Literature review

**Problem Statement**

In this paper, we will test all the search algorithms in a dynamic grid world . Each grid cell g represents a 1gx1g area. g is abstract unit and 1g is the side length of the grid. It is common to cluster the obstacle[citation], so if the grid is occupied, the whole area is occupied. The flight agent occupies one grid cell and can only fly one grid by one grid. The agent can only fly up, left, right and down and the fly rate is 1g/s. Grid cells can have five different statuses: 1. Open space: grids all the agents can pass. We assign a code 0 for open spaces. 2. Permanent no-fly zone: grids are permanently occupied, and agents strictly cannot pass them. We assign a code 99 for the Permanent no-fly zone. 3. Temporary no-fly zone: grids are temporarily occupied, and agents must avoid them unless there is no other way. We assign a code 100 for Temporary no-fly zone. 4. Out-network flight: grids are temporarily occupied or reserved by the flight that are not control by the system. The rule of agents passes those grids are same with the Permanent no-fly zone. We assign a code 101 for Temporary no-fly zone. 5. Reserved zone: grids are reserved by the agents. Like the temporary no-fly zone, other agents will avoid the reserved zone. However, reserved zone will not block the agent, the agent can still pass the reserved zone with large cost. The reserved zone’s code will be the agent number.

A picture containing graphical user interface

Description automatically generatedTable

Description automatically generated with low confidence

Agents will be assigned fly tasks t. The fly task asks agents fly from a start point to a destination. Agents are going to be navigated by the search algorithms to complete the fly task. Before agents move to the next grid cell, agents must submit the fly plan P generated by the search algorithms to the air-space manager. Then, if the air-space manager find that the flying planning is not valid anymore, the airspace will reject the fly plan and ask agent to renavigate. For safety reasons, the fly plan becomes invalid once one of the grids in the fly plan are occupied by the Temporary no-fly zone or Out-network flight. Fly plan also becomes invalid if the first grid of the plan is occupied by another agent. The manager evaluate the flying planning right before agents move. The fly plan must be generated in a short time for agents to complete the tasks safely and efficiently.

If any , or the path is valid.

When the agents fly to the planned grid, the agents whose fly plan are rejected will start navigate.

**Method**

This Multi-Scale A\* (MSA\*) method is inspired by the work in~\cite{lim2020mams}, where a multi-resolution framework is constructed to assist path planning with multiple agents, where each agent's abstract path is improved through the information provided by the other agents. To further increase the efficiency of the algorithm a fine resolution graph representation is constructed around the agents to focus the processing power only on the optimal path it will take. While a coarse resolution representation is kept farther away from the agents to reduce the dimensionality of the search space, which in relation reduces the processing power and time to run the algorithm. For most popular grid search algorithms, the search depth and the search space are the main factors that affect the computational complexity. Multi- Scale A\* can dramatically decrease the search depth and the search space then lower the computational complexity.

The generated grid map will be transformed to a multi-scale grid, which is constructed based on the agent's current location, destination, and map information. Therefore, the multi-scale grid map is varying for different agents or the same agent at different locations. The purpose for using a multi-scale grid is that a lot more simulations can be run at a quicker succession due to the fact that the MSA\* can nearly run in real-time due to its smaller search-space. This allows the collection of new environmental data more efficiently. In this work, the type of environmental data includes how the flight path of the UAS is affected as it changes its reserved airspace volume when encountering any dynamic or static obstacles, and the likelihood of collisions occurring as air traffic density increases.

**Multi-resolution graph**

In this paper, we transfer the grid map to a node based multi- Scale graph Gf = (N,E), where N is the node set, E is the edge set. n ∈ N is a node can contain one grid or multiple grids. where r(n) is the risk value given by the equation (1), and v(n) = (x, y) is the left-bottom vertex position of the node. We use quadtree T to manage those node. T = {N}. The root node of the quadtree, which is the parent for all other nodes, represents all the grids in the grid world.

The quadtree node nl,o ∈ N represents a grad set gn ⊂ G, where l ∈ Z + is the node level in quadtree, o(n) = (x, y)o is the center position of the node. The total size of the represented block set is 2 l ∗ 2 l . Each leaf node (l = 0) represents all the information in a 1x1 block(unit block) such as center, position, and risk. Each non-leaf node(l > 0) represents its 4 direct children at level l−1, and its risk and center position are the average risks and center position of its 4 direct children. Each child represents 1/4 of the blocks its direct parent represents. For example, in Fig. 3(c), the red node represents four unit blocks and the green node only represents one unit block. In practice, some nodes in the full quadtree are not necessary. An agent ai can cut off all the undesired nodes (select desired nodes) and generate a partial tree. The partial tree builds a non-empty multi resolution graph G(Ni ,Ei) where Ni is the set of the selected nodes and Ei is the corresponding edge set. The current agent selects nodes following the hierarchical system and goes down the set nodes in each depth layer. When a node with children is selected by the agent, then its children node and parent node will be excluded from Ni . This ensures that the selected node in Ni completely covers the environment Ω without overlap. Ni is a subset of the full node set N. The node nl,o will be selected if the straight line distance between the center point o of the node, and the current location p of the agent, is greater than the 2 l multiplied by α, which is a parameter set by the user. Otherwise, its children will be selected instead. The larger the α is, the more nodes will be selected.

**Nodes selection:**

Based on object

**Intelligent adaptive parameter controller:**

The size of search space and search depth is high rely on the alpha and beta. If alpha is larger, the results of the search algorithm will be much more reliable because the graph more accurate. However, search will cost much more time. To balance the time cost and reliability of the results, adaptive parameters are introduced. The intelligent parameter **controller** will evaluate the current situation and modify the parameters.

Alpha decrease if distance between current position and destination is smaller than 16g.

Alpha increases temporarily if loop detected.

**A\* search cost function.**

Cost function:

**Experience**

We have a grid map . Each b represents the status of a grid cell. , 0 means open space, 99 means permanent no-fly zone, 100 means temporary no-fly zone. 101 means out-network flight. Other number refers the reserved zone for correspond agent.

Tasks are randomly generated. Straight distance is larger than 200g

Temporary no-fly zone is complete random generated.

Random grids will be no-fly zone for a random duration.

Out-network flight will be fixed.

All the map information is same for each experience.

1. A\*

Contribution:

1, dynamic reservations versus static reservations. Shorter distance, more space utility.

2. adaptability to new information.

3. hibrid

MRA\*

1. A\* corner case
2. MSA\* ,but local case, not optimal, fail to reach the destination
3. New strategy