Centralized and Distributed UTM in Layered Airspace

Leonid Sedov and Valentin Polishchuk
Communications and Transport Systems
ITN, Linköping University, Sweden
firstname.lastname@liu.se

Abstract—**We investigate strategies for management of conflicts among autonomous unmanned aerial vehicles (UAVs) in high-density very low level (VLL) uncontrolled airspace. We consider deconfliction procedures which do not involve horizontal maneuvers, and study two types of airspace structure: single- and multi-layered airspace. We compare different deconfliction paradigms by simulating a busy day of operations over a geographical area. Our main contributions are proposal of new deconfliction schemes for UAVs and assessment of the single- and multi-layered airspace designs. Our work aims to provide regulators and policy-makers with a framework for choosing between resolution strategies for the airspace management.**

I. Introduction

Fast growth of the unmanned aerial vehicles (UAVs) market calls for an effective unmanned traffic management (UTM) system. One immediate task in UTM research is to investigate various airspace organization options and management approaches, in particular, choosing between the centralized and decentralized systems.

While management of drone traffic bears close resemblance to the conventional air traffic management (ATM), the standard ATM approaches may have limited applicability in unmanned traffic management (UTM). One reason is that ATM research focuses mostly on pre-planned airport-to-airport flights whose schedule is known in advance. In contrast, small Unmanned Aircraft Systems (sUAS) traffic is of highly non-deterministic nature, as sUAS operators demands are highly variable and, in addition, a drone flight origin and destination might be virtually anywhere (to quote Dr. Parimal Kopardekar, ”every home will have a drone and every home will serve as an aerodrome” [1]). This unpredictability questions the feasibility and effectiveness of a centralized (”all-know-all”) management entity with perfect situational knowledge and motivates investigating whether unsophisticated (”rules-of-way”) distributed systems could be sufficient for an estimated traffic density and UAVs technological capabilities.

A. Contributions

In this paper we study different deconfliction schemes for management of autonomous sUAS in very low level (VLL) uncontrolled airspace and estimate the capacity under the various approaches to airspace management. We consider two designs of the airspace: the single-layer airspace, where all vehicles occupy the same altitude, and the multi-layer airspace. The latter design is based on the assumption that UAVs have good altimetry capabilities. For each airspace design we compare centralized and decentralized approaches.

TABLE I
Resolution approaches

Ground delay	Sec II-A1	Centralized
Hovering	Sec II-A2	Distributed
Layer assignment	Sec II-B1	Centralized
On-demand descent	Sec II-B2	Distributed

Among our results are the following:

- For the single-layer design we compare the efficiency of ground delay (centralized approach) and hovering strategy (decentralized approach).
- For the multi-layered airspace we consider allocation of UAVs between layers (centralized approach) vs. a simple algorithm based on the descending to the layer below when needed (decentralized approach).

Table I lists the resolution strategies evaluated in the paper.

The rest of the paper is organized as follows. In the reminder of this section we review earlier work. Section II provides details of the resolution algorithms for different airspace configurations. In Section III we report on the simulation setup and experimental results. Finally, in Section IV we draw up conclusion on the results of the paper.

B. Related work

Stochastic modeling has a long history of application to analysis of air traffic organization. Hoekstra in his PhD thesis [2] described the probabilistic setup where aircraft positions are chosen uniformly at random in a given airspace and the aircraft fly with the same speed, heading uniformly distributed directions. The model was further developed by Jardin [3] and explored within the Metropolis project by researchers from TU Delft [4]. In [5] this model is called the *Dutch model*.

An extension of the Dutch model is *Cal model* (the name ”Cal” was chosen in [6] because the model was introduced at ICRAT 2016 by researchers representing University of California Berkeley, going by Cal). In the model, VTOL (vertical takeoff and landing) flights follow straightline segments elevated to the fixed altitude h, with endpoints sampled according to population density.

The setups in both Dutch and Cal models are essentially two-dimensional.

In [4], [7]–[9], the *Layers concept* was studied, in which the airspace is divided into vertically stacked bands and the allowed heading range for aircraft within one layer is restricted. Differently from the earlier heading-based layer assignments, in which some conflicts still remain within the layers, we suggest splitting into *completely* conflict-free layers (Section II-B1) using graph coloring techniques [10] (aiming at minimizing the number of layers). Jang et al. [11] suggested using the layers above the skyline of buildings for fixed-wing and high-speed vehicles whose navigation equipage has limited usability between high buildings, and the layers below the skyline for aircraft with good navigation systems and ability to hover (we use hovering in Section II-A2).

Conflict detection and resolution is one of the most widely studied problems in ATM and UTM (see, e.g., the review [12], and [13], [14] for some recent work); in particular Barnier and Allignol studied GDP for conflict resolution [15]. Muñoz et al. presented a reference implementation of NASA's detect and avoid concept for integration of UAS into the national airspace (NAS), named DAIDALUS [16]. In [5] Bulusu et al. studied a resolution algorithm where UAVs sink to lower level (we explore on-demand descending in Section II-B2; differently from [5], we descend to more than one layer).

Policy makers propose initiatives for building centralized UTM systems [17]–[22]. A distributed protocol for conflict management was suggested in [23] (we study distributed schemes in Sections II-A2 and II-B2).

II. The traffic management paradigms

This section describes the airspace designs and deconfliction strategies studied. We model drones as disks of a given radius r representing the uncertainty zones around the drones. The trajectory of a drone d is a list $d.S$ of consecutive flight segments. If a drone never takes part in a deconfliciton maneuver, the flight directly connects ($|d.S| = 1$) the origin and the destination (Fig. 1, top left).

We define a conflict as an overlapping of two disks at the same altitude. The drones are prioritized based on the start times t_s of the first segment of the flights, i.e. a UAV with the earlier departure time has the priority over UAVs that start their flights later. Whenever two drones are in conflict, the lower-priority drone is responsible for avoiding the conflict.

A. Single-layer airspace design

First we describe algorithms for UAVs deconfliction in an airspace consisting of only one layer (all drones occupy the same altitude during the en-route portion of the flight, as in the Cal model [24]). Each flight segment is a quadruple (s_s, s_e, t_s, t_e), where $s_s, s_e \in \Re^2$ are the the segment's start and endpoint coordinates, and t_s, t_e are the planned times when the drone enters and leaves the segment.

In the following subsections we present the deconfliction algorithms for centralized and decentralized UTM.

1) Ground Delay (Centralized system): Conflict avoidance in a centralized UTM employs the ground delay paradigm (as envisioned in the UTM ConOps'es [17]–[22]). It assumes that the management body has full knowledge of all past and current operations plans and the state of all UAVs at any instant.

The pseudocode for drone flightplan scheduling is presented in Algorithm 1, where $\text{HMD}(d_1, d_2)$ is the *horizontal miss distance* between drones d_1 and d_2 [25] (the distance at the closest point of approach). Whenever an sUAS operator plans to start a flight, the operator files the flightplan to the UTM system. The system checks the flightplan for potential conflicts. If operating according to the flightplan would lead to a loss of separation, the system delays the drone departure by the time τ such that for the delayed drone no conflict is expected, i.e. a drone $d = \{(s_s, s_e, t_s, t_e)\}$ becomes $d^* = \{(s_s, s_e, t_s + \tau, t_e + \tau)\}$. Note that after applying the ground hold, the number of flight segments does not change $|d.S| = |d^*.S| = 1$.

Algorithm 1: Launch drone with Ground Delay Program

Input : A set of launched UAVs L and the planned drone d
Output: The delayed drone d and the updated set L

```
repeat
    conflicting_drone ← ∅;
    is_any_conflict = False ;
    foreach drone l ∈ L do
        if HMD(d, l) ≤ 2r then
            is_any_conflict ← True;
            conflicting_drone ← l;
        end
    end
    if is_any_conflict then
        τ ← delay time needed for d to avoid
          conflicting_drone ;
        delay drone d by τ ;
    end
until is_any_conflict = False;
L ← L ∪ d;
```

GDP may lead to excessive delay times. In the extreme scenario, if two UAVs head directly towards each other, then low-priority drone will be delayed until the high-priority drone ends its flight or passes the low-priority drone origin.

2) Hovering (Decentralized system): For the decentralized approach we employ the hovering capabilities of drones. UAV departures are not delayed, but instead the drones wait in the air by hovering on a lower altitude until the current conflict is resolved (Fig. 1, top right). Note that in order to keep the model as simple as possible, we assume that the vertical movements are instant, i.e. we do not consider the exact trajectory of the maneuver (the time needed for the diving is included into the safety zone radius r); if needed, this can be modeled using an approach from [5].

Specifically, we assume that drones have V2V capabilities to recognize the loss of separation during the flight. When

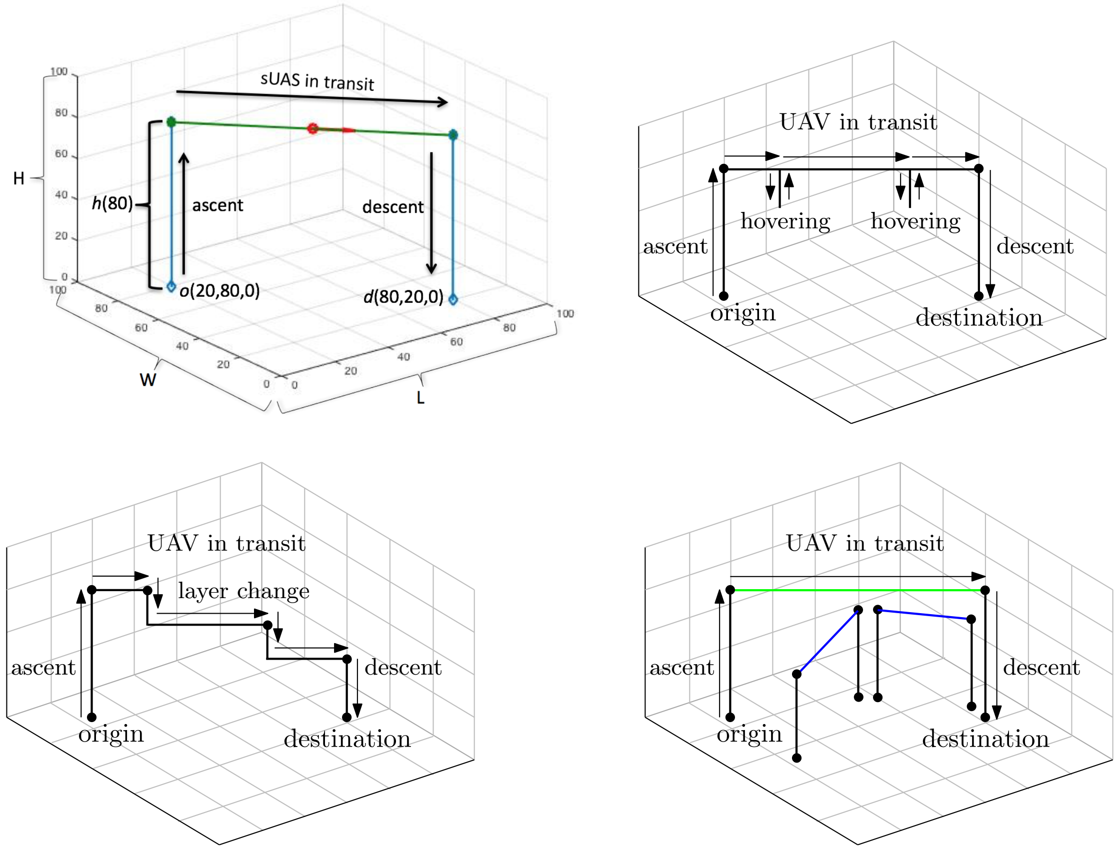

Fig. 1. UAS flight paths. Top left: GDP ([24, Fig. 3] – Cal model). Top right: with hovering maneuvers. Bottom left: with on-demand descending maneuvers. Bottom right: with layer assignment.

two disks overlap, the low-priority UAV descends and waits on the lower altitude until the conflict is eliminated. Hence a drone $d = \{(s_{s_1}, s_{e_1}, t_{s_1}, t_{e_1}), \ldots, (s_{s_n}, s_{e_n}, t_{s_n}, t_{e_n})\}$ that hovers τ seconds starting at time t_c at the point with coordinates $c \in \Re^2$ becomes $d^* = \{(s_{s_1}, s_{e_1}, t_{s_1}, t_{e_1}), \ldots, (s_{s_n}, c, t_{s_n}, t_c), (c, s_{e_n}, t_c + \tau, t_{e_n} + \tau)\}$ (the number of segments increases by 1, $|d^*.S| = |d.S| + 1$).

One of the delimitations of this deconfliction strategy is that waiting in the air leads to additional power consumption; a significant hovering time could even lead to the need in emergency landing due to the lack of charge (this issue could be addressed by giving higher priority to drones with low energy level).

B. *Multi-layer design*

The algorithms, discussed in this section, work with the airspace divided into layers (also called "vertical bands" in the Layers concept proposed in [4]). We assume that sUAS in each layer occupy the same altitude and that all layers have the same thickness. We number the layers from the top layer $l = 0$ to the bottom one; thus, we append an additional parameter—the layer number l—to the flight segments. We remark that the number of layers may be limited due to a variety of reasons, such as imprecise sUAS altitude meters, winds, restrictions on the uncontrolled airspace [26], noise pollution [6] and others.

The layered airspace design allows us to design deconfliction algorithms which eliminate all conflicts without a loss of efficiency. In the following subsections we present such algorithms for centralized and decentralized UTM.

1) Optimal assignment to layers (Centralized system): Here, we assume that a centralized system has full knowledge of all planned airspace operations in advance. The system assigns drones to layers so that each vertical band is conflict-free (i.e. loss of separation never occurs between UAVs within one layer). At the departure, a drone takes off directly to the assigned layer and cruises in the horizontal plane (Fig. 1, bottom right).

The problem of dividing sUAS between layers can be viewed as graph *coloring* – assignment of colors to the vertices so that no pair of adjacent vertices gets the same color. Specifically, let V be a set of drones and E be the set of drones conflicts ($(d_1, d_2) \in E$ iff d_1 and d_2 are in conflict), and let $G = (V, E)$ be the graph in which two vertices are connected iff the corresponding drones are in conflict. Then, the colors in G represent the airspace layers, i.e. an assignment of UAVs to layers, such that no two conflicting drones are assigned to the same layer. Finding the smallest number of colors to color a graph is NP-complete; however, the problem has been widely studied and many heuristic algorithms were developed [27].

Our pseudocode for assigning the drones conflict-free to layers is presented in Algorithm 2.

Algorithm 2: Assignment of drones to layers

Input : A set of planned operations O
Output: Assignment of UAVs to layers C

$V \leftarrow O$;
$E \leftarrow \emptyset$;
foreach *pair of conflicting drones* (d_1, d_2) **do**
  $E \leftarrow E \cup (d_1, d_2)$;
end
$C \leftarrow$ coloring of $G = (V, E)$;

2) On-demand descent (Decentralized system): For the decentralized layered airspace management we assume that all drones start their horizontal flights in the same top layer ($l = 0$ for all UAVs). When a loss of separation occurs, the lower-priority drone descends to the next layer below. Unlike with the hovering, once the drone has moved to the layer below it continues heading to the destination on this layer's altitude (Fig. 1, bottom left). Due to the deconfliction maneuver, the flight segment is separated into two – before and after the conflict: i.e. after a conflict encountered in a point c at time t_c, the drone $d = \{(s_{s_1}, s_{e_1}, t_{s_1}, t_{e_1}, l_1), \ldots, (s_{s_n}, s_{e_n}, t_{s_n}, t_{e_n}, l_n)\}$ becomes $d^* = \{(s_{s_1}, s_{e_1}, t_{s_1}, t_{e_1}, l_1), \ldots, (s_{s_n}, c, t_{s_n}, t_c, l_n), (c, s_{e_n}, t_c, t_{e_n}, l_n{+}1)\}$. The number of occupied layers is $\max_{d \in D}\{d.l\}$, where D is the set of all launched UAVs.

The main advantage of this strategy is that moving to the lower layers during the flight does not lead to efficiency loss because all vertical movements reduce the distance necessary to descend at the destination.

III. Simulations

To compare the centralized/distributed resolution algorithms for the airspace configurations described in the previous section, we simulated a busy day (12 hours) of traffic. The simulations were run for two geographical regions: Norrköping Municipality in Sweden and Bay Area in California, US. The total number N of expected flights during the day varied from 10 to 50000, and the safety zone radius r varied between 50 and 300m (similar to [5], [6], [23], [24], [28]–[30]). Drones velocity $|v| = 90$ km/h $= 25$m/s was constant and the same for all UAVs. We used the same samples of traffic across all our algorithms, to compare the actual performance of the algorithms, so that it is not affected by differences in traffic samples.

In the following subsections we present the simulation results for the algorithms described in Section II. Due to the page limit, in this paper we include graphs only for Bay Area. References to the figures showing the dependence of the metrics on N and r are given in Table II.

A. Single-layer design

For a single-layer airspace, conflict avoidance inevitably leads to loss of flight efficiency – the flights either get delayed (as in ground delay program) or become longer due to the resolution maneuver (as in our hovering scheme). Therefore, for this design our main performance metric is the *Extension of Travel Time* (similar to the Change in Direct Operating Cost metric proposed by Krozel et al. [31]).

Figure 2 shows the average travel time extension for Ground Delay and for Hovering. Note that the Ground Delay Program exhibits faster growth of time extension than the hovering strategy, which may suggest that the domino effect [8], [31], [32] is less pronounced in hovering than in GDP. Still, in both approaches the waiting time grows fast with the radius and the traffic density.

Figure 3 shows the percentage of drones that experienced more than 10% extension of the travel time (which may be considered as substantial loss of efficiency [5] due to waiting on the ground or airborne). Naturally, any other loss of efficiency threshold could be used; e.g., Figure 4 shows the percentage of delayed UAVs (which is the same both in the centralized ground delay scheme and in the distributed hovering strategy), i.e., the percentage of drones that had non-zero travel time extension. As expected, the number of delayed drones grows with N and r as the complexity of the traffic grows.

B. Multi-layer design

In the simulations for multi-layer airspace, we do not restrict the number of layers, in order to compare how many layers are needed for keeping the UAVs separated in centralized and distributed UTM. Figure 5 shows the number of layers used by sUAS. Surprisingly, with the distributed on-demand descending algorithm the drones occupy *fewer* layers than with the centralized layer assignment. This may be due to the centralized approach being overly restrictive as it demands that *all* layers are *completely* free of conflicts by design; on the contrary, the distributed V2V solution handles the conflicts as they arise.

Figure 6 shows the distribution of drones among layers. For the centralized drones-to-layers assignment we used the greedy graph coloring algorithms, and it is clearly visible that top layers are occupied by more vehicles than bottom layers. (Naturally this is also true for the on-demand descent algorithm.) This skew in distribution can be beneficial, for example because it decreases the noise pollution on the ground [6]. If desired, the imbalance can be addressed by algorithms that seek *balanced colorings*, as discussed below in Section IV.

We now compare our centralized layer assignment approach with the assignment based on the vehicles heading directions [4], [7]–[9] (akin to the Alternating Altitude rule for aircraft flight level assignment). In our traffic samples, we observed that there are many conflicts even among drones having similar headings: Figure 7 shows the histogram of heading angle differences $\{|\text{azimuth}(d_1.\vec{v}) - \text{azimuth}(d_2.\vec{v})| \; |d_1, d_2 \in E\}$, where E is

TABLE II
SUMMARY OF THE EVALUATED METRICS

Metric	Ground delay	Hovering	Centralized layer assignment	On-demand descent
Travel time extension	Fig. 2, left	Fig. 2, right	0	0
Percentage of drones lost > 10% time	Fig. 3, left	Fig. 3, right	0	0
Number of layers	1	1	Fig. 5, left	Fig. 5, right
Percentage of drones in vertical conflicts	0	Fig. 10, left	Fig. 10, middle	Fig. 10, right

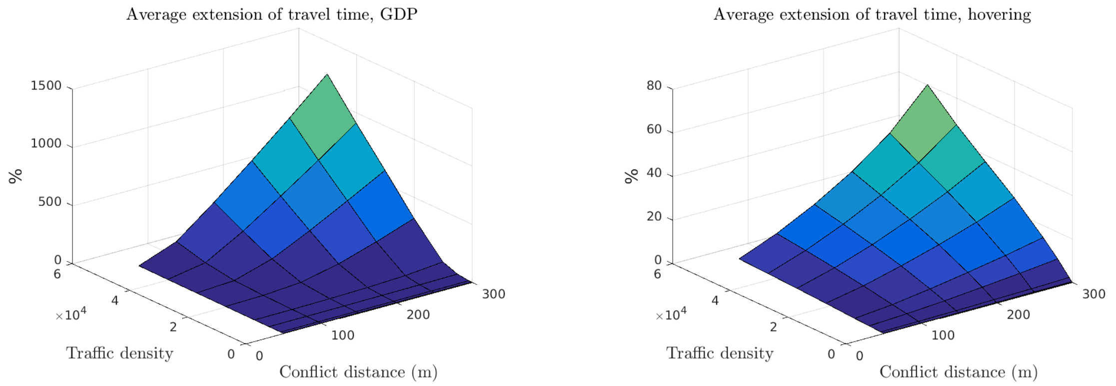

Fig. 2. Average travel time extension as a function of N and r. Left: Ground Delay Program. Right: hovering strategy.

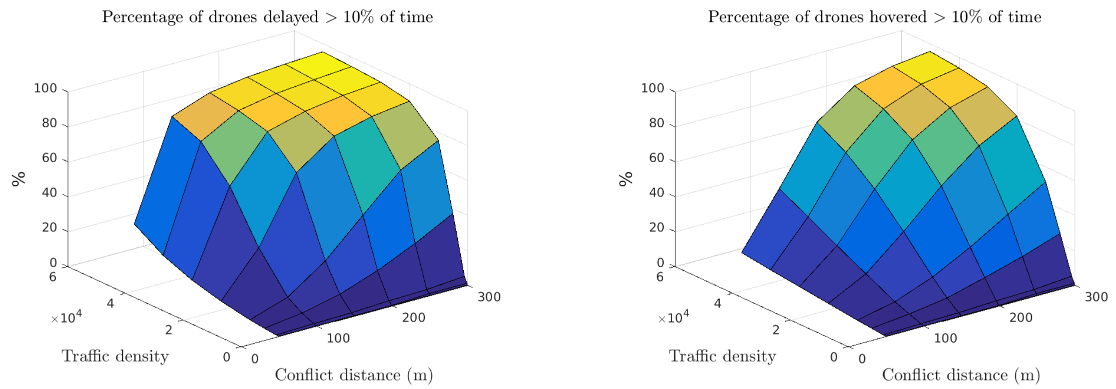

Fig. 3. Percentage of UAVs that experienced increase of travel time by more than 10% as a function of N and r. Left: Ground Delay Program strategy. Right: hovering strategy.

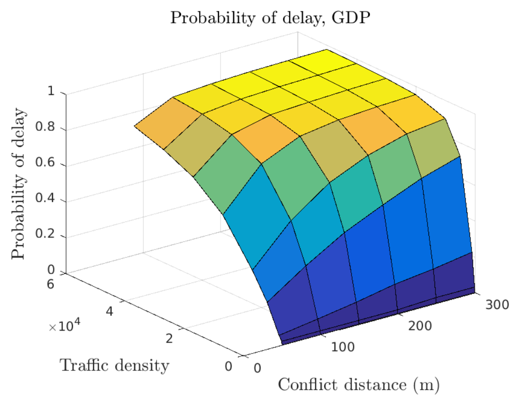

Fig. 4. Delay probability as a function of N and r

the set of pairs of conflicting drones. It can be seen that one would have to subdivide the airspace into too many layers, if one were to achieve conflict-free airspace following the strategy to distribute traffic among the layers based on the headings.

Our algorithm for level assignment is oblivious to the drones headings – it does not forbid UAVs with significantly different headings to fall into the same layer, provided they are not in conflict. Figure 8 shows the standard deviation of drones heading angles for each layer of the centralized assignment to layers (with parameters $N = 50000$ and $r = 300$m). It can be seen that the conflict-free layers do not require drones in the same layer to head in approximately same direction.

It can be concluded that assignment of drones to layers according to their headings may not necessarily be the best

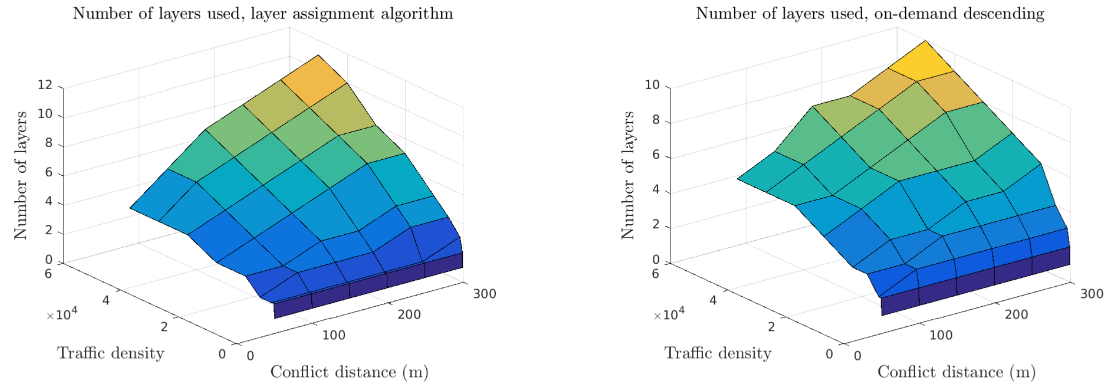

Fig. 5. Number of layers used by UAVs as a function of N and r. Left: centralized layer allocation algorithm. Right: on-demand descending algorithm.

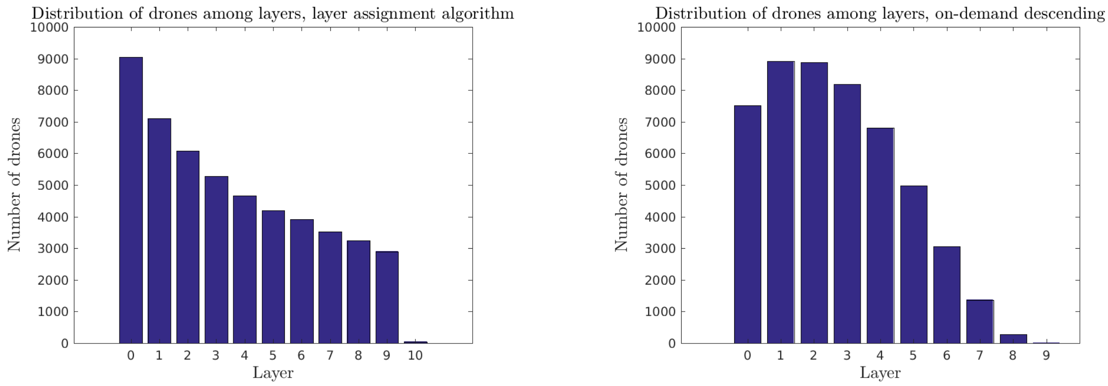

Fig. 6. Distribution of UAVs among layers. Left: centralized allocation algorithm. Right: on-demand descending algorithm.

strategy: even though it may decrease the number of conflicts, it does not eliminate all of them. In particular, if the UTM service provider has advance knowledge of the planned operations, it might be beneficial to consider a *conflict-free* allocation instead of using a simpler strategy of distributing the vehicles based on heading directions.

C. Vertical separation loss

It was suggested that horizontal maneuvers will form the basis for the future UTM deconfliction algorithms [14]. Note that our hovering, centralized layers allocation and on-demand descending strategies employ changing of altitude. Thus, it is vital to estimate the safety issues that may arise from the vertical maneuvers in our deconfliction schemes.

Under the hovering paradigm, a vertical collision may happen when a drone recovers from the diving to a new conflict with a lower-priority drone (Fig. 9, left). In the centralized system each layer is conflict-free, so loss of separation never occurs during a flight; however, a vertical conflict may occur during the VTOL. Finally, under the on-demand descent scheme, vertical conflicts may happen during both VTOL and descending conflict avoidance maneuvers (here we count only conflicts in which the descending drone has lower priority than the drone below, implying that the drone must immediately dive to one extra layer through the higher-priority drone's disk, see Fig. 9, right). Figure 10 shows percentages of the drones that experienced conflicts under the three strategies.

IV. Conclusions and Future Work

We studied several UTM deconfliction paradigms under different airspace models. Our results suggest that a centralized approach does not always exhibit the best performance. We believe that the impact of different conflict avoidance strategies and airspace structures on the capacity and safety should be investigated deeper as it affects the efficiency of operations.

Note that the standard coloring algorithms do not take into account the balance between the colors, so some layers may become much more populated than others. *Balanced colorings* [33] aim to remove the skew between the color classes sizes. E.g., in an *equitable coloring* [34] the sizes of any two color classes differ by at most 1. Finally, *sum coloring* [35]–[38] assigns numbers instead of colors (the numbers may be directly interpreted as the altitudes), and the goal is to minimize the *sum* of the assigned numbers (as opposed to vanilla coloring, which would minimize the *maximum* assigned number); this may be relevant if the goal is to minimize the total (vertical) distance flown by the drones.

One alley of future research is to work with a limited number n of layers. In the centralized UTM, drones can be distributed within n layers by coloring the conflict graph with

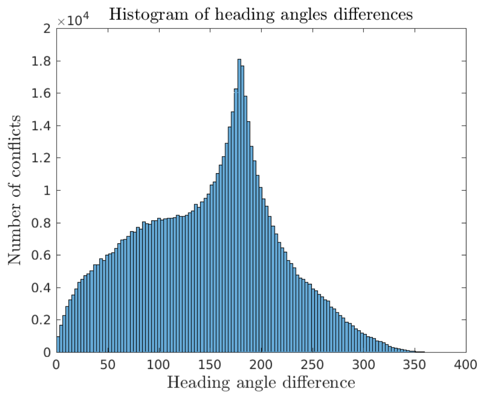

Fig. 7. Histogram of heading angles differences of conflicting pairs of UAVs.

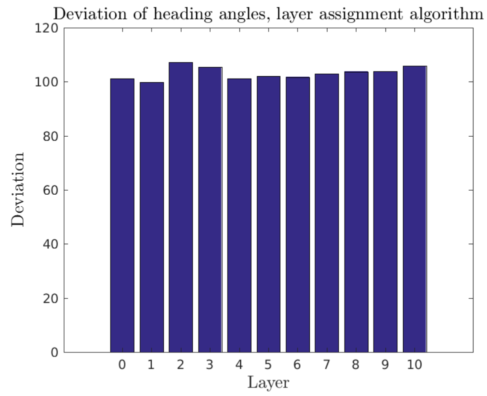

Fig. 8. Standard deviation of heading angles of drones within a layer in centralized allocation model.

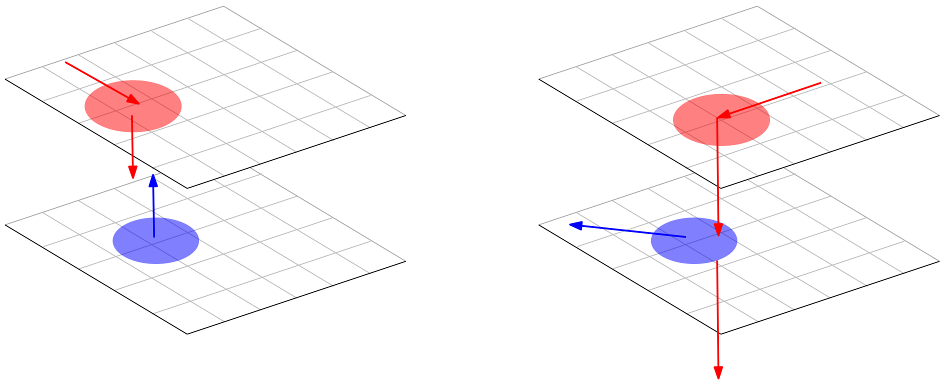

Fig. 9. Schematic pictures of vertical separation loss scenarios. Left: A conflict after a hovering maneuver. The blue UAV returns from a hovering maneuver to a new conflict with the lower-priority red UAV. Right: A conflict after a descending maneuver. The red UAV descending into a new conflict with the higher-priority UAV.

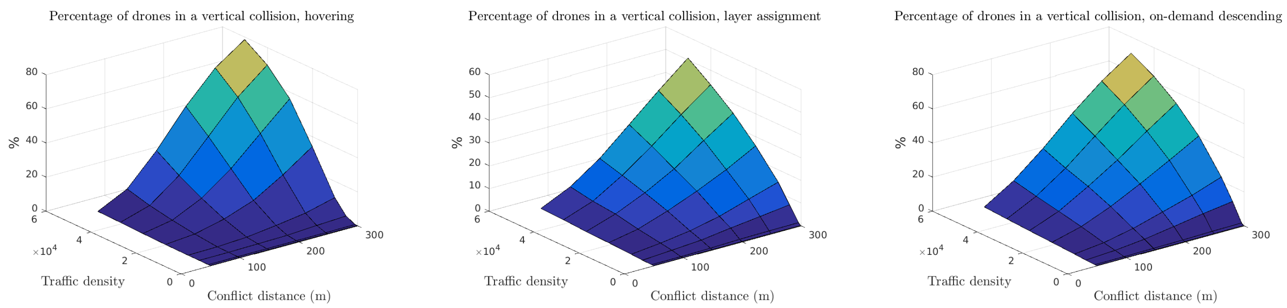

Fig. 10. Percentage of drones involved in vertical conflicts as a function of N and r: hovering (left), centralized allocation (middle), on-demand descending (right).

n colors, while keeping the minimum possible number of monochromatic edges [39] (i.e., minimizing the number of conflicts). In the decentralized system, deconflicting drones within a limited number of layers can be solved by applying both descending and ascending on-demand maneuvers whenever a UAV reaches top or bottom layer (Fig. 11).

Last but not least, the real need for the layers structure arises only in congested spots where the conflicts happen; the rest of the airspace may be single-layer (i.e. layers should rather be an exception than the rule). Our results, together with the Likely UTM (LiU) map [28] showing the density of UAV operations, may provide insights into where the layer restrictions should be established (following Mantra2 of NASA ConOps [17]) – e.g. around areas of frequent on-demand descent.

Acknowledgements

We thank the anonymous reviewers for the helpful comments. This work is part of UTMOK project supported by Swedish Transport Administration (Trafikverket) via Swedish Air Navigation Service provider LFV (Luftfartsverket).

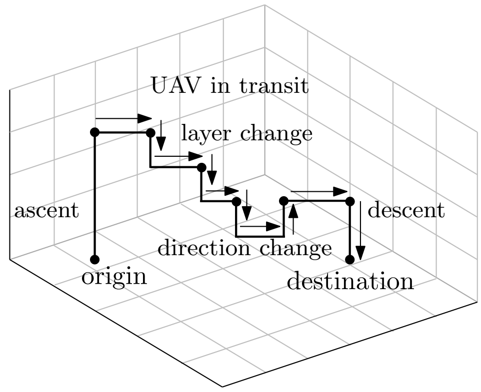

Fig. 11. A flight path with both descending and ascending on-demand vertical maneuvers.

References

[1] P. Kopardekar, "Unmanned Aerial System Traffic Management System," Talks at Google, 2016.

[2] J. Hoekstra, R. van Gent, and R. Ruigrok, "Designing for safety: the free flight air traffic management concept," *Reliability Engineering & System Safety*, vol. 75, no. 2, pp. 215 – 232, 2002.

[3] 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.

[4] 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 R&D Seminar*, 06 2015.

[5] V. Bulusu, V. Polishchuk, R. Sengupta, and L. Sedov, "Capacity estimation for low altitude airspace," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 4266.

[6] V. Bulusu, L. Sedov, and V. Polishchuk, "Noise estimation for future large scale small UAS operations," in *NOISECON*, 2017.

[7] J. M. Hoekstra, J. Maas, M. Tra, and E. Sunil, "How do layered airspace design parameters affect airspace capacity and safety?" in *Proceedings of the 7th International Conference on Research in Air Transportation*, 2016.

[8] E. Sunil, J. Ellerbroek, J. Hoekstra, and J. Maas, "Modeling Airspace Stability and Capacity for Decentralized Separation," *DEP*, vol. 3, p. R1, 2017.

[9] M. Tra, E. Sunil, J. Ellerbroek, and J. Hoekstra, "Modeling the Intrinsic Safety of Unstructured and Layered Airspace Designs," in *2017 ATM R&D Seminar*, 2017.

[10] N. Barnier and P. Brisset, "Graph Coloring for Air Traffic Flow Management," *Annals of Operations Research*, vol. 130, no. 1, pp. 163–178, Aug 2004.

[11] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan, "Concepts of Airspace Structures and System Analysis for UAS Traffic flows for Urban Areas," in *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, 2017.

[12] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, Dec 2000.

[13] S. Balachandran, C. Muñoz, 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*, 2017, p. 4484.

[14] R. E. Tompa, B. Wulfe, M. P. Owen, and M. J. Kochenderfer, "Collision avoidance for unmanned aircraft using coordination tables," in *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*. IEEE, 2016, pp. 1–9.

[15] N. Barnier and C. Allignol, "4d-trajectory deconfliction through departure time adjustment," in *ATM 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar*, 2009.

[16] C. Muoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, and J. Chamberlain, "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, Sept 2015, pp. 5A1–1–5A1–12.

[17] T. Prevot, J. Rios, P. Kopardekar, J. E. Robinson III, M. Johnson, and J. Jung, "UAS traffic management (UTM) concept of operations to safely enable low altitude flight operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3292.

[18] M. Lissone, D. Colin, and P. Barret, "EUROCONTROL RPAS ATM CONOPS," in *Integrated Communications, Navigation and Surveillance Conference*. IEEE, 2017, plenary P3-10.

[19] SESAR, "U-space Blueprint," 2017.

[20] G. U. Association, "UAS Traffic Management Architecture," 2017.

[21] Airservices, "Management of Remotely Piloted Aircraft Systems (RPAS) in ATM operations," 2016.

[22] METI, "Roadmap for Application and Technology Development of UAVs in Japan," 2016.

[23] L. Sedov, V. Polishchuk, and V. Bulusu, "Decentralized self-propagating ground delay for UTM: Capitalizing on domino effect," in *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, April 2017, pp. 6C1–1–6C1–9.

[24] V. Bulusu, R. Sengupta, and Z. Liu, "Unmanned Aviation: To Be Free or Not To Be Free? A Complexity Based Approach," in *7th International Conference on Research in Air Transportation ICRAT*, 2016.

[25] L. Peng and Y. Lin, "Study on the model for horizontal escape maneuvers in TCAS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 392–398, 2010.

[26] "Federal Aviation Administration - Unmanned Aircraft Systems," https://www.faa.gov/uas/, 2016.

[27] P. Galinier, J.-P. Hamiez, J.-K. Hao, and D. Porumbel, "Recent Advances in Graph Vertex Coloring," in *Handbook of Optimization: From Classical to Modern Approach*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 505–528.

[28] L. Sedov, V. Polishchuk, and V. Bulusu, "Sampling-based capacity estimation for unmanned traffic management," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, Sept 2017, pp. 1–10.

[29] 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)*, June 2017, pp. 1673–1681.

[30] V. Bulusu and V. Polishchuk, "A threshold based airspace capacity estimation method for UAS traffic management," in *Systems Conference (SysCon), 2017 Annual IEEE International*. IEEE, 2017, pp. 1–7.

[31] J. Krozel, M. Peters, and K. Bilimoria, "A decentralized control strategy for distributed air/ground traffic separation," in *Guidance, Navigation, and Control and Co-located Conferences*. American Institute of Aeronautics and Astronautics, Aug 2000.

[32] K. Bilimoria, K. Sheth, H. Lee, and S. Grabbe, "Performance evaluation of airborne separation assurance for free flight," in *Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics, 2000.

[33] H. Lu, M. Halappanavar, D. Chavarra-Miranda, A. H. Gebremedhin, A. Panyala, and A. Kalyanaraman, "Algorithms for Balanced Graph Colorings with Applications in Parallel Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1240–1256, May 2017.

[34] K.-W. Lih, "Equitable coloring of graphs," in *Handbook of Combinatorial Optimization*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. Springer New York, 2013, pp. 1199–1248.

[35] M. M. Halldórsson, G. Kortsarz, and H. Shachnai, "Sum Coloring Interval and k-Claw Free Graphs with Application to Scheduling Dependent Jobs," *Algorithmica*, vol. 37, no. 3, pp. 187–209, 2003.

[36] A. Bar-Noy, M. Bellare, M. M. Halldórsson, H. Shachnai, and T. Tamir, "On Chromatic Sums and Distributed Resource Allocation," *Inf. Comput.*, vol. 140, no. 2, pp. 183–202, 1998.

[37] K. Jansen, "Complexity Results for the Optimum Cost Chromatic Partition Problem," *Universität Trier, Mathematik/Informatik, Forschungsbericht*, vol. 96-41, 1996.

[38] M. R. Salavatipour, "On Sum Coloring of Graphs," *Discrete Applied Mathematics*, vol. 127, no. 3, pp. 477–488, 2003.

[39] P. M. Vitnyi, "How well can a graph be n-colored?" *Discrete Mathematics*, vol. 34, no. 1, pp. 69 – 80, 1981.