Lane-Based Large-Scale UAS Traffic Management

David Sacharny, Member, IEEE, Thomas C. Henderson, Life Fellow, IEEE, and Vista Marston

Abstract—The FAA and NASA are developing an Advanced Air Mobility (AAM) capability defining an Unmanned Aircraft Systems (UAS) Traffic Management (UTM) architecture. The combined scale and density of the expected air traffic, as well as the algorithmic complexity of maintaining safe separation, are driving a consensus that a structured airspace will eventually be required. Against this background, a lane-based airspace structure is proposed here whose motivation is to reduce the computational complexity of strategic deconfliction by providing UAS agents with a set of pre-defined airway corridors called lanes. To achieve complexity reduction, an airspace is defined that is composed of a directed graph where every node has either input or output degree equal to one, and flight plans consist of a scheduled sequence of lane traversals. The major results are: (1) the creation and layout of lane structures, (2) an efficient lane-based strategic deconfliction scheduling algorithm, (3) lanenetwork performance analysis tools, and (4) a tactical deconfliction protocol to handle dynamic contingencies (e.g., failure to follow the nominal flight plan). In conclusion, this approach provides efficient scheduling of safe flight paths, straightforward analysis of stream properties of the transportation system, an effective contingency handling protocol, and scalability to thousands of flights over urban areas.

Index Terms—UAS Traffic Management, Lane-Based, UTM, Virtual Highways, Advanced Air Mobility, AAM.

I. INTRODUCTION AND BACKGROUND

ARGE-SCALE UAS traffic management (thousands of flights) for package delivery, air taxis, etc. requires automatic scheduling and safety assurance because humans cannot deal with the complexity of such an operation. This leads to the Optimal UAS Scheduling Problem which is to efficiently provide flight trajectories for UAS as close to the desired times as possible and so no two flights violate minimum separation constraints. Of course, the optimal solution can only be found if all the requests are known ahead of time. The current iteration of the FAA-NASA UTM concept of operations suggests pairwise 4-dimensional conflict resolution [1], which involves high computational complexity for individual agents to resolve feasible trajectories. Additionally, this requires UAS operators (or proxies, such as UAS Service Suppliers) to share flight-operation volumes, and the system may be sensitive to deviations from nominal flight plans or congestion over popular areas.

We propose a lane-based approach (see Figure 1) and show that it has low computational cost for scheduling strategically deconflicted flights, masks operational-intent by minimizing the information required for safe separation, and is less sensitive to behavioral parameters with respect to aggregate system metrics such as delay. These properties are demonstrated in



Fig. 1. UTM airways over the Salt Lake City East Bench area. The left pane contains a 2D map with ground road network highlighted in red. The right pane shows the 3D airspace generated from these roads.

a head-to-head comparison with an implementation of the current FAA-NASA Strategic Deconfliction (FNSD) UTM approach to operations [1]. The two major disadvantages of the lane-based system are (1) UAS are restricted to a fixed set of lanes, and this may result in greater distance traveled, and (2) the UAS may be required to turn more to follow lanes rather than a smooth trajectory. On the other hand, lanes allow for efficient and effective real-time deconfliction to mitigate contingencies [2].

Currently, the most advanced method for coordinating dense heterogeneous airspace operations, under development in the first phase of NASA's Advanced Air Mobility (AAM) National Campaign, is distributed cell-based deconfliction [3], [4]. Under this method, proposed by designers of Advanced Air Mobility (AAM) and small-UAS UTMs, UAS operators and their proxies (Providers of Services for Urban Air Mobility (PSU) and UAS Service Suppliers (USS)) are responsible for contacting other operators in the area, requesting their operational volumes and then designing conflict free trajectories. There are benefits to this approach: the computation for deconfliction is distributed among the PSU and operators, and the resulting paths are optimal with respect to the individual vehicles. However, there are also some downsides that make it untenable for certain operational requirements. For example, in order for a PSU or operator to design a conflict-free trajectory prior to launch (termed Strategic Deconfliction), it must know the precise trajectories of other aircraft within a cell (predefined divisions of the airspace). This has both security and privacy implications since complete trajectory information can reveal the intent of operations. The concept of operational volumes lends a hand in mitigating this issue, however large volumes may be necessary to mask intent and this decreases the efficiency of the system with respect to airspace usage. A completely tactical method, whereby vehicles did not strate-

D. Sacharny is with the School of Computing, University of Utah, Salt Lake City, UT, 84112 USA e-mail: sacharny@cs.utah.edu.

T. Henderson and V. Marston are with the University of Utah.

Manuscript received February 19, 2021; revised December 16, 2021.

gically deconflict prior to launch, would resolve this issue at the expense of safety and the possibility of cascading conflicts [5]. Another issue with this approach is that the information from contingencies, such as mechanical or communication failures, does not follow a uniform trajectory among the agents in the system. This again can result in cascading effects since decisions made by individuals across the system will inevitably be made with limited system observability (imagine all vehicles in an area having to re-plan simultaneously).

See [6] for a survey of some recent proposals for UAS trajectory optimization, and [7], [8], [9], [10] for detailed methods; however, they fail to adequately address the problem. [6] reviews trajectory routing methods and concludes that "Most research on UAV routing does not consider collision avoidance and wind conditions." Moreover, they state that "Only problems with limited size have been solved; e.g., Mixed Integer Linear Programming is the most often proposed approach [8], [10], [11] where the most complex scenarios have from 2 to 6 UAS and a small number of waypoints and take on the order of hours to solve. [7] propose a lane-based method for automatic scheduling of ground vehicles, but use discrete cells, and use a heuristic algorithm because the greedy algorithm they describe is NP-complete. As a more specific comparison, [10] describes a 20 customer, 5 depot, 3 UAS problem which takes 21 minutes to solve, whereas we describe a 4,000-flight, 100-depot UAS experiment in this paper which required under 1 minute to strategically deconflict. Tsourdos et al. have also worked in the area of UTM development [12], [13], [14], culminating in their report on EuroDRONE, a testbed for UTM development and testing. The strategic deconfliction methods used there are similar to the FAA-NASA approach.

Structuring the airspace as corridors, coupled with the Lane-Based Approach to Strategic Deconfliction (LBSD) addresses the flaws in the cell-based FNSD approach ([15], [16], [17], [18], [19]) by enabling a cogent system analysis (all agents, trajectories, algorithms, etc.). The analysis in this paper extends the work in [19] to include static and dynamic spatial network measures that offer a way to compare unstructured to structured approaches.

LBSD eliminates the need for operators and PSU to obtain detailed trajectory information from other operations in the area. Additionally, the lane system directs the flow of contingency information along their flight paths, and agents within the system can have a reasonable expectation of the decisions that other agents will make due to the enhanced system observability. The main limitation of the proposed method is that it restricts the possible trajectories that are allowed in a given airspace, and a coordinated reservation database must be established (although distributed coordination platforms such as Apache Zookeeper are still applicable).

Airspace coordination is, in general, an instance of multirobot motion planning and can draw from a plethora of motion planning algorithms that exist today, many of which rely on some form of discretization (a partition of the configuration space). Popular methods include cell-decomposition and probabilistic sampling, for example Rapidly Exploring Random Trees (RRT) [20], [21]. The most general case of planning for multiple agents is PSPACE-hard; even the more narrow problem of tuning velocity profiles is NP-hard [22]. Problems specifically involving commercial aircraft often pose conflict resolution as an optimization problem (e.g., [23]), which again may be reduced to more general multi-robot motion planning. The linearization procedures that are often used to reduce the complexity are akin to the lanes and headway requirements posed by the lane-based system. If the spatial and temporal dimensions are analyzed separately (as in many decoupled approaches), with time becoming the main decision variable, then the insightful taxonomy of problems under the umbrella of *job-shop scheduling* becomes available.

For a UTM system, operations are scheduled online (i.e., on-demand) and desired release times are unknown to the scheduler until those requests are made. A globally optimal algorithm for efficiently coordinating airspace may not exist, meaning a more efficient use of the airspace could have been realized had the scheduler known all requests in advance. Airspace coordination (in the time domain) may be described as an online job-shop scheduling problem with no-wait constraints [24]. Specifically, this is an online-over-time problem because the scheduler "does not know at any point in time during the process how many more jobs are going to be released in the future and what their release dates are going to be" [24]. It is also classified as *clairvoyent* because all relevant information, such as speed, are available to the scheduler. Minimizing maximum lateness (a measure of the worst violation of due-dates) in this type of system, for a single machine with requested release dates (in Pinedo's nomenclature $1|r_i|L_{max}$), is NP-hard [24]. Any known polynomial-time online algorithm therefore represents an approximation of an optimal algorithm. Another way to describe this fact is to say that all airspace coordination proposals are competing with heuristics, and the lane-based approach presented here is no different.

A major problem for the current FNSD approach for discretizing the airspace is with regard to large-scale contingency mitigation; when a flight becomes non-nominal, it has the potential to disrupt many other flights, which in turn, disrupt even more flights. The flow of contingency information between agents in the unstructured airspace does not follow an explicit path, so predicting system states following these events is difficult. The unstructured airspace also requires careful 4D space monitoring of flights, and high bandwidth, low latency communications between controllers and UAS platforms.

The lane-based approach, however, provides a way to greatly reduce the complexity of both strategic deconfliction (from 4D to 1D) and contingency handling (see [16] where we introduced this approach). In a similar vein, the use of Victor and Jet Routes in commercial air traffic has a long-standing history. Airways are defined as follows [25]:

Airway routing occurs along pre-defined pathways called airways. Airways can be thought of as three-dimensional highways for aircraft. In most land areas of the world, aircraft are required to fly airways between the departure and destination airports. The rules governing airway routing, Standard Instrument Departures (SID) and Standard Terminal Arrival (STAR), are published flight procedures that cover altitude, airspeed, and requirements for entering and leaving the airway.

However, commercial airway lanes are managed by human air traffic controllers, and it is this function which must be automated if large-scale UAS operations (thousands per day in urban areas) are to be achieved. The automation task that lies ahead includes well-defined coordination problems, but a potentially more difficult task will be to automate the instances where human judgement is necessary. The FAA commands that human controllers exercise "common sense" and "best judgement" at least thirty times throughout the air traffic control procedures in scenarios ranging from safety alerts, contingencies (specifically minimum fuel), traffic advisories, and safe separation violations [26]. The UTM structure proposed here provides a foundation for characterizing these issues on the path to complete automation.

Previous work on lane related approaches has mostly been confined to manned aircraft. Devasia et al. [27] choose flight segments along established routes, aiming to decouple the problem of route optimization from safety considerations. For example, in commercial flights, airlines want to choose the best routes whereas the air traffic control system works to maintain safety locally. The authors propose a token-based system for entry into a designated area, and these tokens may be exchanged between airlines. Several aspects of their proposal are problematic, especially with respect to its application to UAS flight management: the "central idea is to hold all aircraft and let the aircraft with the lowest expected time of arrival [i.e., exit from the area] pass through," and this may involve an indefinite hold for some aircraft. Another problem is that as flights are merged, "uncertainty [in time to pass through] in travel time grows linearly with the number of mergers." Devasia et al. [28] again address manned aircraft and present a decentralized Air Traffic Control Method called "Conflict Resolution Procedure (CRP) based on highway-like routes" and give a way to choose flight segments during flight. However, this method does not deconflict segment endpoints (where routes merge), allows only two routes to be in conflict at a time, changes the route structure to resolve the conflict, and does not ensure fairness or liveness. Finally, Yoo and Devasia [29] extend CRP to consider turn rate limitations when routes are modified.

Structured airspaces have been categorized previously; in the layered [30] and "full mix" [31] airspace designs, UAS maintain the discretion to plan individually optimal paths, but must rely on tactical collision avoidance to maintain safe separation. Tactical collision avoidance may involve realtime constraints, therefore it is important to consider the computational complexity. Many heuristic methods have been developed, for example [32], however the success of these methods depends on the simplicity of the scenario and the number of conflicts may be overwhelming (e.g., cascading effects [5]). Conflicts in these airspaces are probabilistic (e.g., [5], [31], [33], [34], [35]), and many risk factors await operators (both human and machine). Lane-based airways were analyzed in [36], however the UAS operations were not deconflicted pre-flight and instead were simulated much like car-following models (e.g., [37]).

A relevant study was performed by Bulusu et al. [38] that considered the capacity of the low-altitude airspace over San Francisco, assuming only tactical deconfliction of small UAS. Their model mirrors our simulation of the FNSD approach with point-to-point flights generated according to a Poisson process and intensity proportional to population density. The loss of safe-separation at various minima was considered, with the most conservative (20m safe-separation) producing a maximum of 5000 flights per day over San Francisco. Even with a suitable strategic deconfliction algorithm (e.g., ground delay), the conflict areas and intensities in an unstructured airspace are inherently probabilistic and depend on the methods used by individual agents to deconflict (Bulusu assumed a simple altitude adjustment). Their conclusion was that higher densities of aircraft would require traffic control, and the lane-based approach presented here assumes this role. Furthermore, the structure and method provided by the lane-based approach does not preclude a tactical analysis as presented by Bulusu et al. because contingent scenarios can fall-back to these methods. To support this claim, we provide a simulation of a similar scenario over San Francisco with approximately 18,000 safely scheduled flights per day.

II. LANE-BASED URBAN AIRWAY SYSTEMS

The lane-based approach defines a set of one-way lanes where each lane is defined by an entry point, an exit point, and a one-dimensional curve between the two. UAS travel in three dimensions, and thus through lanes, requires 3D corridors (e.g., cylindrical-like tubes). The shape of corridors may change dynamically and should be constructed to account for the idiosyncrasies of the vehicles that they are meant to support, for example smaller aircraft in windy environments may require a larger corridor radius than a heavier vehicle with better control dynamics. Further design constraints can be defined in terms of the headway - or safe separation distance - between UAS. The combination of headway and corridor design can support a range of vehicle trajectory constraints, while the directed graph (digraph) imposed on the airspace presents agents with a structured environment for computation (the lanes represent a complete model of the airspace under ideal conditions). Lanes may also have other associated properties (e.g., speed restrictions) specified by the UTM, enabling regulators to communicate requirements effectively to all agents in the system.

Lanes are connected so that every vertex has either in-degree or out-degree equal to one. This permits scheduling to be based on lanes as opposed to vertexes since all flights may be deconflicted based on one incoming or outgoing lane, and simplifies the analysis of congestion because various graphbased measures can be utilized. This contrasts with zone-based deconfliction that presumes vehicles can enter and exit in any direction and the entire zone must be reserved (inefficient for large areas), and cell-based deconfliction that combines zone reservation with general motion planning within each cell (similar to the two-phase decoupled approach in [39]). The choice of the lane spatial layout is key to operational performance. Several alternatives exist:

- 1) airways modeled from ground road networks,
- 2) regular grid networks,
- 3) networks with specific properties (e.g., Delaunay networks, mono- or multi-altitude lane schemes, etc.).

Once the airway network is defined, a lane-based strategic deconfliction algorithm is required to schedule flights into the lanes so as to maintain the required minimum separation at all times during flight; this assumes that every flight follows its approved flight plan. The *Lane-Based Strategic Deconfliction* (LBSD) algorithm is given which allows computationally efficient scheduling. An analysis in the following sections demonstrate that the computational complexity of this algorithm is $\mathcal{O}(k^2)$, where k is the number of flights already scheduled in the proposed lane sequence.

Alternatively, in-flight planning arises due to contingencies, i.e., possible future events, usually causing problems or making further plans and arrangements necessary. Contingency handling may occur at different time-horizons and require different mechanisms, for example tactical (sensor-based) deconfliction. For these scenarios the *Closest Point of Approach* (CPA) algorithm is defined so that UAS can exploit the lane structure to continue their flights while avoiding collisions. This protocol may be based on either individual UAS sensor data or on local inter-UAS communication [2].

The behavior of requests and the strategy for scheduling can have a significant impact on the average density of lanes. Consider a single lane system of length x, with one entry and one exit. Further assume that vehicles consume a oneunit spatial interval within the lane, and requests arrive over time independently for a uniformly random unit interval. In the first scenario, assume that each vehicle either obtains the requested reservation or drops out, a "failure." This scenario mirrors a 1-dimensional sequential interval packing problem, also known as Renyi's parking problem [40]. Renyi showed that as the length of the lane approaches infinity, the mean filling density approaches 0.7476. This property also holds for the lane scheduling approach given here.

The layout of the lane system can also have significant effects on the behavior of the system. A common refrain among air mobility enthusiasts is that the ability to travel point-to-point in a straight line should be maintained and decreases the desirability of structured airspaces. However, a system of agents performing individually optimal trajectories in an unstructured airspace is unlikely to produce an efficient system. This is true in the case where agents can make decisions dynamically based on system-wide conditions, for example, Braess' paradox demonstrates where additional route options can result in an increase in travel time. This also appears to be true when considering conflict counts for a simple cell-based deconfliction experiment (point-topoint flights deconflicted using ground-delay with the FNSD approach). Experiments with 1000 UAS flying point-to-point in an unstructured airspace with uniformly distributed land and launch sites show an increased density of conflicts in the center of the area of travel.

This configuration of trajectories correlates to the structured regular-grid lane network, which exhibits the worst performance in the network comparison experiments described below. A simulation comparison between the point-to-point unstructured airspace and the lane-based approach is demonstrated in Section VI.

III. LANE CREATION

The lane creation process starts by designating a geographic area for UAS operations. Next, a ground network is specified as an undirected graph, G = (V, E), where V is a set of ground position vertexes, and E is a set of undirected edges between the vertexes, which provides the basis for the air lane network. Currently, the ground network may be defined (1) from an existing ground road network, (2) as a regular grid covering the desired area, or (3) as a Delaunay triangulation over the region.

In case (1), road intersections are the nodes of the graph, and road segments between intersections are the edges. For case (2) a spacing is determined, and the grid is produced with evenly spaced nodes. Other lattice configurations are possible, but smaller angles between connected lanes should be avoided as it impacts the safe separation distance. Finally, for case (3) the nodes are randomly placed from a uniform distribution with two user-specified parameters: the number of nodes and the minimum distance between nodes.

Figure 2 shows example networks for the 3 cases for a small set of roads from San Francisco, CA. Next, this undirected ground network is transformed into a 3-dimensional directed graph which specifies the lane airways (see Figure 3).



Fig. 2. Three types of road layouts over the same locale: actual San Francisco roads (left); grid layout (middle); delaunay trianglation (right).

It is also necessary to identify ground vertexes which will be launch or land sites. Since the proposed air lanes are restricted to one-way travel (i.e., airways are digraphs), two-way traffic between vertexes can be achieved by having air lanes next to each other at the same altitude (as for roads), above and below each other, or ensure vertex reachability some other way. Here we demonstrate a two-level scheme.

To create the two-level airways between vertexes, the ground network is duplicated as a set of airway lanes at two altitudes: one for travel in direction $[0, \pi)$, and the other in direction $[\pi, 2\pi)$. Since ground vertexes are road intersections,

each is represented by two roundabouts in the air centered over the vertex; there are up and down lanes between all vertically separated roundabouts above a ground vertex. A larger scale



Fig. 3. An example two-level grid lane layout of San Francisco roads.

example of the Salt Lake East Bench area provides a second example and is shown in Figure 1.

IV. LANE-BASED STRATEGIC DECONFLICTION

To schedule a flight, launch and land sites are selected, as well as a sequence of lanes going from one to the other, along with a desired speed, and a launch time window. The set of lanes may be selected however desired; for example, to minimize distance or weather constraints, or other relevant factors. The launch time window gives the earliest and latest possible launch times (line 1 of LBSD algorithm). Lanes are scheduled individually by flights, and every new flight must respect the headway distance not only in each lane, but also when moving from one lane to another (i.e., with respect to all merging or diverging lanes).

The analysis of flight interactions where all UAS speeds are constant is done with the *Space-Time Lane Diagram* (STLD); there is an STLD for each lane in the UTM system. Figure 4 shows the information representation. Time is represented



Fig. 4. Space-Time Lane Diagram. (a) Trajectories through the Lane of the Proposed Flight Earliest and Latest Possible Entry Times. (b) Trajectory of Already Scheduled Flight, and its Associated Headway Constraint Trajectories.

on the x-axis, and distance along the lane on the y-axis. The left hand side of the figure shows the trajectory of a new flight through the lane if it were to launch at the earliest time (segment $\overline{q_1q_4}$) or at the latest time (segment

 $\overline{q_2q_3}$). The quadrilateral, $q_1q_2q_3q_4$, defines all possible lane traversals for the new flight. The right hand side of the figure shows the representation of an existing flight. The scheduled flight (segment $\overline{t_{i,1}t_{i,2}}$), is sandwiched between two headway trajectories ($\overline{p_1p_4}$ and $\overline{p_2p_3}$) showing the required standoff distance (in y) and time (in x). Any entry time that results in a trajectory that does not cross the $p_1p_2p_3p_4$ quadrilateral is called a *safe entry time* and is produced in line 10 of the LBSD Algorithm (see below).

Algorithm LBSD (Lane Based Strategic Deconfliction) On input:

lanes: lane sequence for requested flight $[q_1, q_2]$: requested launch interval n: number of lanes flights: flights per lane h_t : maximum required headway time On output: Safe time intervals to launch begin

1 possible_intervals $\leftarrow [q_1, q_2]$

2 for each lane $c \in$ lanes

- 3 time_offset \leftarrow time to get to lane c
- 4 possible_intervals ← possible_intervals + time_offset
- 5 for each flight, f, in lane c
- 6 new_intervals $\leftarrow \emptyset$
 - for each interval, i, in possible_intervals
 - $[t_1, t_2] \leftarrow \text{interval } i$
 - $label \leftarrow get_label(p_{f,1}^c, p_{f,2}^c, s_f^c, t_1, t_2, s^r, h_t)$
- 10 f_int \leftarrow get_interval(label, $p_{f,1}^c, p_{f,2}^c, s_f^c, t_1, t_2, s^r, h_t)$
- 11 new_intervals \leftarrow merge(new_intervals,f_int)
- 12 *end*

7

8

9

- 13 end
- 14 possible_intervals \leftarrow new_intervals
- 15 end

16 possible_intervals \leftarrow possible_intervals - time to last lane

In Algorithm LBSD $p_{f,1}^c$ and $p_{f,2}^c$ are the left and right headway start times of the scheduled flight, s^c is the scheduled flight's speed, and s^r is the requested flight's speed. The correctness of this algorithm follows from the discrete number of interactions of flight trajectories and the interval selection is based directly on that; for a detailed discussion, see [15].

To understand the computational complexity of this algorithm, consider the maximum possible number of intervals remaining after each lane is considered. If there are f_k scheduled flights in lane k, then the maximum number of intervals resulting from analysis of lane 1, is $f_1 + 1$ where each existing flight creates a separate sub-interval in the proposed entry time interval, resulting in $f_1 + 1$ sub-intervals. At the next step, the maximum number of intervals is when each of the f_2 flights in Lane 2 creates one new sub-interval, and the max number of total sub-intervals is $f_1 + f_2 + 1$. Thus, the number of pairwise comparisons for each lane k is $f_k(f_1 + f_2 + ... + f_{k-1} + 1)$ making the total number of comparisons, n_c

$$n_c = f_1 + (f_2 f_1 + f_2) + (f_3 f_2 + f_3 f_1 + f_3) \dots$$



Fig. 5. Lane diagram for a single lane, showing six flight reservations, planned trajectories, and simulated telemetry.

$$n_c = \sum_{k=1}^n f_k + \sum_{i \neq j} f_i f_j$$

Since on average, $f_k = \frac{f}{n}$, where f is the number of flights in the lane sequence flight path of the proposed flight and n is the number of lanes, then the worst-case complexity is dominated by the second term, and we have:

$$n_c \propto \binom{n}{2} \frac{f^2}{n^2} \propto f^2$$

Therefore, the complexity is $O(f^2)$. Note that since the number of lanes in a sequence is bounded by the graph diameter, we consider it a constant).

A major requirement of this algorithm is that a complete database of flight reservations must be maintained and used by the algorithm; however, this will generally be required by the flight authorities anyway to allow informed monitoring of airspace usage. The original idea of the FAA was to allow a decentralized approach where each USS maintained its own flight info and shared as necessary; the drawback of this is that if any USS fails, the system fails, and there is the possibility of semantic mismatch in terms of trajectory definition (e.g., meters vs. feet).

The Space Time Lane Diagram (STLD) also provides a straightforward way to visualize the traffic through a lane for monitoring UTM operations. Figure 5 shows a set of planned flights through a lane, where reservations represent a reduced-order model (speed and headway) for the actual or planned trajectory. and their trajectories reflect the accelerations necessary to turn between lanes. Lanes also allow real-time comparison of the UAS' planned flight path and the actual trajectories (e.g., provided by telemetry data).

V. SPATIAL NETWORK AND FLOW MEASURES

Static spatial network measures have been defined to evaluate the quality of a given (ground) transportation network (see [41], [42], [18], [43] for a detailed set of measures), and a set of flow measures (see [18]) as well. A subset of these have been selected to analyze the various road networks used as the basis for airways. For a given graph, G, the particularly useful measures include:

• *Detour Index*: pairwise ratio of straight line distance over length of shortest path. (closer to 1 is straighter)



Fig. 6. Betweenness centrality for a GIS road-based airspace.



Fig. 7. Betweenness centrality for a grid airspace.

- Betweenness Centrality: $bc(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of shortest paths from s to t through node v.
- Closeness Centrality: $C_i^C = \frac{N-1}{\sum_{j=1, i \neq j}^N d_{ij}}$, where *i* is the vertex index, and d_{ij} is the shortest path distance between vertexes *i* and *j*, and *N* is the number of nodes in the graph; measures how close a node is to the other nodes in the network.
- Straightness Centrality (also called accessibility): $C_i^S = \frac{1}{(N-1)} \sum_{j=1, i \neq j}^{N} \frac{d_{ij}^{Eucl}}{d_{ij}}$, where d_{ij}^{Eucl} is the Euclidean distance of *i* to *j*; captures how straight the shortest paths through a node are.

These measures provide clear insight into how the graph affects performance; the Detour Index can help a user select a path, and the last three provide useful information about congestion and flow through the graph. A high measure of betweenness centrality (BC) indicates that a node is prone to congestion since many shortest paths pass through it; high closeness centrality reveals a good site for a launch or land site since the node is close to many nodes; finally, straightness centrality means that shortest paths through this node do not require many turns which can be important for UAS platforms. For example, Figures 6, 7 and 8 show the BC measure for three lane networks and high BC measure corresponds to higher congestion parts of the network.



Fig. 8. Betweenness centrality for a delaunay airspace.

The network measures described above are useful for developing efficient UAS lane networks. However, in order to compare the lane-based and FAA approaches, the following measures provide a strong basis for analysis:

- *Delay Time*: absolute difference between desired and actual launch times
- *Deconfliction Time*: wall-clock time required for deconfliction
- *Failures*: number of flights that could not be scheduled due to conflicts.

The experimental results comparing the LBSD and FAA methods are provided in terms of these measures.

VI. EXPERIMENTS TO DETERMINE PARAMETER IMPACT ON SCHEDULING ALGORITHMS

In a complicated system like a UTM, analytic solutions may not exist, and therefore, simulations are used to explore UTM performance with respect to parameters of interest. The experiments performed here are designed to allow both inter-UTM (e.g., LBSD vs. FNSD) and intra-UTM (e.g., grid vs. Delaunay) structural analysis, as well as a cursory system/behavioral analysis (relating the agents flexibility in scheduling to the overall system performance). The parameters studied here include:

- 1) *Launch Frequency* (flights per hour): comparable to an arrival rate of flights into the system [values: 100 and 1000]
- 2) UAS Speed (m/s): Average UAS speed through lane [values: 5, 10, 15]
- 3) *Headway Distance* (m): Minimum distance allowed between UAS [vaues: 5, 10, 30]
- Flex Time (sec): Interval of possible launch times for flight [values: 0, 300, 1800].

The simulation covers an area of 5 square km (roughly the size of the Salt Lake Valley) with the FAA cells spaced as a 10x10 cell structure. The LBSD grid was chosen to correspond to this as an 11x11 node grid. The 121 launch (land) sites are located near the ground node points in both layouts. The Delaunay networks are generated with the same number of nodes, but they are distributed randomly (sampled from



Fig. 9. Simulation results: averages for launch frequency of 100 flights per hour. The upper row describes the parameter combination enumeration in the lower three rows which give the mean number of failed fights, mean delay, and mean deconfliction for those combinations of parameters.

uniform distribution) in the given area. Road-based networks include an area over San Francisco and an area over Salt Lake City. Ten simulation trials were run for each of the 54 parameter combinations (note that for the Delaunay networks an additional ten trials were run for each due to the random nature of the node locations). The simulation period was set to 4 hours simulated time. The FAA flights are up, over and down trajectories scheduled between randomly selected launch and land sites; the flight altitude was randomly assigned between the min and max altitudes of the LBSD network. For both UTM methods, given the flight frequency, a random set of desired flight times are generated which are uniformly spread across the total simulation time.

Figures 9 and 10 show the mean statistics for launch frequency of 100 flights/hour and 1000 flights/hour, respectively. The means of the maxima over all trials are also given in Figures 11 and 12. The statistics include: delay (calculated as the time between the requested launch time and the assigned launch time), failed flights (flights that could not be accommodated due to time or space constraints), and deconfliction time (the amount of wall-clock time that the computer required to schedule a flight).

This data indicates that all six categories of structures have response characteristics that are most undesirable when the flex is low, the speed is low, and the headway is high. However, the unstructured FAA airspace and the road-based



Fig. 10. Simulation results: averages for launch frequency of 1000 flights per hour.

San Francisco networks are particularly sensitive to these inputs with respect to the mean statistics. The max statistics in regard to delay show a somewhat different story where the FAA structure responded similarly to the others and the San Francisco graph performed the worst. These results indicate that small changes in the policies and behaviors may have dramatic effects on what the average UAS agent experiences accessing the unstructured (FNSD) airspace and complex road networks. Conversely, all the structured airspaces had relatively subdued effects related to these inputs (note that Salt Lake City has a grid-like road system).

VII. CONTINGENCY HANDLING

A contingency occurs when a UAS does not follow its nominal flight path. This may happen due to UAS platform issues (power, control, etc.), or external factors (e.g., weather, other platforms, lane closures, or rogue flight interference). The UTM itself may provide mechanisms to handle contingencies, e.g.: re-planning flight paths, emergency lanes in the air alongside regular lanes, emergency landing lanes, etc. These may exist as part of the static structure of the UTM, or may be created dynamically as the need arises.

Alternatively, it may be more effective to allow the UAS to perform tactical deconfliction by exploiting the lane structure. This can be achieved by having the UAS modify their speeds as they proceed through the prescribed lane sequence [2]. To determine if speed modification is necessary, a flight checks



Fig. 11. Simulation results: maxes for launch frequency of 1000 flights per hour.



Fig. 12. Simulation results: maxes for launch frequency of 100 flights per hour.

the flights in all lanes that share an endpoint with its lane. If at no point along its current lane is is with headway distance of another flight, then it is tactically deconflicted. The Closest Point of Approach (CPA) algorithm can be used to figure this out. Let \mathcal{L}_1 and \mathcal{L}_2 be two lanes that consist of vectors \bar{S}_1 and \bar{S}_2 , where $\bar{S}_1 \equiv \overline{\bar{P}_1 P_2}$ and $\bar{S}_2 \equiv \overline{\bar{Q}_1 Q_2}$, where \bar{P}_1 is the entry point to lane 1, \bar{P}_2 is the exit point of lane 1, \bar{Q}_1 and \bar{Q}_2 are the entry and exit points to lane 2, respectively. Flight f_1 has trajectory $\bar{P}(t) = \bar{P}_1 + t\bar{v}$ in \mathcal{L}_1 , and likewise flight f_2 in lane \mathcal{L}_2 has trajectory $\bar{Q}(t) = \bar{Q}_1 + t\bar{w}$ where the velocities are \bar{v} and \bar{w} , respectively, and t is time in the lane. Since the velocities are $\bar{v} = \frac{s_1(\bar{P}_2 - \bar{P}_1)}{|\bar{P}_2 - \bar{P}_1|}$ and $\bar{w} = \frac{s_2(\bar{Q}_2 - \bar{Q}_1)}{|\bar{Q}_2 - \bar{Q}_1|}$, where s_1 and s_2 are the speeds of f_1 and f_2 , respectively, then the time when the two flights are closest in their trajectories is:

$$t_{min} = \frac{-(P_1 - Q_1) \cdot (\bar{v} - \bar{w})}{|\bar{v} - \bar{w}|^2}$$

If t_{min} is found for $t \in [t_{current}, t_{min_TOA}]$, where t_{min_TOA} is the minimum time of arrival at the end of the lane for flights f_1 and f_2 , then the minimum distance, d_{min} , between the flights across these intervals is just $|\bar{P}(t_{min}) - \bar{Q}(t_{min})|$. The two flights have a conflict if $d_{min} < d_S$. CPA is illustrated in Figure 13.



Fig. 13. CPA Algorithm: two flights at closest points $P_{t_{min}}$ and $Q_{t_{min}}$.

Deconfliction of two flights, f_1 and flight f_2 , can be achieved by:

Deconflict_Pair

while conflict(f_1, f_2) reduce speed, s_1 , of f_1 if $s_1 < s_{min}$ then flight f_1 fails

Based on these, we give the Closest Point of Approach Deconfliction (CPAD) algorithm:

Algorithm 1: Closest Point of Approach			
1	$1 \forall$ active flight f		
1	v active night, j		
2	if f enters a new lane		
3	OR a neighboring flight has slowed		
4	OR f has reduced speed on its own		
5	then call Deconflict_Pair for all flights in neighboring		
	lanes		
6	if f has reduced speed		
7	then f broadcasts this information.		

If each UAS runs the CPAD algorithm, then local deconfliction is guaranteed, and global tactical deconfliction is a result of flights either deconflicting, or leaving the airspace (i.e., resulting in failures). Strategic deconfliction is not guaranteed by CPAD, however, it does assure that at no time are two flights closer than headway distance. CPAD is robust with respect to contingencies and does not require sharing of detailed flight info, there is no centralized flight planning. CPAD may force some flights to fail; however, lane structure can help mitigate this, as well as other UTM policies like limiting the number of flights, or allowing routes to be changed dynamically. CPAD can operate either on the basis of communicated telemetry data between UAS or on-board sensors can be used to determine the location and speed of nearby flights. Over a set of simulation experiments, there was only one flight failure using CPAD.

TABLE I Acronym Table

Acronym	Definition
AAM	Advanced Air Mobility
CRP	Conflict Resolution Procedure
FAA	Federal Aviation Administration
FNSD	FAA-NASA Strategic Deconfliction
LBSD	Lane-based Strategic Deconfliction
NASA	National Aeronautics and Space Administration
PSU	Providers of Services for Urban Air Mobility
RRT	Rapidly exploring Random Trees
SID	Standard Instrument Departure
STAR	Standard Terminal Arrival
STLD	Space-Time Lane Diagram
UAS	Unmanned Aircraft System
USS	UAS Service Supplier
UTM	UAS Traffic Management

VIII. CONCLUSIONS AND FUTURE WORK

Lane-Based Strategic Deconfliction has been given as a solution to the UAS Scheduling Problem. The approach is based on a pre-defined set of lanes which reduces the deconfliction complexity from an NP-hard 4-D problem (x,y,z,t) to a 1-D problem (delay time). The method will produce a solution if there is one, and ensures safe vehicle separation while in flight. The algorithm is $O(n^2)$ where *n* is bounded by the number of flights in the flight path set of lanes. The Space-Time Lane Diagram provides an insightful visualization of the state of the system which facilitates detecting rogue UAS or those which are off course. Finally, the experimental results show that LBSD outperforms FNSD over a variety of structural networks.

Future areas of research include: (1) dynamic lanes, (2) anomaly detection, and (3) multi-modal, heterogeneous fleet coordination. The virtual nature of lanes means that designers have the additional control ability to add, subtract, and transform lanes to adapt to changing conditions. The input variables to these methods could range from weather to congestion to technology advances.

Since lanes represent an ideal model of the airspace, encoding many of the properties of the system from possible trajectories to occupancy and communication, the lane-based approach offers a straightforward way to detect anomalous flights. Additionally, recent experiments have shown the potential of classifying anomalous behaviors, for example malfunctions or malicious activities.

Additional possibilities leveraging the lane-based approach include coordination between ground, sea, or space based assets and autonomous aircraft. Coupled with dynamic lanes, a range of scenarios may be constructed, for example: planning missile trajectories, communication bandwidth sharing, satellite reservations, cellular coverage, and moving-vehicle launch and landing.

REFERENCES

- Federal Aviation Administration, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations, V2.0," Federal Aviation Administration, Tech. Rep., 2020.
- [2] D. Sacharny, T. Henderson, and E. Guo, "A DDDAS Protocol for Real-Time UAS Flight Coordination," in *InfoSymbiotics/Dynamic Data Driven Applications Systems Conference*, Boston, MA, October 2020.

- [3] AirMap, "Five Critical Enablers or Safe, Efficient, and Viable UAS Traffic Management (UTM)," AirMap, Santa Monica, CA, Tech. Rep., January 2018.
- [4] J. Rios, I. Smith, P. Venkatesan, D. Smith, V. Baskaran, S. Jurack, S. Iyer, and P. Verma, "UTM UAS Service Supplier Development, Sprint 2 Toward Technical Capability Level 4," NASA, Moffet Field, CA, Tech. Rep. NASA/TM-2018-220050, December 2018.
- [5] M. R. Jardin, "Analytical Relationships Between Conflict Counts and Air-Traffic Density," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1150–1156, 2005.
- [6] W. Coutinho, M. Battara, and J. Fliege, "The Unmanned Aerial Vehicle Routing and Trajectory Optimisation Problem, A Taxonomic Review," *Computers and Industrial Engineering*, vol. 120, pp. 116–128, 2018.
- [7] A. Giridhar and P. Kumar, "Scheduling Automated Traffic on a Network of Roads," *IEEE-T Vehicular Technology*, vol. 55, no. 5, pp. 1467–1474, September 2006.
- [8] A. Fluegenschuh and D. Muellenstedt, "Flight Planning for Unmanned Aerial Vehicles," in 4th IMA Conference on Mathematics in Defence, Harwell Oxford, UK, November 2015.
- [9] C. Murray and R. Raj, "The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones," *Trnsportation Research Part C Emerging Technologies*, January 2020.
- [10] M. Torabbeigi, G. Lim, and S. Kim, "Drone Delivery Scheduling Optimization Considering Payload-Induced Battery Consumption Rates," *Jnl* of Intelligent Robotics and Systems, vol. 97, pp. 471–487, 2020.
- [11] A. Weinstein and C. Schumacher, "UAV Scheduling via the Vehicle Routing Problem with Time Windows," Air Force Research Lab, Wright-Patterson AFB, OH, Tech. Rep. AFRL-VA-WP-TP-2007-306, January 2007.
- [12] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, and S. Wang, "Solving Trajectory Optimization Problems in the Presence of Probabilistic Constraints," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4332–4345, 2020.
- [13] A. Alharbi, A. Poujade, K. Malandrakis, I. Petrunin, D. Panagiotakopoulos, and A. Tsourdos, "Rule-Based Conflict Management for Unmanned Traffic Management Scenarios," in *AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, San Antonio, TX USA, October 2020.
- [14] V. Lappas, G. Zoumponos, V. Kostopoulos, H. Shin, A. Tsourdos, M. Tantarini, D. Shmoko, J. Munoz, N. Amoratis, A. Maragkakis, T. Machairas, and A. Trifas, "EuroDRONE, A European Unmanned Traffic Management Testbed for U-Space," *Drones*, vol. 6, no. 53, 2022.
- [15] T. C. Henderson, D. Sacharny, and M. Cline, "An Efficient Strategic Deconfliction Algorithm for Lane-Based Large-Scale UAV Flight Planning," University of Utah, Tech. Rep. UUCS-19-005, September 2019.
- [16] D. Sacharny and T. Henderson, "A Lane-based Approach for Largescale Strategic Conflict Management for UAS Service Suppliers," in *IEEE International Conference on Unmanned Aerial Systems*, Atlanta, GA, June 2019.
- [17] —, "Optimal Policies in Complex Large-Scale UAS Traffic Management," in *IEEE International Conference on Industrial Cyber-Physical Systems*, Taipei, Taiwan, May 2019.
- [18] D. Sacharny, T. Henderson, M. Cline, B. Russon, and E. Guo, "FAA-NASA vs. Lane-Based Strategic Deconfliction," in *IEEE Multisensor Fusion and Integration Conference*, Karlsruhe, Germany, September 2020.
- [19] D. Sacharny, T. Henderson, and M. Cline, "Large-Scale UAS Traffic Management (UTM) Structure," in *IEEE Multisensor Fusion and Integration Conference*, Karlsruhe, Germany, September 2020.
- [20] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion Theory, Algorithms, and Implementation*. Cambridge, Massachusetts: MIT Press, 2005.
- [21] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.
- [22] D. Alejo, J. M. Díaz-Báñez, J. A. Cobano, P. Pérez-Lantero, and A. Ollero, "The Velocity Assignment Problem for Conflict Resolution with Multiple Aerial Vehicles Sharing Airspace," *Journal of Intelligent* and Robotic Systems: Theory and Applications, vol. 69, no. 1-4, pp. 331–346, 1 2013.
- [23] M. Soler, M. Kamgarpour, J. Lloret, and J. Lygeros, "A Hybrid Optimal Control Approach to Fuel-Efficient Aircraft Conflict Avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1826–1838, 2016.
- [24] M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems, 5th ed. New York, NY, USA: Springer International Publishing, 2016.
- [25] J. Barbagello, "Instrument Procedures Handbook," Federal Aviation Administration, Washington, DC, Tech. Rep. FAA-H-8083-16B, August 2017.

- [26] Federal Aviation Administration, "Order JO 7110.65Z, Air Traffic Organization Policy," 2021.
- [27] S. Devasia, M. Heymann, and G. Meyer, "Automation Procedures for Air Traffic Management: A Token-based Approach," in *Proceedings of* the American Control Conference, Anchorage, AK, May 2002.
- [28] S. Devasia, D. Iamratanakul, G. Chatterji, and G. Meyer, "Decoupled Conflict-Resolution Procedures for Decentralized Air Traffic Control," *IEEE-T on Intelligent Transportation Systems*, vol. 12, no. 2, June 2011.
- [29] J. Yoo and S. Devasia, "Provably Safe Conflict Resolution with Bounded Turn Rate for Air Traffic Control," *IEEE-T on Control Systems Tech*nology, vol. 21, no. 6, November 2013.
- [30] L. Sedov and V. Polishchuk, "Centralized and Distributed UTM in Layered Airspace," in 8th International Conference on Research in Air Transportation, 2018, pp. 1–8.
- [31] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern, "Metropolis : Relating Airspace Structure and Capacity for Extreme Traffic Densities," in 11th USA/Europe Air Traffic Management Research and Development Seminar, Lisbon, Portugal, 2015.
- [32] S. Balachandran, C. Munoz, and M. C. Consiglio, "Implicitly Coordinated Detect and Avoid Capability for Safe Autonomous Operation of Small UAS," in *17th AIAA Aviation Technology, Integration, and Operations Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 6 2017.
- [33] M. Tra, E. Sunil, J. Ellerbroek, and J. Hoekstra, "Modeling the Intrinsic Safety of Unstructured and Layered Airspace Designs," in 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, Washington, USA, 2017.
- [34] V. Bulusu, R. Sengupta, V. Polishchuk, and L. Sedov, "Cooperative and Non-Cooperative UAS Traffic Volumes," in 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, Miami, FL, USA, 2017.
- [35] H. Blom and G. Bakker, "Conflict Probability and Incrossing Probability in Air Traffic Management," in *Proceedings of the 41st IEEE Conference* on Decision and Control, 2002, Las Vegas, NV, USA, 2002, pp. 2421– 2426.
- [36] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan, "Concepts of Airspace Structures and System Analysis for UAS Traffic flows for Urban Areas," in *AIAA Information Systems-AIAA Infotech @ Aerospace.* Grapevine, Texas: American Institute of Aeronautics and Astronautics, 1 2017.
- [37] G. Newell, "A Simplified Car-Following Theory: A Lower Order Model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195–205, 3 2002.
- [38] V. Bulusu, V. Polishchuk, R. Sengupta, and L. Sedov, "Capacity Estimation for Low Altitude Airspace," in *17th AIAA Aviation Technology*, *Integration, and Operations Conference*, Denver, CO, USA, June 2017.
- [39] M. Saha and P. Isto, "Multi-Robot Motion Planning by Incremental Coordination," in *IEEE International Conference on Intelligent Robots* and Systems, 2006, pp. 5960–5963.
- [40] A. Renyi, "On a One-Dimensional Problem Concerning Random Space Filling," Publications of the Mathematical Institute of the Hungarian Academy of Sciences, vol. 3, pp. 109–127, 1958.
- [41] M. Barthelemy, Morphogenesis of Spatial Networks. Berlin, Germany: Springer Verlag, 2018.
- [42] M. Patrikson, The Traffic Assignment Problem: Models and Methods. Garden City, NY: Dover Publications, 2015.
- [43] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Germany: Springer Verlag, 2013.



David Sacharny David Sacharny received a BS in Electrical Engineering from California Polytenchnic State University, San Luis Obispo in 2010. He is a PhD candidate in computing and robotics at the University of Utah, and a managing member of GeoRq LLC. His research interests concern geospatial intelligence, large-scale UTM, and AI. His research spinoff *GeoRq* was one of a few challengers in the Advanced Air Mobility National Campaign Developmental Test (NC-1) airspace simulations.



Thomas C. Henderson Thomas C. Henderson received his BS in Math with Honors from Louisiana State University in 1973 and his PhD in Computer Science from the University of Texas at Austin in 1979. He is currently a full Professor in the School of Computing at the University of Utah. His research interests include autonomous cognitive systems, robotics and computer vision, and his ultimate goal is to help realize functional androids. enjoys good dinners with friends, reading, playing basketball and hiking.



Vista V. Marston Vista V. Marston received a BS in Anthropology from the University of Utah in 2019. She is currently working on a BS degree in Computer Science at the University of Utah. Her interests are in artificial intelligence, robotics, and computer vision. When she isn't working on laboratory projects, she is enjoying the outdoors whether that is mountain biking, hiking, backpacking, or skiing.