Large-Scale UAS Traffic Management (UTM) Structure

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

Abstract— **The advent of large-scale UAS exploitation for urban tasks, such as delivery, has led to a great deal of research and development in the UAS Traffic Management (UTM) domain. The general approach at this time is to define a grid network for the area of operation, and then have UAS Service Suppliers (USS) pairwise deconflict any overlapping grid elements for their flights. Moreover, this analysis is performed on arbitrary flight paths through the airspace, and thus may impose a substantial computational burden in order to ensure strategic deconfliction (that is, no two flights are ever closer than the minimum required separation). However, the biggest drawback to this approach is the impact of contingencies on UTM operations. For example, if one UAS slows down, or goes off course, then strategic deconfliction is no longer guaranteed, and this can have a disastrous snowballing effect on a large number of flights. We propose a lane-based approach which not only allows a one-dimensional strategic deconfliction method, but provides structural support for alternative contingency handling methods with minimal impact on the overall UTM system. Methods for lane creation, path assignment through lanes, flight strategic deconfliction, and contingency handling are provided here.**

I. INTRODUCTION AND BACKGROUND

The goal of industry and governing agencies is to provide a safe, effective and efficient Unmanned Aircraft System (UAS) Traffic Management (UTM) system in order to achieve (see [1]):

- beyond visual line of sight operations
- air traffic within high density urban environments
- autonomous vehicle-to-vehicle (V2V) internet connected UAS
- large-scale contingency mitigation, and
- news gathering, deliveries and personal use services.

To date, most UTM developers, e.g., AirMap [2], have focused on UAS registry, GIS, flight communications, traffic monitoring, and user interfaces. However, all of this has been positioned within the FAA-NASA strategic deconfliction framework [3] based on a geographic grid layout wherein new flights must be pairwise deconflicted with all scheduled flights in common grid elements. Smith et al. state [4]:

> When a UAS Operator submits a flight geography, USS adds a conformance buffer and computes a conformance geography. The operation is expected to stay within the conformance geography at all times; a violation results in automated actions such as alerts. USS also computes its protected geography. Proposal Handler uses this geography for de-conflicting operation plans and constraints. Protected geographies from separate operations should never overlap in space and time.

NASA states [3]:

> 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.

Such 4D trajectory deconfliction is, in general, PSPACE hard, and therefore it is vital to consider the computational consequences of the selected method. A major problem for the current approach 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. This 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 [5] 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 [6]:

> 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 UTM structure proposed here allows this development.

Previous work on lane related approaches has mostly been confined to manned aircraft. Devasia et al. [7] 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 (ATCS) works to maintain safety locally. The authors propose a token-based system for entry into a designated area. These

[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

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. [8] 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 [9] extend CRP to consider turn rate limitations when routes are modified.

II. LANE-BASED UAS TRAFFIC MANAGEMENT

A UTM is described which supports efficient and effective:

- *lane creation*: the definition of a set of lanes (airways) to allow flight from one ground location to another.
- *flight path determination*: given launch and land ground locations, return a sequence of lanes which goes from the launch location to the land location.
- *flight reservations*: given a lane sequence defined by a flight path, and a time interval of possible launch times, find the set of possible launch times that stay safely separated from any scheduled flights.
- *contingency mitigation*: lanes may be either pre-defined (e.g., emergency side lanes) or dynamically created (e.g., emergency landing lanes) in order to handle real-time departures from nominal flight paths.

In addition to the description of these, we provide a set of MATLAB functions to deliver these capabilities; these can be found at http://www.cs.utah.edu/ tch/notes/UAM.

A. Lane Creation

Airways are defined by giving a set of ground locations and edges between them. For example, this may be directly obtained from GIS data by finding roads (the edge) and their intersection points (the vertexes), or by manual specification of the desired locations and their connectivity. In urban environments, it may be desirable to locate the airways above roads. For example, the Utah Department of Transportation, Aeronautics Division, which is developing the UTM system in Utah, wants airways above roadways since these are public spaces, and a great deal of infrastructure is already in place on the roadways to support UTM operations (existing access to power and networking for new radar and GPS systems, etc.). Moreover, NASA supports this idea [10]:

> 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 social concerns.

However, the lane-based approach does not require that airways be placed about roadways.

Thus, let $V = \{x_i, y_i\}$ be the ground vertexes and $E = \{i, j\}$ be the edges where i and j are indexes into V. Launch and land vertexes must be specified (as indexes into V). Other required information includes the upper and lower altitudes for airway lanes, as well as a minimum length lane for roundabout structures. The airway constructed from this data defines the 3D vertexes created for the airway lanes, the airway lanes (as directed 3D line segments), and the indexes into the launch and land lanes.

All airway lanes are one-way, and in order to allow two-way movement between ground vertexes requires two separate lanes. These are separated vertically at some safe distance. Roundabouts are created at intersections to allow flights to choose outbound directions from a vertex. Thus, there are 3 types of basic lanes: (1) launch/land lanes, (2) roundabout lanes, and (3) between ground vertex lanes. Other types of lanes may be introduced for contingency handling. We first describe roundabouts as other lanes all connect to them.

B. Roundabouts

In order for UAS traffic to safely move through intersections, we propose to use roundabouts (see Figure 1). A

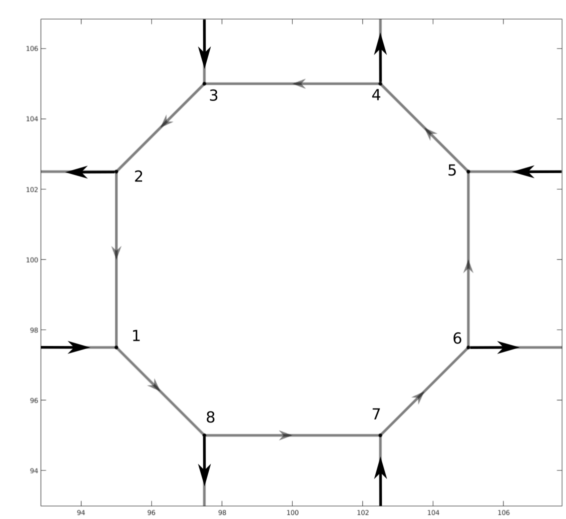

Fig. 1. An Example 4-Way Airway Roundabout.

roundabout is created directly above each ground vertex and consists of a set of points located on a circle centered above the ground vertex. The radius of the circle is chosen so as to ensure a minimal length for roundabout lanes. There will be a roundabout vertex corresponding to each edge from the associated ground vertex, as well as vertexes for any launch or land lanes. Two such roundabouts are positioned above each ground vertex to facilitate bi-directional movement between ground vertexes. Figure 2 shows an example lane layout. Note that the convention used here for choosing between ground vertex airway directions is that all travel in directions with angle in the range $[0, \pi)$ is at the upper altitude, while those in the range $[\pi, 2\pi)$ are at the lower altitude. Also, note that there are lanes connecting the two vertically separated roundabouts above a ground vertex. Finally, it is important to note that all vertexes in the graph must have either in-degree 1, or out-degree 1 in order to exclude UAS conflicts at a node; thus, all conflicts must occur in the lanes.

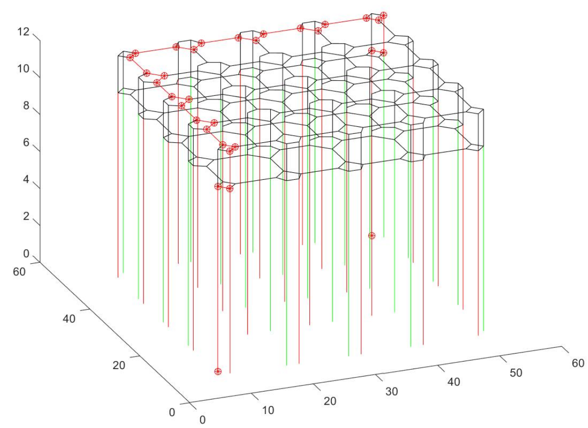

Fig. 2. An Example 5x5 Grid Airway with Launch and Land Lanes to Every Ground Vertex. A Flight Path is shown in red from Ground Vertex 1 to Ground Vertex 25.

III. FLIGHT PATH DETERMINATION

Given an airway structure and the launch and land ground vertexes for a proposed flight, then a sequence of lanes, $(s_1, s_2, \ldots, s_n)$, must be found that starts at the launch vertex and ends at the land vertex. This is achieved at the moment with an A^* algorithm using lane length as the cost, and it finds a lowest cost path through the lanes. One issue to be addressed is the need to develop a variety of flight sequences between the launch and land vertexes; e.g., there may be multiple shortest paths due to the grid nature of the airways, or it may be useful to incorporate costs for weather, closed lanes, system congestion, etc. Figure 2 shows a path (in red) going from ground vertex 1 to ground vertex 25.

IV. LANE-BASED STRATEGIC DECONFLICTION

We propose a flight reservation system which given a proposed flight plan consisting of a sequence of lanes, and the speed in each lane, chooses a launch time that maintains safe separation (termed "headway" within lanes) at all times between the proposed flight and already scheduled flights. This process is based on the use of the Space Time Lane Diagram (see Figure 3). The x-axis is time, and the y-axis is distance along the lane (such diagrams are used for ground highway traffic analysis). Figure 3a shows the flight trajectories through the lane of the earliest (q_1) and latest (q_2) lane entry times for a proposed flight. Figure 3b shows the trajectory of an already scheduled flight (with start time $t_{i,1}$), as well as its two headway trajectory constraints (with lane entry times p_1 and p_2, respectively). To ensure safety, the trajectory of the proposed flight may not cross the headway constrained parallelpiped.

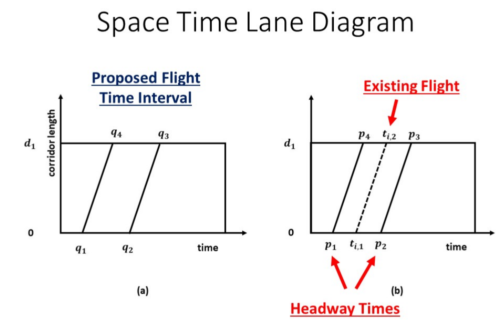

Fig. 3. The Space Time Lane Diagram. (a) Trajectories through the Lane of the Proposed Flight Earliest and Latest Possible Entry Times. (b) Trajectory of Already Scheduled Flight, and its Associated Headway Constraint Trajectories.

To perform strategic deconfliction, we next assign labels to the intervals defined by the proposed flight trajectories (see Figure 4). This labeling strategy makes the combinatorics

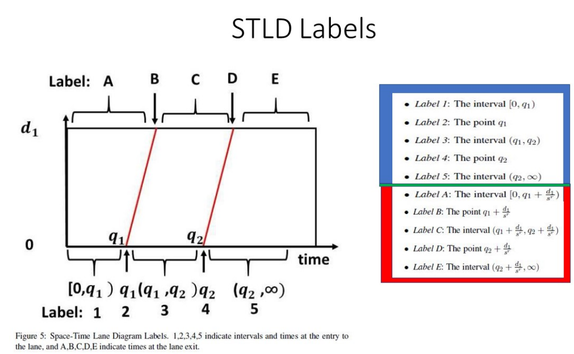

Fig. 4. Entry-Exit Labels for Flight Trajectories.

explicit, while also providing for a practical program structure. At the entry to the lane, the interval up to q_1 is labeled '1'; the point q_2 is labeled '2'; the open interval (q_1, q_2) is '3'; the point q_2' is '4'; and the interval from q_2 onward is '5.' The lane exit point labels are similarly defined based on the lane exit times, and are 'A' through 'E,' respectively.

To determine if a conflict exists between the proposed flight time through the lane and an existing flight, the two headway constraint trajectories are each given a label pair assignment based on the location of the trajectory endpoints with respect to the labels. Thus, if the start time in the lane is to the left of q_1, and the exit time to the left of q_4, then that is labeled '1A.' Likewise a label pair is assigned to the other constraint trajectory. These four labels (or two label pairs) suffice to determine any possible entry time intervals for the proposed flight. Figure 5 shows some example cases of this. In the figure, case 1 shows a scheduled flight which

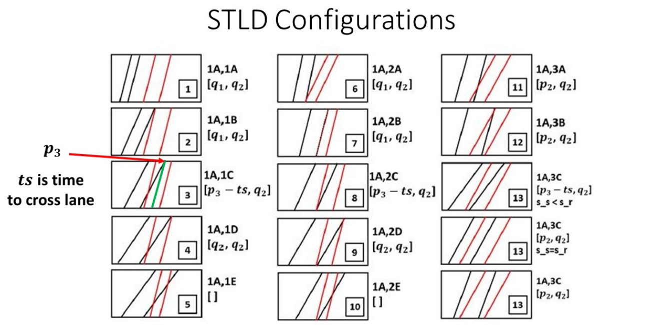

Fig. 5. Example Cases of Flight Trajectory – Proposed Flight Interval Results.

takes place much earlier and does not intersect the proposed entry interval, and thus, the remaining possible interval is just the original interval. Case 3, on the other hand, shows that the rightmost headway constraint crosses into the proposed interval; the green line shows the earliest possible entry time for the proposed flight, resulting in a reduced interval $[p_3 - ts, q_2]$, where ts is the time for the proposed flight to cross the lane. It turns out that there are only 139 possible cases (see Figure 6), and thus, the label pairs serve as indexes to produce the possible entry times into the lane.

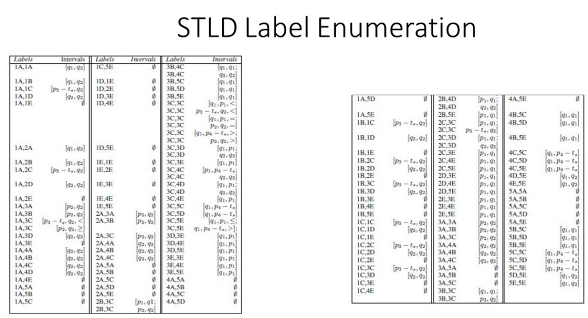

Fig. 6. Enumeration of all 139 Cases of Flight Interactions and Resulting Possible Time Entry Interval.

To determine the possible launch times for a proposed flight, it is necessary to apply the above process to each flight in the launch lane which produces a set of possible intervals. Each of these intervals is then in turn applied to the next lane in the sequence, which may further reduce the number of possible entry intervals, and so on, to the landing lane. If a non-empty set of intervals results after the entire lane sequence is considered, then a launch time may be selected from that, and the flight is deconflicted for the entire flight.

The Lane Strategic Deconfliction (LSD) Algorithm is:

Algorithm LSD (Lane 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($p^c_{f,1}, p^c_{f,2}, s^c_f, t_1, t_2, s^r, h_t$)
      f_int $\leftarrow$ get_interval(label,$p^c_{f,1}, p^c_{f,2}, s^c_f, t_1, t_2, s^r, h_t$)
      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

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

$$n_c = f_1 + (f_2 f_1 + f_2) + (f_3 f_2 + f_3 f_1 + f_3)...$$

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

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, 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 $O(f^2)$.

V. CONTINGENCY HANDLING

Contingencies represent the safety and logistics issues that arise in real-world systems, serving major design and operational constraints. They also represent a considerable computational challenge, since the representation and planning for all possible contingencies is generally intractable. In fact, contingencies are the primary impediment to the large-scale integration of autonomous systems.

The lane-based approach offers several ways to mitigate contingencies that arise during a UAS flight. However, the main benefit stems from the ability to perform a cogent analysis of contingencies, avoiding the complexity of the free-flight design. For an example of how the lane-based approach supports contingency analysis, consider the single-lane example in Figure 7 (adapted from [11]). In this model,

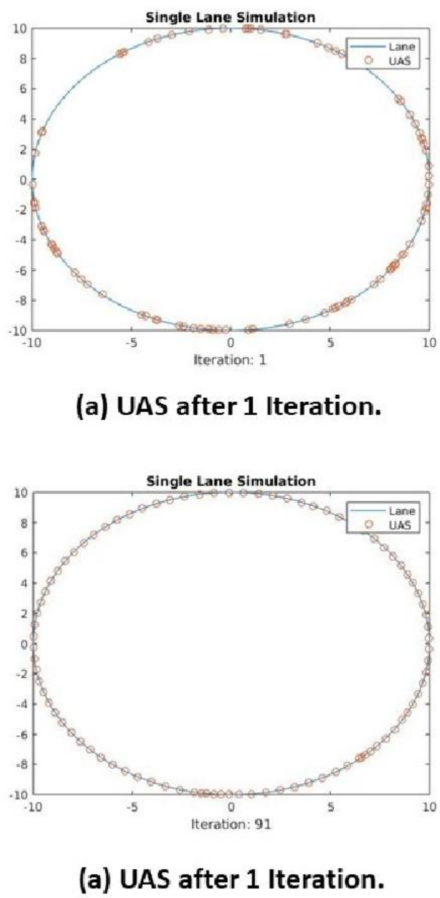

Fig. 7. Single Lane Contingency Effects.

the lane is a one-dimensional curve represented by an array of length L. Each element of the array can be in one of seven states: it may be empty, or occupied by a UAS having an integer speed from 0 to 5. The speed value represents the number of array elements that the UAS moves forward in the next step of the simulation. The behavior of each UAS is defined as follows, calculated at each step of the simulation simultaneously for all UAS (*gap* is the number of unoccupied array elements in front of the UAS):

1) **Acceleration:** Each vehicle with speed $v < v_{max}$ and $gap \geq v + 1$ gets speed $v \leftarrow v + 1$
2) **Deceleration:** Each vehicle with $gap \leq v - 1$ gets speed $v \leftarrow gap$
3) **Move:** Each vehicle moves forward v elements

The simulation in Figure 7 begins with 85 UAS placed randomly across the lane. Here, a contingency is defined as a deceleration event (item 2 in the behavior described above). At iteration 500 a contingency is forced on a number of UAS in the lane. Figure 8 shows aggregated contingency events for 10 runs of the simulation (each color is a different run) with 85 and 100 UAS (Figures 8a and b, respectively). The figures show two dramatically different system responses: in Figure 8a the number of contingencies returns to a settled value before the forced contingency, while in Figure 8b each run produces a different and hence unpredictable[1] outcome.

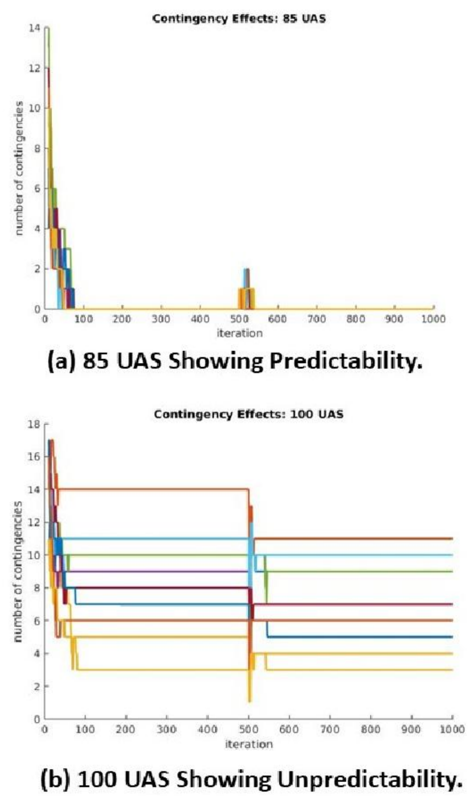

Fig. 8. Contingency Effects on a System State and Individual Behavior in a Lane.

The lane-based approach provides a single control variable to account for these two scenarios, the density of UAS in the lane structure. A more complex individual behavior could potentially guard against such unpredictable effects, but with this approach the trade-offs are made explicit for

[1]As noted by Nagel and Rasmussen, this model can be treated analytically [12], but the analytical results are "more difficult to obtain" than measurements from simulation [11].

the policy maker. Other mitigation strategies include pre-calculated emergency paths, prioritization schemes and local real-time lane scheduling protocols.

VI. EXPERIMENTS

The lane-based approach has been tested on a number of scenarios. Figure 9 shows a set of airway lanes above part of Salt Lake City which has been used to help determine UAS Air Mobility structure and parameters for the Utah UTM. The red spheres in the airways represent UAS flights scheduled through the lanes using the LSD algorithm. The goal is to run simulations to determine the optimal placement of launch and land sites, as well as airway lanes. Simulations involving large numbers of UAS have also been run as shown in Figure 10.

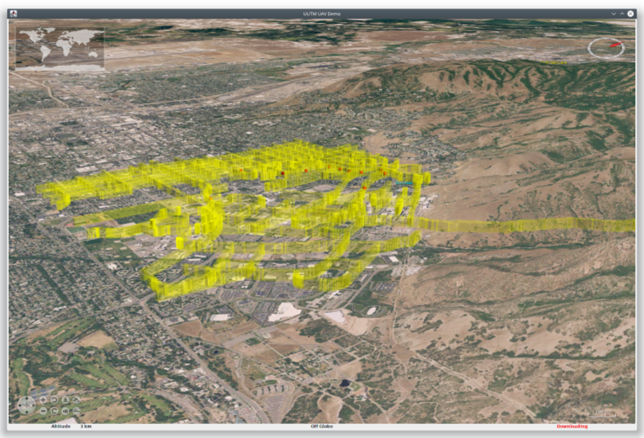

Fig. 9. Set of Airway Lanes Derived from GIS Data for Salt Lake City, UT.

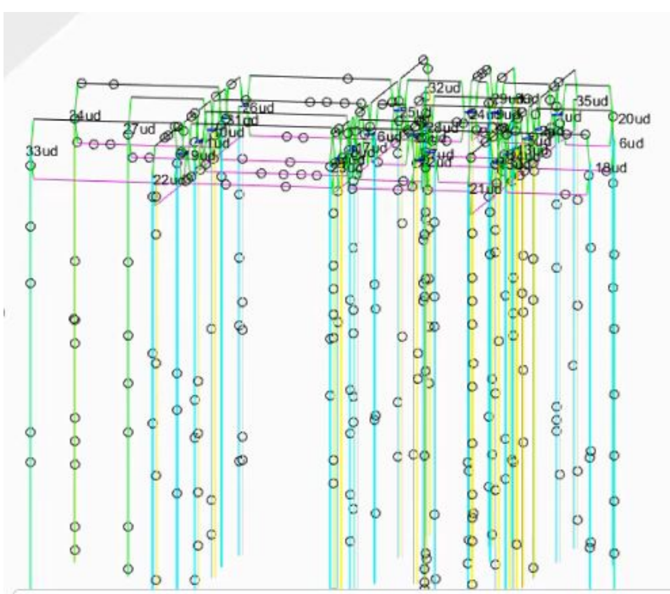

Fig. 10. One thousand Strategiclly Deconflicted UAS Flights through a Small Network.

VII. CONCLUSIONS AND FUTURE WORK

We have demonstrated the efficiency and effectiveness of a lane-based approach to large-scale UAS Traffic Management. A methodology is described for the creation of the lane structure, as well as for efficient path selection and strategic deconfliction. Moreover, we have indicated how the lane-based approach supports contingency mitigation. All of these are far superior to the current FAA-NASA arbitrary flight path creation and deconfliction approach.

There are a number of things to consider in future work:

- dynamic lane creation and deletion
- UTM parameter optimization (e.g., lane speeds, lane connectivity, airway volume around lane segments, etc.)
- inclusion of weather, congestion, and other parameters for path selection
- dynamic UAS flight parameters and path selection
- role of communications in UAS flight path planning (e.g., connectivity of UAS, relay support for communications outages, etc.).

We are currently developing methods for several of these.

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] AirMap, "Five Critical Enablers or Safe, Efficient, and Viable UAS Traffic Management (UTM)," AirMap, Santa Monica, CA, Tech. Rep., January 2018.

[3] 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.

[4] I. Smith, J. Rios, P. McGuirk, D. Mulfinger, P. Ventatesan, D. Smith, V. Baskaran, and L. Wang, "UTM TCL 2 Software Requirements," NASA, Moffet Field, CA, Tech. Rep. NASA/TM-2017-219513, April 2017.

[5] 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.

[6] J. Barbagello, "Instrument Procedures Handbook," Federal Aviation Administration, Washington, DC, Tech. Rep. FAA-H-8083-16B, August 2017.

[7] 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.

[8] 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.

[9] J. Yoo and S. Devasia, "Provably Safe Conflict Resolution with Bounded Turn Rate for Air Traffic Control," *IEEE-T on Control Systems Technology*, vol. 21, no. 6, November 2013.

[10] 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.

[11] K. Nagel and S. Rasmussed, "Traffic at the Edge of Chaos," in *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living systems*, R. Brooks and P. Maes, Eds., Boston, MA, July 1994, pp. 222–235.

[12] A. Schadschneider and M. Schreckenberg, "Cellular Automation Models and Traffic Flow," *Journal of Physics A: Mathematical and General*, vol. 26, no. 15, pp. L679–L683, August 1993.