i.
- $P_i^r = P \cap r_i$: Intersection between P and row-r_i.
- $P_{i,j}^c = P \cap C_{i,j}$: Intersection between P and columns $C_{i,j}$.
- $P_{i,j}^r = P \cap R_{i,j}$: Intersection between P and rows $R_{i,j}$.
- $D(P)$: Dominating vertices in path P.
- $C(P)$: Connecting vertices in path L.
- $D_i^c(P)$: Dominating vertex in P_i^c.
- $D_i^r(P)$: Dominating vertex in P_i^r.
- $D_{i,j}^c(P)$: Dominating vertices in $P_{i,j}^c$.
- $D_{i,j}^r(P)$: Dominating vertices in $P_{i,j}^r$.
- $C_{i,j}^c(P)$: Connecting vertices that connect vertices in $D_{i,j}^c(P)$.
- $C_{i,j}^r(P)$: Connecting vertices that connect vertices in $D_{i,j}^r(P)$.
- $C_i^{con}(P)$: Connecting vertices that connect vertices between $D_{1,i}^c(P)$ and $D_{i+1,n}^c(P)$.
- $\gamma(G)$: The size of minimum dominating set of G.
- $\gamma_c(G)$: The size of minimum connected dominating set of G.
- $\gamma_p(G)$: The size of minimum dominating path of G.

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

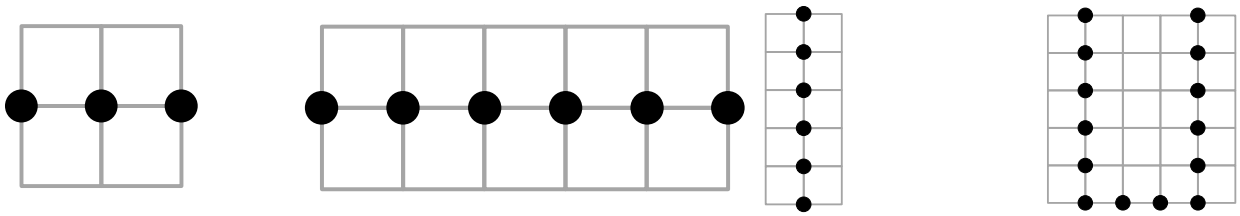
1) How many vertices are needed for constructing the minimum dominating path in $L_{m,n+3}$: The grid $L_{m,n+3}$ can be divided into two parts: (1) the left part grid from the 1st to the n-th columns, and (2) the right part from the $(n+1)$-th to the $(n+3)$-th columns. The left part is basically a smaller grid of $L_{m,n}$.

Given the grid $L_{m,n+3}$, we will derive the minimum number of vertices needed to add into the minimum dominating path of left part grid (i.e., $L_{m,n}$), so as to construct the minimum dominating path in $L_{m,n+3}$.

Lemma 2.

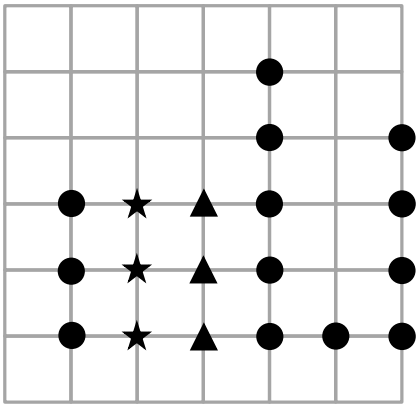
Let P^* denote minimum dominating path of $L_{m,n}$, and P denote a minimum dominating path of $L_{m,(n+3)}$. Then there exists at least one minimum dominating path P in $L_{m,(n+3)}$, such that the number of dominating vertices of path P from the 1st to the n-th column is no less than the number of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

Proof: We have $|D_{1,n}^c(P)| \geq |D(L_{m,n})|$ if all vertexes

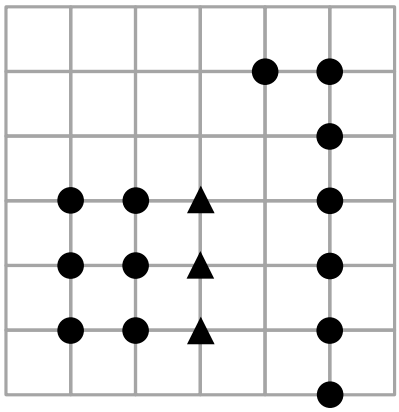


(a) A minimum dominating path of $L_{3,3}$; (b) A minimum dominating path of $L_{3,6}$; (c) A minimum dominating path of $L_{6,3}$; (d) A minimum dominating path of $L_{6,6}$;

Fig. 4: A complete process from minimum dominating path in $L_{3,3}$ to minimum dominating path in $L_{6,6}$



(a) 3 continuous vertexes in c_n dominated by $D^c_{n+1}(P)$. Circle vertices are vertices in P, star vertices are vertices in $v_{k'}$, triangle vertices are vertices in v_k.



(b) Partial of P' that could replace P.

Fig. 5

in $C_{1,n}$ is dominated by $D^c_{1,n}(P)$ since $C_{1,n} = L_{m,n}$. Therefore, if $|D^c_{1,n}(P)| < |D(P^*)|$, some vertexes in c_n must be dominated by $D^c_{n+1}(P)$ and do not have neighbors in $D^c_{1,n}(P)$.

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

If $k \geq 2$, since these k vertexes are not dominated by $D^c_{1,n}(P)$, the corresponding k vertexes $v_{k'}$ which are in the same row with v_k in c_{n-1} can not belong to $D^c_{1,n}(P)$. Therefore, k vertexes $v_{k''}$ in c_{n-2} should belong to $D^c_{1,n}(P)$ 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 P which makes P irregular. So we could use the corresponding vertexes in c_{n-2} to replace them so as to shorten P. Then, we could use $v_{k''}$ to construct, as shown in Figure 5a.

Since $C_{n+1,n+3}$ may have multiple connected components, P may step into $C_{n+1,n+3}$ and finish trajectory or move out $C_{n+1,n+3}$.

In the first case, we could construct as Figure **??** and construct P' such that $|P'| = |P|$ and none of vertexes in c_n is dominated by $D^c_{n+1}(P')$.

In the second case, when $D^c_{n+1}(P)$ come from $D^c_{n+2}(P)$, we have the following three cases. When $|v_k| > 3$, P would need more vertexes in c_{n+3} to dominate vertexes in c_{n+2}. When $|v_k| < 3$, P will need more connecting vertexes. For these two cases, we use same structure in Figure **??** to construct P'. When $|v_k| = 3$, assume P move back to $C_{1,n}$ from row r. If $v_{r,n+3} \notin P$, we could use same structure in Figure **??**. Otherwise, we consider the path extend from r and vertexes forms a dominating path for $L_{6,k}$ partial but can not reach the minimum. Therefore, P could not be the minimum dominating path when $|v_k| = 3$.

If $k = 1$, then the vertex must lay in boundary otherwise it will need extra connecting vertexes between R_{n+1} and R_{n+2}. Therefore, we assume $v_{1,n+1} \in P$. Then $v_{1,n}, v_{1,n-1} \notin P$ and one of $v_{1,n-2}$ and $v_{2,n-1}$ must belong to P to dominate $v_{1,n-1}$. If $v_{1,n-2} \in P$, P will turn to r_{n-1} to dominate vertexes in r_n and it will bring more vertexes than the following condition. If $v_{2,n-1} \in P$, we have structure shown in Figure 7b. This case could only exist once. We could transform $L_{m,n+3}$ symmetrical. Then, $|D^c_{1,n}(P)| \geq |D(L_{m,n})|$.

■

Given P as the minimum dominating path of the grid $L_{m,n+3}$, let $D^c_{n+1,n+3}(P)$ denote the set of dominating vertices of path P from the $n+1$-th column to the $n+3$-th column of grid $L_{m,n+3}$; and let $C^{con}_n(L)$ denote the set of connecting vertices of path P between $D^c_{1,i}(P)$ and $D^c_{i+1,n}(P)$. Then, we have the following lemma.

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

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

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

If r_i in $C_{n+2,n+3}$ is dominated by two connected components in $P^c_{n+2,n+3}$, 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, r_i is dominated by two end vertices which can be transformed by extending one vertex to dominate all vertices dominated by two components. Under the second scenery, r_i is dominated by one start point and one intermediate vertex. This is the unique case that could not be replaced. But we could merge them into one part since the union of two components follows the result.

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

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

■

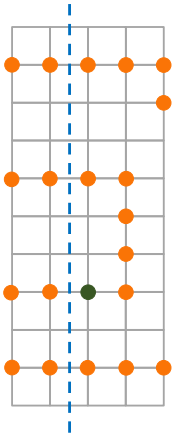
Since the left part of $L_{m,n+3}$ is a grid $L_{m,n}$, the following theorem tells the relationship between the minimum dominating paths of $L_{m,n+3}$ and $L_{m,n}$.

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

Proof:

In Lemma 2 and Lemma 3, we know that there exist at least one minimum dominating path P in $G = L_{m,n+3}$ and $|D^c_{n+1,n+3}(P)| + |C^{con}_n(P)| + |D^c_{1,n}(P)|$ could fulfill additive part in result. Therefore, if the result is false,

(a) Typical structure of minimum dominating path on the boundary.



(b) 2-period coverage and 3-period coverage of specific structure.

Fig. 6

$|C^c_{1,n}(P)| + |C^c_{n+1,n+3}(P)|$ in G must be less than $C(P^*)$ which is the connecting vertices of a minimum dominating path P^* in $L_{m,n}$.

Connectivity on the boundary: Since connectivity depends on structure of r_n and there are only two start vertices in P, we have structures in r_n like figure 6a. We will consider different structures of P in r_n.

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

If there are two connecting vertices in one connected component, when we consider them separately, result is the same as one connecting vertex. If we consider them together, the size of P of the component in r_n must be 4. We assume two vertices are $v_{i,n}$ and $v_{i+1,n}$. Also, we have two

cases. In the first case, we have $v_{i-1,n+1}$, $v_{i-1,n+2}$, $v_{i-1,n+3}$, $v_{i,n+3}$, $v_{i+1,n+3}$, $v_{i+2,n+1}$, $v_{i+2,n+2}$, $v_{i+2,n+3} \in P$, which hold $|C^c_{1,n}(P)| + |C^c_{n+1,n+3}(P)| = |C(P^*)|$. In the second case, we have $v_{i-1,n+1}$, $v_{i-1,n+2}$, $v_{i,n+2}$, $v_{i+1,n+2}$, $v_{i+2,n+1}$, $v_{i+2,n+2} \in P$, and $|C^c_{1,n}(P)| + |C^c_{n+1,n+3}(P)|$ decreases 2, $|C^{con}_n(P)|$ add 2 and $|D^c_{n+1,n+3}(P)|$ 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 construct that structure, r_{i-2} and r_{i+3} must be dominated by other components. This structure destroys the 3 periodic structure which will need more vertices in further extend. Specifically, it has the 2-period coverage and 3-period coverage as shown in figure 6b. If there are two 2-period coverage, the dominated rows will decrease more than gain in the dominating vertices. And if there are two 3-period coverage, it could be simple replaced by regular structure. Therefore, there are only two structures exist. In the first possibility, when $m \equiv 2 \pmod 3$ and $m \geq 11$, we have the same structure in figure 7c. Consider this structure in $L_{m',n'}$. We assume the structure first appear when $n' = b$. Since $|P^c_{n'-2,n'}| = m'$, 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 could not obey the result. In the second possibility, $n' \equiv 1 \pmod 3$ which could be simplified till $n' = 1$ which is also out of range.

Connectivity in the middle: In this case, we split $P^{*c}_{1,n}$ on middle and connect two connected components in $P^c_{n+1,n+3}$ and generate P. We assume the origin endpoints in P^* lay in the $n - th$ column of $L_{m,n}$, otherwise we need more connecting vertices between $P^c_{1,n}$ and $P^c_{n+1,n+3}$. If $C^c_{1,n}(P)| + |C^c_{n+1,n+3}(P)$ could be less than $|C(P^*)|$, we consider $P^c_{1,3}$. From Lemma 2, we could infer that $|P^c_{1,3}| \geq m$ and if $3 \nmid m$, $|P^c_{1,3}| \geq m + 1$. 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 the $4-th$ of $L_{m,n}$ and we could reduce the extend from minimum dominating path in $P_{m,n-3}$ to P^*. Then, we could construct a minimum dominating path P^{**} from $P^c_{4,n+3}$ and $|P^{**}| < \gamma_p(L_{m,n})$ and this makes contradiction.

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

2) Meta structs for constructing the minimum dominating path in $L_{m,n+3}$: In Theorem 4, we have proof the minimum number of vertices needed to add into the minimum dominating path in $L_{m,n}$ is m, which can be achieved in certain structures of the grid. In other structures, we need either $(m+1)$ or $(m+2)$ vertices to construct the minimum dominating path in $L_{m,n+3}$ from that in $L_{m,n}$. We call these certain structures in grids as three types of *meta struct*.

Struct 1: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m$. There are two cases for meta struct 1.

1) When $m \equiv 0 \pmod 3$, we could select vertices (red ones in Figure **??**) in rows r_2, r_5..., $r_m - 1$ in columns in $C_{n+1,n+3}$ into minimum dominating path, plus the vertices (blue ones in Figure **??**) shifted from column c_n to column c_{n+3}, as shown in Figure **??**. This struct can be repeated in bigger-size grids.
2) When $n = 1$, when $m \equiv 3 \pmod 1$ we could construct as Figure **??** and when $m \equiv 3 \pmod 2$ we could construct as Figure **??**. Both cases could only exist once.

Struct 2: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m + 1$. There are five cases for meta struct 2.

1) When $m \equiv 2 \pmod 3$, we could construct the minimum dominating path using similar struct as shown in Figure **??**, which can be also repeated in bigger-size grids.
2) When $v_{1,n}$ belong to the minimum dominating path in $L_{m,n}$, we have the meta struct as shown in Figure **??**.
3) When $c_n \setminus v_{1,n}$ belong to minimum dominating path in $L_{m,n}$, we have another meta struct as shown in Figure **??**.
4) When $n \equiv 0 \pmod 3$, we could rotate direction of minimum dominating path of $L_{m,n}$ and construct with meta struct 1. If the rotation only brings one extra vertex, we have the meta struct as shown in Figure **??**.
5) When $n \equiv 2 \pmod 3$ and $n \geq 8$, we could use the meta struct as shown in Figure **??** to construct the minimum dominating path in $L_{m,n+3}$.

Struct 3: $\gamma_l(L_{m,n+3}) - \gamma_l(L_{m,n}) = m + 2$. There are many cases for meta struct 3, and we only show the most common case here. When there are no vertices in column c_n that belong to the minimum dominating path P^* in $L_{m,n}$, vertices in column c_{n-1} must belong to P^*. Then, we could construct the minimum dominating path P in $L_{m,n+3}$ with the meta struct as shown in Figure 9, by adding $(m+2)$ vertices (red ones in Figure 9).

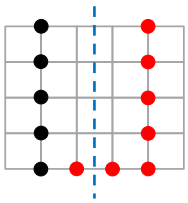
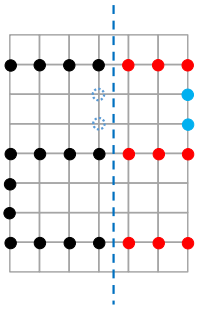


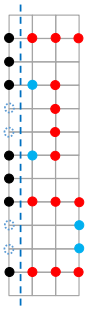
Fig. 9: An example of meta struct 3 when expanding a 5×3 grid to a 5×6 one.

C. Optimal Trajectory Planning Algorithm

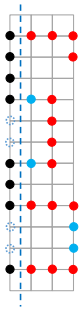
Based on the previous meta structs, we will elaborate the trajectory planning procedure in the following—i.e., construct the minimum dominating path P in grid $L_{m,n}$: (1) $m < 4$ and $m < n$; (2) $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$ or $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$; (3) $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$ or $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$ or $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$; and (4) $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$.



(a) The 9×4 grid expanded to the 9×7 grid;

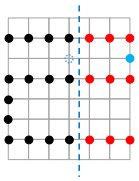


(b) The 13×1 grid expanded to the 13×7 grid;

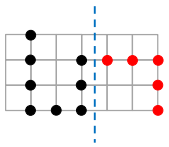


(c) The 14×1 grid expanded to the 14×7 grid;

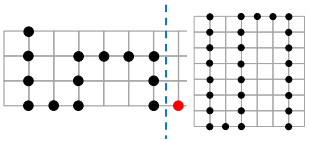
Fig. 7: Different cases of meta struct 1 by adding m vertices to construct the minimum dominating path in the expanded grid. Red vertices are extra vertices in the minimum dominating path of $L_{m,n+3}$, blue vertices are vertices shifted from blue dashed-line vertices in the n-th column, black vertices are part of vertices in the minimum dominating path of $L_{m,n}$. The dashed line separates columns in $C_{1,n}$ and columns in $C_{n+1,n+3}$.



(a) The 8×4 grid expanded to the 8×7 grid;

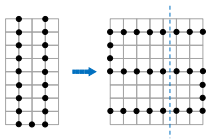


(b) The XXX8×4 grid expanded to the 8×7 grid;



(c) The XXX8×4 grid expanded to the 8×7 grid;

(d) (d)



(e) (e)

Fig. 8: Different cases of meta struct 2 by adding $(m+1)$ vertices to construct the minimum dominating path in the expanded grid.

Fig. 10: The minimum dominating path for $L_{1,n}$, $L_{2,n}$ and $L_{3,n}$

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

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

2) Case 2: Since the minimum dominating path in $Lm, 1$ contains $m - 2 - \lceil \frac{m}{3} \rceil$ connecting vertices, when $m \equiv 0 \pmod 3$ and $n \equiv 1 \pmod 3$, we could use the meta struct 1 to construct $L_{m,n}$, and we can construct the minimum dominating path as shown in Figure **??**. Therefore, when $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$, we have $\gamma_p(L_{m,n}) = 3ab+3a-2$.

Similarly, when $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$, we use meta struc 1 when $n = 1$, then we use meta struct 2 as n increases. We can construct the minimum dominating path as shown in Figure **??**. Therefore, when $m \equiv 2 \pmod 3$, $n \equiv 1$

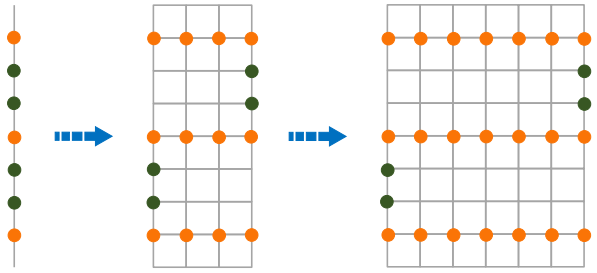


Fig. 11: Minimum dominating path when $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$.

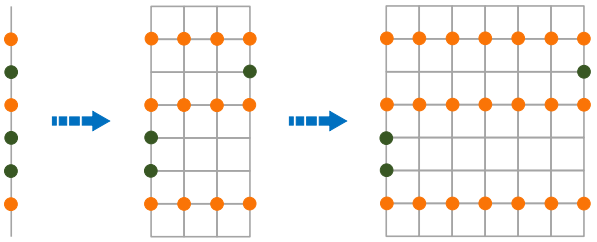


Fig. 12: Minimum dominating path when $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$.

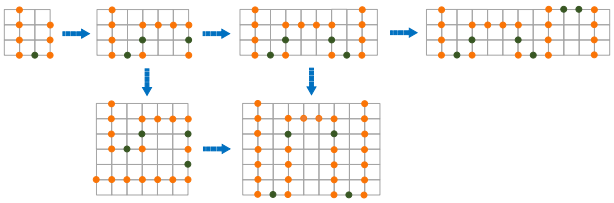


Fig. 16: Minimum dominating path when $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$.

$(\text{mod } 3)$, we have

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

.

3) Case 3: XXX We use the reduction on the structure to construct the minimum dominating path of $L_{m,n}$ in this case.

When $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$, if $m = 6$, we could construct $L_{6,n}$ using meta struct 1. Assume the result holds for $m = k$, then we could get the minimum dominating path for $L_{k+3,k}$ from $L_{k,k+3}$, since none of meta struct 1 or meta struct 2 will fit in the structure, we could use meta struct 3 to get the minimum dominating path for $L_{k+3,k+3}$. Then, we could use meta struct 1 to get $L_{k+3,n}$ when $n \geq k+3$ as shown in Figure **??** which holds the structure for $k+3$. We could use the same approach to construct P, when $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$ as shown in Figure **??**.

When $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$, we have two structures (a) and (b) as shown in Figure **??** and for different n we also have two structures shown in **??**. Assume the result holds for $m = k$, and the critical point between structure (a) and structure (b) is $n = l$ when $m = k$. Specifically, when $n \leq l$, $L_{k,n}$ has structure (a) and otherwise, $L_{k,n}$ has structure (b). For $L_{k,n}$ with structure (a), we could construct P XXX from $L_{k+3,l}$. Since structure (a) could only use meta struct 3 or meta struct 2, we use meta struct 3 on XXX $L_{k+3,l}$ until it satisfies the condition of meta struct 2. XXX And we hold result for $m = k+1$.

Therefore, we have following results.

When $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$, assume $m \leq n$, we have $\gamma_p(L_{m,n}) = 3ab+2a-2$. When $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$, assume $m \leq n$, we have

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

. When $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$, we have

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

.

4) Case 4: We could easily get the minimum dominating path of $L_{4,4}$ in Figure **??**. Therefore, we use two meta structs 2 to construct the minimum dominating path of $L_{4,7}$ and we use meta struct 2 to construct the minimum dominating path of $L_{7,7}$. Besides, we construct minimum dominating path of $L_{4,n}$ when $n \geq 10$ from $L_{1,n}$ with a meta struct 1. Finally, for $m \geq 7$, $n \geq 10$, we could only use meta struct 3 to construct the minimum dominating path of $L_{m,n}$ from $L_{m-3,n}$, which leads to the result in Figure **??**.

Therefore, when $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$, we have

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

.

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

In a real-world 3D space that might be irregular, we give an approach to concatenate 2D trajectories to a 3D trajectory.

- First, we divide 3D space into cuboids and obtain a 3D irregular grid; and then we further divide the 3D grid into multiple layers of 2D irregular grids.
- Then, for each irregular 2D grid, we could divide it into a number of regular 2D grids. In each regular 2D grid, we can use the proposed algorithm to construct the minimum dominating path for each of them.
- Finally, we could simply concatenate the minimum dominating paths for those regular grids because their starting points always lie on the grid boundary, which produce a trajectory for the drone in the 3D space.

V. Finding the COLs from the minimum dominating path

A. Consider total time consuming

Since we formulate this problem as a combination of a minimum dominating path problem and a constrained minimum dominating set problem, we could use the same method in subsection **??** to construct minimum dominating path of four cases and then select the least number of OLs along the path as COLs. However, dominating vertices in the minimum dominating path is the minimum dominating set of path from definition. Therefore, we could simply select dominating vertices in minimum dominating path as COLs which are shown in black vertices of figures in **??**. We use the same cases in **??** and get $|V_C|$.

1) Case 1: From figure 10, we have

1) when $m = 1$, $|V_C| = \lfloor \frac{n+2}{3} \rfloor$.
2) when $m = 2$ or $m = 3$, $|V_C| = \gamma_l(L_{m,n})$.

2) Case 2: From figure 11, when $m \equiv 0 \pmod 3$, $n \equiv 1 \pmod 3$, we have $|V_C| = 3ab+a$.

From figure 12, When $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$, we have $|V_C| = 3ab+a+3b+1$.

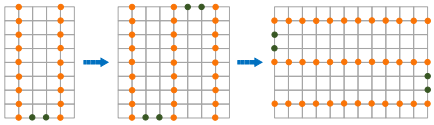


Fig. 13: Minimum dominating path when $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$.

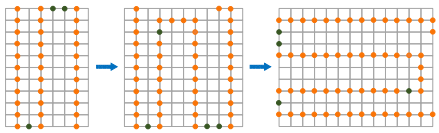


Fig. 14: Minimum dominating path when $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$.

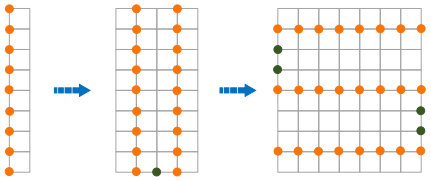


Fig. 15: Minimum dominating path when $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$, middle one with structure (a) and right one with structure (b).

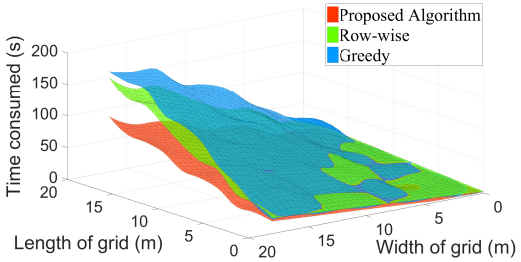


Fig. 21: Comparison of the Adaptive Monitoring Algorithm, Greedy Algorithm and Sequential Selection.

3) Case 3: From figure 13, when $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$, we have $|V_C| = 3ab$.

From figure 14, when $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$, assume $m \leq n$, we have $|V_C| = 3ab + 2a + 3b + 2$.

From figure 15, when $m \equiv 0 \pmod 3$, $n \equiv 2 \pmod 3$, we have

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

.

4) Case 4: From figure 16, when $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$, assume $m \leq n$, we have

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

.

B. Optimal Trajectory Planning Algorithms

VI. Evaluation

A. Experiment Setup

Prototype: The prototype consists of two parts: (1) a drone with an API that can configure its trajectory; and (2) a measurement sensor board or a smartphone is installed into a plastic box with vent-holes made by a laser engraving machine, which is attached at the bottom of the drone. The drone we use is a DJI Phantom 3 Quadcopter, as shown in Figure **??**. There is an intelligent flight mode that allows the customization of the trajectory by configuring a number of stop locations such that the drone will autonomously follow this trajectory to hover over each stop and complete the flight. The GPS sensor, ultrasonic sensor, and the motion sensors collectively determine the current observation location of the drone and guide the drone to the next stop in the trajectory. At each stop, the drone hover for a few seconds, T_M, to collect sufficient measurement data, before traversing to the next stop.

Compared algorithms: We compare the proposed algorithm with two other conventional algorithms. In the proposed algorithm, we choose the minimum dominating path in grids first, and then select COLs from path. In contrast, the conventional algorithms we considered will select COLs in grids first, and then choose a path to connect the selected COLs. The two conventional algorithms differ by the way of selecting COLs, as introduced below.

1) *Row-wise COL selection*: We select the minimum dominating set in grids first, and select an OL at boundary as start point. Then, we use a row-first COL selection strategy to find a row by row, along the longer side of the grid.
2) *Greedy COL selection*: We select the minimum dominating set in grids first, and then select a COL at boundary as start point. Then, we follow the greedy algorithm that chooses the next vertex v as the closest one that can lead to the maximum coverage, such as:

$$\min_{d(v,v')} \max |N[v] \backslash N[P]|\ v,$$

where $v' \in \mathcal{G}$ is the last vertex of path of $P \subset \mathcal{G}$.

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

- *Time consumed*: The time consumed is defined as total time of flight and measurement of the drone to cover the space of the trajectory under a specific algorithm.
- *Coverage space*: The coverage space is defined as the maximum coverage area in the 3D space during entire battery life, by following the trajectory under a specific algorithm.

B. Experimental Results

We choose an open space in a university campus which has a size of 45 m in length, 35 m in width, and 10 m in height. It is divided into 3D grids formed by 5 by 5 by 5 m cuboids. Since the time consumed is the sum of flight time and measurement time, we consider the impact of different measurement times in

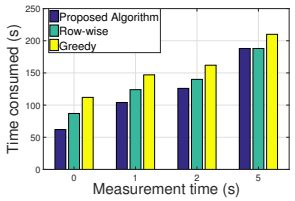


Fig. 17: Time consuming of three algorithms with different granularity of measurement time.

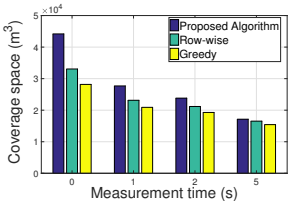


Fig. 18: Battery coverage of three algorithms with different granularity of measurement time.

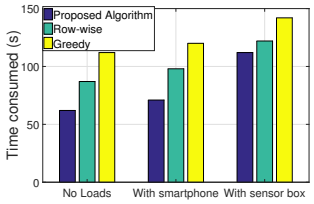


Fig. 19: Time consuming of three algorithms with different loads when considering flight time only.

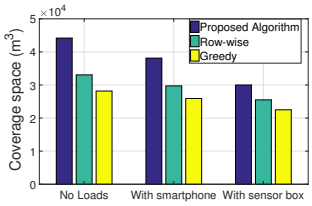


Fig. 20: Battery coverage of three algorithms with different loads when considering flight time only.

experiments, which are set as 0, 1, 2, 5 seconds at each COL. Note that the case when the measurement time is 0 second is equivalent to the case when we consider the flight time only.

Figure **??** show the time consumed under compared algorithms with different granularity of the measurement time. The sensing area is a 2D grid of size 9 by 7 m. We observe that our proposed algorithm consumes less time when the measurement time is small. The gap between our algorithm and the row-wise COL selection algorithm decreases as the measurement time increases. When the measurement time is 5 seconds at each COL, the total time consumed by the proposed and the row-wise COL selection algorithms are equal.

Figure **??** show the coverage space in the 3D grid under compared algorithms with different granularity of the measurement time during the battery life. We fix the width of the grid as 6 m, we observe that our proposed algorithm could cover more space than other algorithms in all cases, during the whole battery life. As the measurement time increases, the coverage area by any of three algorithms decreases, since the measurement time consumes part of the battery. Meanwhile, the difference in the coverage space among three algorithms decreases when the measurement time is set greater.

In a real-world application, drones might be used to delivery packages or other goods. Hence, we also evaluate the time consumed and the coverage space with different loads when we only consider the flight time. Results in Figure **??** and **??** show that as the load increases, the time consumed of three algorithms to complete the sensing task of a given grid increases; meanwhile, the coverage space of three algorithms during the battery life decreases. Note that the proposed algorithm performs the best in all scenarios under different loads on the drone.

C. Results in different-size spaces

In experiments, we only test the performance of compared algorithm in a give 3d space with the fixed size. Next, we can evaluate the algorithm performance in different-size spaces.

Figure **??** show the flight time in spaces with different width and length. We observe that our algorithm consumes less time to complete the sensing task in each grid. When fixing the width (or length) of the grid, the gap among algorithms increases as the length (or width) increases.

Similarly, we observe in Figure **??**, our proposed algorithm has the maximum coverage space during the battery life; When fixing the width (or length), our algorithm could cover more space than other compared algorithms.

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 2D grids when considering 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 except0 vertices in G_1^c into V_C;
        put G_8^r except vertices in G_1^c and G_3^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;