Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)

Density-Adapting Layers towards PBN for UTM

Vincent Duchamp
Ecole Nationale de l'Aviation Civile (ENAC)
Toulouse, France

Leonid Sedov and Valentin Polishchuk
Communications and Transport Systems
ITN, Linköping University
Norrköping, Sweden

***Abstract*—We study separating urban unmanned aerial vehicles (UAV) traffic into altitude levels, using a PBN-inspired approach in which low-density airspace has few layers while congested areas in the city center are split into a larger number of layers. Navigating in the many-layers environment may require better vehicle equipage to support higher performance in terms of altimetry precision; our work thus follows the stakeholders encouragements to use performance-based navigation (PBN) in UAV traffic management (UTM). We present results for several traffic volume scenarios over Norrköping municipality in Sweden, demonstrating applicability of our solutions in a city setting.**

***Keywords*—Airspace design; Levels structure; Unmanned traffic management; Performance-based navigation; Very low level urban airspace**

I. Introduction and Related work

The use of levels for aircraft separation prevails in ATM, with practices ranging from quadrantal table of cruising levels to semi-circular (alternating altitudes) rule [17] to reduced vertical separation minima (RVSM) [18]. A thorough investigation of several air traffic structuring concepts by the Metropolis project [23], [24], [39]–[42] revealed that layered airspace is the most successful design. The general assumption in Metropolis was that the aircraft are distributed uniformly, so it was natural to use the same layers throughout the whole geographical spread of the airspace. Contrary to this, in UTM, the distribution of drone traffic over a city will likely be highly non-uniform [14], [25], [31], [35], with congested city center and lower-density outskirts. Thus, using the same layers over the whole city could be overly conservative, as it may suffice to establish more layers for traffic in high-density areas where large conflicts happen, and have the number of layers go down towards the suburbs. In this paper, we study the design of such an airspace with the number of layers in different areas adapted to the area-specific traffic conflict size.

A. Connection to policy makers views

a) Follow the Mantras and Key Principles with PBN!: The adaptive airspace design, investigated in this paper, follows Mantra 1 "Flexibility where possible, structure only where necessary" of NASA UTM ConOps [32] by establishing more layers only where traffic deconfliction requires it. It also adheres to Mantra 2 "Risk-based approach where geographical needs and use cases determine the airspace performance requirements", as it relies on PBN to separate traffic classes based on their performance (which is also reflected in one of the key principles in SESAR's U-space Blueprint, encouraging "To follow a risk-based and performance-driven approach when setting up appropriate requirements..." [37]). Indeed, the altitude band for very low level (VLL) urban airspace operations may be quite thin, due to noise considerations, upper ceiling of 400-500ft, and other factors. This means that the distance between the layers will be small, implying, in its turn, that only well equipped drones, with good vertical precision performance (and, possibly lower noise, if the levels will have to go closer to the ground) will be allowed into the areas with many layers – a PBN requirement.

b) Software-Defined Airspace Anyone?: Technologically, airspace structure in UTM may be enabled via geofencing: EUROCONTROL's ConOps [30] assumes that for VLL operations geofencing is in place, SESAR's U-space Blueprint [37] lists geofencing as one of the 3 foundation services due for deployment already in U1, and CORUS ConOps [11] speaks about loading geofence directly into drones. Clearly, legacy ATM practices, like maintaining Aeronautical Information Publication (AIP) on Aeronautical Information Regulation And Control (AIRAC) cycles, are too slow for defining airspace boundaries for the much more versatile drone traffic: updates to the UTM layers structure can arrive on the hourly basis (keeping, say, an extra layer for leftover traffic from the previous hour) or even more frequently. The frequent updates can be supported by the higher level of automation, envisioned in UTM both on the users and the service provider side. In particular, airspace layer definitions can be regularly recomputed, with little or no human oversight – something which may be dubbed *Software-Defined Airspace (SDA)*. Algorithms in this paper cater to the SDA by providing automated procedures for generation of the restricted multi-level areas, based on the actual or near-term projected traffic pattern. As a couple of concrete examples for application of our methods we mention that (1) the higher-density, higher-performance requirement areas in the city center, computed by our algorithms, may serve as the Amber airspaces of CORUS ConOps [11], and (2) our

Part of the research was conducted during summer internship of VD in Linköping University, funded by the French Government. LS and VP are supported by the Swedish Research Council and the Swedish Transport Administration.

algorithms can be used to segment the airspace "into cells of similar requirements", as per DLR Concept for Urban Airspace Integration [19] (which develops concepts to "form a basis for further research in the area of separation and performance-based operations for UAS").

c) PBN for UTM: The need for PBN in UTM has been long promoted by Dr. Parimal Kopardekar [27] who encouraged to adopt different regulations in different airspaces (including restrictions for drones with weaker capabilities from flying through congested urban spaces) and to establish restrictions (cruising, lanes, corridors, altitude separation) only when and where necessary. These views are, naturally, reflected in Mantras of NASA UTM ConOps [32] and key principles of SESAR U-space Blueprint [37], which we follow in this paper. Dr. Kopardekar also highlighted the need to "match the geography to the required performance to operate in the airspace" [28], which is exactly what our algorithms do.

Importance of performance-based regulatory regime and risk-based analysis for urban airspaces is stressed also by the FAA [16]; AirMap requires U-space to be "performance-based to encourage innovation" [2]. One of the ideas in PBN is to acknowledge the diversity of the users and airspaces, and provide air navigation services based, in particular, on the equipage level and airspace specifications. Note that in ATM, PBN has to fit into the existing well evolved system, which is a sometimes painful process. To avoid the same type of retrofitting complications in UTM, it may be beneficial to be proactive and establish PBN for UTM as early as possible, especially given the diversity of UTM users (drones and drone operators).

d) Layers and PBN: Eurocontrol's RPAS ConOps [30] mentions the need to define classification of drones capabilities; a categorization of the drone PBN capabilities, including the vertical performance, was done in [33]. It follows from the findings in [33] (and is widely acknowledged in general) that layers may be a scarce resource in UTM. For instance, all three airspace issues identified at CORUS Exploratory Workshop [3] were related to the height; in particular, the overall view on the specific question of airspace division into bands was that many layers should be an exception, for drone-busy areas, rather than the rule. This is exactly the approach we take in this paper.

B. Overview of the approach and Roadmap

The results in this paper contribute to creation of decision support tools which help in transforming generic mandates from policy makers (like "only better equipped drones should be permitted in a city center") into concrete operational numbers and locations (which area exactly is the "city center" in which drones with low performance are forbidden, what are those performance characteristics, etc.). Our methods are inspired by algorithms for perfect graphs [21] and push off earlier literature on UTM capacity thresholds [6], [35]. The novelty of this paper in comparison with earlier work on the thresholds is that we focus on *spatial distribution* of the conflicts. Specifically, the main finding in [6], [35] was that the probability of observing a large conflict over a city is essentially either 0 or 1 (depending on the traffic density and the conflict radius r). It should be emphasized that these results are oblivious to the locations of the conflicts, i.e., they assert that the conflict either will not be observed *anywhere* in the city or, on the contrary, there will be a conflict *somewhere* in the city – without looking into the conflicts "geography" in the latter case. In this paper, we extend the studies in [34], [35] by examining *where* the conflicts happen. Our geofences then separate areas of large-, medium- and small-size conflicts from each other.

In the next section we give the details of our solutions. Section III presents a sample of the algorithms output for Norrköping municipality in Sweden. Section IV discusses extensions of our methods. Section V concludes the paper.

II. Algorithms

We first describe notions from graph algorithms relevant for our study, and then recapitulate and slightly extend the results on UTM capacity thresholds. We then use the thresholds to determine the number of airspace layers needed in different parts of the city and to delineate the boundaries of different-layers parts.

A. Graph-theoretic preliminaries

We model drones as disks of a given radius r representing the safety zones around the drones; two UAVs are in conflict if their disks overlap. Deconflicting the drones by distributing them to different airspace layers can be viewed as graph *coloring* – assignment of colors to the vertices of the graph 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 pairwise conflicts ($(d_1, d_2) \in E$ iff d_1 and d_2 conflict with each other at some point during their flights), and let $G = (V, E)$ be the drones conflict graph in which two vertices are connected iff the corresponding drones 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. The *chromatic number* χ of G is the minimum number of colors needed to color the graph.

A *clique* in G is a subset $Q \subseteq V$ of pairwise-connected vertices ($(d_1, d_2) \in E \ \forall d_1, d_2 \in Q$); Q is called *maximal* if it is not possible to add vertices to it while keeping it a clique ($Q \cup v$ is not a clique $\forall v \in V$). A *maximum* clique is a clique of largest size; let ω denote this size (Fig. 1, left).

Finding both χ and ω are computationally hard (NP-complete) problems, but many heuristics exist. For experiments in this paper we used JGraphT's implementation [38] of Dsatur greedy coloring algorithm [4], and Bron–Kerbosch clique enumeration algorithm [5] for listing *all* maximal cliques (which computes ω as a by-product, since a maximum clique

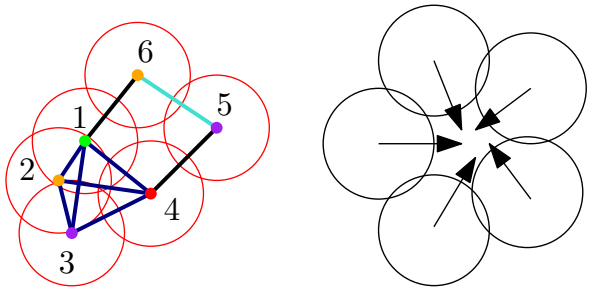

Fig. 1. Left: G with $\chi = \omega = 4$. {1,2,3,4} (navy) is a maximum clique; {5,6} (cyan) is a maximal clique – no vertex can be added to the set while keeping it a clique. Right: If, during the motion, 5 disks formed a cycle, then it is likely that some disk overlapped (or will overlap) also with a disk which is not its neighbor in the cycle (we put an interactive applet at https://www.geogebra.org/graphing/t3dw75ja to give a feeling of how finely the trajectories must be adjusted in order to have a chordless cycle in G.)

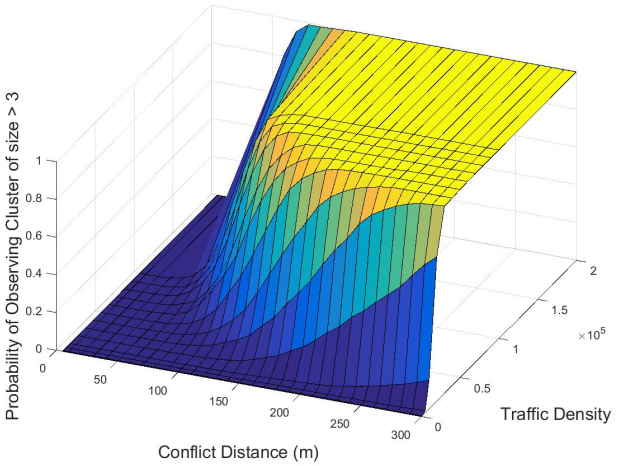

Fig. 2. Fig. 9 from [6], showing how $P_3(N, r)$ sharply grows at certain N and r.

is, of course, maximal).[1] In almost every drone conflict graph G that we encountered, the greedy coloring found a coloring either with exactly ω or $\omega + 1$ colors. Since the vertices of a clique must be colored with different colors, ω is a lower bound on χ ($\chi \geq \omega$), which implies that in our case the greedy coloring performed very well – either finding the true χ or making an error of at most 1 color.[2] We will therefore slightly abuse the terminology and identify ω with the number of colors found by the greedy – this way we may be off by at most 1 (at most 1 extra layer).

B. Thresholds for monotone properties of random graphs

UAV traffic will be highly non-deterministic, as many drones may fly without any pre-defined schedule. In this paper, we account for the probabilistic component using the *Cal model* (the name "Cal" was chosen in [7] because the model was introduced at ICRAT 2016 [8] by researchers representing University of California Berkeley, going by Cal), in which traffic demand is proportional to the population density and the flights are direct. Because of the stochastic traffic, the conflict graph G is a *random* graph. Consequently, it makes sense to estimate *probabilities* of having certain structures (subgraphs) in G as functions of the number of expected drone operations N and the safety radius r. In particular, [6] studied the probability $P_3(N, r)$ of G having a connected component of size at least 3. Clearly, the probability grows both as a function of N and as a function of r; the interesting finding of [6] was that the functions exhibit *thresholds*: $P_3(N, r)$ jumps from essentially 0 to essentially 1 as (N, r) passes over quite a thin *threshold* curve in the (N, r)-plane (Fig. 2).

As explained in [6], the threshold behavior of $P_3(N, r)$ is not entirely surprising: a fundamental result in random graph theory [20] states that random geometric graphs exhibit thresholds for any *monotone* property (a graph property is called monotone if it continues to hold when edges are added to the graph; e.g., "having a connected component of size at least 3" is a monotone property – if a graph had such a connected component and an edge is added, the graph still has it). However, this theory result is established only for the case when the graph vertices are distributed *uniformly* at random, and the experiments in [6] were needed to verify existence of the thresholds in G, given that the drones distribution over an urban area is highly non-uniform.

The next step in studying the thresholds was taken in [35], which computed the distribution of drones over a city and stored it in the *Likely UTM (LiU)* map. The map allows one to obtain samples of the conflict graph G much faster than with the simulation-based approach from [6]; in particular, [35] computed the probability functions $P_k(N, r)$ (having a component of size at least k in G) for several values of $k >$ 3. The probabilities again exhibited the thresholds (perhaps unsurprisingly, since having a component of size at least k is a monotone property), with the threshold curves shifting into larger N and r, as k was growing (Fig. 3).

C. From thresholds to layers via sampling

We repeated the calculations from [35] for another monotone property of G – "having a clique of size at least q". The resulting probability functions $P^q(N, r)$ are shown in Fig. 4; it can be seen that the property also has a threshold for any q. For the sake of brevity we will abuse the numbers, and instead of "The probability of observing a clique of size at least q

[1] As a side remark, these approaches have complementary properties: the greedy coloring runs in polynomial time but does not guarantee to find χ; the clique enumeration algorithm finds ω (and not only the maximum clique, but also all maximal cliques), but has no running time guarantee.

[2] We speculate that the algorithms perform so well because any 5-cycle (or more generally, any odd cycle) in G is likely to have a *chord*, i.e., an edge that connects non-neighboring vertices of the cycle (Fig. 1, right), implying that G may be *perfect* [21]. In a perfect graph $\omega = \chi$ and both can be found in polynomial time, That is, even though the greedy coloring is not guaranteed to be optimal, in perfect graphs, the coloring and the clique problems are "easy".

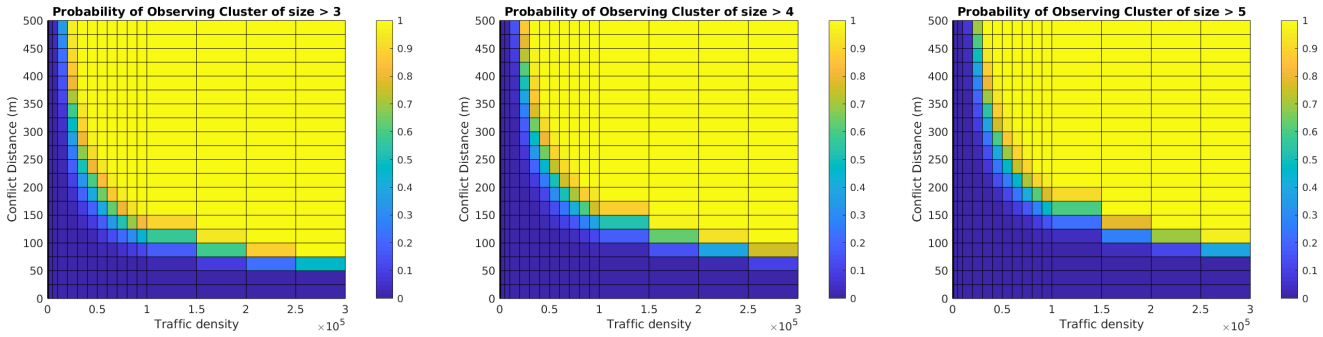

Fig. 3. Top view of graphs of $P_k(N, r)$ (darkblue is 0, yellow is 1). From left to right: $k = 3, 4, 5$.

is close to 0" will say simply "There is no clique of size q"; otherwise, we will say that "There is a clique of size q".

In this paper, we view the probability graphs slightly differently from the previous work: (instead of looking at the probabilities as a functions of N and r for a *fixed* q,) we watch when the threshold curves pass over a *fixed* point of interest $(N.r)$ when q increases (refer to Fig. 4). This value $q^* = \max\{q$: there is a clique of size $q\}$ gives us the maximum number of layers needed in the city.

Next, using LiU map [35], we produce 10000 sample snapshots of drone traffic and build 10000 conflict graphs G for a given pair (N, r). In each conflict graph, we find all maximal cliques with the JGraphT's implementation [38]. We record the set of points S_{q^*} where the cliques of size at least q^* were observed.[3] The area, in which our airspace will have q^* layers is the convex hull of S_{q^*}.[4] Borrowing from ATM terminology, we call our areas with the different number of layers the *sectors* (we are not aware of any existing name for such a concept): an area with q layers will be called a q-sector.[5]

The q-sectors for $q < q^*$ are build recursively, similarly to the q^*-sector: we record the locations S_q of (maximal) cliques with size at least q, output by JGgraphT on our 10000 samples of G, and define the q-sector as the convex hull of points in S_q. Note that the q-sector contains points where *at least* q layers are needed, not only where *exactly* q layers are needed. This way our sectors form nested, isolines-like structure, with q-sector being a subset of the q'-sector for $q > q'$ (refer to Fig. 5 in Section III). This is reasonable in terms of PBN: drones with higher performance are allowed to fly also in sectors requiring lower performance.

The pseudocode for our algorithm is given below.

Algorithm 1: Sectorization algorithm

Input : A set Q of traffic snapshots from LiU map (each sample $P \in Q$ is a set of points).
Output : A list of sectors' boundaries B.

$q^* \leftarrow \max\{q$: there is a clique of size $q\}$;
// determined from the graphs $P^q(N, r)$
$S \leftarrow$ an empty list;
foreach *sample* $P \in Q$ **do**
  $G \leftarrow$ conflict graph of drones in P;
  $C \leftarrow$ set of all maximal cliques in G;
  foreach $q \in \{1 \dots q^*\}$ **do**
    $S_q \leftarrow S_q \cup \{c \in C : size(c) \geq q\}$;
  end
end
$B \leftarrow$ an empty list;
foreach $q \in \{1 \dots q^*\}$ **do**
  $B_q \leftarrow$ convex_hull(S_q);
end
return B;

III. Results

We computed the sectors for a range $N = 10000 \dots 200000$ of expected daily operations and a range $r = 50 \dots 300$m of safety radii, for the urban area of Norrköping Municipality, Sweden. Figure 5 shows maps with sectors for different values of parameters N and r [6]. Using our maps, the amber airspace class of CORUS ConOps [11] may be defined, e.g., as the airspace where the number of layers exceeds some limit q^a; the rest of the airspace is green. The parameter q^a can be decided based on the drones' operators capability to solve complex conflicts (cliques in our model), which, in its turn, can be

[3]Strictly speaking, a clique is not a single point (it consists of q^* radius-r disks); still, at the geographical scale of the urban area, we viewed the disks as points (if one is concerned with the clique falling into two neighboring sectors, one may enlarge each of our sectors by r, i.e., take the Minkowski sum of the sector and radius-r disk).

[4]Of course, very rarely our conflict graphs G had $\omega > q^*$ (i.e., featured cliques of size larger than q^*): according to the computed probability functions (refer to Fig. 4)), $P^q(N, r)$ is close to 0 but is not exactly 0, even for $q > q^*$. Indeed, given the stochasticity of drone traffic, it is impossible to fully exclude the possibility that a 100 drones would want to fly through the same point. Still, the thresholds suggest that this would be a very rare event, in which case other UTM measures (ground delay, etc.) will have to be invoked to handle the large conflict – these are outside this paper's scope.

[5]We emphasize that our sectors are merely subsets of the city and are *not* meant for human oversight – otherwise, it could be a little provocative to suggest that a controller for q-sector controls a superset of what a controller for q'-sector controls for $q' > q$.

[6]Pictures for other values of N and r are available at https://tiny.cc/atm_sem_pics

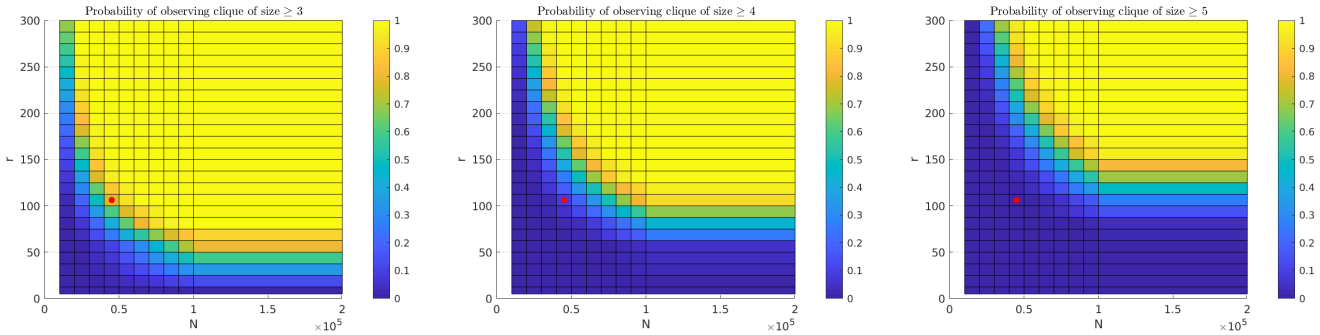

Fig. 4. Top view of the graphs of $P_q(N, r)$ for some q (darkblue is 0, yellow is 1). From left to right: $q = 3, 4, 5$. For the pair $(N, r) = (40000, 100)$ (red dot), there is a clique of size 3, but no clique of size 4.

estimated by taking into account, e.g., the rules of air, human factor, or other characteristics. Figure 6 shows an example of possible airspace division into green and amber zones for $q^a = 4$.

IV. Extensions

In this section we use generalizations of the convex hull (CH) to define the sectors.

A. k-hulls

The *k-depth contour* aka *k-hull* [10] of S_q is the kth level set of the *location depth* – one of the many depth functions developed in the rich field of statistical data depth [1]; the idea of k-hull is to identify areas of statistically meaningful points that lie "deep inside" S_q:

Definition 1. *Let $k \geq 0$ be an integer. The k-hull of S_q consists from all points p such that for any line ℓ through p there is more than k points of S_q on each side of ℓ (Fig. 7).*

CH is the 0-hull (i.e., k-hull is a generalization of CH); for larger values of k the k-hull goes "deeper" into the set. Figure 8 shows how k-hulls, for $k > 0$, more tightly enclose areas of "denser" conflicts.

B. α-shapes

The *α-shape* [15] is a formal way of capturing the intuitive notion of the "shape" of the point set:

Definition 2. *Let $\alpha \geq 0$ be a number (not necessarily integer). An α-disk is a disk of radius α. The α-shape of S_q connects two points $u, v \in S_q$ whenever there exists an α-disk that has u, v on its boundary and contains no points of S_q in its interior (Fig. 9).*

CH is the ∞-shape (i.e., α-shape is a generalization of CH); for smaller values of α the α-shape provides a "finer" reconstruction of the shape of S_q. α-shapes were recently used in UTM [9] for airspace capacity evaluation.

Figure 10 shows how an α-shape, for $\alpha < \infty$ encloses only the areas where the conflicts actually happen, leaving out the areas in between. This implies that only a smaller area is occupied by the more restrictive airspace. Note however that the α-shape is not convex, meaning that a straight segment of a drone flight may enter and leave the shape more than once – this may be undesirable (in conventional ATM, significant effort is put into keeping the sectors convex [12], [13], [22], [26], [36], [43]).

C. k-order α-shape

Last but not least, we used a recent generalization of both k-hull and α-shape—*k-order α-shape*—which combines the advantages of both k-hull and α-shape (Fig. 11). k-order α-shape is defined analogously to α-shape but allowing the α-disks to contain up to k points of S_d in the interior (see [29] for the formal definition). k-order α-shape combines features of both k-hull (which identifies statistically meaningful points inside S_q) and α-shape (which allows the shape to have several connected components – clusters).

Figure 12 shows a sectorization using k-order α-shapes on the same sample data as in Figure 5 (top-left). It can be observed that k-order α-shapes both removed outliers and used less space than CH. Such parsimonious reservation of the airspace for the sectors widens the geographical area where the less equipped drones are permitted to operate.

V. Conclusion and Future Work

We presented a way to guide the design of layered UTM airspace structure over higher-density environment in the city center, while keeping lower number of layers in less dense areas. Our ideas closely follow UTM ConOpses and Blueprints in paving the way towards using PBN for UTM in (what we called) a Software-Defined Airspace. In such a flexible airspace, modifications to the layered structure may be supported on demand when the traffic pattern changes due, e.g., to some event like police activity, search-and-rescue operations etc.: the new projected flightplans may be taken as the input to our algorithms and the new layers recomputed.

While our algorithms ensure that there is enough layers for deconflicting the drones in any sector, they do not suggest how to redistribute the drones among the layers when going

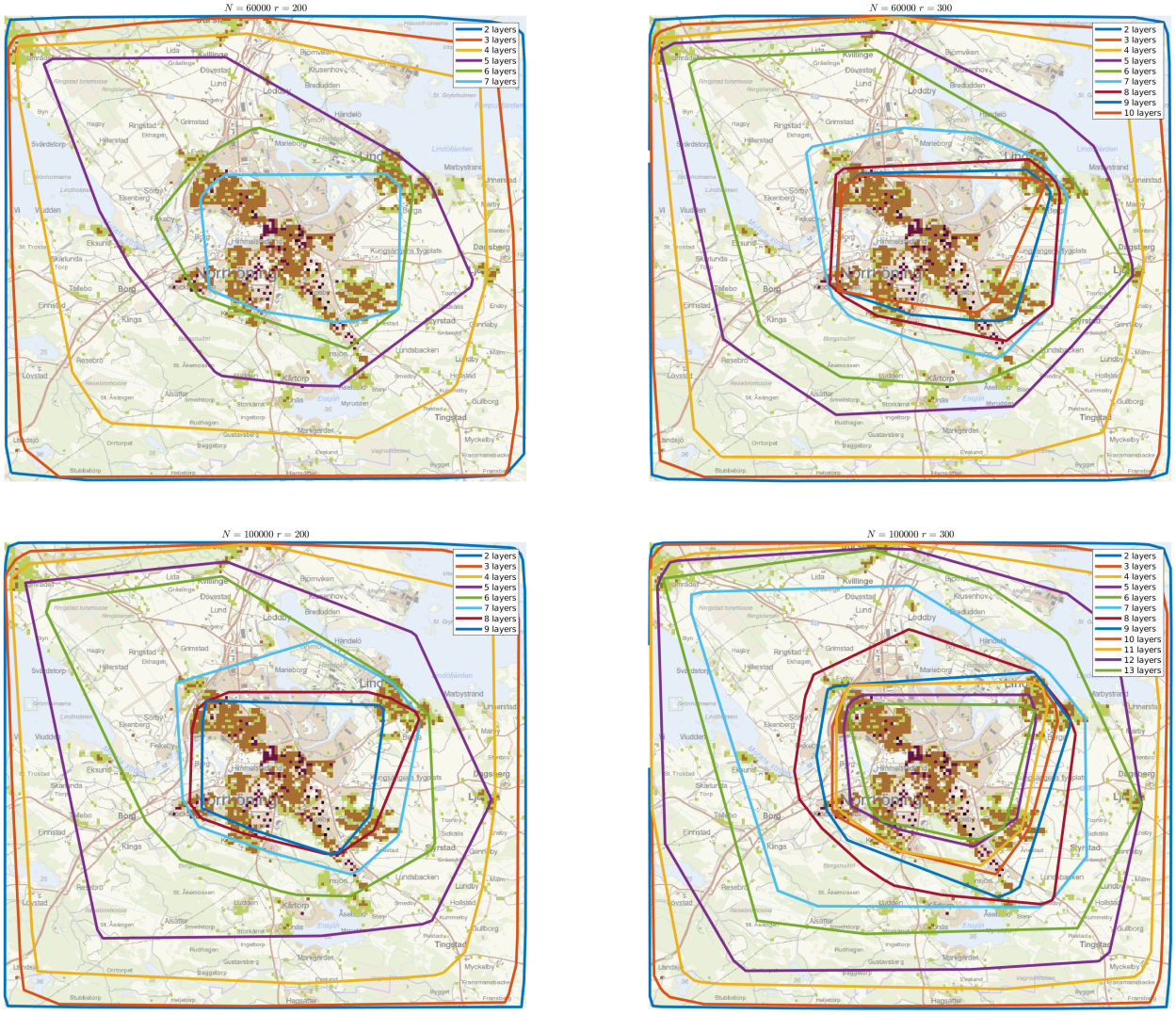

Fig. 5. From top to bottom: $N = 60000, 100000$. From left to right: $r = 200, 300$. Only boundaries of the sectors are drawn. The legends show the color coding of the number of layers in the sectors.

between the sectors so as to minimize the level changes. One possible way to attack this problem is by formulating it as an integer program. Also, it can be of interest to see what impact the presence of *geovectoring* [23] or no-fly zones in the city will have on the sectors.

Acknowledgment: We thank Vishwanath Bulusu (NASA) and Parker D. Vascik (MIT) for discussions, and Billy Josefsson, Pierre Ankartun and Roger Li (UTM project manager, LFV) for the practical insight.

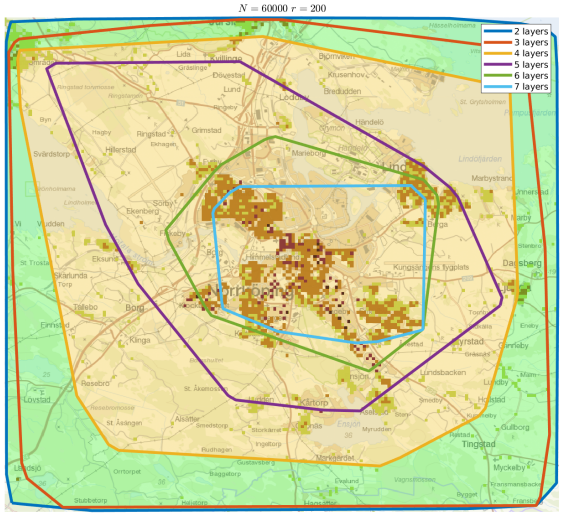

Fig. 6. A possible division of the airspace into green and amber zones for $(N, r) = (60000, 200)$ and $q^a = 4$.

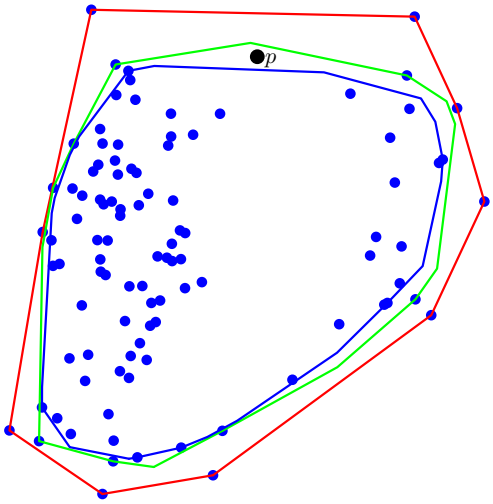

Fig. 7. Any line through p contains more than 1 point of S_q on each side; hence p belongs to the 1-hull (green). The 2-hull is blue and the 0-hull (i.e. the convex hull) is red.

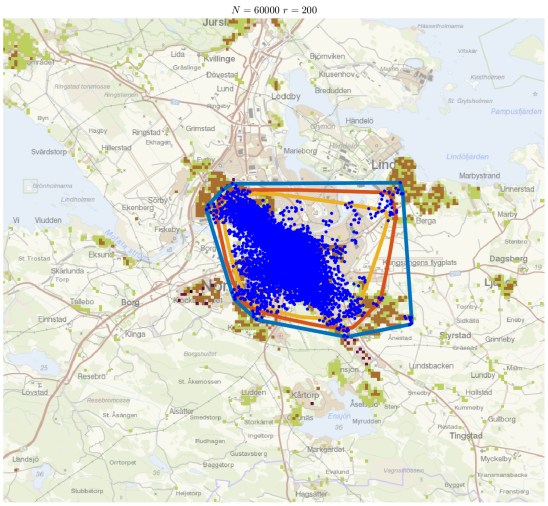

Fig. 8. S_7 (for $(N, r) = (60000, 200)$) and its CH (i.e., 0-hull), 10-hull and 30-hull.

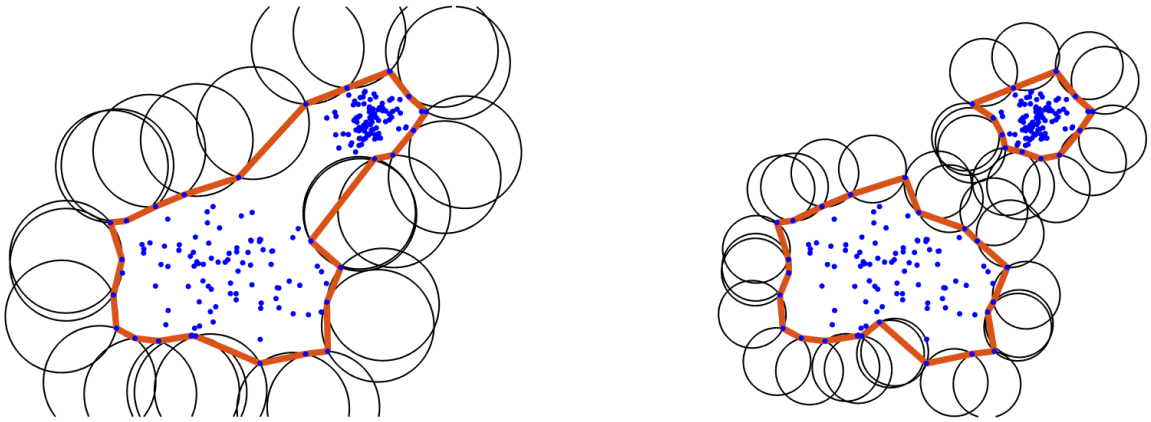

Fig. 9. Left: An α-shape (some empty α-disks are shown); points come in two clusters. Right: α-shape for a smaller α.

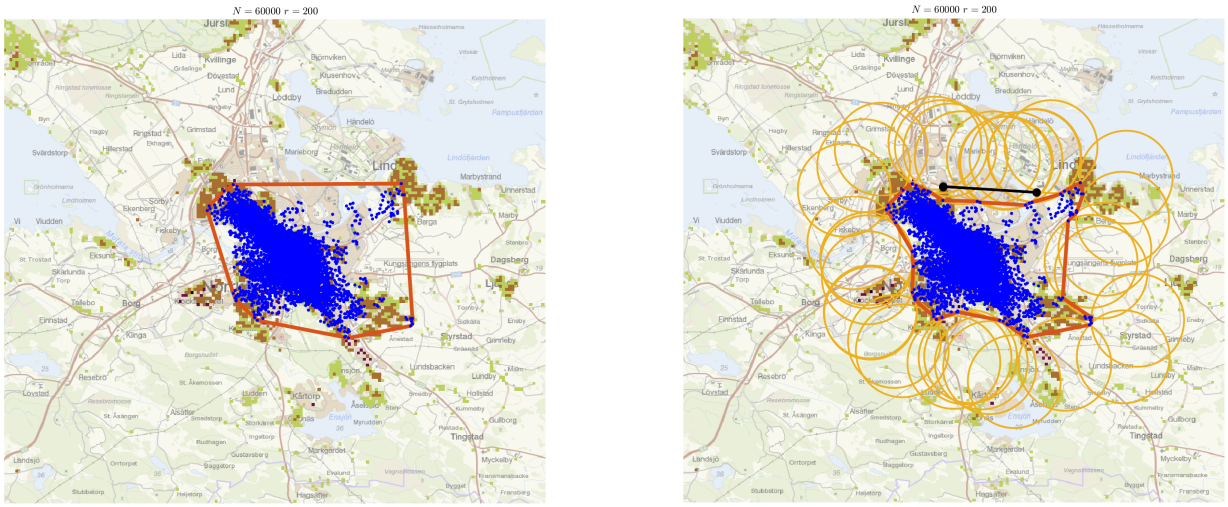

Fig. 10. Left: S_7 (for $(N, r) = (60000, 200)$) and its CH (i.e., ∞-shape). Right: S_7 (for $(N, r) = (60000, 200)$) and its α-shape for $\alpha < \infty$; some empty circles are shown. If α-shape is used for 7-sector instead of the CH of S_7, then, for example, the black path can be taken by a less-equipped drone while staying inside S_6 sector.

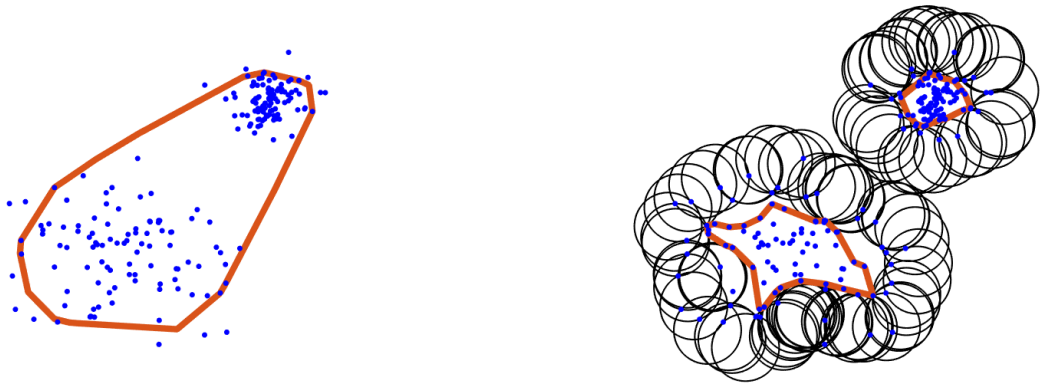

Fig. 11. Points come in two clusters. Left: The 2-hull uses too much empty space. Right: order-2 α-shape identifies shapes of both clusters and removes outliers.

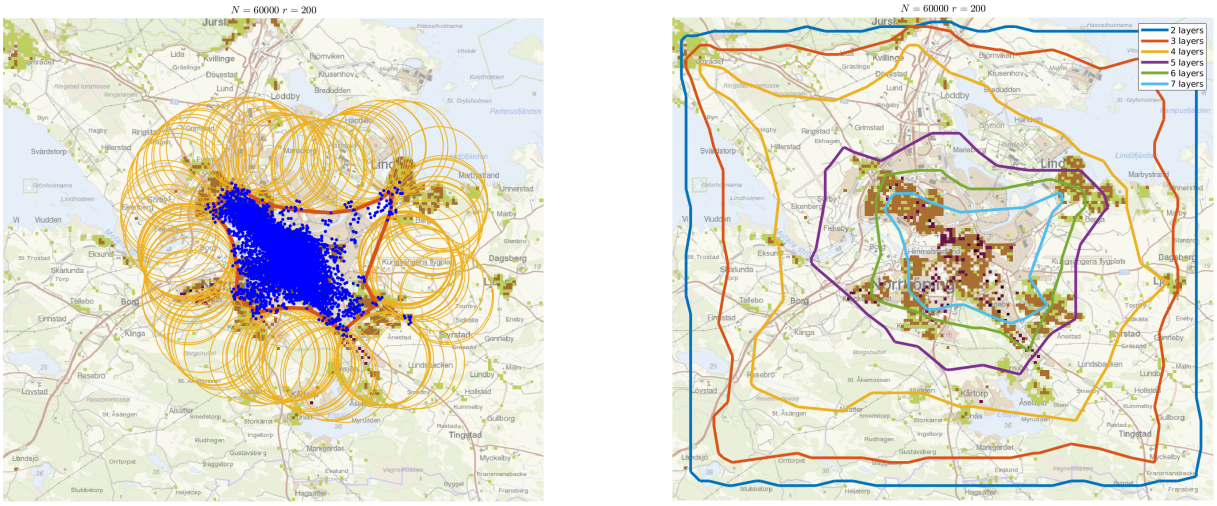

Fig. 12. Left: S_7 (for $(N, r) = (60000, 200)$) and its order-5 α-shape for $\alpha < \infty$. Right: order-5 α-shapes of S_q for $q = 2 \ldots 7$, $N = 60000, r = 200$, and $\alpha < \infty$.

REFERENCES

[1] Statistical Data Depth at Tufts. http://www.cs.tufts.edu/research/geometry/depth/.

[2] S. Babiraz. U-space: From blueprint to reality.

[3] L. Bellesia, B. Clark, E. Malfliet, Y. Seprey, C. Ronfle-Nadaud, A. Wennerberg, C. Barrado, W. Van Leare, L. Brucculeri, and A. Hately. CORUS 1st workshop early findings, 2018.

[4] D. Brélaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, Apr. 1979.

[5] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, Sept. 1973.

[6] V. Bulusu and V. Polishchuk. A threshold based airspace capacity estimation method for UAS traffic management. In *IEEE Systems Conference*, 2017.

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

[8] 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.

[9] J. Cho and Y. Yoon. How to assess the capacity of urban airspace: A topological approach using keep-in and keep-out geofence. *Transportation Research Part C: Emerging Technologies*, 92:137 – 149, 2018.

[10] R. Cole, M. Sharir, and C. K. Yap. On k-hulls and related problems. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 154–166, New York, NY, USA, 1984. ACM.

[11] CORUS. Concept of Operations for U-space, 2018.

[12] D. Delahaye, J. M. Alliot, M. Schoenauer, and J. L. Farges. Genetic algorithms for partitioning air space. In *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*, pages 291–297, March 1994.

[13] D. Delahaye, M. Schoenauer, and J. M. Alliot. Airspace sectoring by evolutionary computation. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 218–223, May 1998.

[14] S. Devasia and A. Lee. Scalable Low-Cost Unmanned-Aerial-Vehicle Traffic Network. *Journal of Air Transportation*, pages 74–83, 2016.

[15] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, July 1983.

[16] D. Elwell. UberElevate summit, 2017.

[17] Eurocontrol. Guidance notes for pilots, 2009.

[18] FAA. Reduced vertical separation minimum, 2017.

[19] D. Geister and B. Korn. Concept for urban airspace integration, 2017.

[20] A. Goel, S. Rai, and B. Krishnamachari. Monotone properties of random geometric graphs have sharp thresholds. *Annals of Applied Probability*, pages 2535–2552, 2005.

[21] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, Courant Institute of Mathematical Science, New York University, 1980.

[22] T. A. Granberg, T. Polishchuk, V. Polishchuk, and C. Schmidt. A novel MIP-based airspace sectorization for TMAs. In *USA/Europe Air Traffic Management Research and Development Seminar (ATMSeminar)*, 2017.

[23] J. M. Hoekstra, J. Ellerbroek, E. Sunil, and J. Maas. Geovectoring: Reducing traffic complexity to increase the capacity of uav airspace. In *International Conference for Research in Air Transportation, Barcelona, Spain*, 2018.

[24] 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.

[25] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan. Concepts of Airspace Structures and System Analysis for UAS Traffic Flows for Urban Areas. In *AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0449. 2017.

[26] B. Josefsson, V. Polishchuk, and L. Sedov. Towards simplified optimal sectorization. In *7th SESAR Innovation Days*, 2017.

[27] P. Kopaderkar. Safely enabling UAS operations in low-altitude airspace. In *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, page 33, 2016. Plenary talk.

[28] P. Kopardekar. Unmanned aerial system traffic management system. Talks at Google, 2016.

[29] D. Krasnoshchekov and V. Polishchuk. Order-k α-hulls and α-shapes. *Inf. Process. Lett.*, 114(1):76–83, Jan. 2014.

[30] M. Lissone, D. Colin, and P. Barret. EUROCONTROL RPAS ATM CONOPS. In *Integrated Communications, Navigation and Surveillance Conference*. IEEE, 2017. Plenary P3-10.

[31] M. F. B. Mohamed Salleh, C. Wanchao, Z. Wang, S. Huang, D. Y. Tan, T. Huang, and K. H. Low. Preliminary Concept of Adaptive Urban Airspace Management for Unmanned Aircraft Operations. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 2260. 2018.

[32] 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*, page 3292, 2016.

[33] L. Ren, M. Castillo-Effen, H. Yu, E. Johnson, Y. Yoon, N. Takuma, and C. A. Ippolito. Small unmanned aircraft system (sUAS) categorization framework for low altitude traffic services. In *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*, pages 1–10. IEEE, 2017.

[34] L. Sedov and V. Polishchuk. Centralized and Distributed UTM in Layered Airspace. In *8th International Conference on Research in Air Transportation*, 2018.

[35] 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)*, pages 1–10, Sept 2017.

[36] M. Sergeeva, D. Delahaye, C. Mancel, L. Zerrouki, and N. Schede. 3d sectors design by genetic algorithm towards automated sectorisation. In *SESAR Innovation Days*, 2015.

[37] SESAR. U-space blueprint, 2017.

[38] J. Sichi, J. Kinable, D. Michail, B. Naveh, and Contributors. Jgrapht - graph Algorithms and Data Structures in java (Version 1.3.0). http://www.jgrapht.org, 2018.

[39] E. Sunil, J. Ellerbroek, J. Hoekstra, and J. Maas. Modeling Airspace Stability and Capacity for Decentralized Separation. *DEP*, 3:R1, 2017.

[40] E. Sunil, J. Ellerbroek, and J. M. Hoekstra. Camda: Capacity assessment method for decentralized air traffic control. 2018.

[41] 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.

[42] 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.

[43] H. Trandac, P. Baptiste, and V. Duong. Airspace sectorization with constraints. *RAIRO - Operations Research - Recherche Opérationnelle*, 39(2):105–122, 2005.

BIOGRAPHIES

Vincent Duchamp is completing a double Master's Degree at the ENAC (the French Civil Aviation University). He was invited in 2018 to work at the Communications and Transport Systems division at the Linköping University on Air Traffic Management.

Dr. **Valentin Polishchuk** is an Associate Professor in the Department of Science and Technology of Linköping University, leading the Academic Excellence in ATM (and UTM) Research group.

Leonid Sedov is a PhD student in Linköping University in Sweden. He works on airspace design and Unmanned Traffic Management, in particular on airspace capacity estimation and route optimization.