An Efficient Strategic Deconfliction Algorithm for Large-Scale UAS Traffic Management

David Sacharny, Thomas C. Henderson and Michael Cline
University of Utah

UUCS-20-010

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

30 June 2020

Abstract

We propose a lane-based airway navigation framework wherein each lane is one-way, and intersections are handled by means of polygonal lane roundabouts; it is possible to assign flight plans so that the set of all such plans is strategically deconflicted. That is, no two Unmanned Aerial Systems (UAS) will ever get closer in a lane than the minimum allowed headway time (or distance) of each other. We describe here a method to determine all allowable launch times (i.e., strategically deconflicted) given a requested launch time interval and a set of scheduled flights. Scheduling a new flight has low complexity in the number of scheduled flights. Note that the method proposed here applies to all lane-based scheduling problems, e.g., in Urban Air Mobility (UAM) systems, automated warehouses, etc.

An Efficient Strategic Deconfliction Algorithm for Large-Scale UAS Traffic Management

David Sacharny[1] Thomas C. Henderson[2] Michael Cline[3]

***Abstract*— We propose a lane-based airway navigation framework wherein each lane is one-way, and intersections are handled by means of polygonal lane roundabouts; it is possible to assign flight plans so that the set of all such plans is strategically deconflicted. That is, no two Unmanned Aerial Systems (UAS) will ever get closer in a lane than the minimum allowed headway time (or distance) of each other. We describe here a method to determine all allowable launch times (i.e., strategically deconflicted) given a requested launch time interval and a set of scheduled flights. Scheduling a new flight has low complexity in the number of scheduled flights. Note that the method proposed here applies to all lane-based scheduling problems, e.g., in Urban Air Mobility (UAM) systems, automated warehouses, etc.**

I. INTRODUCTION

Currently, a framework is being developed to support large-scale (thousands) of UAS flights per day over urban areas, and NASA has proposed a UAS deconfliction strategy that requires service providers (UAS Service Suppliers or USS's) to exchange full flight path information and to mutually find a deconflicted set of flights. This approach has high complexity and sacrifices UAS operator privacy. We propose a lane-based deconfliction strategy which reduces the shared information to be simply lane entry and exit times and UAV speed through the lane. Then given a requested launch time interval, it is possible to determine the set of all allowable (deconflicted) time intervals within the requested interval.

Techniques proposed for flight planning, include full mix and layered methods [1], [2] for which safe separation is maintained by tactical collision avoidance methods in otherwise unconstrained flights. While several heuristic methods have been developed for this problem (e.g., [3]), it is still possible that the number of conflicts may overwhelm the algorithms (see [4] for an analysis of cascading effects of conflict resolution). There has been a large amount of research into quantifying the risk of conflict in this type of system (e.g., [1], [4], [5], [6], [7]), indicating that there are numerous risk factors that an operator would need to consider in order to reduce the risk of collision. Lane-based airways were analyzed in [8], however the UAS operations were not deconflicted pre-flight and instead were simulated much like car-following models (e.g., [9]). Recently, a report published by NASA detailing negotiations among stakeholders regarding requirements for USS's described the following overarching requirement for operations within the UAS Traffic management (UTM) system: "A UTM Operation should be free of 4-D intersection with all other known UTM Operations prior to departure and this should be known as *Strategic Deconfliction* within UTM" [10]. Furthermore, they discuss the requirement that any scheme for strategic deconfliction must be mandated by the airspace regulator.

The *Strategic Deconfliction Problem* **is to produce a set of scheduled flight paths such that no two aircraft ever get closer than a specified safety distance (specified either in space or time).**

Strategic deconfliction, or strategic conflict management, refers to the first of three layers of conflict management defined by the International Civil Aviation Organization (ICAO), "achieved through the airspace organization and management, demand and capacity balancing, and traffic synchronization" [11]. The next layers are applied in order of the shrinking conflict horizon, and are *tactical* in nature and termed "separation provision" and "collision avoidance." Broadly speaking, strategic conflict management deals with planning collision free paths, which in the most general case of planning for multiple agents is PSPACE-hard [12]. Even the more narrow problem of tuning velocity profiles is NP-hard [13]. We consider the simpler, but more realistic scenario given the UTM architecture, of scheduling UASs in real-time within lanes, reducing the configuration space of the UAS to a single dimension for each flight. The result is a practical, computationally tractable algorithm for strategic conflict management. The theoretical contribution of this paper provides an efficient algorithm for strategic deconfliction. The experimental section of this paper considers the capacity constraints imposed by this system, which enables airspace regulators to make informed decisions about how to address the demand from users.

The majority of motion planning algorithms rely on some form of discretization, e.g., cell-decomposition or probabilistic sampling such as Rapidly Exploring Random Trees (RRT) [12], [14]. The algorithms that don't rely on discretization either assume a functional representation of trajectory (e.g., a spline) or are tactical because they apply to controls directly. The decisions related to discretization are vital in determining the effectiveness and complexity of a motion planning problem. For instance, in the RRT algorithm the line connecting sampled locations must be discretely sampled to determine if any conflicts exist. If the sample resolution is too fine, then computation resources suffer. If the sample

[1]David Sacharny is with the School of Computing, University of Utah, Salt Lake City, UT, USA sacharny@cs.utah.edu

[2]Thomas C. Henderson is with the School of Computing, University of Utah, Salt Lake City, UT, USA tch@cs.utah.edu

[3]Michael Cline is with the School of Computing, University of Utah, Salt Lake City, UT, USA m.cline@utah.edu

resolution is too coarse, then there is the possibility that a conflict exists that would not be discovered until it was too late.

The strategic conflict management problem shares characteristics with many application areas, as well as theoretical work in discrete mathematics (see [15] in the context of scheduling) and topology (see the chapter on configuration spaces in [14]). This includes the *Air Traffic Flow Management Problem* (TFMP) [16], [17], [18], *The Job-Shop Scheduling Problem* [18], [19], [20], [21], [22], *The Multi-Robot Motion Planning Problem* [23], [24], [25], *The Traffic Assignment Problem* [6], [26], and *Optimization Problems* [27], [28]. The FAA expects tens of thousands of UASs to utilize the airspace in close proximity over urban areas, therefore the problem model composition is important to ensure that safety requirements are met. There are two ways in general to represent the safety requirements: using constraints, or with an objective function. The objective is to maximize the separation (or headway) between UAS's. Assuming the solution is optimal, the question of whether it meets the safety requirement is determined by a threshold, e.g., "the minimum separation is at least 10 meters," or "the minimum separation is at least 10 meters with 99.9% probability." We only consider the constraint model which casts the objective as a function of the time between desired release times and scheduled release times.

Lanes, as we propose them, are created as desired by USS's and approved by the UTM authorities. Each lane has an entry point and an exit point and allows one-way travel from entry to exit. Where lanes intersect, we introduce an airspace structure inspired by roadway roundabouts. In addition, we provide a computationally tractable trajectory scheduling algorithm for UAS Service Suppliers (USS) within this structure. A capacity analysis follows the description of the airway structure to provide a baseline for further research. Prior research into the capacity of airspaces does not simultaneously consider the complexity of planning the operations; however, both concepts must be considered together since the airspace regulator is expected to manage both. We analyze the relationship between airspace capacity and such a lane-based structure. Over dense urban areas which are of primary concern here, there will most likely be a limited set of lanes possible, and understanding the capacity of the lane system is important to urban planners.

The lane-based method proposed here can be seen as an extension of Victor and Jet Airways used in manned air traffic management [29]. However, these were rigidly defined off of VOR systems (Very High Frequency Omnidirectional Range) in the 1960's. Moreover, such routes were under visual flight rules, and at intersections required human deconfliction. The innovation in our approach is the dynamic nature of lane creation and deletion, as well as the introduction of roundabouts which permit efficient strategic deconfliction. Finally, we note that in the following, it is not assumed that UAS have the same speed in a lane, multiple levels (altitudes) are used in the lanes, and although lanes may be above roadways, this is not required; on the latter subject, it should be noted that NASA has stated (emphasis added) [30]:

> With regard to the routes that UAM will traverse between two vertiports, a *natural starting point* for emergent UAM operations is to fly along defined helicopter routes ... These helicopter routes tend to overlay highways and freeways on the ground to mitigate societal concerns"

In the experiments described here, shortest route lane sequences are generated over urban areas, (although arbitrary lane sequences may be used), but these are not large distance interstate routes, and the altitudes of lanes are somewhat arbitrary but are safely separated.

II. Strategic Deconfliction Algorithm

The current approach proposed by the FAA/NASA is shown in Figure 1, where USS_1 and USS_2 have a number of scheduled flights (USS_1 flights in green and USS_2 in blue). These flights have already been deconflicted for operation

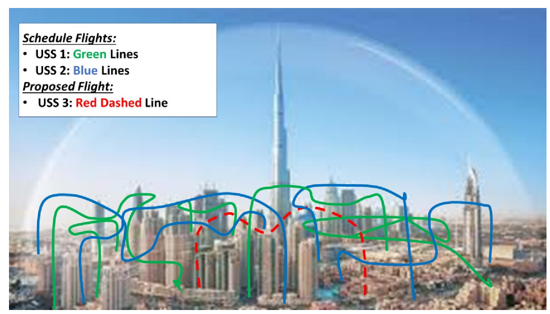

Fig. 1. Sketch of Current FAA-NASA Proposed Strategic Deconfliction.

over some time interval. USS_3 wants to schedule a flight (red dashed line), but in order to do so, must deconflict flights pairwise with both USS_1 and USS_{12} which means all flight paths must be shared. If some other USS manages to deconflict and schedule a flight in this space-time before USS_3, then USS_3 must start the process all over from the beginning [31]. Note that thousands of flights a day are envisioned, thus making the complexity of strategic deconfliction overwhelming.

We propose an alternative approach in which lanes are created and reserved. Before giving the details, a simple example is provided; consider Figure 2. The three line segments represent lanes, and all flights must move along these lanes. Nodes 1 and 4 are on the ground while nodes 2 and 3 are some distance up in the air. The lanes allow only one direction of travel. A flight must schedule its entry-exit times through a sequence of lanes, where the exit time from the previous lane equals the entry time of the following lane. In order to determine whether flights have a conflict, we propose the Space-Time Lane Diagram (STLD) as shown in Figure 3, where the horizontal axis is the time axis, and the vertical axis is distance along the lane. An STLD is created for each lane as shown. In the leftmost figure, there are two scheduled flights (1 and 2) with entry (start)

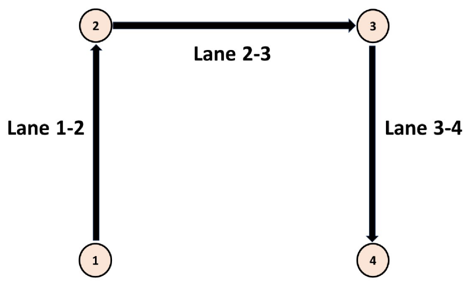

Fig. 2. Simple Three-Lane Layout.

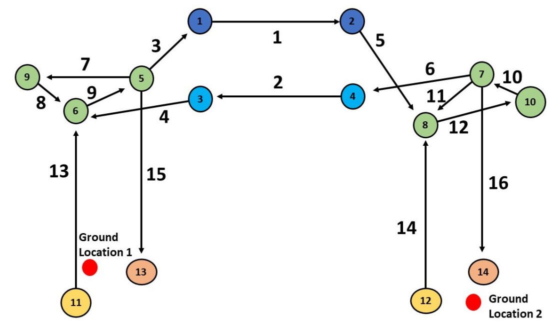

Fig. 4. The Lanes (and Vertexes) for the Two Ground Locations Case.

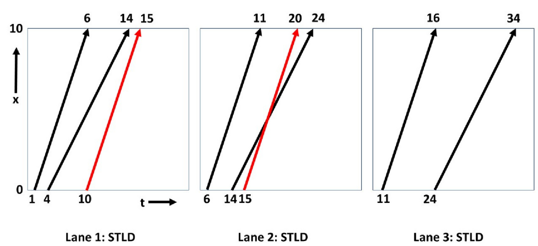

Fig. 3. STLD's for 2 Scheduled Flights on the Three Lanes (black) as well as a Proposed Flight (red).

times of 1 and 4, and speeds of 2 and 1, respectively. The STLD's show their progress through the 3 lanes; it can be seen that there is always a time headway of at least 1 unit, and that the flights get further apart as they proceed along the lanes. Suppose a new flight must be scheduled, with speed 2, and the possible launch time interval is [0,21] (meaning any start time in this interval is acceptable). The proposed algorithm will find all possible sub-intervals in [0,21] in the lane that are deconflicted with the scheduled flights. In this case, the solution is $\{[0, 0], [2, 3], [20, 21]\}$. For example, if the proposed flight starts at time 0, then it is 1 time unit from flight 1, and since they go the same speed, they never get any closer. Moreover, for Lane 2-3 and Lane 3-4 the proposed flight is always 1 unit to the left of flight 1, so it is allowed. On the other hand, consider a flight start at time 10; then it exits Lane 1-2 and enters Lane 2-3 at time 15; the figure shows that this flight then crosses the path of flight 2 and therefore is disallowed.

Next consider the more general layout of lanes; Figure 4 represents the airway corridors (lanes) between two ground locations. Launch and land nodes exist for both locations in this example; these are nodes 11 and 13, and 12 and 14, respectively, and vertical lanes exist between these and the roundabout. Nodes 5, 6, and 9 lie on a circle above the first ground location and form a (polygonal) roundabout. Lane 1 (going from node 1 to node 2) provides a way from Ground Location 1 to 2, while Lane 2 (located below Lane 1) provides a return corridor. A flight from Ground Location 1 to 2 follows the sequence of lanes: 13, 9, 3, 1, 5, 12, 10, and 16, and can be viewed as a polyline. In this example, Lane 1 is at altitude 534 feet, Lane 2 is at 467 feet, and the roundabouts are at 500 feet. These may be set to other values as desired by the system designers. An airway lane constrains the trajectory of the UAS to the center-line of the airway, referred to as the longitudinal direction of the aircraft trajectory in prior research (e.g., [4]). The vertical and lateral directions are assumed to be under control so the vehicle remains inside the lane, and uncertainty is accounted for in the design of the width and height of the lane. The critical aspect of this formulation is that there are no crossing-conflicts. In previous work [32] we gave a discrete time slot algorithm for launch time selection, whereas here the solution is over continuous intervals.

To better utilize intersections, only merging or diverging conflicts should exist because crossing conflicts require that the scheduler manage nodes as well as segments. This would add additional constraints on UAS requesting time within an intersection that would be independent otherwise. Since each segment is defined by exactly one schedule that manages UAS arrivals, organizing the airspace in this way removes the need for intersection management such as the signalized intersections in [8]. In Figure 5, Node 2 is an example of a diverging conflict, where incoming traffic is split into two traffic streams. Node 1 is an example of a merging conflict, where two traffic streams are joined into one. Crossing conflicts may be eliminated by implementing a roundabout, a concept borrowed from ground traffic engineering [33]. Figure 5 displays the graph model for a roundabout, which includes unidirectional edges between eight nodes (each node represents the endpoint of a segment) in a counter-clockwise direction.

The primary safety concern is to schedule flights so that no two UAS are ever closer than the minimal specified headway time; here we use the maximum of all headway times to ensure safety, and call it h_t (we can also plan using headway distance). On the other hand, optimal resource utilization requires packing as many flights as possible into the lanes. Assume that requested flight launch times are uniformly distributed across a fixed-length time interval, say from $[0, x]$. Then this problem has been studied by Renyi [34], [35] as a parking problem (i.e., cars of unit length are parked in a $[0, x]$ interval at uniformly distributed locations), and it was shown that the parking density, $M(x)/x$, Renyi's constant, is

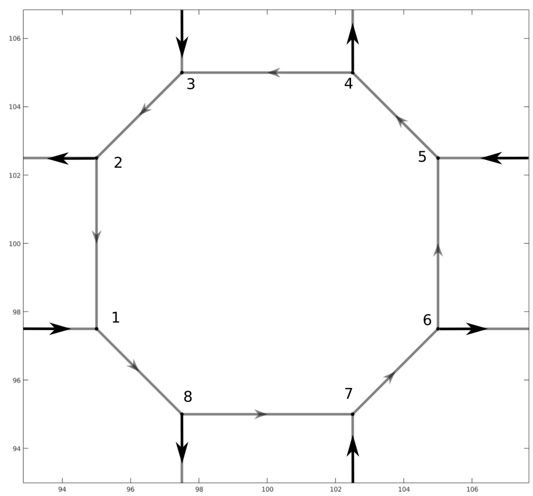

Fig. 5. Airway Roundabout.

0.74759, in the limit as x goes to infinity, where $M(x)$ is the mean of a number of trials with sampling from the uniform distribution. This provides a useful tool for analyzing flight densities through the lanes. For example, in our simulations on this problem with $x = 100$ (time units), the flight packing density was found to be 0.743, consistent with Renyi's constant and shows that consistent analysis of lane traffic is feasible. Of course, denser packing is possible if only the earliest available launch time is assigned as opposed to the closest to the desired time.

The other main issue is the determination of whether a proposed flight conflicts with any scheduled flight. The STLD is used to solve this problem. Suppose that there exists a set of scheduled flights which are represented in terms of lane enter-exit times and speed through each lane (the speed of a UAS is assumed constant along a lane, but speeds may differ across UAS and lanes, or vary to a small degree while maintaining minimum headway). Let $F(c)$ be a set of n scheduled flights through lane c defined as:

$$F(c) \equiv \{(t_{1,1}, t_{1,2}, s_1), ..., (t_{n,1}, t_{n,2}, s_n)\}$$

where for flight i, $t_{i,1}$ is the lane entry time, $t_{i,2}$ is the lane exit time, and s_i is the speed of the flight through the lane. Furthermore, let a flight request interval be specified as:

$$R \equiv [q_1, q_2, s]$$

where q_1 is the first possible launch time, q_2 is the latest possible launch time, and s is the proposed speed which may differ from scheduled flights' speeds, but is assumed constant (variations in speed are allowed, but must be accommodated in the headway considerations). What must be determined is the set of (possibly disjoint) intervals in $[q_1, q_2]$ which are allowable launch times (i.e., strategically deconflicted). In order to determine this, the requested launch time interval is put in the Lane 1 STLD as shown in Figure 6(a), where d_1 is the length of Lane 1, q_3 is $q_2 + \frac{d_1}{s}$, and q_4 is $q_1 + \frac{d_1}{s}$. Each flight in Lane 1 is compared separately to

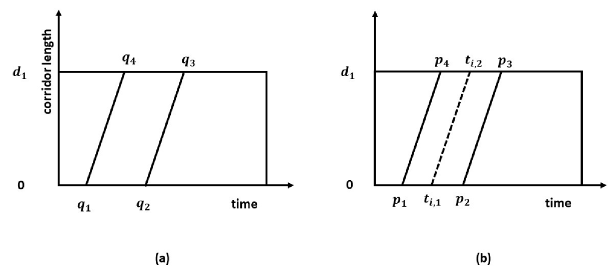

Fig. 6. Space-Time Lane Diagrams: (a) trajectory boundaries for requested launch time interval $[q_1, q_2]$. (b) The headway boundary trajectories for a scheduled flight which enters the lane at time $t_{i,1}$ and exits at time $t_{i,2}$. The requested flight must start at some time between q_1 and q_2 and its path through the STDL may not enter the $p_1p_2p_3p_4$ parallelpiped.

the requested time interval to ensure that the time headway, h_t, is respected. Figure 6(b) shows the exclusion zone (the $p_1p_2p_3p_4$ quadrangle) for flight i, where $p_1 = t_{i,1} - h_t$, $p_2 = t_{i,1} + h_t$, $p_3 = t_{i,2} + h_t$, $p_4 = t_{i,2} - h_t$.

To determine safe launch time intervals, first consider the labeling of the STLD shown in Figure 7. The labels are

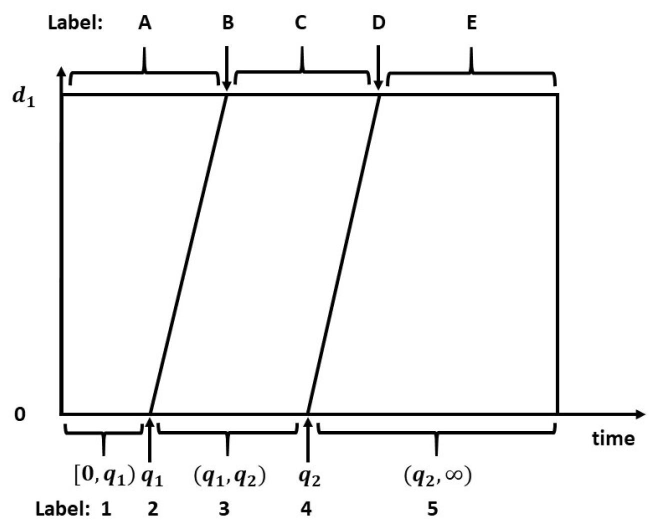

Fig. 7. Space-Time Lane Diagram Labels. 1,2,3,4,5 indicate intervals and times at the entry to the lane, and A,B,C,D,E indicate times at the lane exit.

defined as follows:

- *Label 1*: The interval $[0, q_1)$
- *Label 2*: The point q_1
- *Label 3*: The interval (q_1, q_2)
- *Label 4*: The point q_2
- *Label 5*: The interval (q_2, ∞)
- *Label A*: The interval $[0, q_4)$
- *Label B*: The point q_4
- *Label C*: The interval (q_4, q_3)
- *Label D*: The point q_3

- *Label E*: The interval (q_3, ∞)

The two (leading and trailing) headway trajectories arising from the scheduled flight are labeled according to where their endpoints lie with respect to the requested launch interval. For example, if $p_2 < q_1$ and $p_3 < q_4$, then the label for that trajectory is 1A since both start and end points are in the first intervals at distances 0 and d_1, respectively. The relation of a previously scheduled flight in a lane to the requested launch time interval is determined by the labels of the two scheduled flight headway trajectories; the requested launch interval is shown in red. Figure 8 shows 15 possible combinations. For example, 1A,1A is the case where both headway trajectories are completely to the left (i.e., before in time) of the first possible launch time trajectory through the lane. Note that in the figures, $ts = \frac{d_1}{s}$ is the time for the requested flight to cross the lane. Also, the square brackets ([]) in the figure indicate the empty interval. Although there are 625 total

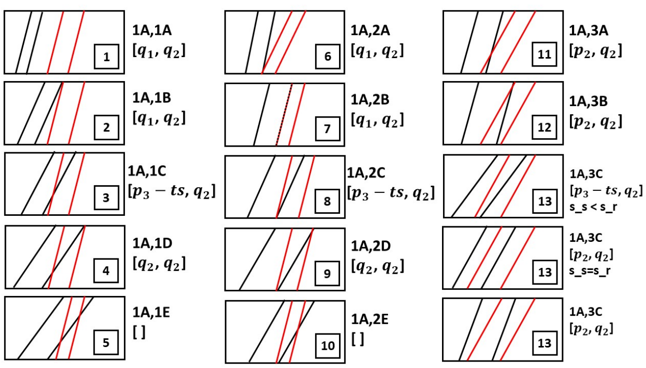

Fig. 8. Space-Time Lane Diagrams for 15 Possible Label Combinations, where [] represents the empty set. Note that Table II Provides the Complete Set of Interval Possibilities; these are the Foundation to Solve Strategic Deconfliction.

label combinations, only 139 are physically possible; for example, no start time can be greater than the end time; Table II gives the complete enumeration which underlies the SD Algorithm. For each combination, it is possible to give the safe launch intervals contained in the requested interval (see the figure for some examples). In some cases, there is no possible safe launch time (e.g., 1A,1E in the figure). For other combinations, the resulting safe intervals depend on the relative speeds of the two UAS. An example of this is 1A,3C where a scheduled flight slower than the requested flight has a different interval as when the scheduled flight is equal or greater in speed. To determine the viability of a flight through the complete sequence of lanes, each lane is considered in order as described by Algorithm SD (Strategic Deconfliction).

Algorithm SD (Strategic Deconfliction)

On input:
  lanes: lane sequence for requested flight
  $[q_1, q_2]$: requested launch interval
  n_c: number of lanes
  flights: flights per lane
  h_t: maximum required headway time
On output:
  Safe time intervals to launch
begin
possible_intervals $\leftarrow [q_1, q_2]$
for each lane $c \in$ lanes
  time_offset $\leftarrow$ time to get to lane c
  possible_intervals $\leftarrow$ possible_intervals + time_offset
  for each flight, f, in lane c
    new_intervals $\leftarrow \emptyset$
    for each interval in possible_intervals
      $[t_1, t_2] \leftarrow$ interval i
      label $\leftarrow$ get_label($t_{f,1}, t_{f,2}, s_f, t_1, t_2, s, h_t$)
      f_int $\leftarrow$ get_interval(label,$t_{f,1}, t_{f,2}, s_f, t_1, t_2, s, h_t$)
      new_intervals $\leftarrow$ merge(new_intervals,f_int)
    end
  end
  possible_intervals $\leftarrow$ new_intervals
end
possible_intervals $\leftarrow$ possible_intervals - time to last lane

where *get_interval* uses Table II to get the possible intervals for this combination, *merge* intersects and adds new intervals resulting from the newly considered flight, and *get_label* produces the STLD label from the set $\{1, 2, 3, 4, 5, A, B, C, D, E\}$.

The key computational cost of this algorithm is the determination of f_int; each instance of this can be done in constant time; call it operation $\mathcal{I}$. Then given n lanes, f_k flights in lane k, and f is the total number of flights in the lane sequence, then the total number of $\mathcal{I}$ operations is less than or equal to:

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

The second sum dominates the complexity, and assuming f_k is on average $\frac{f}{n}$, and since there are $\binom{n}{2}$ terms, then the big O complexity is $O(f^2)$.

A. Lane Stream Properties: Occupancy, Density and Flow

For the discussion here, we assume an airway lane of length d and consider a time interval of length t_{max}, call it $[0, t_{max}]$. Also assume that all UAS fly through the lane with a constant speed, s. A flight scheduler assigns start times for flights to go through the lane; let S be a set of such start times. Then, to satisfy constraints, it must be the case that no two start times are closer than headway time, h_t, of each other. This is equivalent to packing segments of length h_t into the lane (time) interval. The maximum number of UAS possible in the lane during the time it takes a UAS to traverse the lane, n^t_{max}, is then:

$$n^t_{max} \equiv \lfloor \frac{d}{s \cdot h_t} \rfloor$$

Clearly, achieving n^t_{max} depends on obtaining a perfectly packed requested start time sequence. There is also a maximum number of UAS in terms of spatial packing, and this is given by:

$$n^d_{max} \equiv \lfloor \frac{d}{h_x} \rfloor$$

where $h_x = s \cdot h_t$ is the headway distance.

Suppose that flight request start times are sampled from a uniform distribution across the given time interval $[0, t_{max}]$. The *time occupancy*, $\Theta_t(\mathcal{A})$, is a function of the scheduling algorithm $\mathcal{A}$ and is defined as:

$$\Theta_t(\mathcal{A}) \equiv \frac{\mu_{\mathcal{A}}}{n^t_{max}}$$

where $\mu_{\mathcal{A}}$ is the mean number of flights through the lane during the time interval $[0, \frac{d}{s}]$ of several trials with algorithm $\mathcal{A}$. If the scheduler has no choice but to assign the requested start time if possible and otherwise reject the request (call this algorithm $\mathcal{A}_0$), then this is an example of Renyi's Parking Problem [35], and $\Theta(\mathcal{A}_0) \rightarrow 0.74759$ as $t_{max} \rightarrow \infty$. In the experiments below, we compare algorithms and lane parameter sets by means of their observed time and space occupancy measures.

Next consider standard ground traffic stream properties: density, occupancy and flow (see [36] for a detailed discussion). The *spatial density* of the lane at time t, $k_s(t)$, is defined as:

$$k_s(t, \mathcal{A}) \equiv \frac{\mu_{\mathcal{A}}}{d}$$

that is, the average number of vehicles in the lane over the length of the lane. *Spatial occupancy* can then be defined as:

$$\Theta_s(t, \mathcal{A}) \equiv \frac{\Theta_t(\mathcal{A}) \cdot n^d_{max}}{d}$$

Finally, *spatial flow*, $q(t, \mathcal{A})$, is defined as:

$$q_s(t, \mathcal{A}) \equiv k_s(t, \mathcal{A}) \cdot s$$

These traffic stream properties are used to characterize the performance of a set of algorithms compared in the experimental section.

III. EXPERIMENTS

The first experiment establishes a baseline for Renyi's constant; our lane packing implementation obtains $\Theta_t(\mathcal{A}_0) = 0.7448$ for 1000 trials on $(0, 100]$, and $\Theta_t(\mathcal{A}_0) = 0.7447$ for 10000 trials. The first experiment related to lanes aims to establish whether Renyi's constant holds for these types of intervals. To ascertain this, a USS is called each time step for the first 3700 time units, and it issues a uniformly sampled request in the time interval $(4000, 4100]$. For 10 experiments, the result is $\Theta_t(\mathcal{A}_0) = 0.75$. This shows that the system is acting as expected.

We now consider three scheduling algorithms, each of which receives a requested launch interval, and a desired launch time:

- Algorithm $\mathcal{A}_0$: Only accepts a flight plan with the desired launch time.
- Algorithm $\mathcal{A}_1$: chooses the closest allowable time to the desired time in the interval (and the earlier time in case of a tie).
- Algorithm $\mathcal{A}_2$: selects the earliest allowable time, if any, in the interval.

TABLE I

RESULTS OF 3X3 GRID SIMULATIONS.

	Algorithm $\mathcal{A}_0$	*Algorithm* $\mathcal{A}_1$	*Algorithm* $\mathcal{A}_2$
Test 1	1552	3094	3312
Test 2	1570	3089	3344
Test 3	1572	3073	3324
Test 4	1560	3092	3315
Test 5	1577	3081	3332
Test 6	1515	3138	3311
Test 7	1541	3108	3353
Test 8	1584	3095	3336
Test 9	1542	3109	3367
Test 10	1550	3073	3323
Mean($\mu_{\mathcal{A}}$)	1556.3	3095.2	3331.7

The parameters of the scenario are as follows: a 3x3 grid with distance 50 units between each node is used, and there is a launch and land lane at each road intersection (see Figure 9). The air roundabouts are located at 50 units in the air, while upper and lower lanes are at 53 and 46 units, respectively. The speed of each UAV is set to 1 unit per time step, and h_t is also set to 1. There is 1 USS issuing 5 flight requests each time step, and these are in 100-time unit intervals with a random desired time uniformly sample in the interval. The simulation is run 1000 time steps for each algorithm, with 10 trials of each.

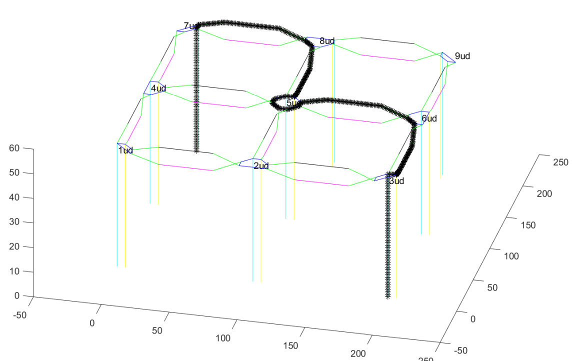

Fig. 9. 3x3 Grid Airway Layout and Example Flight Path.

The results are given in the Table I, showing that algorithm $\mathcal{A}_2$ schedules the most flights. However, deeper insight into the algorithms performance can be gained from the lane stream properties. For example, Figure 10 compares the time occupancy of the three algorithms in Lane 121.

It can be seen that Algorithm $\mathcal{A}_2$ outperforms the other two, but in terms of flight policy, it can be seen that

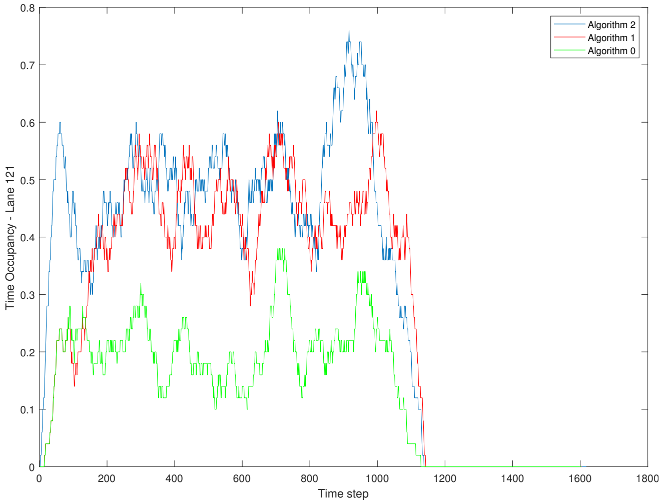

Fig. 10. Lane 121 Time Occupancy for the Three Algorithms. $\mathcal{A}_2$ Outperforms the Other Two.

Algorithm $\mathcal{A}_1$ does not perform very much worse than $\mathcal{A}_2$ while it assigns launch times closer to the desired times of the operators. Thus, a small time occupancy difference may be acceptable given the higher satisfaction of users. This kind of cost-benefit analysis is quite useful to UAM administrators.

IV. Conclusions and Future Work

We propose a lane-based strategic deconfliction scheduling method. *Algorithm SD*, which is more efficient and effective than current state-of-the-art large-scale UAS flight management approaches (e.g., NASA-FAA's proposal that USS's exchange flight data details). In addition, *Algorithm SD* can be used without sacrificing the privacy of operators flight data. We are working with the Utah Department of Transportation, Aeronautics Division, to make this system available in actual practice, and have built and simulated flights in airways over large sections of Salt Lake City, Utah.

Further developments are underway:

- the current approach only considers the shortest path through the lanes, and a variety of paths should be considered,
- dynamic re-scheduling must be included to account for UAS deviations from the prescribed plans (i.e., contingencies), or for emergency situations,
- flight planning should include environmental and other factors, and
- *Algorithm SD* may be applied to any type of lane-based, large-scale UAV environment, for example, automated warehousing.

Techniques for lane-based contingency management are currently under study (e.g., emergency lanes next to regular lanes), and dynamic creation of lanes allows any advantages for handling contingencies (e.g., creating landing lanes). In addition, Software Defined Networking ideas are begin applied to allow real-time distributed routing. These methods show great promise in making large-scale UAS traffic management possible.

TABLE II

Enumeration of All Possible Scheduled Flights vs Requested Launch Times Interactions.

Labels	Intervals	*Labels*	*Intervals*	*Labels*	*Intervals*
1A,1A	$[q_1,q_2]$	1C,5E	$\emptyset$	3B,4C	$[q_1,q_1;$
				3B,4C	$q_2,q_2]$
1A,1B	$[q_1,q_2]$	1D,1E	$\emptyset$	3B,5C	$[q_1,q_1]$
1A,1C	$[p_3-t_s,q_2]$	1D,2E	$\emptyset$	3B,5D	$[q_1,q_1]$
1A,1D	$[q_2,q_2]$	1D,3E	$\emptyset$	3B,5E	$[q_1,q_1]$
1A,1E	$\emptyset$	1D,4E	$\emptyset$	3C,3C	$[q_1,p_1,<;$
				3C,3C	$p_3-t_s,q_2,<]$
				3C,3C	$[q_1,p_1,=;$
				3C,3C	$p_2,q_2,=]$
				3C,3C	$[q_1,p_4-t_s,>;$
				3C,3C	$p_2,q_2,>]$
1A,2A	$[q_1,q_2]$	1D,5E	$\emptyset$	3C,3D	$[q_1,p_1;$
				3C,3D	$q_2,q_2]$
1A,2B	$[q_1,q_2]$	1E,1E	$\emptyset$	3C,3E	$[q_1,p_1]$
1A,2C	$[p_3-t_s,q_2]$	1E,2E	$\emptyset$	3C,4C	$[p_1,p_4-t_s;$
				3C,4C	$q_2,q_2]$
1A,2D	$[q_2,q_2]$	1E,3E	$\emptyset$	3C,4D	$[q_1,p_1;$
				3C,4D	$q_2,q_2]$
1A,2E	$\emptyset$	1E,4E	$\emptyset$	3C,4E	$[q_1,p_1]$
1A,3A	$[p_2,q_2]$	1E,5E	$\emptyset$	3C,5C	$[q_1,p_4-t_s]$
1A,3B	$[p_2,q_2]$	2A,3A	$[p_2,q_2]$	3C,5D	$[q_1,p_4-t_s]$
1A,3C	$[p_3-t_s,q_2,<]$	2A,3B	$[p_2,q_2]$	3C,5E	$[q_1,p_1,\leq;$
1A,3C	$[p_2,q_2,\geq]$			3C,5E	$q_1,p_4-t_s,>];$
1A,3D	$[q_2,q_2]$	2A,3C	$[p_2,q_2]$	3D,3E	$[q_1,p_1]$
1A,3E	$\emptyset$	2A,4A	$[q_2,q_2]$	3D,4E	$[q_1,p_1]$
1A,4A	$[q_2,q_2]$	2A,4B	$[q_2,q_2]$	3D,5E	$[q_1,p_1]$
1A,4B	$[q_2,q_2]$	2A,4C	$[q_2,q_2]$	3E,3E	$[q_1,p_1]$
1A,4C	$[q_2,q_2]$	2A,5A	$\emptyset$	3E,4E	$[q_1,p_1]$
1A,4D	$[q_2,q_2]$	2A,5B	$\emptyset$	3E,5E	$[q_1,p_1]$
1A,4E	$\emptyset$	2A,5C	$\emptyset$	4A,5A	$\emptyset$
1A,5A	$\emptyset$	2A,5D	$\emptyset$	4A,5B	$\emptyset$
1A,5B	$\emptyset$	2A,5E	$\emptyset$	4A,5C	$\emptyset$
1A,5C	$\emptyset$	2B,3C	$[p_1,q1;$	4A,5D	$\emptyset$
		2B,3C	$p_2,q_2]$		
1A,5D	$\emptyset$	2B,4D	$[p_1,q_1;$	4A,5E	$\emptyset$
		2B,4D	$q_2,q_2]$		
1A,5E	$\emptyset$	2B,5E	$[p_1,q_1]$	4B,5C	$[q_1,q_1]$
1B,1C	$[p_3-t_s,q_2]$	2C,3C	$[p_1,q_1;$	4B,5D	$[q_1,q_1]$
		2C,3C	$p_3-t_s,q_2]$		
1B,1D	$[q_2,q_2]$	2C,3D	$[p_1,q_1;$	4B,5E	$[q_1,q_1]$
		2C,3D	$q_2,q_2]$		
1B,1E	$\emptyset$	2C,3E	$[p_1,q_1]$	4C,5C	$[q_1,p_4-t_s]$
1B,2C	$[p_3-t_s,q_2]$	2C,4E	$[p_1,q_1]$	4C,5D	$[q_1,p_4-t_s]$
1B,2D	$[q_2,q_2]$	2C,5E	$[p_1,q_1]$	4C,5E	$[q_1,p_4-t_s]$
1B,2E	$\emptyset$	2D,3E	$[p_1,q_1]$	4D,5E	$[q_1,q_2]$
1B,3C	$[p_3-t_s,q_2]$	2D,4E	$[p_1,q_1]$	4E,5E	$[q_1,q_2]$
1B,3D	$[q_2,q_2]$	2D,5E	$[p_1,q_1]$	5A,5A	$\emptyset$
1B,3E	$\emptyset$	2E,3E	$[p_1,q_1]$	5A,5B	$\emptyset$
1B,4E	$\emptyset$	2E,4E	$[p_1,q_1]$	5A,5C	$\emptyset$
1B,5E	$\emptyset$	2E,5E	$[p_1,q_1]$	5A,5D	$\emptyset$
1C,1C	$[p_3-t_s,q_2]$	3A,3A	$[p_2,q_2]$	5A,5E	$\emptyset$
1C,1D	$[q_2,q_2]$	3A,3B	$[p_2,q_2]$	5B,5C	$[q_1,q_1]$
1C,1E	$\emptyset$	3A,3C	$[p_2,q_2]$	5B,5D	$[q_1,q_1]$
1C,2C	$[p_3-t_s,q_2]$	3A,4A	$[q_2,q_2]$	5B,5E	$[q_1,q_1]$
1C,2D	$[q_2,q_2]$	3A,4B	$[q_2,q_2]$	5C,5C	$[q_1,p_4-t_s]$
1C,2E	$\emptyset$	3A,4C	$[q_2,q_2]$	5C,5D	$[q_1,p_4-t_s]$
1C,3C	$[p_3-t_s,q_2]$	3A,5A	$\emptyset$	5C,5E	$[q_1,p_4-t_s]$
1C,3D	$[q_2,q_2]$	3A,5B	$\emptyset$	5D,5E	$[q_1,q_2]$
1C,3E	$\emptyset$	3A,5C	$\emptyset$	5E,5E	$[q_1,q_2]$
1C,4E	$\emptyset$	3B,3C	$[q_1,q_1;$		
		3B,3C	$p_2,q_2]$		

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] D.-S. Jang, C. 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.

[9] 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.

[10] J. Rios, "NASA UTM Strategic Deconfliction Final Report," NASA Ames Research Center, Tech. Rep., 2018.

[11] ICAO, "Doc 9854 AN/458 - Global Air Traffic Management Operational Concept," *International Civil Aviation Organization*, p. 82, 2005.

[12] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[13] D. Alejo, J. Díaz-Báñez, J. 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.

[14] H. Choset, K. M. L. 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.

[15] H. Friedman, D. Rothman, and J. Mackenzie, "Solution to: An Unfriendly Seating Arrangement (Problem 62-3)," *SIAM Review*, vol. 6, no. 2, pp. 180–182, 1964.

[16] D. Bertsimas and S. S. Patterson, "The Air Traffic Flow Management Problem with Enroute Capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 6 1998.

[17] J. Rios and J. Lohn, "A Comparison of Optimization Approaches for Nationwide Traffic Flow Management," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 8 2009.

[18] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, New York: Dover Publications, 1988.

[19] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th ed. New York, NY, USA: Springer International Publishing, 2016.

[20] A. Arisha, P. Young, and M. El Baradie, "Job Shop Scheduling Problem: an Overview," in *International Conference for Flexible Automation and Intelligent Manufacturing (FAIM 01)*, Dublin, Ireland, 2001, pp. 682–693.

[21] A. Jones, L. C. Rabelo, and A. T. Sharawi, "Survey of Job Shop Scheduling Techniques," in *Wiley Encyclopedia of Electrical and Electronics Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 12 1999.

[22] J. C. Beck and N. Wilson, "Proactive Algorithms for Scheduling with Probabilistic Durations," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 28, 2005, pp. 1201–1206.

[23] J. van den Berg, J. Snoeyink, M. Lin, and D. Manocha, "Centralized Path Planning for Multiple Robots: Optimal Decoupling Into Sequential Plans," in *Robotics: Science and Systems V*, 2009.

[24] 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.

[25] J. Rios and K. Ross, "Delay Optimization for Airspace Capacity Management with Runtime and Equity Considerations," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, p. 10.

[26] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*. Dover Publications, 2015.

[27] G. Neuhauser, A. R. Kan, and M. Todd, *Optimization*. New York, NY: North-Holland, 1989.

[28] I. Zelinkaet, V. snasel, and A. Abraham, *Handbook of Optimization*. Berlin, Germany: Springer Verlag, 2013.

[29] J. S. Duncan, "Instrument Procedures Handbook," Federal Aviation Administration, Washington, DC, Tech. Rep. FAA-H-8083-16B, 2017.

[30] D. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. Goodrich, J. Homola, H. Idris, P. Kopardekar, J. Lachter, N. Neogi, H. Ng, R. Oseguera-Lohr, M. Patterson, and S. Verma, "Urban Air Mobility Airspace Integration Concepts and Considerations," in *2018 Aviation Technology, Integration and Operations Conference*. Atlanta, GA: AIAA, June 2018.

[31] Google, "InterUSS Platform," 2018. [Online]. Available: https://github.com/interuss/dss/tree/master/assets

[32] D. Sacharny and T. Henderson, "A Lane-Based Approach fo Large-Scale Strategic Conflict Management for UAS Service Suppliers," in *IEEE International Conference on Unmanned Aircraft Systems*, Atlanta, GA, 2019.

[33] E. National Academies of Sciences and Medicine, *Roundabouts: An Informational Guide*, 2nd ed. Washington, D.C.: Transportation Research Board, 12 2016.

[34] M. P. Clay and N. Simanyi, "Renyi's Parking Problem Revisited," *Stochastics and Dynamics*, vol. 16, pp. 1–12, June 2014.

[35] A. Rényi, "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.

[36] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Germany: Springer Verlag, 2013.