12th International Conference on Advanced Computational Intelligence (ICACI)
Dali, China, August 14-16, 2020

Priority-based Multi-Flight Path Planning with Uncertain Sector Capacities

Sudharsan Vaidhun
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, United States
sudharsan.vaidhun@knights.ucf.edu

Zhishan Guo
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, United States
zsguo@ucf.edu

Jiang Bian
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, United States
bjbj11111@knights.ucf.edu

Haoyi Xiong
Baidu Inc.
China
xionghaoyi@baidu.com

Sajal K. Das
Department of Computer Science
Missouri University of Science and Technology
Rolla, United States
sdas@mst.edu

***Abstract*—The United States National Airspace System is currently operating at a level close to its maximum potential. The workload on the system, however, is only going to increase with the influx of unmanned aerial vehicles and soon, commercial space transportation systems. The traffic flow management is currently managed based on the flight path requests by the airline operators; while the minimum separation assurance between flights is handled *strategically* by air traffic control personnel. A more *tactical* approach would be to plan for a longer time horizon which is non-trivial given the uncertainties in the airspace due to weather. In this work, we consider a simplified model of the airspace as a grid of sectors and the uncertainties in the airspace are modeled as *blocked* sectors. In the modeled airspace with uncertainties, we schedule multiple flights using a dynamic shortest path algorithm. A novel cost function based on potential energy fields is proposed for use in the path planning algorithm to handle blocked sectors. A priority-based contention resolution scheme is proposed to extend the solution to multiple flights. We then demonstrate the proposed framework using a simulated test case.**

I. Introduction

The Next Generation Air Transportation System (NextGen) [1] is a collection of new technologies and tools provided by the Federal Aviation Administration (FAA) to improve the safety and efficiency of the National Airspace System (NAS). Managing the capacity and demand in the national airspace is one of the primary focuses of the NextGen. However, current techniques to manage the capacity are at best tactical rather than strategic. Developing strategic decisions is naturally a harder problem which involves resolving conflicts in flight paths, diverting traffic away from weather affected regions of the airspace, maintaining a minimum separation between flights.

In this work, majority of the focus in on the capacity of the airspace sectors. Sector capacity refers to the maximum number of flights that can be handled by the air traffic controllers in a given interval of time. This sector capacity is largely determined by the workload incurred by the air traffic controllers that manage the sector to preserve separation assurance among several other safety constraints [2]. Separation assurance is the safety invariant that assures a minimum distance required to be maintained between any two aircrafts. Several works [2], [3] have established a workload model that characterizes the different categories of work that is required to be handled by the controllers. Among all workloads, the traffic dependent workloads are *transit workload*, *conflict workload* and *recurring workload*. Further, weather induced flow constrained areas in the air space have degraded sector capacities due to the additional workload on the air traffic controllers to guide to flights. In addition to all the disturbances, even under normal weather conditions, human factors such as fatigue can reduce the actual sector capacity from the maximum sector capacity. To address this problem, recent research has focused on getting accurate position and heading of the aircrafts, better weather prediction models, data communication instead of voice communication for effective information exchange, several flow management programs like ground delay, en route delay programs. We aim to directly reduce the traffic dependent workload by developing a path planning framework to strategically design flight paths considering the uncertain dynamics of the weather.

There exist several preliminary works to handle the path-planning problem. A^* search [4] and geodesic computation [5] are some of the widely used approaches to tackle the problem. The presence of uncertainties have also been considered in the path planning procedures [6], [7], however, those approaches rely on the A^* search algorithm [8]. Aircraft routing can not be modeled as a shortest path routing problem because of the dynamic changes in the weather, which affects the path costs. A change in the path cost may make the planned route infeasible and in worst cases, the aircraft may not be able to reach its destination in time. There are two desirable properties

978-1-7281-4248-7/20/$31.00 ©2020 IEEE

expected from the routing algorithm. The first property is to perform rerouting with minimal overhead and not drastically change the flight path. The second desired property is avoiding the local minima problem. Avoiding local minima problem guarantees that the flight reaches its destination [9]. When multiple flights and dynamic obstacles such as the weather is considered, the local minima problem can occur with certain cost functions.

Graph-based path planning algorithms work for simple cases, and as discussed earlier, it does not capture the complexity of the air traffic problem. Zhong [10] has provided a detailed survey of the various optimal path planning techniques with various objectives such as fuel savings, delay minimization, reducing emissions and many more. Schilke and Hecker [11] proposed a route optimization framework under adverse weather data by first aggregating the airspace sectors to avoid and then planning a route based on $A*$ search. In these approaches, the path is to be updated in intervals when there is an update in the long-term weather forecast data.

However, these works focus on dealing with the path finding problem of a single flight or do not consider rapidly adapting the flight path to changes in the weather along the path. Because of the push towards autonomous drones for civilian and logistic purposes, there will be an increased demand on air traffic controllers. This work aims at automating the routing process, such that the air traffic controllers can act as an oversight instead of manually decongesting the air space. The multi-target case models multiple target at the same time while the interaction between the different flights is handled by priorities. The priority assignment is not in the scope of the paper and it could be assigned based on the flight dependent information such as time of flight, fuel capacity, flight speed etc.

Our Contributions. The following is a summary of contributions in this work.

1) A novel cost function based on potential energy fields is proposed for the graph search algorithm. The cost function captures the information about the blocked sectors and contending flights.
2) A priority-based contention resolution is proposed to support path planning for multiple flights in a shared airspace.

Organization. Section II introduces the models used to represent the airspace, the sector capacity and the flight model. In Section III we propose a path planning framework for multiple flights. A heuristic cost function is proposed in Subsection III-B. Using the cost function, we show the path planning for a single flight in Subsection III-C. Later, in Subsection III-D we extend the framework to include multiple flights. Simulation results to demonstrate the framework using a set of sample flights is shown in Section IV. Finally, we present our conclusions and thoughts on future works in Section V.

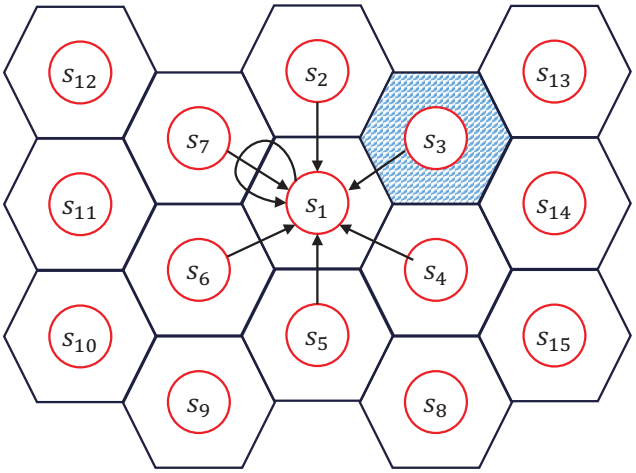


Fig. 1: A sample representation of the airspace with sectors $s_1, s_2, \ldots, s_{15}$ considering an hexagonal grid, which shows the weather model. A flight can only traverse between adjacent sectors.

II. Models and Problem Statement

In this section, the airspace model and the sector capacity model is stated. Based on the models, the air traffic scheduling problem is then formulated.

A. Airspace Sector Model

The airspace is represented by a contiguous set of sectors. The set of all sectors $S = \{s_1, s_2, \ldots, s_n\}$ represents the available sectors and each sector is managed by its own control center. The airspace as a whole is therefore represented using a graph G with the sectors as vertices with adjacent sectors connected by edges. For any sector $s_i \in S$, the subset $N(s_i) \subset S$ represents the neighbors of s_i, i.e., $\forall s_j \in N(s_i)$, an aircraft can transit *directly* from s_i to s_j represented by the edge e_{ij} in the graph.

An example of the graph model is as shown in Figure 1. The altitude of the flight in a sector is not considered in this model, since altitude changes are infrequent and assumed to be handled by the air traffic controllers. Since the routing of the flight is automated at a finer granularity, majority of the workload for the air traffic controllers is reduced. Due to the simpler structure of our airspace, the distance traveled by a flight transiting from one sector to the other is assumed to be equal to the *euclidean distance* between the sectors.

B. Sector Capacity Model

Sector capacity refers to the number of flights that can be handled by the air traffic controllers in any given sector. For simplicity, we assume that each sector can handle the same amount of flights and therefore normalize the sector capacity. The sector capacity can further be reduced by adverse weather conditions for a short period of time. The capacity of a sector s_i at time k is represented by $x_i[k]$, where $k \in \mathbb{N}$. The capacity $x_i[k]$ takes a real value in the range $[0, 1]$; A value of 0 means

that the sector is blocked and has no more capacity, while a value of 1 means that the sector capacity is at its maximum and there are no flights or adverse weather in the sector.

Weather impact on strategic air traffic management can be modeled by a networked Markov process model called the influence model. The influence model is shown to be able to generate realistically model the weather dynamics [12]. We use the influence model to model the dynamics of the sector capacity being affected by the weather conditions. According to the influence model, the probability of any sector $s_j \in V(G)$ influencing the sector s_i's next capacity (i.e., $x_i[k+1]$) is given by $p_{i,j}$, where $0 \leq p_{i,j} \leq 1$ and $\sum_j p_{i,j} = 1$. Once the influencing sector s_j is chosen, the sector s_i's next capacity $x_i[k+1]$ is determined by the 2×2 local transition matrix $\boldsymbol{A_{i,j}} = \begin{bmatrix} A_{i,j_{0,0}} & A_{i,j_{0,1}} \\ A_{i,j_{1,0}} & A_{i,j_{1,1}} \end{bmatrix}$, where $A_{i,j_{m,n}} = p(x_i[k+1] = n \mid x_j[k] = m)$.

An example is shown below for understanding. From Figure 1, let s_3 be the influencing sector for s_1 and $x_1[0] = 1$; $x_3[0] = 0$ be the sector capacities at the two sectors at time $k = 0$. And assuming $A_{1,3} = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix}$ implies

$$\begin{aligned} p[x_1[1] = 0 \mid x_3[0] = 0] &= 0.9 \\ p[x_1[1] = 1 \mid x_3[0] = 0] &= 0.1 \end{aligned}$$

Therefore, according to the influence model, there is a 10% chance that the capacity of the sector s_1 will be blocked due to the weather in the next time interval. The transition probabilities $p_{i,j}$ and the influence matrix $A_{i,j}$ is calculated from historic weather data. Later in the work, we will use the influence model to generate weather data for simulation purposes.

C. Flight Model

We consider a set of flights $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$. Each flight $f_l \in \mathcal{F}$ has an origin sector o_l and destination sector d_l, where $o_l, d_l \in S$. A basic requirement of air traffic scheduling is to allocate a path from the origin sector to the destination sector. In this work, we assume that the sectors are of the same size and the flight time between sectors is assumed to be equal to its euclidean distance. For each flight $f_l \in \mathcal{F}$, D_l represents its maximum flight distance or *deadline* by which it should reach the destination. The flight path p_l for flight f_l is a path in the airspace graph G from the origin sector o_l to the destination sector d_l. The edge weights in the path represent the cost to travel from one sector to the next. Given a flight path $\{o_l, \ldots, s_{l_1}, s_{l_2}, s_{l_3}, \ldots, d_l\}$, if the flight is at sector s_{l_1} at time k, then at time $k+1$ the flight will be at sector s_{l_2} and at time $k+2$ at sector s_{l_3} and so on.

D. Problem Statement

The state of the airspace system at time k is represented by the position (i.e., sector) of all the flights $f_l \in \mathcal{F}$ and the capacity $x_i[k]$ of each sector. The number of flights in a sector s_i at time k is given by $\eta_k(s_i)$. The edge weight $c_k(s_i, s_j) \geq 0$ in the airspace graph G represents the cost of transition from s_i to s_j at time k for flight f_l. The goal of the work is to find $\forall f_l \in \mathcal{F}$, a flight path p_l such that

$$\underset{\{p_l \mid \forall f_l \in \mathcal{F}\}}{\arg\min} \quad \eta_k(s_i). \tag{1}$$

III. Solution Framework

In this section, we give an overview of the D* Lite path planning algorithm [13]. Next, we provide a cost function for the search algorithm using a heuristic based on potential fields. Then, we model the sectors blocked by weather as obstacles and show the implementation of D* Lite algorithm for a single flight. After the application for a single flight is shown, we extend the framework to include multiple flights simultaneously.

A. D* Lite Algorithm

For any given position in the graph, D* Lite algorithm maintains two different cost estimates to the destination. The first estimate g is based on the information about the map that is already known. The second estimate rhs is a look-ahead estimate. When the two estimates g and rhs are different, the node in the graph is marked as *inconsistent* and is placed in a priority queue. The nodes in the queue with a lower cost estimate to the destination take a higher priority in the queue.

Initially, every node in the map assumes that its cost to the destination is infinite since the map has not been explored yet. The map exploration in a D* Lite algorithm starts from the destination until the exploration reaches our current position. The exploration is a repetitive update of inconsistent nodes in the priority queue. The node with the highest priority is updated to make its estimates equal and, then the neighboring nodes are informed of the new cost g. If the neighboring node finds a new path with a lower cost, its look-ahead cost rhs is updated to reflect that which makes it inconsistent with its g value and hence placed in the queue. This process is repeated until the estimated cost from the current node is not infinite and the current node is *consistent*. The exploration process is terminated, and no further exploration is necessary. For further details on the D* Lite algorithm, refer [13] by Koenig and Likhachev.

Two important features of D* Lite algorithm that make it suitable for our problem are as follows:

1) Exploration of the entire graph is not necessary. The map exploration procedure starts from the destination and terminates when a path is found from the current node. Since the airspace is a vast region, it is unlikely that a good path lies extensively far away from the source and the destination.
2) When the map changes due to updates in the weather forecast, the changes are propagated to the current node. This propagation eliminates the need for recomputing the path from scratch and allows a relatively very quick recomputation of an updated path.

B. Cost Function

Graph-based path planning algorithms require a defined cost for a transition from one node to another. The heuristics behind the cost function are application specific. In this subsection, first, potential fields are introduced, and then a cost function for the path planning problem is defined based on the designed potential fields.

Potential functions have been extensively used in global path planning algorithms [14]. Potential function approach involves constructing a potential energy surface over the area considered for path planning with the goal being the point(s) with the lowest potential energy. The points in the energy surface with the highest energy represents the obstacles. There are two major sub-components that combine to form the potential energy surface, namely the *attractive potential* and the *repulsive potential*. Let the potential function associated with an aircraft at sector s_i at time instant k be given as shown in Equation (2)

$$U_{tot}(s_i, k) = U_{att}(s_i) + U_{rep}(s_i, k), \tag{2}$$

where, U_{att} is the attractive potential and U_{rep} is the repulsive potential.

Attractive potential. The attractive potential $U_{att}(s_i)$ at s_i is proportional to the square root of the Euclidean distance $r(s_i, s_d)$ between s_i and the destination sector s_d. The $U_{att}(s_i)$ is calculated as shown in Equation (3).

$$U_{att}(s_i) = k_{att}\ \sqrt{r(s_i, s_d)}, \tag{3}$$

where, k_{att} is the attractive potential constant. The attractive potential is chosen to be quadratic so that the magnitude of the slope increases as the flight approaches the destination.

At destination d_l for flight f_l, $r(d_l, d_l) = 0$ which implies $U_{att}(d_l) = 0$. Also, since the attractive potential depends on the goal sector, the $U_{att}(d_l)$ at any given sector s_i does not change due to obstacles.

Repulsive potential. The repulsive potential U_{rep} is used to repel the aircraft away from obstacles, which in our case are the blocked sectors. The repulsive potential must be designed to prevent the aircraft from approaching the blocked sectors. The repulsive potential is chosen to be a 2-dimensional Gaussian function with the constrained sector as its center. The amplitude and the spread of the Gaussian function are design parameters to adjust the intensity of the repulsion and its range of influence in the energy surface. The repulsive potential at sector s_i at time k due to *all* the sectors in the airspace is given by $U_{rep}(s_i, k)$ as shown in Equation (4).

$$U_{rep}(s_i, k) = \sum_{s_j \in S} k_{rep,j}\ x_i[k]\ exp\left(\frac{r(s_i, s_j)}{\sigma_{rep,j}}\right), \tag{4}$$

where, $k_{rep,j} \in \mathbb{R}^+$ is the repulsive potential amplitude, $\sigma_{rep,j} \in \mathbb{R}^+$ is the spread of the gaussian function corresponding to the sector capacity at sector s_j, $x_i[k]$ is a binary value indicating the sector capacity of sector s_i at time k as defined in Subsection II-B. A sample potential field is shown in Figure 2, with blocked sectors at $(2,2), (7,3), (8,2)$ and the destination sector at $(7,9)$.

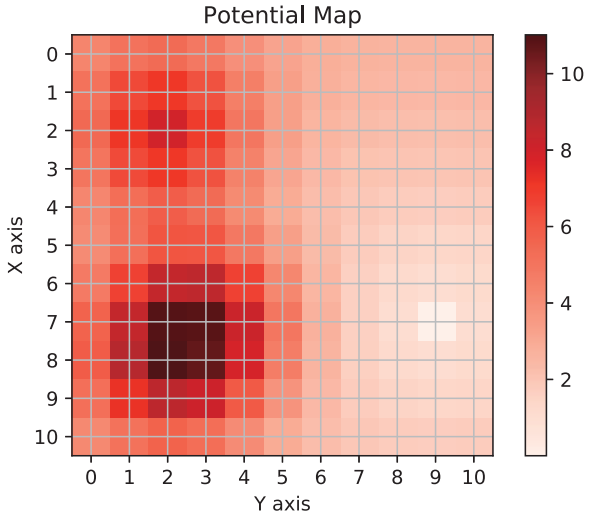


Fig. 2: A sample potential field shown in Cartesian coordinates with the destination at (7,9) and a blocked sector at $(2,2), (7,3)$, and $(8,2)$. The two adjacent sectors at $(7,3)$ and $(8,2)$ contribute a higher magnitude due to their proximity to each other. The color bar on the right indicates the magnitude of the potential in the constructed potential energy surface.

It should be noted that a simple *gradient descent* approach on the potential surface might seem to be an obvious approach. However, potential field functions suffer from local minima problem. The occurrence of local minima in the potential energy surface can prevent the gradient descent approach from reaching the global minimum which in our case is the destination. Graph-based path planning algorithms, however, have no such problem and are guaranteed to find a path to the destination, if there exists one.

Cost function. With the potential energy surface constructed using the attractive and repulsive potentials, we map a relationship between the potentials and the edge costs for the path planning algorithm. Based on the definition of the edge cost for D* Lite in [13], the edge cost to transit from s_i to s_j is $0 < c(s_i, s_j) \leq \infty$. Considering the requirements, for our application, we define the cost function as follows

$$c(s_i, s_j) = \begin{cases} r(s_i, s_j) & \text{if } U_{tot}(s_i, k) \geq U_{tot}(s_j, k) \\ r(s_i, s_j) + U_{tot}(s_j, k) & \text{otherwise} \\ \qquad - U_{tot}(s_i, k) & \end{cases} \tag{5}$$

where the total potentials $U_{tot}(s_i)$ and $U_{tot}(s_j)$ are calculated as shown in Equation (2) and $r(s_i, s_j)$ is the distance between s_i and s_j.

Local minima problem. An edge from a sector at a higher potential to a sector at a lower potential has a cost that corresponds directly to the distance between the sectors. This cost function is similar to following a gradient descent. The distance cost restricts the path from being infinitely long. However, the second case of the cost function, when the edge

is from a lower potential to an higher potential sector leads to a local minima problem. When following gradient descent, such an edge would be overlooked and therefore could result in being stuck at the local minima. The cost function for the path planning is designed to incur a penalty to traverse to an higher potential sector with the penalty being proportional to the difference in potential between the sectors. An undesired effect of avoiding the local minima problem is that, it is now possible to move into a blocked sector given sufficient potential difference. An example of this is when the path to avoid the blocked sector(s) is too long compared to the penalty incurred to travel through the blocked sector(s). It also permits moving from one blocked sector s_j to an another s_k, if $U_{tot}(s_j) \leq U_{tot}(s_k)$. To address these issues, we propose a modified cost function

$$c(s_i, s_j) = \begin{cases} r(s_i, s_j) & \text{if } s_i \text{ is not blocked and} \\ & U_{tot}(s_i, k) \geq U_{tot}(s_j, k) \\ r(s_i, s_j) + U_{tot}(s_j, k) & \text{if } s_i \text{ is not blocked and} \\ \quad - U_{tot}(s_i, k) & U_{tot}(s_i, k) < U_{tot}(s_j, k) \\ r(s_i, s_j) + U_{tot}(s_j, k) & \text{otherwise} \\ \quad - U_{att}(s_i) & \end{cases} \tag{6}$$

C. Path Planning for a Single Flight

With the transition cost between adjacent sectors defined, we apply the path planning algorithm for the case of a single flight in the airspace. As mentioned earlier in Subsection III-A, there are two cost estimates g and rhs. The cost estimate $g(s_i)$ is the cost to destination from sector s_i. The cost estimate for every sector is dynamically calculated using D* Lite.

Dynamic Updates. The generated path is based on the current information of the blocked sectors in the airspace. However, as the path is being traversed, updates to the sector capacities may require recomputation of the path.

Example 1. *To demonstrate the working for a single flight, an airspace of size* 41×41 *is considered with* 200 *randomly chosen blocked sectors. The origin sector of the flight is randomly chosen as the sector corresponding to* $(11, -10)$ *in the cartesian coordinates and similarly the destination sector corresponding to* $(-16, 5)$*. The flight path is overlayed on the potential surface* U_{tot} *in the Figure 3. Also, assuming the weather is static, the calculated heuristic cost* $g(s_i)$ *is shown for each sector* s_i *in Figure 4. It should be noted that, due to the dynamic computation of the D* Lite, the cost is calculated only at relevant sectors as seen in Figure 4.*

D. Path Planning for Multiple Flights

In this subsection, we extend the framework to include multiple flights. The air traffic scheduling is a resource sharing problem with multiple flights competing for the same set of air space sectors. We use priority based content resolution to resolve conflicts among flights.

Priority levels. As mentioned in the flight model in Section II-C, each flight f_l has a *deadline* $D_l \in \mathbb{R}^+$. Based on

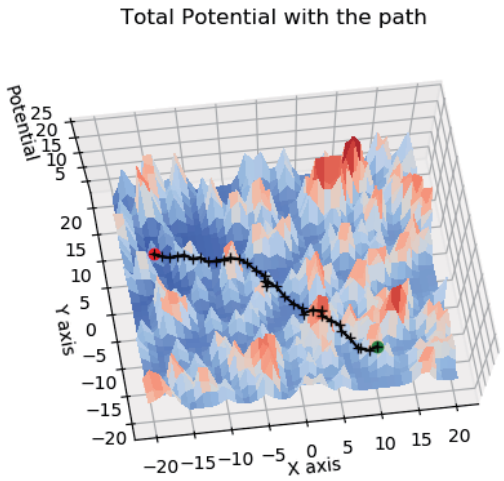


Fig. 3: The potential surface with randomly chosen 200 obstacles, $(11, -10)$ as origin sector and $(-16, 5)$ as destination sector.

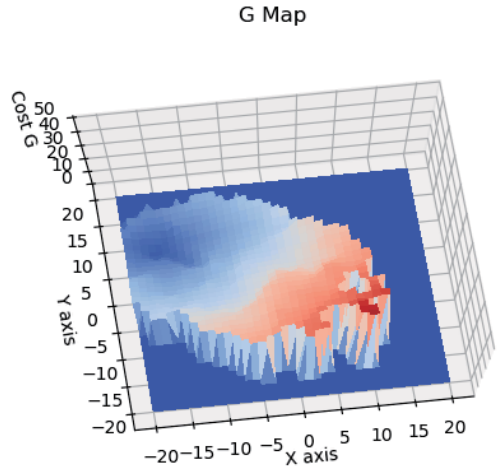


Fig. 4: A map of g values for the map shown in Figure 3

the origin and destination, assuming there is no weather interference or interference from other flights, it is possible to calculate the minimum distance $M_l \in \mathbb{R}^+$ to the destination. Then, the slack time of the flight is defined as the difference between the deadline D_l and the minimum distance M_l for the flight f_l.

$$slack_l = D_l - M_l, \quad \forall f_l \in \mathcal{F} \tag{7}$$

The slack time here refers to the amount of time the flight can afford to *waste* on its route to the destination. Based on the slack time of the flight, a priority level $p_l \in \mathbb{Z}^+$ is assigned; lower value has the higher priority. The flight with a *lower slack time* receive a *higher priority level*. Therefore the priority assignment to the flights must satisfy

$$\forall f_l, f_m \in \mathcal{F}, \quad slack_l < slack_m \iff p_l < p_m \tag{8}$$

In case, two flights have the same slack, then it is assumed that the ties are arbitrarily broken.

Flights as obstacles. The blocked sectors are not under the control of the flights and are therefore treated as obstacles and a path is planned avoiding the obstacles. Similarly, under the given priority assignment, a flight with a lower priority has no control over the actions of a flight with a higher priority and therefore treats the higher priority flight as an obstacle and constructs a path to its destination.

The overall path planning framework for multiple flights is given as a pseudo-code in Procedure 1.

Procedure 1. Path planning for multiple flights

```
Input: Origin and Destination for all flights
for each flight do
    Calculate slack using Equation (7);
    Assign priority following Condition (8);
    Current sector = Origin;
    Calculate potential fields using Equations (2, 3, 4);
    Calculate the edge costs using Equation (6);
end
for each flight do
    while Current Sector ≠ Destination do
        Compute shortest path using D* Lite;
        Current Sector = Next sector;
        Update obstacles;
        Update potential fields;
        Update edge costs;
    end
end
```

Computational Complexity. Since the underlying path-planning algorithm has the algorithm is D* Lite, the computational complexity of path planning follows the algorithm. Updating the cost function at a sector depends on each other other sector in the map. Therefore, the complexity of recalculating the cost function is $\mathcal{O}(|S|^2)$.

IV. Simulation Results

In this section, we demonstrate the working of the multiple flight path planning framework by considering an airspace with 11×11 sectors and 2 flights and few randomly chosen blocked sectors as shown in Figure 5. The flights f_1, f_2 have their origins at $(0, 4), (5, 0)$ and destinations at $(9, 5), (8, 9)$ respectively. The assigned deadlines of the flights are $D_1 = 15$ and $D_2 = 20$. The slack is calculated to be $slack_1 = 5.57$, $slack_2 = 10.94$ according to Equation (7) and therefore f_1 gets the higher priority. At time $k = 0$, the initial flight path is planned after resolving the priority levels for each flight. After the flight path is established, the flights continue along their paths at time $k = 1$ and sector capacity is continuously changing. At time $k = 3$, it can be seen that due to the blocked sectors near $(8, 6)$ and the proximity to flight f_2, the planned path for flight f_1 is updated drastically. It should also be noted that the re-computation updates only the sectors that are relevant to the path (e.g. the sectors behind the flight).

To show the time complexity of the framework, we compare the run times of the algorithm for varying sector size and for number of flights to be scheduled. We generate flights with randomly chosen origin and destinations and the airspace is generated with blocked sectors at pseudo-random locations. The algorithm is run on a Windows PC running on Intel Core $i7-3770$ CPU @ 3.40 GHz and 16 GB of RAM. Although, the CPU is quad-core, only one core was used for the simulation. The results are presented in Table I. Each entry in the table is in seconds, represents the average cumulative computation time for the initial path planning and the subsequent updates until all flights reach their destinations.

TABLE I: Average running time (in seconds) for varying airspace size and number of flights

Number of flights	Number of airspace sectors		
	25	100	400
5	0.09	0.36	2.91
10	0.16	0.65	6.02
20	0.25	1.55	19.07
50	-	5.85	57.35
100	-	-	172.21

V. Conclusions and Future Work

The air transportation system traditionally included commercial passenger aircrafts and military aircrafts as its major consumer. More recently, unmanned aerial vehicles and commercial space transportation are growing fields under the FAA. A robust decision support system that is capable of scaling up with load is required to address the future needs. Towards that larger goal, this work proposes a framework to handle multiple flights under a simple airspace model. The application of this framework is not limited to air transportation system. It can be transferred to a range of cyber-physical systems which require decision support systems such freight logistics systems. Adjusting priority dynamically during run-time is another direction that will improve the travel time.

Acknowledgement

This work is partially supported by NSF grants (CNS 1850851, C-Accel 1937833, CNS-1545050, and CCF-1725755.).

References

[1] "Modernization of U.S. Airspace," https://www.faa.gov/nextgen/, accessed: 2018-04-10.

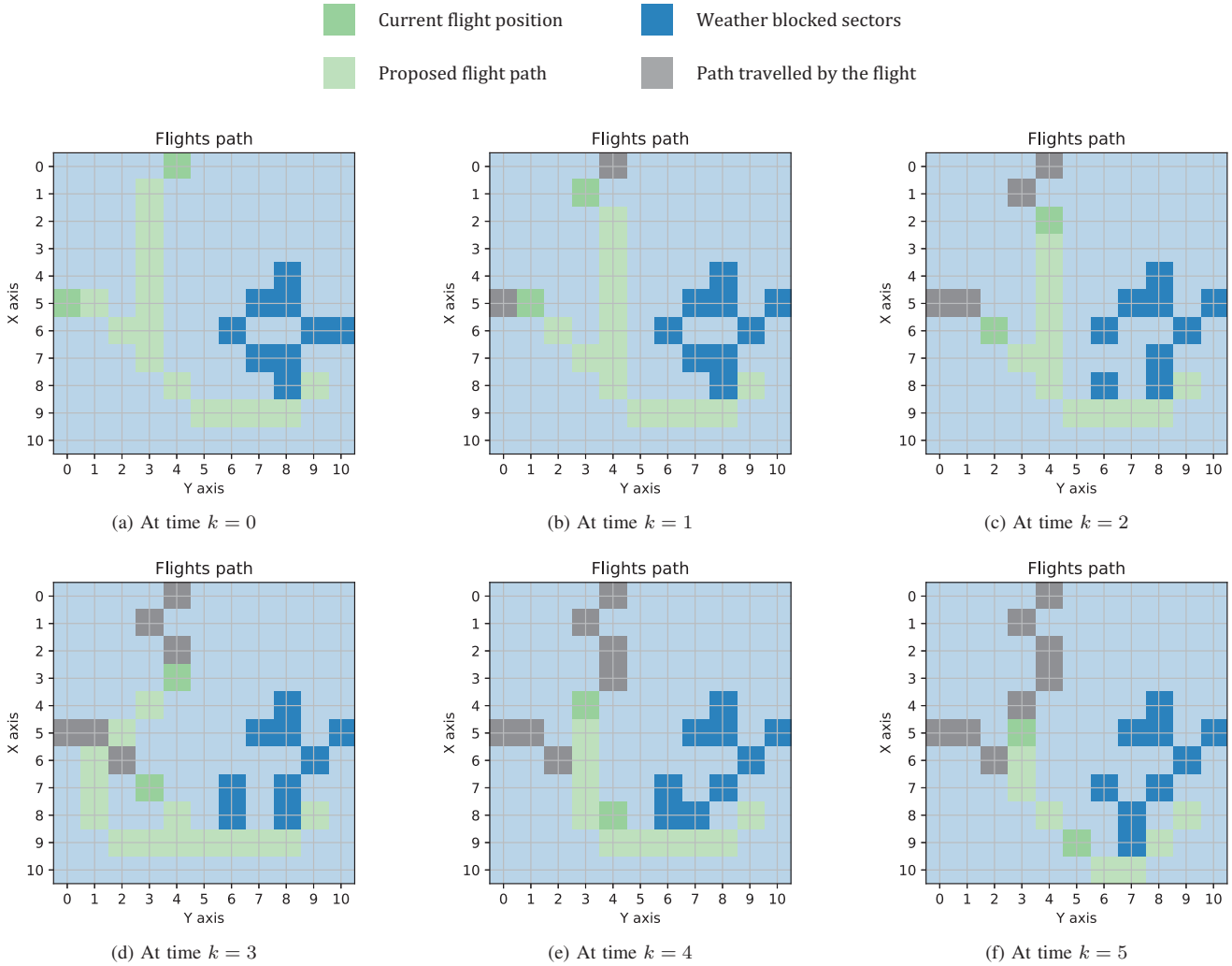
[2] J. Welch, J. Cho, N. Underhill, and R. DeLaura, "Sector Workload Model for Benefits Analysis and Convective Weather Capacity Prediction," in *Proceedings of the 10th USA/EUROPE Air Traffic Management R&D Seminar*, 2013, p. 10.

[3] D. Kulkarni, "Models of Sector Flows under Local, Regional and Airport Weather Constraints," in *Proceedings of the 36th Digital Avionics Systems Conference*, 2017, p. 7.

[4] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for a," *International journal of geographical information science*, vol. 23, no. 4, pp. 531–543, 2009.

[5] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *Journal of Global Optimization*, vol. 56, no. 3, pp. 873–895, 2013.

[6] J. C. H. Cheung, "Flight planning: Node-based trajectory prediction and turbulence avoidance: Flight planning: Node-based trajectory prediction and turbulence avoidance," *Meteorological Applications*, vol. 25, no. 1, pp. 78–85, 2018.



(a) At time $k = 0$ (b) At time $k = 1$ (c) At time $k = 2$

(d) At time $k = 3$ (e) At time $k = 4$ (f) At time $k = 5$

Fig. 5: Two flights start from sectors $(0, 4)$ and $(5, 0)$.The flight paths at consecutive time instants with varying sectors blocked by weather (dark blue). The flight paths are shown as planned at each step where the future paths (green) update as the weather changes. The grey sectors mark the actual path taken by the flights in the previous steps.

[7] Z. Qin and X. Luo, "Research on Dynamic Route Planning under Adverse Weather," in *Proceedings of 4th International Conference on Information Science and Control Engineering (ICISCE)*, 2017, pp. 1153–1157.

[8] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill Pub. Co., 1971.

[9] A. Gardi, R. Sabatini, S. Ramasamy, and T. Kistan, "Real-time trajectory optimisation models for next generation air traffic management systems," in *Applied Mechanics and Materials*, vol. 629. Trans Tech Publ, 2014, pp. 327–332.

[10] Z. W. Zhong, "Overview of recent developments in modelling and simulations for analyses of airspace structures and traffic flows," *Advances in Mechanical Engineering*, vol. 10, no. 2, Feb. 2018.

[11] C. Schilke and P. Hecker, "Dynamic Route Optimization Based on Adverse Weather Data," p. 8, 2014. [Online]. Available: https://www.sesarju.eu/newsroom/brochures-publications/dynamic-route-optimization-based-adverse-weather-data

[12] M. Xue, S. Zobell, S. Roy, C. Taylor, Y. Wan, and C. Wanke, "Using stochastic, dynamic weather-impact models in strategic traffic flow management," in *Proceedings of the 91st American Meteorological Society Annual Meeting*, vol. 15, 2011.

[13] S. Koenig and M. Likhachev, "D* lite," in *Proceedings of the 18th National Conference on Artificial Intelligence*, 2002, pp. 476–483.

[14] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, 2012, pp. 1227–1232.