

# FAA-NASA vs. Lane-Based Strategic Deconfliction

David Sacharny<sup>1</sup>   Thomas C. Henderson<sup>2</sup>   Michael Cline<sup>3</sup>   Benjamin Russon<sup>4</sup>   EJay Guo<sup>5</sup>

**Abstract**—The Federal Aviation Administration (FAA) and NASA have provided guidelines for Unmanned Aircraft Systems (UAS) to ensure adequate safety separation of aircraft, and in terms of UAS Traffic Management (UTM) have stated[1]:

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 ... A UTM Operator must have a facility to negotiate deconfliction of operations with other UTM Operators ... There needs to be a capability to allow for intersecting operations.

The latter statement means that UTM Operators must be able to fly safely in the same geographic area. The current FAA-NASA approach to strategic deconfliction is to provide a set of geographic grid elements, and then have every new flight pairwise deconflict with UTM Operators with flights in the same grid elements. Note that this imposes a high computational burden in resolving these 4D flight paths, and has side effects in terms of limiting access to the airspace (e.g., if a new flight is deconflicted and added to the common grid elements during this analysis, then the new flight must start all over).

We have proposed a lane-based approach to large-scale UAS traffic management [2], [3] which uses one-way lanes, and roundabouts at lane intersections to allow a much more efficient analysis and guarantee of separation safety. We present here the results of an in-depth comparison of FAA-NASA strategic deconfliction (FNSD) and Lane-based strategic deconfliction (LSD) and demonstrate that FNSD suffers from several types of complexity which are generally absent from the lane-based method.

## I. INTRODUCTION AND BACKGROUND

Small Unmanned Aircraft Systems (UAS) are to be integrated into the low altitude (Class G) airspace, and initial concepts have been provided by the NASA UAS Traffic management (UTM) project [4]. A set of four Technical Capability Levels (TCL) have been defined, and the current focus is TCL 4 which addresses “an urban environment and includes handling of high density environments, large-scale off-nominal conditions, vehicle-to-vehicle communications, detect-and-avoid technologies, communication requirements, public safety operations, airspace restrictions, and other related goals.” Figure 1 shows the UTM framework proposed by NASA. The UAS Service Supplier (USS) provides key functions in managing the airspace, an in particular, is

charged with ensuring strategic deconfliction (SD) of flights, and other services usually provided by the Air Navigation Service Provider (ANSP) in manned aviation. In addition, USS are charged with monitoring flight operations. All of these functions are to be achieved in a distributed, cooperative manner. NASA’s vision is that USS perform SD by making sure that any proposed flight has no 4D (space and time) conflict with any scheduled flight in its area of operation. This also permits arbitrary flight paths.

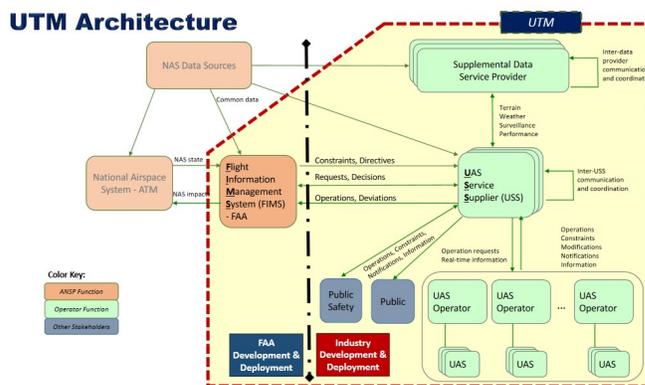


Fig. 1. NASA’s UAS Traffic Management Framework.

The FAA-NASA SD approach has some glaring problems, including the computational complexity of arbitrary 4D path planning, as well as its susceptibility to monopoly control by organizations with large-scale resources. We have proposed an alternative approach using well-defined lanes [2], [3] in which a (generally) fixed set of lanes (airways) are established, and then flights are scheduled through these lanes. This means that SD becomes a 1D problem that is solved much the same as with manned flight, that is by delay of the takeoff time.

We provide here the first detailed set of experimental results which allows analysis and comparison of the two alternative approaches. The results indicate that the lane-based approach is superior in most aspects.

## II. FAA-NASA STRATEGIC DECONFLICTION

The FNSD is based on a gridded approach in which the area of flight operations is divided into a number of grid elements, and each flight scheduled by a UTM Operator (henceforth called a UAS Service Supplier or USS), keeps track of the grid elements over which it operates. Then when a new flight is being scheduled, it only needs to deconflict with the flights with which it has common grid elements.

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

<sup>1</sup>David Sacharny is with the School of Computing, University of Utah, Salt Lake City, UT, USA sacharny@cs.utah.edu

<sup>3</sup>Michael Cline is with the School of Computing, University of Utah, Salt Lake City, UT, USA m.cline@utah.edu

<sup>4</sup>Benjamin Russon is with the School of Computing, University of Utah, Salt Lake City, UT, USA benrusson@gmail.com

<sup>5</sup>EJay Guo is with the School of Computing, University of Utah, Salt Lake City, UT, USA ejay.guo@gmail.com

Simulation parameters here are set up to conform to a 2-mile by 2-mile area divided at its highest resolution into a 100x100 grid. Thus, each 10 units corresponds to about 500 feet; this allows flights to operate at between 10 and 12 units above the ground. Figure 2 shows the grid layout with a 4x4 set of grid elements (i.e., each is about 25 ft<sup>2</sup>).

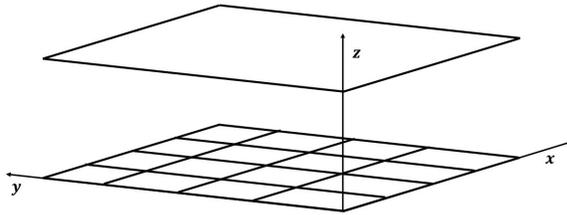


Fig. 2. The Basic Experimental Layout with a 4x4 Grid.

For the flights considered here, a flight path consists of a polyline with three line segments:

- 1) *segment 1*:  $[pt_1, pt_2]$ , a segment going straight up from a launch site to a randomly chosen altitude in the range of 10-12 units,
- 2) *segment 2*:  $[pt_2, pt_3]$ , a segment going across the workspace at a fixed altitude, and
- 3) *segment 3*:  $[pt_3, pt_4]$ , a segment going straight down to the ground.

Each flight path is comprised of a randomly selected launch site, land site, and flight altitude. In addition, each flight has a designated start time, and fixed speed for the entire flight (randomly assigned between 0.1 and 0.31 units space per unit time which corresponds to 3-10 mph, respectively).

Given a set of flights, the convention is that they are requested and deconflicted in the order of the list. That is, the first flight is scheduled as specified since there are no flights scheduled before it, the second flight must only deconflict with the first, etc. The deconfliction strategy used is based on ground delay of the flight until it has no conflicts with scheduled flights in its grid elements; this allows a fair comparison to the lane-based method which is also based on setting a conflict free launch time; moreover, this is the way standard air traffic control is accomplished. The FNSD algorithm used here is:

```
On input: scheduled_flights,
          flight_request,
          delta_t,
          headway_time
On output: new_flight
```

```
if flight_request shares no grid
  elements with scheduled_flights
then new_flight is requested flight
```

```
(earliest time); return

if there are no flight segments in the
  scheduled_flights within headway
  distance of the flight_request
  segments
then new_flight is requested flight
  (earliest time); return

pinch_pts = all segment pairs of
  scheduled flights that are within
  headway distance of the flight
  request segments

while (any pinch point segments have the
  two flights within headway distance
  during their traversal of the segment)

  shift the start time of the flight
end
```

Although this is just an example of a deconfliction method, the statistics accumulated will be somewhat independent of the particular method. This is due to the fact that the complexity is related to the number of scheduled flights which share grid elements with the flight request and the nature of the segment interactions in the grid element. This method first eliminates from consideration any scheduled flight which shares no grid elements. Next, it eliminates any which share grid elements, but none of whose flight segments are within headway distance of the flight request segments. Finally, to determine whether segment pairs that are within headway distance actually pose a problem, the time of passage of the two flights must be considered. For example, if the entry-exit time through the scheduled flight pinch segment does not overlap the entry-exit time interval of the flight request segment, then there is no conflict. Finally, if these time intervals overlap, then an analysis is performed to see if the flights get within headway distance while crossing their respective segments; if so, the start time for the flight request is delayed a fixed amount, and the impact re-analyzed.

The pinch point segments are determined as follows. Let  $P_0$  and  $P_1$  be the endpoints of segment 1,  $s \in [0, 1]$ , and  $Q_0$  and  $Q_1$  be those of segment 2. Define:

$$P(s) = P_0 + s(P_1 - P_0)$$

$$Q(t) = Q_0 + s(Q_1 - Q_0)$$

$$w(s, t) = P(s) - Q(t)$$

Then the distance squared between  $P(s)$  and  $Q(t)$  is:

$$|w(s, t)|^2 = w(s, t) \cdot w(s, t)$$

When the distance is less than the allowed headway, then the pair of segments is recorded.

### III. LANE-BASED STRATEGIC DECONFLICTION

Detailed descriptions of the lane-based UAS Traffic Management (UTM) System and strategic deconfliction in that context have been given elsewhere (see [2], [3]). Therefore, a brief overview is given here. Figure 3 shows a 5x5 layout in (a) a 3D view, and (b) an x-y view of the lanes. As can be seen, lanes go up and down to the urban airspace, there are lanes between the ground locations, and there are roundabouts to allow lane changes at intersections. Note that there is a set of lanes at two altitudes which allows for travel in both directions. Lanes are one-way, and every node in the lane graph has either in-degree 1 or out-degree 1; this prevents flights crossing at a node, and means that deconfliction can be performed on just the lanes.

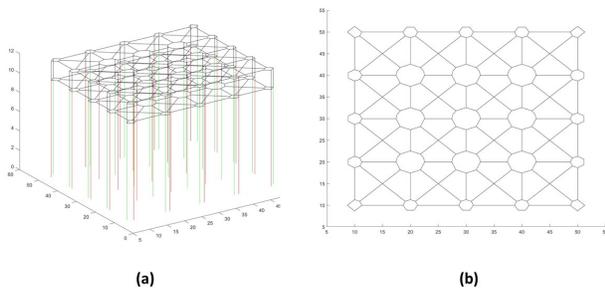


Fig. 3. Example Lane-Based UTM. (a) 3D View of Lanes. (b) x-y View of Lanes.

An additional concern with the headway distance in lanes is that the linear distance along two consecutive lanes that have an angle less than  $\pi$  radians between them allows two UAS to be closer in actual 3D space than the required minimum headway. Figure 4 shows how this can be accounted for. If  $d_S$  is the minimum 3D spatial distance allowed between

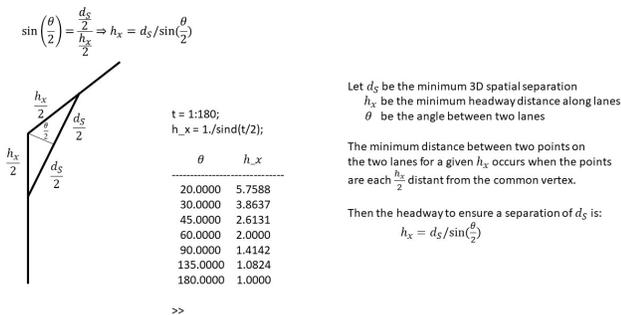


Fig. 4. How to Achieve 3D Space Minimum Distance in Terms of Minimum Headway along the Lanes.

aircraft, then given a headway distance,  $h_x$  along the lanes, then the minimum distance between the two points occurs when each aircraft is distance  $\frac{h_x}{2}$  from the common endpoint

of the lanes. Therefore, to ensure that this distance is greater than or equal to  $d_S$ , we solve for  $h_x$ :

$$h_x = \frac{d_S}{\sin(\frac{\theta}{2})}$$

A portion of a Matlab run is shown where  $h_x$  is computed when  $d_S = 1$ . For example, when the lanes are at right angles (90 degrees),  $h_x=1.4112$ , and when they lie on the same line,  $h_x = 1$ .

The strategic deconfliction method is given a sequence of lanes, and a possible start time interval, and finds all deconflicted times in that interval (this may be several sub-intervals). Figure 5 shows a set of Space Time Lane Diagrams (STLD) for a 3-lane sequence. Each STLD has time as the x-axis and distance along the lane as the y-axis. A flight through the lane is shown by a line segment whose first endpoint (lower) is the flight entry time into the lane, and whose second endpoint (upper) is the flight exit time from the lane. Blue line segments represent scheduled flights, and the red dashed line shows a possible placement to schedule a new flight. The angle of the segments indicates the speed through the lane, and as can be seen, need not be the same for every lane in the sequence, although it is assumed constant in any given lane. Note that not all flights in the first lane go on to the second lane, since they may take a different next lane at a lane exit.

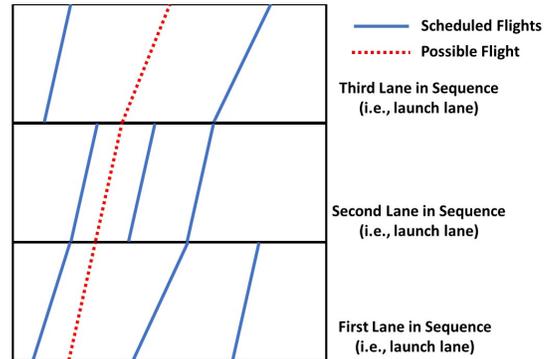


Fig. 5. Example of Lane Scheduling. Blue line segments are already scheduled flights, and the red dashed segments represent a possible new flight placement.

Lane deconfliction is done with the Lane Strategic Deconfliction (LSD) algorithm, and its computational complexity is proportional to the square of the number of flights scheduled in the proposed lane sequence. However, the operation that is performed is a simple lookup in a table, and is very efficient. For example, Figure 6 shows the lane structure above the East Bench area of Salt Lake City with some simulated flights (black circles) starting up the launch lanes, wherein thousands of lanes exist, and thousands of flights are scheduled and deconflicted.

### IV. EXPERIMENTS

Given the two approaches described above, which we call FNSD and LSD, the goal is to perform simulation experi-

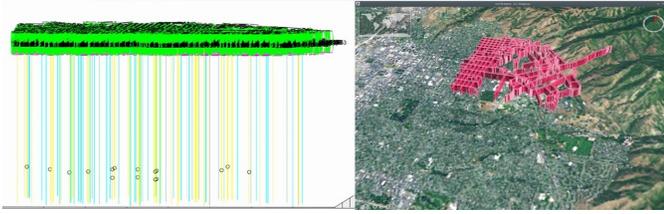


Fig. 6. Example of Large-Scale Lane Structure above East Bench of Salt Lake City, UT. (a) Set of Simulated Flights (Black Circles) Starting up the Launch Lanes. (b) GIS View of Lanes over East Bench Area.

ments to better understand their respective advantages and disadvantages. To achieve this, some performance measures are defined. In addition to providing comparison metrics for the two methods, a set of measures for evaluation of different lane-based strategies is also described. Both of these are studied in terms of the following framework. A 100x100 unit area is considered, where 1 unit corresponds roughly to 10 feet. For the lane based system, a 6x6 grid of ground locations is defined, and the subsequent airways based on that; nearest eight-neighbors are connected, and every ground vertex has both launch and land lanes. For the FAA-NASA flights, a 5x5 grid is defined (i.e., grid elements are 20x20 square units). UAS speeds are in the interval [0.1,0.31], as these correspond to 3-10 mph. The altitude for the lanes is between 10-12 units, while for the FAA-NASA flights, it is set to 11 units. All flights are specified as between two ground vertexes, and lane-based flights take a shortest route the the lanes, while the FAA-NASA flights follow a 3-polyline trajectory of up, over and down. Note that this makes these latter routes shorter than the lane-based routes. A set of 1,000 flights is scheduled in each scenario; however, if the deconffliction takes more than 30 seconds for some flight, then data from the first 75 flights is used to interpolate a result for all 1,000 flights.

#### A. FAA-NASA vs. Lane-Based

Within this context, we consider three scenarios:

- *Scenario 1*: The launch and land ground vertexes are selected to be the two most distant (i.e., 1 and 36). Figure 7 shows the lane-based and direct routes for this. The speed of every flight is fixed to be 0.12, and all flights follow the same trajectory. The start times interval for these flights is [0,2000]; that is, a flight should be assigned the earliest possible launch time in this interval. Finally, the minimum headway distance is set at 1 unit.
- *Scenario 2*: This is the same as Scenario 1, except that the launch and land ground vertexes are chosen randomly.
- *Scenario 3*: In this scenario, the launch and land ground vertexes are chosen randomly, as are the UAS speeds, and the start times interval. Each UAS has its own speed which us constant across the whole flight, and the initial start time is randomly selected in the interval [0,1000].

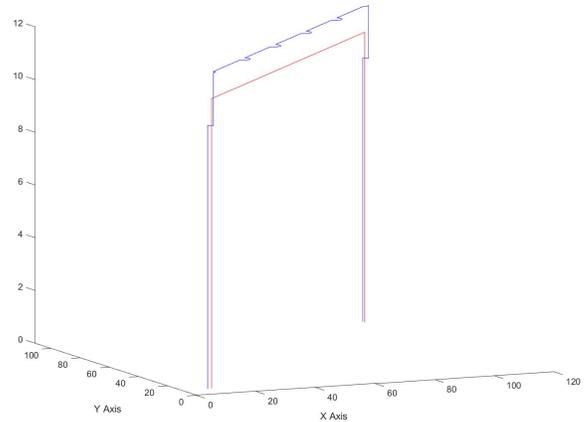


Fig. 7. The Lane-Based (blue) and FAA-NASA (red) Routes from Ground Vertex 1 to 36.

Table 1 gives the data collected from the experiments. As can be seen, the lane-based method does better in Scenarios 1 and 3, while the FAA-NASA approach performs better on average delay, but not maximum delay, on Scenario 2. Scenario 1 represents the scheduling problem on a heavily used route while Scenario 3 is more representative of a random arrival process; thus, we believe that these are more reflective of actual operational situations. Another observation is that these results are achieved in the context of all flights being nominal, that is, no contingencies occur. Since the lane-based approach has a distinct advantage with respect to contingencies, then these results indicate the overall superiority of the lane-based approach. Also, note that cost of deconffliction in the lane-based approach in Scenarios 1 and 3 is 2 to 4 orders of magnitude lower.

<i>Scenario 1</i>	LSD	FAA-NASA
Avg Delay	494.97	583.61
Max Delay	989.95	1,167.20
Avg Flight Time	1,527.00	1,318.00
Avg Comparisons	1,249.50	1.49x10 <sup>7</sup>
Avg Decon Time	0.0063	138.53
<i>Scenario 2</i>	LSD	FAA-NASA
Avg Delay	19.87	10.73
Max Delay	67.72	917.60
Avg Flight Time	879.49	705.68
Avg Comparisons	61.20	27.40
Avg Decon Time	0.0212	0.2524
<i>Scenario 3</i>	LSD	FAA-NASA
Avg Delay	1.65	16.85
Max Delay	30.39	325.20
Avg Flight Time	532.88	451.65
Avg Comparisons	49.84	28.62
Avg Decon Time	0.0014	0.3579

Another advantage of the lane-based method is that it is possible to easily visualize the flight schedules through the lanes. Let's consider an example from each scenario. First,

consider flight 10 in Scenario 1. Figure 8 shows the complete lane sequence for the flight with all other scheduled flights. Flight 10 is shown in red, and since all the flights follow the same lane sequence and have the same speed, they are all represented as parallel line segments where the lower end point represents the launch time, and the upper end point represents the landing time. It is also clear that this representation makes it easy for a flight operations center controller to visually determine if flights are off course by overlaying telemetry data on top of this graph. Figure 9 shows the corresponding Space Time Lane Diagram for Scenario 2. Here it can be seen that Flight 10 is the only flight scheduled along this specific route, but that other flights are scheduled at various times on some of the lanes in the route. Finally, Figure 10 demonstrates how readily system-wide type information is made evident by these graphs; note that the number of segments in the upper lanes (later part of the flight) indicate that there may be some congestion in that region, and it might be wise to find alternate routes so as to avoid that. The variety of slopes in the graph indicates the different speeds of the flights through the lanes.

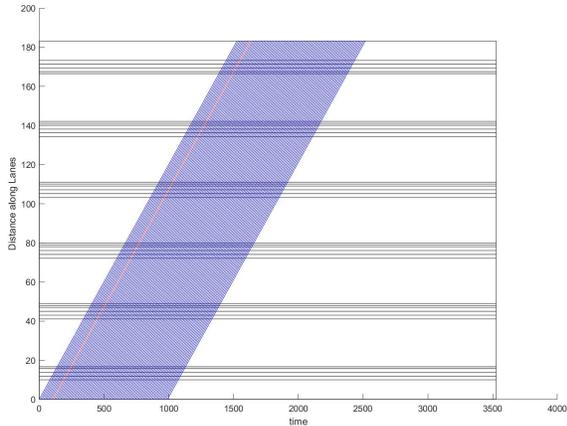


Fig. 8. The Stacked Set of Space Time Lane Diagrams for Flight 10 in Scenario 1.

### B. Lane Stream Properties

We now define properties specific to the lane-based approach. To do so, 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. Note that  $h_x = s \cdot h_t$  is the headway distance. The maximum number of UAS possible in the lane at one time,  $n_{max}^t$ , is then:

$$n_{max}^t \equiv \lfloor \frac{d}{s \cdot h_t} \rfloor + 1$$

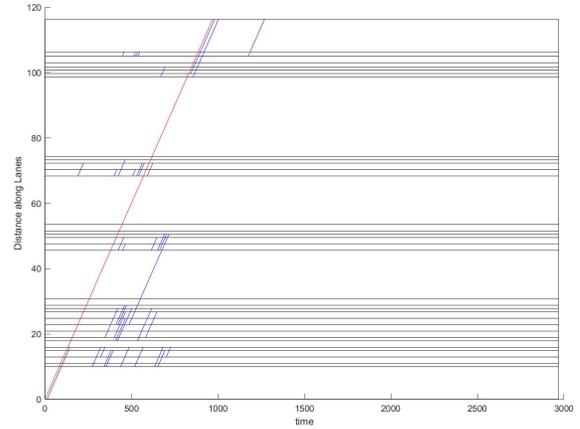


Fig. 9. The Stacked Set of Space Time Lane Diagrams for Flight 10 in Scenario 2.

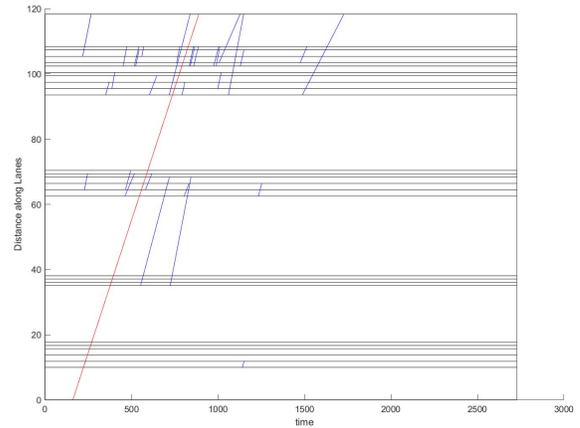


Fig. 10. The Stacked Set of Space Time Lane Diagrams for Flight 10 in Scenario 3.

Clearly, achieving  $n_{max}^t$  depends on obtaining a perfectly packed requested start time sequence.

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:

$$\Theta_t(\mathcal{A}) \equiv \frac{\mu_{\mathcal{A}}}{n_{max}^t}$$

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 [5], [6], [7], 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 [8] 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:

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

Finally, *spatial flow*,  $q_s(t, \mathcal{A})$ , 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.

These measures are given as a means of comparing the effectiveness of alternative lane scheduling algorithms. Since that problem is not addressed here where we compare the FAA-NASA approach to lanes, we simply give the values for these measures for Scenario 1, where the flights are most densely packed. The following values result for Senario 1 for the launch lane:

$$n_{max}^t = 8$$

since  $h_x = 1.41$

$$\Theta_t(\mathcal{A}) = \frac{8}{8} = 1$$

looking at the time interval  $[0, 83.333]$  (since  $\mu_{\mathcal{A}} = 8$  and  $83.333 = \frac{10}{0.12}$ ).

$$k_s(t, \mathcal{A}) = \frac{8}{10} = 0.8$$

$$\Theta_s(t, \mathcal{A}) = \frac{1 \cdot 8}{10} = 0.8$$

$$q_s(t, \mathcal{A}) = 0.8 \cdot 0.12 = 0.096$$

## V. CONCLUSIONS

A direct comparison of performance characteristics has been made between the FAA-NASA and lane-based UAM approaches. A variety of scenarios were examined, and measures defined on the computational and other requirements over a set of flights. The lane-based method was found to outperform the FAA-NASA approach in the most likely actual conditions which will be encountered in large-scale UAS traffic management. Although the lane-based method requires flights of slightly longer route, there are multiple advantages in terms of management.

In future work, we intend to explore the use of Agent Based Modeling and Simulation (ABMS) to determine more optimal UAM parameters related to lane properties and their layout, as well as lane speeds, and auxiliary lane support structures (e.g., emergency lanes alongside regular lanes, emergency landing lanes, etc. In addition, we are looking into real-time adaptive lane scheduling by the UAS themselves. This may be particularly useful locally in contingency situations. We are exploring the formal verification of the safety

aspects of such protocols. Finally, we are working with the Utah Department of Transportation to realize a version of lane-based UAS traffic management in urban regions (e.g., the Salt Lake City Valley) in order to effectively meet the challenge of large-scale UAS deployment for deliveries and other services.

## REFERENCES

- [1] J. Rios, "UAS Traffic Management (UTM) Project Strategic Deconfliction: System Requirements Final Report," NASA, Moffet Field, CA, Tech. Rep. NASA Report, July 2018.
- [2] D. Sacharny and T. Henderson, "A Lane-based Approach for Large-scale Strategic Conflict Management for UAS Service Suppliers," in *IEEE International Conference on Unmanned Aerial Systems*, Atlanta, GA, June 2019.
- [3] T. C. Henderson, D. Sacharny, and M. Cline, "An Efficient Strategic Deconfliction Algorithm for Lane-Based Large-Scale UAV Flight Planning," University of Utah, Salt Lake City, UT, Tech. Rep. UUCS-19-005, September 2019.
- [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] 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.
- [6] M. Clay and N. Simanyi, "Renyi's Parking Problem Revisited," *Stochastics and Dynamics*, vol. 16, no. 2, June 2014.
- [7] M. Gargano, A. Weissenseel, J. Malerba, and M. Lewinter, "Discrete Renyi Parking Constants," Pace University, New York, NY, Tech. Rep., May 2005.
- [8] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Germany: Springer Verlag, 2013.