Dynamic Path Planning for Unmanned Aerial Vehicles Under Deadline and Sector Capacity Constraints

Sudharsan Vaidhun, *Member, IEEE*, Zhishan Guo, *Senior Member, IEEE*, Jiang Bian, *Member, IEEE*, Haoyi Xiong, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—The US National Airspace System is currentlyoperating at a level close to its maximum potential. The limitation comes from the workload demand on the air traffic controllers. Currently, the air traffic flow management is based on the flight path requests by the airline operators, whereas the minimum separation assurance between flights is handled strategically by air traffic control personnel. In this paper, we propose a scalable framework that allows path planning for a large number of unmanned aerial vehicles (UAVs) taking into account the deadline and weather constraints. Our proposed solution has a polynomial-time computational complexity that is also verified by measuring the runtime for typical workloads. We further demonstrate that the proposed framework is able to route 80% of the workloads while not exceeding the sector capacity constraints, even under dynamic weather conditions. Due to low computational complexity, our framework is suitable for a fleet of UAVs where decentralizing the routing process limits the workload demand on the air traffic personnel.**

***Index Terms*—Air traffic, routing, conflict avoidance, simulation, unmanned aircraft.**

I. Introduction

THE Next Generation Air Transportation System (NextGen) [22] 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). In addition to commercial, private, and military aircrafts, the NextGen system considers commercial unmanned aerial vehicle (UAV) or drone fleets. In this work, we use the generic term 'flights' to refer to UAVs and aircrafts.

Each flight in the airspace requires efforts from air traffic personnel for safe operation. Consider a scenario where a set of flights are navigating a given region of airspace under the supervision of a control center. These flights may require rerouting, adjusting elevation, and other run-time flight operations. These flight operations are handled by air traffic controllers and the capacity of the airspace is represented by the number of aircraft operations per hour [5]. The overall system demand is reflected by the total workload generated by these flights.

Workload models [16], [34] have been established to characterize the workload experienced by the controllers, at the granularity of individual sectors. Traffic-dependent workload can be classified as *transit workload*, *conflict workload*, and *recurring workload*. It has been observed that the conflict workload, typically resulting from collision avoidance operations, is the dominant contributor to the workload and is most notable in small sectors [35]. For routing a fleet of UAVs, the conflict workload can be a bottleneck preventing scalability. Additionally, weather-induced constraints may increase the system workload. Ensemble techniques have been adopted to obtain better estimates of weather dynamics and, in turn, their impacts on flight planning [4]. Nevertheless, the weather-impacted airspace regions contribute to the demand and hinder scalability. Developing strategic decisions to minimize the workload is challenging because it involves long-term planning, dynamically resolving conflicts in flight paths, diverting traffic away from weather-affected regions of the airspace, and maintaining minimum separation between flights.

This work focuses on the capacity of airspace sectors (called *sector capacity*), which refers to the maximum number of flights that can be handled by air traffic controllers in a time interval. As mentioned earlier, the sector capacity is largely determined by the workload incurred by air traffic controllers managing the sector to maintain safety [34]. Among the safety constraints, *separation assurance* is the safety invariant that requires a minimum distance be maintained between any two flights at all times. The *conflict workload* largely arises to meet the separation assurance requirement. We combine the effects of weather and the flight-related workload into a single parameter, based on which the objective function is defined.

Specifically, we aim to develop a dynamic path planning framework under uncertain weather dynamics such that the utilized sector capacity is maximized. Informally, this can be achieved by controlling two factors: (1) avoiding weather-affected sectors where capacity is degraded and (2) assigning non-interfering flight routes such that the workload stays within

Manuscript received 18 February 2021; revised 31 May 2021 and 8 September 2021; accepted 5 October 2021. Date of publication 15 November 2021; date of current version 22 July 2022. This work was partially supported by NSF under Grants CNS-1545050, CCF-1725755, CNS-1818942, CNS-1850851, and OAC-2104078. *(Corresponding author: Zhishan Guo.)*

Sudharsan Vaidhun, Zhishan Guo, and Jiang Bian are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mail: sudharsan.vaidhun@knights.ucf.edu; zsguo@ucf.edu; bjbj11111@knights.ucf.edu).

Haoyi Xiong is with the Baidu Inc., Beijing 100085, China (e-mail: haoyi.xiong.fr@ieee.org).

Sajal K. Das is with the Department of Computer Science , Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Digital Object Identifier 10.1109/TETCI.2021.3122743

the sector capacity. While our proposed framework reduces the demand on the air traffic controllers by avoiding situations that require close monitoring, it complicates the routing process significantly. Our preliminary investigation [32] addressed this problem for a small region with constant and uniform flight speeds. In [32], route planning for flights with uncertain sector capacities was proposed using a D* Lite search to reduce the re-computation costs. The proposed work provides a solution that is scalable to a larger region as well as with constrained flight speeds. Although we consider weather and speed constraints in this paper, we believe other constraints can be easily integrated into the proposed framework.

Due to the emerging trend towards autonomous UAVs for civilian and logistic purposes, the demand on air traffic controllers is expected to grow rapidly. However, most of the existing works on flight path planning suffer from the following limitations. They either consider only a single flight [11], [25] or do not consider dynamically changing constraints [11], [20], [25] or are not scalable to a large number of flights [10]. To address these shortcomings of the existing works, we aim at automating the routing process, such that the air traffic controllers can act as an oversight instead of manually de-congesting the air space.

Contributions: This paper proposes a novel framework for routing UAVs while considering constraints such as the shared airspace, allowed flight speeds and weather-affected interference. We also evaluate the framework in a simulation environment of a autonomous fleet of UAVs delivering packages between a set of locations. Specifically,

1) We define a novel cost function based on potential energy fields, which captures information about the weather-affected sectors and the contending flights.
2) We propose a priority-based contention resolution mechanism to support path planning for multiple flights on a shared airspace.
3) We design a scalable scheme for global routing, utilizing the speed profile and temporal information of flight paths.
4) We evaluate the performance of the proposed framework by simulation experiments and explore the effects of airspace size, number of flights, maximum flight speed on runtime and routing success.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III describes models to represent the airspace and flights, and defines the routing problem with constrained sector capacities. Section IV considers a simplified flight model and adapts traditional path planning algorithms to address the problem. Considering a more general case, Section V proposes a scalable algorithm for much wider airspace region. Section VI evaluates the performance of the proposed framework through simulation experiments. The final section concludes the paper.

II. Related Work

Extensive research has been conducted to optimize the air-traffic routing. Here we categorize them into three major groups—graph search [9], mathematical programming [40], and reinforcement learning [24], [37]. Graph theory has been use to solve routing problems such as de-conflicting optimal trajectories [19], [31] and avoiding convective weather [18]. Then, graph search techniques such as A^* search [39] and geodesic computation [6] are used to solve the objective. Similar to our proposed work, route optimization framework under adverse weather data was proposed by Schilke and Hecker [29] for larger area with a long time horizon. However, such an approach is not suitable for smaller sectors where time horizon is much shorter. In addition to the routing objective, several constraints have been considered, such as fuel burn and emissions [7], [23], [29] and weather risk [38]. The presence of uncertainties have also been investigated in the path planning procedures [3], [26].

Besides graph theory, integer programming techniques and it has been used for optimizing the airspace based on different criteria such as flight costs, travel delays, and throughput [28], [40], [41]. Another alternative approach is the use of machine learning approaches such as long short-term memory (LSTM) [30], reinforcement learning [14] and self-organizing maps [12]. Reinforcement learning techniques such as Q-Learning [33] and their derivative algorithms such as multi-agent deep deterministic policy gradient (MADDPG) [24] and dueling double deep Q-Networks (D3QN) [37] have also been proposed. In learning-based approaches, the optimality of the solution depends on the convergence rate and error tolerance, which may lead to higher computational requirement. However, learning-based approaches have the benefit of updating the path continuously whenever there is an update in the weather forecast data.

Besides routing frameworks, recent research has also focused on the use of accurate position and heading of flights, better weather prediction models [15], [16], the usage of data communication instead of voice communication [27], and flow management programs like ground delay or en-route delay [38]. A detailed survey of optimal path planning techniques with various objectives (e.g., fuel savings, delay minimization, emission reductions, etc.) is proposed in [42] and more recently in [10]. As a more general case, the trend towards automation in unmanned vehicles is surveyed in [21].

III. System Model and Problem Statement

This section introduces the models to represent the airspace and flights and formulates the air traffic scheduling problem.

A. Airspace Sector Model

The airspace consists of contiguous sectors from the set $S = \{s_1, s_2, \ldots, s_n\}$. It can be represented by a graph $G = (V, E)$ where the vertices are sectors and the edges are formed by adjacent sectors. For any vertex or sector $s_i \in S$, the subset $N(s_i) \subset S$ represents the neighbors of s_i. That is, $\forall s_j \in N(s_i)$, an aircraft can transit *directly* from s_i to s_j following the edge $e_{i,j} = (s_i, s_j) \in E(G)$. Each sector has an associated sector capacity, which represents the upper limit on the workload that can be managed by the air traffic controllers. Sectors impacted by weather conditions constraints reduce the capacity, and flights in a sector consume the available sector capacity. To simplify the demand-capacity relationship, we treat the capacity reduction

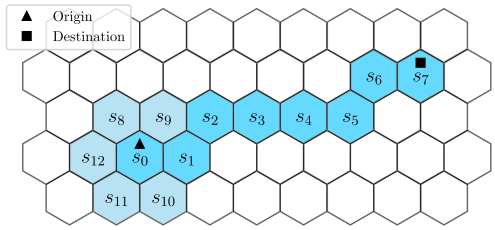


Fig. 1. A sample representation of the airspace with sectors $s_1, s_2, \ldots, s_{15}$ considering hexagonal grid. The highlighted sectors represent a flight path from s_0 to s_7.

due to weather interference as the capacity being consumed by the weather. Then, we quantify the relationship using a parameter $x_i(k)$ which denotes the sector capacity of s_i utilized at time k.

Although the altitude of the flight is not explicitly considered, the airspace is modeled as a graph and does not restrict the airspace to be two-dimensional. If the airspace is vertically divided to represent multiple altitudes (also referred to as *flight levels*), then the adjacent vertical sectors are neighboring sectors and represented with an edge in the graph. In this work, the airspace sectors are considered to be regular hexagons (i.e., with equal sides and equal angles), and therefore the distance between neighboring sectors is assumed to be equal to the *Euclidean distance* $r(\cdot,\cdot)$ between the center of the sectors.

B. Flight Model

We consider a set of flights $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$. Each flight $f_l \in \mathcal{F}$ is represented as

$$f_l = \{o_l, d_l, D_l, v_l^{min}, v_l^{max}\} \tag{1}$$

where o_l and d_l respectively represent the origin and destination sectors of the flight f_l, such that $o_l, d_l \in S$. The deadline D_l denotes the maximum allowed time to reach the destination d_l. The speeds v_l^{min} and v_l^{max} denote the minimum and maximum speeds of the flight f_l. It is assumed that the flight does not operate outside its speed range during the entire journey. A flight path p_l of f_l from origin o_l to destination d_l is represented by a sequence of nodes in the graph G:

$$p_l = \langle o_l \equiv s_1^l, s_2^l, \ldots, s_i^l, \ldots, s_{|p_l|}^l, s_{|p_l|+1}^l \equiv d_l) \rangle$$

such that the edge $(s_i, s_{i+1}) \in E(G)$, for $1 \leq i \leq |p_l|$, where $|p_l|$ is the path length.

Example 1: Fig. 1 illustrates an airspace with a flight f_1, origin sector $o_1 = s_0$ and destination sector $d_1 = s_7$. The neighbors of the origin is given by $N(s_0) = \{s_1, s_8, s_9, s_{10}, s_{11}, s_{12}\}$. A possible flight path is given by $p_1 = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$.

When multiple flights share the airspace, it is possible that the flight paths overlap in space, although not necessarily in time. For safety reasons, a minimum separation distance is maintained between flights at all times during the journey. For flight f_l, let the minimum separation distance be X_l.

TABLE I
SUMMARY OF NOTATIONS

Set of sectors	$S = \{s_1, \ldots, s_i, s_j, \ldots, s_n\}$
Set of flights	$\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$
Airspace	$G(V, E)$
Sector capacity of s_i utilized at time k	$x_i(k)$
Neighboring sectors of s_i	$N(s_i)$
Distance between sectors s_i and s_j	$r(s_i, s_j)$
Length of the path from s_i to s_j along p_l	$r_l(s_i, s_j)$
Cost to move from s_i to s_j at time k	$c(s_i, s_j, k)$
Potential functions	U_{tot}, U_{att}, U_{rep}
Earliest and latest time for f_l to reach s_i	t_{i-}^l, t_{i+}^l

C. Weather Model

The exact dynamics of weather and its impacts are unknown. However, we assume that the impact on sector capacity follows the weather-impact model [36] designed for analysing traffic flow management problems. In the weather-impact 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 $A_{i,j_{m,n}} = p(x_i[k+1] = n | x_j(k) = m)$.

For example, let s_0 be the influencing sector for s_1 and $x_1[0] = 1; x_0[0] = 0$ be the sector capacities at the two sectors at time $k = 0$. And assuming $A_{1,0} = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix}$ implies

$$p[x_1[1] = 0 | x_0[0] = 0] = 0.9$$
$$p[x_1[1] = 1 | x_0[0] = 0] = 0.1$$

Therefore, according to the weather-impact 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. We use the weather-impact model to generate weather data for simulation purposes as well.

D. Problem Statement

Given a graph G and a set of flights $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$, the goal is to develop a dynamic path planning framework for each flight $f_l \in \mathcal{F}$ that generates paths p_l with the objective

$$minimize \quad \sum_{1 \leq i \leq n, k \in \mathbb{Z}^+} \max(x_i(k) - 1, 0) \tag{2}$$

$$\text{such that,} \quad |p_l| / v_l^{max} \leq D_l \tag{3}$$

Inequality 3 is a necessary constraint for the proposed path p_l to be feasible. It states that the flight f_l must be able to reach the destination d_l within the deadline D_l, subject to the speed limit of v_l^{max}. The sector capacity $x_i(k)$ is affected by both weather as well as the flights. The objective is to minimize the utilized sector capacity from exceeding above 1 for each sector. The objective function can be minimized by avoiding weather-affected sectors as well as by avoiding multiple flights from being in any sector. By minimizing the objective function, the workloads for the

air traffic controllers arising from flight interference as well as weather-impact are both minimized.

IV. Path Planning in a Local Region

In order to solve the flight path planning problem, spatial and/or temporal conflicts are to be resolved when the flight paths overlap in case of multiple flight routing. In this section, we first consider a simplified flight model and map the problem to the shortest path problem in graph theory. Next, we propose a cost function to represent dynamic obstacles and a suitable graph search algorithm. Finally, we propose a solution to handle multiple flights as well as weather-affected sectors.

A. Simplified Flight Model

We make two simplifications to the flight model in (1) and map the problem to the shortest path problem. Specifically, we restrict the flight speed to a single fixed value and eliminate the deadline requirement. That is, $v_l^{min} = v_l^{max} = v$ is the constant flight speed and the deadline $D_l = \infty$. (1) is then re-written as:

$$f_l = \{o_l, d_l, \infty, v, v\}, \forall f_l \in \mathcal{F}$$

Note that we only simplify the flight model, but consider the same weather model. Since the flight speeds and deadlines do not have any influence in the routing procedure, we will use the simplified flight model defined in (4) for the remainder of the section.

$$f_l = \{o_l, d_l\}, \forall f_l \in \mathcal{F} \tag{4}$$

For this simplified formulation, traditional routing techniques can be used although they must be modified to account for sector capacity while planning paths. The following subsections introduce the D* Lite path planning algorithm and our modifications to meet the sector capacity requirements.

B. Cost Function

Graph-based path planning algorithms require to define (application-specific) cost function $c(s_i, s_j)$ for transition from a sector s_i to another sector s_j. This subsection first describes the requirement for cost function and drawbacks of a static cost function. Then it introduces potential fields to allow dynamic cost functions, and defines a cost function for path planning problem based on the potential fields.

The cost function in graph-based path planning techniques require the condition: $c(s_i, s_j) \geq 0$. Traditionally, the cost function in routing algorithms is the Euclidean distance metric, with the exception that the distance to the obstacle vertices (sectors) carries an infinite cost. Such a metric is suitable when the underlying graph parameters are static. However, in our application, the obstacles are dynamic and uncertain. In the reminder of the subsection we propose a novel cost function that captures the uncertainty of the obstacles. Later, in Section IV-C, we adapt a dynamic path planning algorithm to utilize our cost function. Fig. 3(b) demonstrates the improvement in the flight path using our proposed cost function.

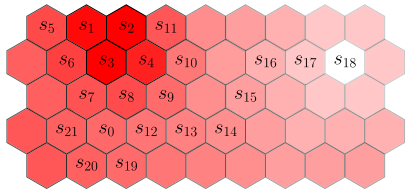


Fig. 2. A sample potential field for the example graph in Fig. 1. The potential is the highest at the obstacle sectors and the lowest at the target sector. The intensity of the color represents the potential with a brighter color representing higher potential than the pale regions.

Potential functions have been extensively used in global path planning algorithms [17]. The potential function approach is to construct a potential energy surface over the area considered for path planning with the goal to find the point(s) with the lowest potential energy. The points in the energy surface with the highest energy represent the obstacles. Two major components that combine to form the potential energy surface are: the *attractive potential* (U_{att}) and the *repulsive potential* (U_{rep}). Let the potential function associated with an aircraft at sector s_i at time instant k be given as:

$$U_{tot}(s_i, k) = U_{att}(s_i) + U_{rep}(s_i, k) \tag{5}$$

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 d_l of flight f_l. Thus,

$$U_{att}(s_i) = k_{att}\sqrt{r(s_i, d_l)} \tag{6}$$

where $k_{att} \in \mathbb{R}^+$ is the attractive potential constant. The attractive potential is chosen to be a quadratic function so that the magnitude of the slope increases as the flight approaches the destination. At destination d_l for flight f_l, we have $r(d_l, d_l) = 0$ which implies $U_{att}(d_l) = 0$. Moreover, since the attractive potential depends on the goal sector, $U_{att}(d_l)$ at any given sector s_i does not change due to obstacles. Note that the attractive potential is independent of the time instant k, since it only depends on the distance to destination.

Repulsive Potential: The repulsive potential $U_{rep}(s_i, k)$ 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; it is chosen to be a two-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 other sectors in the airspace is given by

$$U_{rep}(s_i, k) = \sum_{s_j \in S} k_{rep,j} x_i[k] exp\left(\frac{r(s_i, s_j)}{\sigma_{rep,j}}\right) \tag{7}$$

where $k_{rep,j} \in \mathbb{R}^+$ is the repulsive potential amplitude and $\sigma_{rep,j} \in \mathbb{R}^+$ is the spread of the Gaussian function.

The parameters k_{rep} and σ_{rep} together minimize the sector capacity $x_i(k)$ used for any s_i at time k. A larger value of $\sigma_{rep,j}$ denotes a wider spread and is proportional to the uncertainty of

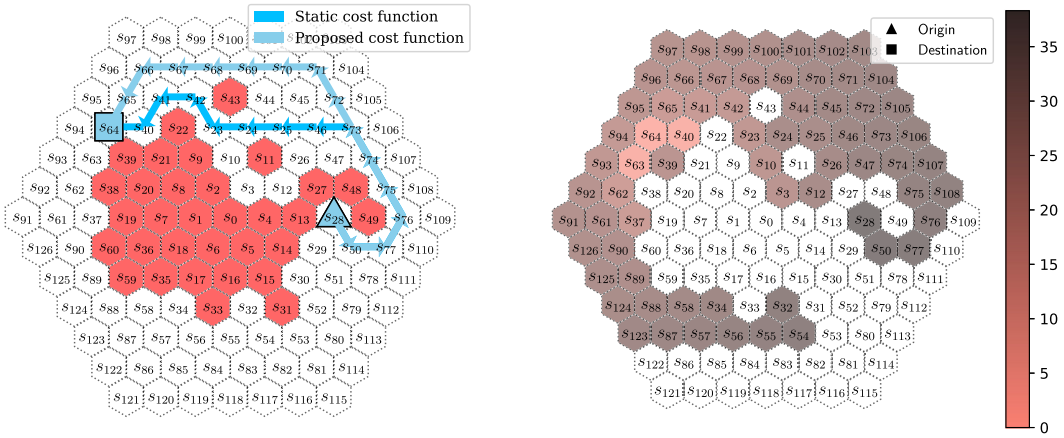
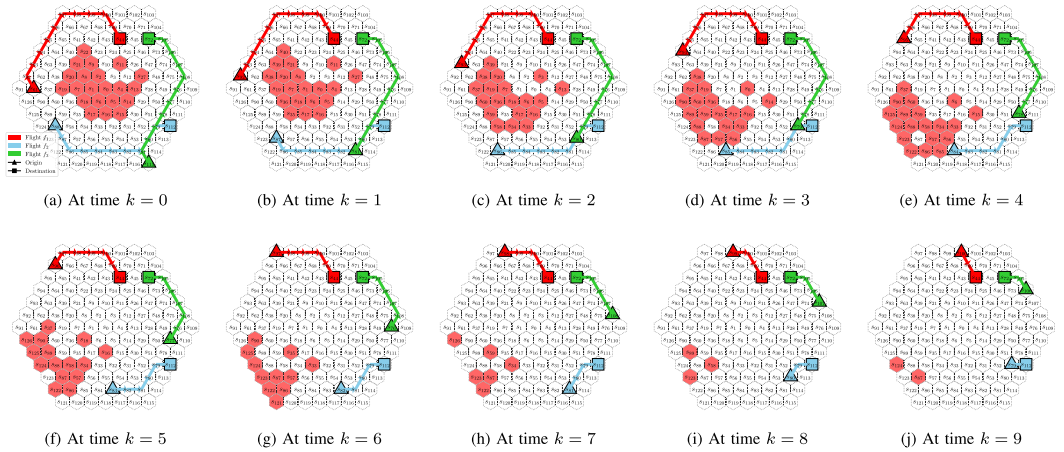


Fig. 3. Paths generated using a static cost function and our proposed cost function for a flight navigating in an airspace with 127 sectors and randomly generated obstacles. (a) The flight paths resulting from using a static cost function and the proposed cost function is shown along with weather-affected sectors highlighted in red. (b) A map of g values showing the cost to the destination using the proposed cost function.



(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 $k = 5$ (g) At time $k = 6$ (h) At time $k = 7$ (i) At time $k = 8$ (j) At time $k = 9$

Fig. 4. A sequence of snapshots of an airspace showing the dynamic updates using Procedure 1 to route 3 flights. The projected flight paths from the current sector to the destination are represented by arrows. The weather-affected sectors are represented by colored sectors and their dynamics (from Sub-section III-C) causing them to move directionally with time.

the weather at a given sector s_j. Larger values force a larger separation between flights and the weather-affected sectors, thereby minimizing $x_i(k)$. Note that σ_{rep} is chosen to be at least 1 to guarantee a minimum separation of at least 1 sector between the flights and obstacles.

Example 2: Consider a flight f_2 with origin o_2 at s_0 and destination d_2 at s_{18}. Sectors s_1, s_2, s_3 and s_4 are congested or restricted to fly. In such a scenario, the flight path $p_2 = \{s_0, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}\}$ avoids congested region in the airspace. Fig. 2 illustrates the potential field.

Note that a simple *gradient descent* approach on the potential surface might seem natural to compute the shortest path. However, the potential field functions suffer from the *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 edge cost for D* Lite in [13], the edge cost to transit from s_i to s_j is $0 < c(s_i, s_j) \leq \infty$. In our application, the edge cost also depends on time k; and is denoted as (s_i, s_j, k). Based on the requirements for our application, we define the cost function as follows:

$$c(s_i, s_j, k) = \begin{cases} r(s_i, s_j) \\ \text{if } x_i(k) = \quad \& \quad U_{tot}(s_i, k) \geq U_{tot}(s_j, k) \\ \\ r(s_i, s_j) + U_{tot}(s_j, k) - U_{tot}(s_i, k) \\ \text{if } x_i(k) \neq 1 \quad \& \quad U_{tot}(s_i, k) < U_{tot}(s_j, k) \\ \\ r(s_i, s_j) + U_{tot}(s_j, k) - U_{att}(s_i) \\ \text{otherwise} \end{cases} \tag{8}$$

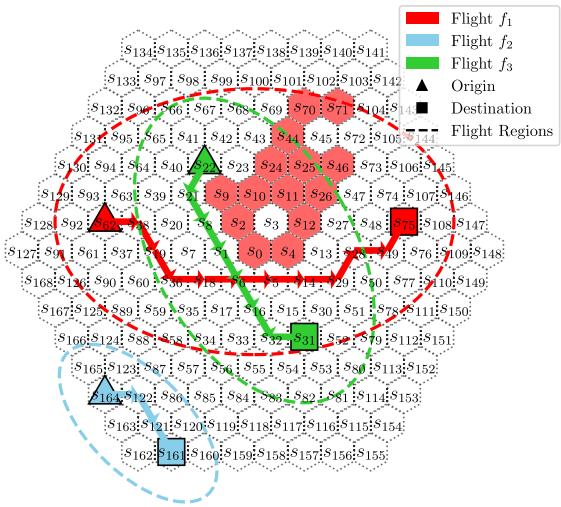


Fig. 5. Multi-flight path planning in a global region showing 3 flights along with their valid regions (dashed ellipses) and their flight paths (shown using arrows) with obstacle sectors.

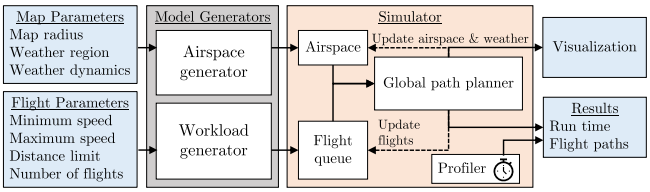
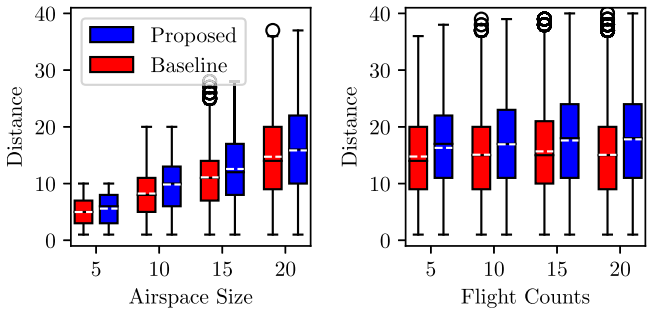


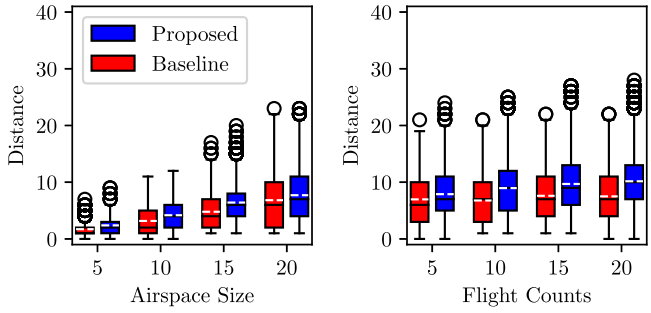
Fig. 6. Overview of the simulation setup.



(a) Varying airspace size (b) Varying number of flights

Fig. 7. Distribution of the distance between any pair of flights at any point during their journey.

If the current sector s_i is blocked and is at a higher potential compared to the next sector s_j, then the cost to travel is equal to the distance cost. This first condition is to enforce the path to travel away from block sectors. The second condition is the case when the current sector s_i is unblocked and the next sector s_j is at a higher potential. Then the cost to travel to s_j is the difference in potential in addition to the cost of the distance. This second condition is to prevent being stuck in a local minima where the additional cost penalty is paid to escape the minima. For all other case, the cost is distance cost in addition to a potential difference. Note that the repulsive potential of the current sector is ignored.



(a) Varying airspace size (b) Varying number of flights

Fig. 8. Distribution of *minimum* distance between any flight and weather-affected sectors at any point during the journey.

Having the cost function defined, we are ready to propose our path planning algorithm to compute the flight paths. Before proceeding further, let us briefly mention the the D* Lite algorithm. (For further details, please refer to [13].)

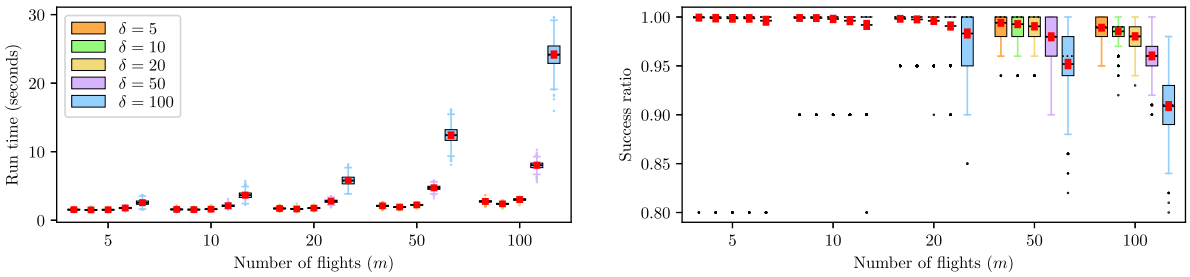
C. D* Lite Algorithm

For any given sector $s_i \in S$ in the airspace graph, the D* Lite algorithm maintains two cost estimates. The first estimate $g(s_i)$ is the shortest path from the origin sector to s_i. Since we do not know the actual cost of shortest path to s_i, we use the Euclidean distance as the estimate: $g(s_i) = r(s_o, s_i)$ where s_o is the origin. The second estimate $rhs(s_i)$ (refer to (1) in [13]) for flight f_l is a 'look-ahead estimate' given by

$$rhs(s_i) = \begin{cases} 0 & \text{if } s_i = o_l \\ \min_{s' \in N(s_i)} g(s') + c(s', s_i) & \text{otherwise} \end{cases}$$

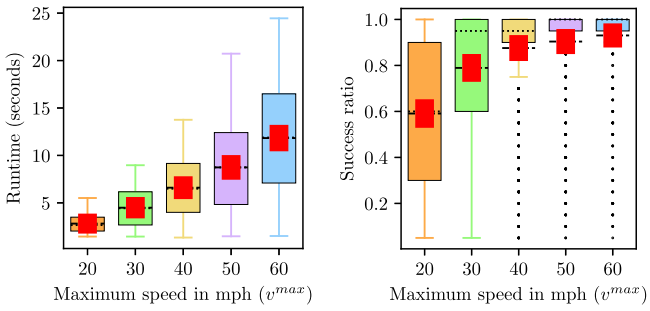
where $c(s', s_i)$ is the cost to travel from s' to s_i. When the two estimates $g(s_i)$ and $rhs(s_i)$ are different, the sector s_i in the graph is marked as *inconsistent* and is placed in a priority queue. The shortest paths through inconsistent sectors are calculated, and the sector with a shorter path gets a higher priority in the queue. Updating the g and rhs estimates of the inconsistent sectors may result in additional inconsistent sectors, which are then placed in the priority queue. The queue is iteratively processed until the shortest path to the destination is found. If the priority queue is empty and there is a path to the destination, the estimated value of g reaches its actual value of g^*. When an edge cost is changed, the g estimate is updated, causing it to deviate from the rhs estimate. As mentioned earlier, the inconsistencies are resolved iteratively until all sectors are consistent again.

Two essential features of the D* Lite algorithm that make it suitable for our route planning problem are: (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 sector. The exploration process only updates the neighboring sectors and, among those, only the inconsistent sectors are further processed. (2) When the map changes due to updates in the weather information, the changes are propagated to the current sector. Such propagation eliminates the need for



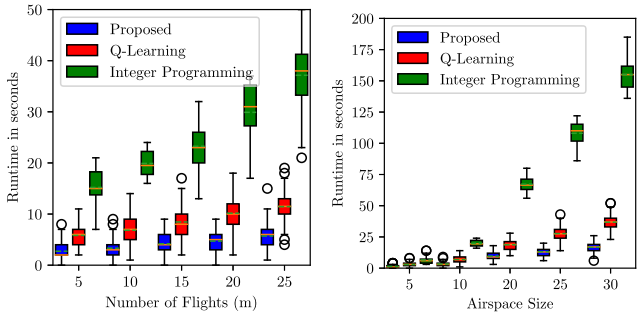
(a) Variation of runtime (b) Variation of success ratio

Fig. 9. Simulation runtime of and success ratio of routing the flights for varying number of flights and distance thresholds. The subfigures share the same legends. The means are represented by the horizontal dashed line and the median by the horizontal dotted line – 95% confidence intervals of the means are represented by the red boxes.



(a) Variation of runtime (b) Variation of success ratio

Fig. 10. Runtime and success ratio with the maximum speed allowed for the flights, where the minimum speed is restricted to 10 mph. The means are represented by horizontal dashed lines – their 95% confidence interval is represented by red boxes. The medians are represented by horizontal dotted line. The legends for the sub-figures is the same as sub-figure 9(a).



(a) Varying the number of flights (m)(b) Variation of airspace size when the when the number of sectors $n = 133$ number of flights $m = 10$

Fig. 11. Total runtime to route for varying number of flights in airspaces of different radius.

recomputing the path from scratch and updates only a minimal set of sectors to get the updated path.

D. 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 in the previous subsection, there are two cost estimates g and rhs, where $g(s_i)$ is the cost from the origin to s_i. The cost estimate for every sector is dynamically calculated using the D* Lite algorithm.

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

Example 3: To illustrate the working of the algorithm for a single flight, we consider an airspace with 127 sectors with randomly chosen blocked sectors. The origin and destination sectors of the flight are randomly chosen as s_{28} and s_{64}, respectively. In Figure 3(a), the flight path using distance-based static cost function is represented in dark blue, while the path using the proposed cost function is in light blue. Note that the proposed cost function leads to a longer path, which maintains a safe distance from the weather sectors, should they move towards the planned flight path. However, the red path does not maintain a safe distance and thus is susceptible to frequent path changes. Based on the weather at a fixed time, the estimated cost $g(s)$ for a complete map is shown in Figure 3(b), in which the cost is calculated only at relevant sectors due to the dynamic computation of D* Lite.

E. Path Planning for Multiple Flights

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

Priority Levels: As mentioned in the flight model in Section III-B, each flight f_l has a *deadline* $D_l \in \mathbb{R}^+$. Based on the origin and destination, assuming there is no interference due to weather or from other flights, it is possible to calculate the shortest distance to destination and the corresponding time taken $M_l \in \mathbb{R}^+$. Then, the *slack time* of the flight $f_l \in \mathcal{F}$ is defined as the difference between the deadline D_l and the shortest time M_l. That is,

$$slack_l = D_l - M_l \tag{9}$$

The slack time 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 $\pi_l \in \mathbb{Z}^+$ is assigned; the lower value

Procedure 1. Path planning for local region.

```
Input: Origin and Destination for all flights
for each flight do
    Calculate slack using Equation (9);
    Assign priority following Condition (10);
    Current sector = Origin;
    Calculate potential fields using Equations (5, 6, 7);
    Calculate the edge costs using Equation (8);
end
for each flight do
    while Current Sector ≠ Destination do
        Compute shortest path using D* Lite;
        Current Sector = Next sector;
        Update obstacles, potential fields, edge costs;
    end
end
```

has the higher priority. The flight with a *lower slack time* receives a *higher priority level*. Therefore the priority given to the flights must satisfy

$$\forall f_l, f_m \in \mathcal{F}, slack_l < slack_m \iff \pi_l < \pi_m \tag{10}$$

If two flights have the same slack, the tie is arbitrarily broken.

Flights as Obstacles: Generally, blocked sectors are not under the control of the flights and are therefore treated as obstacles, and a path is planned to avoid the obstacles. Under a given priority assignment, the path of a flight with higher priority is unaffected by the lower priority flights. Alternatively, the priority levels imply that a lower priority flight cannot occupy a sector occupied by a higher priority flight. For this reason, the sectors occupied by higher priority flights are treated as obstacles by the lower priority flights.

Procedure 1 presents the overall path planning framework for multiple flights in a local region. In this procedure, the D* Lite is the underlying path planning algorithm. As shown by experimental results in [13], the D* Lite algorithm is at least as efficient as the D* algorithm. Thus, the computational complexity of the initial setup and subsequent runtime adaptations in Procedure 1 are the same as D* Lite which is $\mathcal{O}(n)$ following Theorem 1 in [13]. The computational complexity of recalculating the cost function is $\mathcal{O}(n^2)$.

Let us illustrate the multiple flight path planning framework with an example that applies the D* Lite method.

Example 4: Consider an airspace with a radius of 7 sectors resulting in 127 sectors, and assume 3 flights. The flights f_1, f_2, f_3 have their origins at sectors s_{62}, s_{31}, and s_{164} respectively and their destinations at s_{73}, s_{22}, and s_{161} respectively. The flights and their initial flight paths according to Procedure 1 are shown in Fig. 4(a). The blocked sectors represent the weather-affected sectors, whose dynamics follow the assumptions made in Subsection III-C.

Flight f_1 gets a higher priority and has its shortest path p_1 to the destination. Whereas, flight f_2 has a lower priority, and the total cost of the flight path to go around f_1 is larger than the cost to go through it. This results in a path p_2 for f_2 which intersects with p_1. The third flight f_3 is relatively unaffected by the higher priority flights due to its distance from the flight paths. However, following Procedure 1, the flight path calculation for p_3 considers the paths p_1 and p_2 even though they never interfere directly. At time $k = 0$, the initial flight paths are planned after resolving the priority levels for each flight. According to the simplified flight model, the flights travel at the same speed. At time $k = 3$, due to the blocked sector at s_2, the flight path for f_2 is dynamically updated to avoid s_2 and follow via s_1. Note that the re-computation updates only the sectors that are relevant to the path (e.g., the sectors behind the flight). The sequence of sub-figures in Fig. 4 shows the entire traversal.

V. Path Planning for a Global Region

In Section IV, we considered a simplified model and adapted existing path planning solutions to solve the problem. However, for the D* Lite-based approach, the cost function for each edge depends on all vertices in the graph. This is not scalable for larger graphs and can only be used for smaller geographical regions. In this section, we consider the original flight model (see Sub-section III-B), show how to restrict the considered airspace for each flight, and design a scalable routing algorithm. The proposed framework works in three stages: (1) to limit the airspace considered by a flight, we calculate the feasible region that reduces the map size and memory consumption; (2) we identify the pairs of flights that interfere spatially; and (3) we attempt to isolate the flights spatially, and if that fails, we temporally isolate the flights. If temporal isolation is not feasible, the flight will interfere; and it is the only scenario where the air traffic controllers interfere.

A. Routing Feasibility

The local routing algorithm requires information about the surrounding sectors. However, not all sectors can interfere with the flight. This is also evident from the map of g values as illustrated in Example 3 (Fig. 4). Therefore, to improve scalability, we identify the valid region of the flight. The flight model restricts the speed of the flight by an upper limit and also has a deadline by which the flight has to reach the destination. Combining these two restrictions help decide the maximum distance the flight can move away from either the source or the destination. We first define a valid path for a flight such that the flight can reach the destination subject to those restrictions, and then define a valid region within which the path resides.

Valid Path: Consider a flight path p_l given by

$$p_l = \langle (s_1, s_2), (s_2, s_3) \ldots, (s_k, s_{k+1}) \rangle$$

Then p_l is a valid path if

$$\exists T_i^l \in \left(\frac{r(s_i, s_{i+1})}{v_l^{max}}, \frac{r(s_i, s_{i+1})}{v_l^{min}} \right) \text{s.t.} \sum_{1 \le i \le k} T_i^l \le D_l \tag{11}$$

where T_i^l is the time traverse the edge (s_i, s_{i+1}) by flight f_l.

Informally, for flight f_l, any path p_l is a valid path if the edges in the path can be traversed while maintaining the flight speed $v \in [v_l^{min}, v_l^{max}]$.

Definition 1 (Valid region): A valid region for a flight f_l is represented by a subset of sectors $R_l \subseteq S$ such that each sector $s_i \in R_l$ follows the following condition

$$\frac{d(o_l, s_i) + r(s_i, d_l)}{v_l^{max}} \leq D_l \tag{12}$$

Remark 1: The valid region R_l forms an elliptical region with the origin sector o_l and destination sector d_l as the foci of the ellipse.

Remark 2: Any valid path p_l of a flight f_l lies within its valid region R_l.

B. Flight Interference

The repulsive potential used in the D* Lite-based approach in Section IV is due to other flights and sectors blocked owing to the weather. However, for routing, not all flights need to be considered. In this subsection, we attempt to spatially isolate the flights. If spatially isolation is not possible, then we identify such flights for further processing. Due to the dynamic nature of these obstacles, the sector capacities affected by these obstacles also vary in time. Since we are modeling the flights as obstacles, to guarantee minimum separation, the information of the entire graph is required when using traditional path planning algorithms. However, according to Condition 12, the valid flight has to exist within the valid region. Following Remark 2, two flights cannot obstruct each other if their valid regions do not overlap. Thus, it is sufficient to consider only the flights whose valid regions overlap to account for interference during path planning.

Interfering Flights: The interfering flights for flight $f_l \in \mathcal{F}$ is defined as

$$I_l = \{f_i | R_i \cap R_l \neq \phi\} \tag{13}$$

Equation (13) gives the set of flights interfering in the spatial domain. To calculate the window of temporal interference between two flights, we first calculate the time interval (t_{i-}^l, t_{i+}^l) at which a flight can exist in any sector in its valid region.

Calculating the Time Interval: Given a path p_l, the earliest time when the flight can reach the sector s_i is given by t_{i-}^l. This earliest time is the time taken to reach s_i from the flight's origin at $o_l \equiv s_0$ at its maximum speed v_l^{max} along the path p_l. Note that the distance calculation is pessimistic since the flight may not necessarily travel at its highest speed, implying the flight will arrive later than expected. Additionally, the assurance of separation has to be guaranteed as mentioned in Subsection III-B. To account for separation assurance, a flight f_l is said to have *reached* s_j, if it has reached its previous node s_i in the path. This is because when the flight is along the edge (s_i, s_j), there can be no other flight in sector s_j. Formally, the earliest time is given by

$$t_{i-}^l = \frac{r_l(o_l, s_j) - r_l(s_i, s_j)}{v_l^{max}} \tag{14}$$

where $s_j \equiv s_j^l$; $s_i \equiv s_{j-1}^l$ and $\{s_j^l, s_{j-1}^l\} \in p_l$. The function $r_l(s_i, s_j)$ is the distance between s_j and s_i along the path p_l.

Procedure 2. Path planning for global region.

Input: f_l, S, G

$\mathcal{F} = \phi$;
for *each flight* f_l **do**
  Calculate valid region R_l using Condition (12);
  Calculate interfering flights I_l using Equation (13);
  p_l = CalculateFlightPath(f_l, I_l);
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_l\}$;
end

Similarly, the latest time t_{i+}^l a flight can be in sector s_i such that it can meet its deadline, is given by

$$t_{i+}^l = D_l - \frac{r_l(s_i, d_l) - r_l(s_i, s_j)}{v_l^{max}} \tag{15}$$

where $s_i \equiv s_i^l$; $s_j \equiv s_{i+1}^l$ and $s_i^l, s_{i+1}^l \in p_l$.

Consider two interfering flights f_l, f_m, where $f_m \in I_l$. Let $s_i \in R_l \cap R_m$ be a sector where both flights interfere. Either of the following conditions are sufficient to guarantee temporal isolation (i.e., two flights do not overlap temporally):

$$t_{i+}^l < t_{i-}^m \forall f_m \in I_l, s_i \in S, \tag{16}$$

$$t_{i+}^m < t_{i-}^l \forall f_m \in I_l, s_i \in S. \tag{17}$$

C. Weather Interference

The sector capacity can be affected for reasons other than interfering flights, e.g., restricted no-fly zones and weather conditions. In case of bad weather conditions, based on an ensemble forecast, the time interval(s) during which the sectors can be affected can be interpreted. By representing blocked sectors in terms of time intervals, the weather-affected sectors can all be treated as one large high-priority flight.

The weather is represented as flight f_w whose path is unknown. The sectors affected by the weather are represented by the region R_w. Note that the sectors in the region R_w may not be contiguous. A single weather entity can represent multiple weather clusters. When calculating flight interference, the weather-affected regions are treated as high-priority interfering flight. As the weather information is updated, the region R_w is updated accordingly. Correspondingly, the values t_{r-}^w and t_{r+}^w are calculated for $s_r \in R_w$.

D. Algorithm for Global Routing

The algorithm for global routing starts by processing flights based on their priority ordering, which is first-in-first-out. The weather is treated as the flight with the highest priority. Procedure 2 presents the algorithm for global routing. For each flight f_l, the first step in the procedure calculates its valid region R_l given by Condition (12). The second step is to find a set of interfering flights. Since the flights with lower priority are not routed yet, only the flights with higher priority than f_l are considered for calculation. With the interfering flights calculated, Procedure 3 determines the flight path.

Procedure 3. CalculateFlightPath.

Input: f_l, I_l, R_l

1 $X_l \leftarrow R_l \cap \left(\bigcup_{f_x \in I_l} p_x \right)$;
2 Construct G_l such that $V(G_l) = R_l$;
3 **for** $s_i \in X_l$ **do**
4 $\quad p_l^{i-} \leftarrow$shortest path from o_l to s_i in G_l;
5 $\quad$ Calculate t_{i-}^l given the path p_l^{i-};
6 $\quad p_l^{i+} \leftarrow$shortest path from s_i to d_l in G_l;
7 $\quad$ Calculate t_{i+}^l given the path p_l^{i+};
8 **end**
9 **for** $f_x \in I_l$ **do**
10 $\quad V_x \leftarrow \phi$;
11 $\quad$ **for** $s_i \in X_l$ **do**
12 $\quad\quad$ Calculate t_{i-}^x, t_{i+}^x;
13 $\quad\quad$ **if** $(t_{i+}^l < t_{i-}^x)$ *or* $(t_{i+}^x < t_{i-}^l)$ **then**
14 $\quad\quad\quad V_x \leftarrow V_x \cup s_i$;
15 $\quad$ **end**
16 $\quad R_l \leftarrow R_l - V_x$
17 **end**
18 Construct G_l such that $V(G_l) = R_l$;
19 **if** *path* p_l *exists from* o_l *to* d_l *in* G_l **then**
20 $\quad$ return p_l;
21 **else**
22 $\quad$ return False;
23 **end**

The sub-procedure to determine the flight path p_l for a flight f_l, given the set of interference flights, is presented in Procedure 3. The first step is to find the set of nodes where the flight paths interfere with the valid region R_l. Next, for all these interfering nodes, the earliest time to reach and the latest time to leave are calculated for each flight. A new subgraph G_l is then constructed with only the nodes that do not interfere temporally. If a path exists from the origin o_l to the destination d_l, then it is returned.

Computational Complexity: The computational complexity of the global routing algorithm is $\mathcal{O}(mn^2(m + \log n))$, where m is the number of flights, and n is the number of nodes (sectors) in the airspace. The computational complexity depends mainly on the size of the valid region R_l. Although the valid region can cover the entire airspace, suitably chosen deadlines, and flight speeds reduce the size of the valid region significantly. This directly influences the complexity contributed by lines 3-8 in Procedure 3. Lines 5 and 7 calculate the shortest path using Djikstra's shortest path algorithm.

Example 5: Using Example 4 in Section IV, let us illustrate Procedures 2 and 3. The origin of the three flights considered are at sectors s_{62}, s_{164}, and s_{22} respectively; and their destinations are at s_{75}, s_{161}, and s_{31} respectively. The min. and max. speeds of the flights were fixed as 1 and 2 respectively. The deadlines for the flights are 5.5, 3, and 4.5 time units. Fig. 5 illustrates the valid regions of each flight. It can be seen that flight f_2 from s_{164} to s_{161} has a valid region that does not interfere with flights f_1 and f_3. Therefore, the path planning for flight f_2 can be performed independently with no other information required about other flights. However, for flights f_1 and f_3, the valid regions intersect and their paths temporally intersect. Flight f_1 has a higher priority based on the flight indices, and the path for flight f_3 is curved around to temporally isolate from the flight f_1. For example, at s_6, both flight paths intersect. The time interval for flight f_1 at s_6 is $[2, 3]$ and for flight f_3 it is $[3, 3.5]$. These are non-overlapping time intervals. Flight f_1 leaves the sector s_6 and reaches s_5 before flight f_3 reached sector s_6.

VI. Performance Evaluation

In this section, we present the simulation environment and the workload generation procedures used to evaluate the proposed framework. The parameters for workload generation and the simulation settings are selected based on specifications of existing delivery drones. Next, we conduct two sets of experiments. The first experiment is to evaluate the impact of the proposed cost function against a baseline cost function. For the second experiment, we setup two solution frameworks representative of the existing solutions. We use multiple computation and routing metrics to compare our proposed framework against the existing frameworks.

A. Simulation Environment

As shown in Fig. 6, the simulator generates the airspace and the flight workload. The airspace generator generates a representative airspace model based on the required dimensions and the set of blocked regions. The airspace model is a graph structure implemented using `Networkx` [8]. The `Networkx` package performs most of the required graph operations, such as finding the shortest paths. For simulating the workload, the flight parameters are provided as input, including the minimum and maximum speed of the flights, the number of flights to be simulated, and the maximum allowable distance between the origin and destination sectors. The workload generator generates the required number of flights and adds them to a FIFO queue. (The parameters for workload generation are discussed in the next subsection.) The simulation module implements the global path planner with the airspace model and the flight queue as input, as discussed in Procedure 2. The generated flight paths are optionally visualized using a graphical interface. The simulator also keeps track of the simulation time of various parts of the simulation using Python's inbuilt `time.process_time`.

Size of the Airspace: The size of the hexagonal airspace chosen in the simulation is designed to have an apothem of 16 miles which translates to an area of 887 square miles.[1] By choosing the distance between adjacent sectors to be 0.16 miles, the airspace is represented by a set of 30,301 sectors.

Deadline and Speed Limits: According to the operating rules for drones provided by the FAA's part 107, the maximum speed of a drone cannot exceed 100 mph. The Amazon Prime Air [2] and the Airborg [1] are rated for a maximum horizontal speed of up to 50 mph and can fly up to 10 and 7 miles, respectively. Given that the distance between sectors is 0.16 miles and the

[1]In comparison, the largest city by area (Jacksonville, Florida) in the contiguous United States is 875 square miles.

flight range is up to 10 miles, a flight path in our model can be as long as 63 sectors. In the simulation, we assume a cruise speed between 10 and 20 mph. We use the parameter δ to control the distance of the destination from the origin while generating flight models to reflect real-world capabilities.

The simulator is executed on a Linux machine with 8-core 16-thread Intel Core i9-9900 K CPU @3.6 GHz and 32 GB system memory. Although the CPU is multi-core, the simulator is not designed to use multiple cores. Since the simulation is CPU-intensive, 8 parallel instances of the simulator are run to maximize the system resource utilization.

B. Workload Generation

A set of flights is generated following the flight model described in Subsection III-B. We introduce a derived parameter δ in addition to the flight model parameters. Although the flights are randomly generated, the origin and destination sectors satisfy the following condition.

$$r(o_l, d_l) \leq \delta \tag{18}$$

The parameter δ effectively limits the length of the flight paths. If no limits are in place, the origin and destination sectors of the flight could be at the opposite ends of the entire airspace and do not reflect a realistic scenario. The following parameters control the workload generation.

- Number of flights in the system $m \in \{5, 10, 20, 50, 100\}$.
- Distance threshold $\delta \in \{5, 10, 20, 50, 100\}$.
- Minimum flight speed is chosen as $v^{min} = 10$ mph.
- Maximum flight speed is chosen from $\{20, 30, 40, 50, 60\}$mph with 20 mph as the default.
- Deadline of the flights is assigned $\alpha \cdot r(o_l, d_l)$ where $\alpha \in [0.75, 1.25]$.

For weather, we have used the dynamic stochastic weather model proposed for air traffic management [36]. The parameters of the model are chosen to provide a random directional bias to the movement of the weather.

We perform two sets of experiments. The first set evaluates the proposed cost function and its effect on the distance separation during flight using the local path planner. This is evaluated under a varying number of flights and airspace size, and the aggregate results of 20 iterations under each setting are reported. The second set of experiments is performed for 100 iterations under each combination of the settings using the global path planner.

C. Experimental Results

First, we present the performance of the local planner proposed in Procedure 1, where we use a novel cost function including the weather information and distance metrics. Traditional path planning algorithms consider infinite cost for moving into an obstructed space – it is treated as a baseline in our work. We evaluate the distance to other flights in the airspace as well as to the weather-affected region.

Figs. 7(a) and 7(b) show the distribution of the distance between any pair of flights in the airspace. Figs. 8(a) and 8(b) show the distribution of the *closest* distance of any flight to a weather-affected sector at any point in its journey. The general trend is that the proposed approach maintains a safer distance while ensuring the distance between flights never reaches 0. However, the distance to weather cannot be guaranteed to be non-zero. This can be explained by the fact that the weather dynamics are known probabilistically at best, and can result in a situation where weather-affected sectors surround a flight. It is also observed that as the airspace gets larger, so does the average distance between the flights. This is due to the fact that the density of flights in the region decreases, and there is more room for path planning without interference.

Since the proposed methods are designed for unmanned aircraft and we aim to perform the routing locally on the aircraft, we choose the runtime of the algorithm as a performance metric. Additionally, we measure the ratio of flights successfully scheduled. In Fig. 9(a), using a boxplot, we exhibit the runtime with varying number of flights (m) and the distance threshold (δ). It can be observed that the overall trend remains the same with increased number of flights. However, as δ increases, the trend also gets amplified. Flights with larger δ value have a larger valid region and as a result have a higher chance of interfering with other flights. In the extreme case of $\delta = 100$, the flight path distance is equal to the airspace radius, resulting in interference with almost every other flight in the airspace. Increased interference diminishes the advantage of performing routing in a local region. However, it is also seen that over 90% of the workload is schedulable within 5 seconds. Here the runtime is the sum of computations for *all* the flights.

Using a box plot, Fig. 9(b), shows the success rate of routing the flights for varying number of flights (m) and distance limits (δ). Similar to Fig. 9(a), as the distance limit increases, the success rate begins to drop drastically, but the overall trend remains the same as the number of flights is increased. From the box plot, it is observed that the majority of the workload is routed except for the few outliers. From Procedure 2, the main reason for a routing failure is that there is no path where temporal overlap does not happen. This is critical in a two-dimensional map since there is no way to avoid a spatial and temporal overlap in flight paths. Such failed flights can be added back to the queue for re-routing, or the altitude levels can be considered by adding vertical dimensions to the graph to include the altitude levels. In either case, the demand for sectors due to these flights can be managed by air traffic controllers.

Figs. 10(a) and 10(b) show the variation of runtime and success ratio with the maximum speed. The minimum speed for a set of 20 flights is set a 10 mph, and the deadline for the flights is chosen as described in the workload generation procedure. The simulation runtime increases with increasing maximum speed. Such increase can be attributed to the widening speed range as the maximum speed limit increases, leading to increased interference while routing. Fig. 10(b), demonstrates that as the maximum speed limit is increased, the success ratio increases sharply. This trend can be attributed to the resulting larger valid region for each flight. A larger valid region allows for more possibilities to identify non-interfering flight paths.

Next, we compare our proposed framework against two commonly used frameworks for multi-flight routing.

1. Reinforcement Learning: Reinforcement learning-based frameworks have been proposed for routing by similar existing works [24], [37]. To compare against such frameworks, we chose the Q-learning algorithm [33] with a fixed state space. We did not choose its neural network-based extensions to avoid bias from the size of the neural network. A larger network size may have better performance, but at the cost of memory usage. Since we are evaluating the scalability of the algorithms, we chose an approach with minimal variation in memory consumption. The state-space of the Q-learning formulation is of size $n + 2 \cdot m$ which includes the sector capacity $x_i(k)$ of n sectors, current sector index and the destination sector index of m flights. The output of the Q-learning algorithm gives the index of the next sector for m flights. We trained the Q-learning algorithm until the convergence is under 0.1 for all settings. The algorithm was tested under dynamic weather conditions and online learning was enabled.

2. Integer Optimization: We use a mixed-integer programming formulation to represent the routing problem. The optimization problem is implemented using the `CVXPY` framework and the Gurobi solver is used to solve the problem. We also optimized the solver by trimming the variables and constraints to account of unreachable sectors and weather-affected sectors. As an example, for $n = 331$ sectors and $m = 10$ flights, the solver had 49650 variables and 97665 constraints after trimming. We limited the optimality gap of the solver to 1% and enforced a time limit of 10 minutes before declaring failure on a processor with 8 logical cores.

Finally, Fig. 11 compares our proposed framework against the Q-learning algorithm and a mathematical programming technique. As the number of flights increased, the runtime increased linearly, however, with an increased variance. Interestingly, as the number of sectors in the airspace increased in Fig. 11(b), the runtime increased dramatically with occasional outliers.

VII. Conclusion and Future Works

The air transportation system traditionally includes commercial passenger flights and military flights as its primary consumer. More recently, unmanned aerial vehicles (UAVs) and commercial space transportation are growing fields under the FAA (Federal Aviation Administration). A robust decision support system capable of scaling up with load is required to address future needs. This paper proposed a novel framework to route multiple flights with speed constraints in an airspace with sector capacities affected by weather. First, we adapt traditional path planning algorithms for a simplified problem. Next, with additional constraints in the routing problem, we propose a way to restrict the interference between flights. This allows for a scalable implementation where the required information for routing can be sent to the flight, and the computation can be done on board in small aerial vehicles such as drones. In this work, we established a method for scalable routing for a large number of unmanned flights, with some constraints such as a deadline for the flights and sector capacity for the airspace.

Our future work will aim to generalize the framework by including additional constraints such as wind speed, and optimizing flight paths by proactively adjusting the speed profiles for each flight depending on the workload in the local region. We will also evaluate the influence of utility. We plan to relax the airspace model to allow path overlaps with flights at different flight altitudes along with different vertical and horizontal velocities for the flight. These considerations will improve the fidelity and applicability of the airspace and flight models considered.

Acknowledgment

The authors are grateful to the Editor and anonymous reviewers for insightful comments and constructive suggestions that significantly helped improve the quality of the manuscript.

References

[1] *Media information - Airborg H8 10 K - top flight technologies*, [Online]. Available: www.tflighttech.com/products/media-airborg-h8-10k.html

[2] *Authorization for outdoor testing research and development for prime Air - FAA exemption under section 333*. [Online]. Available: www.faa.gov/uas/advanced_operations/section_333/authorizations_granted/media/Amazon_com_11290.pdf

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

[4] R. Dhal, S. Roy, C. P. Taylor, and C. R. Wanke, "Forecasting weather-impacted airport capacities for flow contingency management: Advanced methods and integration," in *Proc. Aviation Technol., Integration, Operations Conf.*, 2013, pp. 2013–4356. [Online]. Avilable: https://arc.aiaa.org/doi/10.2514/6.2013-4356

[5] G. L. Donohue, "A simplified air transportation capacity model," *J. Air Traffic Control*, Jun. 1999, pp. 8–15.

[6] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *J. Glob. Optim.*, vol. 56, no. 3, pp. 873–895, 2013.

[7] R. S. Félix Patrón and R. M. Botez, "Flight trajectory optimization through genetic algorithms coupling vertical and lateral profiles," *Proc. ASME 2014 Int. Mech. Eng. Congr. Expo. Volume 1, Adv. Aerospace Technol.*, Montreal, Quebec, Canada. Nov. 14–20, 2014. [Online]. Avilable: https://doi.org/10.1115/IMECE2014-36510

[8] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in G. Varoquaux, T. Vaught, and J. Millman, Eds., *Proc. 7th Python Sci. Conf.*, 2008, pp. 11–15.

[9] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Math. Operations Res.*, vol. 17, no. 1, pp. 36–42, 1992.

[10] M. T. R. Khan, M. Muhammad Saad, Y. Ru, J. Seo, and D. Kim, "Aspects of unmanned aerial vehicles path planning: Overview and applications," *Int. J. Commun. Syst.*, vol. 34, no. 10, 2021, Art. no. e4827.

[11] S.-H. Kim, G. E. G. Padilla, K.-J. Kim, and K.-H. Yu, "Flight path planning for a solar powered UAV in wind fields using direct collocation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1094–1105, Apr. 2020.

[12] C. Klüver, J. Klüver, and D. Zinkhan, "A. self-enforcing neural network as decision support system for air traffic control based on probabilistic weather forecasts," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2017, pp. 729–736.

[13] S. Koenig and M. Likhachev, "D* lite," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 476–483.

[14] T. Kravaris, C. Spatharis, K. Blekas, G. A. Vouros, and J. M. C. Garcia, "Multiagent reinforcement learning methods for resolving demand - capacity imbalances," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, 2018, pp. 1–10.

[15] T. N. Krishnamurti, L. Bounoua, and L. Bounoua, *An Introduction to Numerical Weather Prediction Techniques*. Boca Raton, FL, USA: CRC Press, May 2018, pp. 1–293.

[16] D. Kulkarni, "Models of sector flows under local, regional and airport weather constraints," in *Proc. 36th Digit. Avionics Syst. Conf.*, 2017, p. 1–7. [Online]. Avilable: https://ntrs.nasa.gov/api/citations/20170011240/downloads/20170011240.pdf

[17] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, 2012, pp. 1227–1232.

[18] W.-X. Lim and Z.-W. Zhong, "Re-planning of flight routes avoiding convective weather and the "three areas"," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 868–877, Mar. 2018.

[19] Q. Lin, H. Song, X. Gui, X. Wang, and S. Su, "A shortest path routing algorithm for unmanned aerial systems based on grid position," *J. Netw. Comput. Appl.*, vol. 103, pp. 215–224, Feb. 2018.

[20] A. Mardani, M. Chiaberge, and P. Giaccone, "Communication-aware UAV path planning," *IEEE Access*, vol. 7, pp. 52609–52621, 2019.

[21] J. Mańdziuk, "New shades of the vehicle routing problem: Emerging problem formulations and computational intelligence solution methods," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3 no. 3, pp. 230–244, Jun. 2019.

[22] Modernization of U.S. Airspace, Accessed on: Oct. 04, 2018. [Online]. Available: https://www.faa.gov/nextgen/

[23] R. F. Patron, A. Kessaci, and R. M. Botez, "Flight trajectories optimization under the influence of winds using genetic algorithms," in *Proc. AIAA Guid., Navigation, Control (GNC) Conf. Amer. Inst. Aeronaut. Astronaut.*, 2013. [Online]. Avilable: https://arc.aiaa.org/doi/pdf/10.2514/6.2013-4620

[24] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 146264–146272, 2019.

[25] Y. Qiming, Z. Jiandong, and S. Guoqing, "Modeling of UAV path planning based on IMM under POMDP framework," *J. Syst. Eng. Electron.*, vol. 30, no. 3, pp. 545–554, 2019.

[26] Z. Qin and X. Luo, " Research on dynamic route planning under adverse weather," in *Proc. 4th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, 2017, pp. 1153–1157.

[27] M. R. Manesh and N. Kaabouch, " Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ADS-B) system," *Int. J. Crit. Infrastructure Protection*, vol. 19, pp. 16–31, Dec. 2017.

[28] O. Rodionova, H. Arneson, B. Sridhar, and A. Evans, "Efficient trajectory options allocation for the collaborative trajectory options program," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, 2017, pp. 1–10.

[29] C. Schilke and P. Hecker, "Dynamic route optimization based on adverse weather data," in *Proc. 4th SESAR Innovation Days*, 2014. [Online]. Avilable: https://scholar.google.com/scholar?cluster=9070448371967290049&hl=en&as_sdt=40005&sciodt=0,10

[30] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, " LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–8.

[31] T. Stollenwerk *et al.*, " Quantum annealing applied to de-conflicting optimal trajectories for air traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 285–297, Jan. 2020.

[32] S. Vaidhun, Z. Guo, J. Bian, H. Xiong, and S. K. Das, "Priority-based multi-flight path planning with uncertain sector capacities," in *Proc. 12th Int. Conf. Adv. Comput. Intell. (ICACI)*, 2020, pp. 529–535.

[33] C. J. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 279–292, 1992.

[34] J. Welch, J. Cho, N. Underhill, and R. DeLaura, " Sector workload model for benefits analysis and convective weather capacity prediction," in *Proc. 10th USA/EUROPE Air Traffic Manage. R&D Seminar*, 2013, p. 10.

[35] J. D. Welch, "En route sector capacity model final report," MIT Lincoln Lab., Lexington, MA, USA, Project Rep. ATC-426, 2015.

[36] 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 *Proc. 91st Amer. Meteorological Soc. Annu. Meeting*, vol. 15, 2011. [Online]. Avilable: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=EDR7vRkAAAAJ&citation_for_view=EDR7vRkAAAAJ:Wp0gIr-vW9MC

[37] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments," *J. Intell. Robot. Syst.*, vol. 98, pp. 297–309, 2019.

[38] Y. Yoon, M. Hansen, and M. O. Ball, "Optimal route decision with a geometric ground-airborne hybrid model under weather uncertainty," *Procedia - Social Behav. Sci.*, vol. 17, pp. 551–571, Jan. 2011.

[39] W. Zeng and R. L. Church, " Finding shortest paths on real road networks: The case for A," *Int. J. Geographical Inf. Sci.*, vol. 23, no. 4, pp. 531–543, 2009.

[40] Y. Zhang, R. Su, Q. Li, C. G. Cassandras, and L. Xie, " Distributed flight routing and scheduling for air traffic flow management," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2681–2692, Oct. 2017.

[41] Y. Zhang, R. Su, G. G. N. Sandamali, Y. Zhang, C. G. Cassandras, and L. Xie, "A hierarchical heuristic approach for solving air traffic scheduling and routing problem with a novel air traffic model," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3421–3434, Sep. 2019.

[42] Z. W. Zhong, "Overview of recent developments in modelling and simulations for analyses of airspace structures and traffic flows," *Adv. Mech. Eng.*, vol. 10 no. 2, Feb. 2018, Art. no. 1687814017753911.