

# CONVEX SECTORIZATION—A NOVEL INTEGER PROGRAMMING APPROACH

*Christiane Schmidt, Tobias Andersson Granberg, Tatiana Polishchuk, Valentin Polishchuk, Communication and Transport Systems, ITN, Linköping University, Norrköping, Sweden*

## Abstract

We present a MIP-based airspace sectorization framework for Terminal Maneuvering Areas that can enforce convex sectors. The approach integrates an airspace complexity representation, and the resulting sectorizations have a balanced taskload. We present results for Stockholm TMA; and compare our results to convex sectorizations obtained by enumerating all possible topologies for a given number of sectors.

## Introduction

Over the last decades air traffic volumes have increased, and projections indicate that the growth will continue: the International Air Transport Association (IATA) [1] estimates that the number of passengers will double until 2034, and the Statistics and Forecasts (STATFOR) unit of EUROCONTROL [2] predicts an increase of 40-120% in flight movements in Europe from 2010 to 2030 (where the range stems from various scenarios from limited resources to strong economic growth). The increased volumes lead to congestion, in particular, in Terminal Maneuvering Areas (TMAs), that is, the airspace surrounding one or several aerodromes. An optimized design of the control sectors can increase capacity. The human factor is a major challenge for this design, as each sector is monitored by an air traffic controller (ATCO). The mental workload associated with working in such a complex system leads to the major constraints for sectorization: the workload should be balanced and below thresholds for every single ATCO. Moreover, the sector design must be valid w.r.t. the sector shape and how the sector boundaries interact with standard flows and critical points.

One common requirement put on sectorization, mentioned in a variety of sectorization papers (see, for example, [3],[4],[5]), is that the resulting sectors should be convex. The rationale behind desiring the convexity is that convex shapes are easy to "grasp"

(learn, comprehend) by human controllers. Moreover, a (straight-line) flight cannot enter and leave a convex sector multiple times, which is a useful property of airspace design, since changing sectors involves communication overhead and switching the radio frequency. Many authors suggested automatic design methods for sectorization, but the vast majority cannot enforce the sector convexity. (The exception are Voronoi-diagram methods, where a set of points are chosen as Voronoi sites and the sectors are their Voronoi cells; however, such methods can provide only rather crude control over the sector boundaries and do not give enough flexibility to ensure balanced workload among the sectors.) In addition, most of the sectorization work concentrated on en-route airspace.

In this paper, we present sectorization methods that allow one to incorporate the convexity constraint, while also taking care of all "usual" load balancing requirements. We apply our techniques to split Stockholm terminal maneuvering area (TMA) into convex sectors of approximately the same workload. To obtain our solution, we use integer programming (IP). Formulating the sectorization problem as an IP is by far not new—many earlier works employed IP or constraint programming to produce sectors. However, all such prior work used synthesis methods in which the IPs had a variable per elementary piece of airspace, the IP solution glued the pieces together to form the sectors. In contrast, we use a novel IP formulation in which there is a variable per potential edge of the sector boundary. Our IP is an extension of a prior framework that integrates an airspace complexity representation in the computation of a workload-balanced sectorization [6].<sup>1</sup>

Our grid-based MIP formulation discretizes the search space by laying out a square grid in the airspace, and connecting each node to its 8 neighbors. We can easily enforce any of the options (a) general

<sup>1</sup>An extended abstract of the complete MIP (with convexity constraints) appears in the informal workshop EuroCG 2017 [7].

sectors and (b) convex sectors. To the best of our knowledge, no other sectorization approach generates this flexibility, e.g., Gerdes et al. [8] start with convex Voronoi cells, and then adapt the cells to be non-convex. Adding the convexity constraint in the IP is possible, but we only have a limited number of edge directions, which yields that each sector is at most an octagon. In addition, it is computationally expensive for many sectors. If we restrict ourselves to TMAs, which have a small number of sectors, we observe that only a limited number of topologies is possible. Thus, we compare to enumerating all these topologies and presenting an operationally feasible solution.

### ***Roadmap***

In the remainder of this section we review related work. In the following section, we discuss taskload and workload, and formally introduce our problem in the section thereafter. Next we review our grid-based mixed integer programming (MIP) formulation of the sectorization problem with various constraints. Then we present the constraints for convex sectors, and review the complete MIP. The next section shows how the low number of TMA sectors allows us to enumerate the best balanced solutions for all different topologies. Then we present experiments for both the IP and the topologies for Stockholm TMA. The paper ends with conclusion and discussion.

### ***Related work***

Various papers considered automated airspace sectorization, for an extensive survey see Flener and Pearson [3]. Most research concentrated on sectorization of en-route airspace. Authors used fairly different definitions of taskload/workload, as no universal workload metric has been agreed on so far. US-based studies often have a focus on convective weather, which plays a smaller role in European studies.

The approaches can mostly be split in graph- or region-based models, where the former builds on partitioning a graph representing existing trajectories and constructing sector boundaries based on the partition, and the latter partitions the airspace into regions. Various sets of constraints on the resulting sectorization are considered, including constraints on the workload, the sector's geometry and size, and on the interaction with routes, e.g., each trajectory must intersect a sector for a minimum distance.

Kostitsyna [9] proves that most formulations of the airspace sectorization problem are NP-hard. In addition, she presents a method to redesign sectors that improves a given sectorization by locally adjusting sector boundaries.

Sabhnani et al. [4] present a flow conforming design, where they consider constraints on flow-sector boundary crossings, on flow-flow crossings, on convexity, and constraints that forbid too acute sector angles; in addition, they integrate constraints on the interplay with Special Use Airspace (SUA). The authors then use a discretized search space—a uniform grid plus nodes approximating the medial axis of standard flows—, and search for cuts in the complete graph on these nodes that conform to the constraints. Recently, Gerdes et al. [8] presented an approach that first clusters flight data using fuzzy clustering, then computes a Voronoi diagram based on the resulting cluster centers, and then takes respect to the controller workload using an evolutionary algorithm. They in particular make sure that the convex Voronoi cells are also able to handle non-convexity.

Other authors also presented geometry-based approaches. Xue's [10] design extends a pure Voronoi diagram computation. Brinton et al. [11] give a three stage algorithm that grows cells into clusters and straightens out boundaries in the final step. Their workload definition is based on dynamic density, see Kopardekar and Magyarits [12] for a comparison of four different dynamic density metrics. Conker et al. [13] present another three stage algorithm that uses a modified  $k$ -means clustering to obtain an initial sectorization, followed by an SLS heuristic to improve the workload balance and a final phase that straightens out sector boundaries.

Gianazza [14] combines elementary airspace modules for sectors, predicting the workload of configurations with a neural network. Leiden et al. [15] give a method based on a monitor alert parameter: once this threshold is exceeded for a sector it is split, and the algorithm picks the better of the solutions from a greedy bottom-up and a greedy top-down phase. The authors evaluate their sectorizations based on a transition cost, but do not include that cost in computing new sectors. Both Bloem and Gupta [16] and Kulkarni et al. [17] propose approximate dynamic programming methods. Drew [18] gives a MIP-based procedure that combines under-utilized sectors to

larger sectors, without integrating shape constraints.

An approach using constraint programming is presented by Jägare [19]: hexagonal cells are merged to build sectors using constraints on the workload, entry points etc..

A graph-based constraint programming method is presented by Trandac et al. [5]. We refer to the survey of Allignol et al. [20] for constraint programming approaches in air traffic management. Various further graph-based approaches have been proposed. One of the oldest works on sectorization by Delahaye et al. [21] uses 3D Voronoi diagrams in postprocessing to construct the sector boundaries. Martinez et al. [22] assign spatial cells to the nodes of a flow network, and partition the flow network subsequently until all subgraphs comply with an upper bound on the workload (measured as a peak traffic count).

Zelinski and Li [23] present a comparison of seven algorithms to a baseline, the current solution: they compare the delay resulting from the constructed sectorization, and the traffic pattern and reconfiguration complexity. All but one algorithm achieve better results for the delay.

In contrast to all these approaches, we focus on a sectorization of a TMA, which has fewer sectors than the en-route airspace of air traffic control centers (ATCC). Thus, it is feasible to choose an approach that may become more computationally complex with increased number of sectors. On the other hand, our method can integrate various constraints, in particular, in future work we aim to show that it incorporates a spectrum of constraints for convexity (from strictly enforcing convex sectors as shown here, over limiting the angle deviation from strictly convex vertices, to allowing arbitrary vertices with the original IP), and is flexible as it can be based on different complexity representations.

## Taskload/Workload

ATCOs must first of all ensure safe separation of aircraft (i.e., ensure a minimum safety distance between aircraft). In addition, they enable aircraft to reach their destinations in a timely manner. To do so, they permanently monitor air traffic, anticipate and detect (potential) conflicts and perform various other tasks that contribute to the airspace's complexity and drive an ATCO's mental workload. Both taskload and

workload reflect the demand of the air traffic controller's monitoring task (the latter measures objective demands, the former reflects the subjective demand experienced during a task); in the remainder of this paper, we will refer to both as taskload. Loft et al. [24] give an overview on different methods that study the elaborate problem of determining the taskload associated with a sector.

Recently, Zohrevandi et al. [25] presented a model for relating ATCO's taskload to the airspace complexity. They quantify the taskload as a weighted combination of ATCOs' clicks on the radar screen (weight  $\approx$  time for the task). Using linear regression the authors were able to explain terminal airspace complexity, given by eight complexity factors, about 70% better than the model by Djokic et al. [26] who used controller pilot data link communication and controller-pilot voice communications (regression analysis factor  $R^2 = 0.84$ ). Thus, the weighted radar screen clicks is a very good model for terminal airspace complexity. Zohrevandi et al. presented heat maps that visualize the density of weighted clicks. We use the heat maps for weighted clicks ([25]) as input for our sectorization. Our model does not depend on these specific maps, it is a general model that integrates complexity. In particular, it can be used as a building block in a common design of routes (with resulting complexity) and sectors.

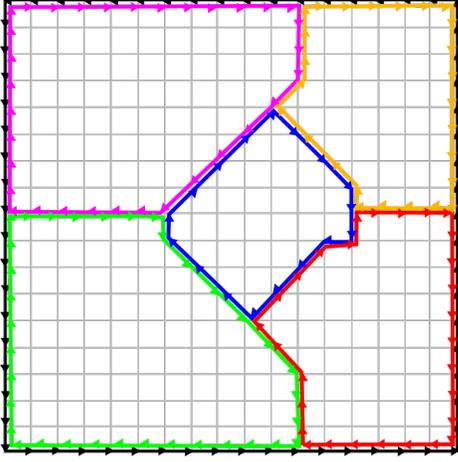
## Notation and Preliminaries

A *simple polygon*  $P$  is given by a set of  $n$  vertices  $v_1, v_2, \dots, v_n$  and  $n$  edges  $v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1$  such that no pair of non-consecutive edges share a point.  $P$  is the closed finite region bounded by the vertices and edges. A *sectorization* of a simple polygon  $P$  is a partition of the polygon  $P$  into  $k$  disjoint subpolygons  $S_1 \dots S_k$  ( $S_i \cap S_j = \emptyset \ \forall i \neq j$ ), such that  $\cup_{i=1}^k S_i = P$ . The subpolygons  $S_i$  are called *sectors*.

### Sectorization Problem:

**Given:** The coordinates of the TMA, defining a polygon  $P$ , the number of sectors  $|\mathcal{S}|$ , a set  $\mathcal{C}$  of constraints on the resulting sectors, and possibly an objective function  $\mathcal{F}$ .

**Find:** A sectorization of  $P$  with  $k = |\mathcal{S}|$ , fulfilling all constraints in  $\mathcal{C}$ , and possibly optimizing  $\mathcal{F}$ .



**Figure 1.** Artificial sector  $S_0$  (black) and a sectorization with  $|\mathcal{S}| = 5$ . Edges are slightly offset to enhance visibility.

## Grid-based MIP formulation

We start with a review of our prior MIP-formulation for airspace sectorization ([6],[7]).

We discretize the search space by laying out a square grid in the TMA. Every grid node has directed edges to its 8 neighbors ( $N(i) =$  set of neighbors of  $i$  (including  $i$ )), resulting in a bidirected graph  $G = (V, E)$ , i.e., for any two neighbors  $i$  and  $j$  both edge  $(i, j)$  and  $(j, i)$  are included in the edge set  $E$ . The length of an edge  $(i, j) \in E$  is denoted by  $\ell_{i,j}$ .

The main idea for the sectors is to use an artificial sector,  $S_0$ , that encompasses the complete boundary of  $P$ , using all counterclockwise (ccw) edges. That is, we use sectors in  $\mathcal{S}^* = \mathcal{S} \cup S_0$  with  $\mathcal{S} = \{S_1 \dots S_k\}$ . For all edges  $(i, j)$  used for boundary of any sector, we enforce that also the opposite edge,  $(j, i)$ , is used for another sector, see Figure 1. Thus, all edges of an (interior) sector are clockwise (cw).

Our model is an integer program, which in general is NP-hard, but we are able to solve relevant instance sizes. We use decision variables  $y_{i,j,s}$ , where  $y_{i,j,s} = 1$  indicates that edge  $(i, j)$  is a boundary edge for sector  $s$ . We add:

$$y_{i,j,0} = 1 \quad \forall (i, j) \in S_0 \quad (1)$$

$$\sum_{s \in \mathcal{S}^*} y_{i,j,s} - \sum_{s \in \mathcal{S}^*} y_{j,i,s} = 0 \quad \forall (i, j) \in E \quad (2)$$

$$y_{i,j,s} + y_{j,i,s} \leq 1 \quad \forall (i, j) \in E, \forall s \in \mathcal{S}^* \quad (3)$$

$$\sum_{s \in \mathcal{S}^*} y_{i,j,s} \leq 1 \quad \forall (i, j) \in E \quad (4)$$

$$\sum_{(i,j) \in E} y_{i,j,s} \geq 3 \quad \forall s \in \mathcal{S}^* \quad (5)$$

$$y_{i,j,s} \in \{0, 1\} \quad \forall (i, j) \in E, \forall s \in \mathcal{S}^* \quad (6)$$

Equation (1) ensures that all ccw boundary edges belong to  $S_0$ . Consistency between edges is given by Equation (2): if  $(i, j)$  is used for some sector, edge  $(j, i)$  has to be used as well. Equation (3) ensures that a sector cannot contain both edges  $(i, j)$  and  $(j, i)$ , that is, enclose an area of zero. Together with Equation (2) it ensures that if an edge  $(i, j)$  is used for sector  $S_\ell$ , the edge  $(j, i)$  has to be used by some sector  $S_x \neq S_\ell$ . Equation (4) enforces that one edge cannot participate in two sectors. Equation (5) enforces a minimum size for all sectors. Moreover, we add constraints on the degree of vertices on sector boundaries:

$$\sum_{l \in V: (l,i) \in E} y_{l,i,s} - \sum_{j \in V: (i,j) \in E} y_{i,j,s} = 0 \quad \forall i \in V, \forall s \in \mathcal{S}^* \quad (7)$$

$$\sum_{l \in V: (l,i) \in E} y_{l,i,s} \leq 1 \quad \forall i \in V, \forall s \in \mathcal{S}^* \quad (8)$$

Equation (7) ensures that indegree and outdegree coincide for all nodes. By Equation (8) a node has at most one incoming edge per sector.

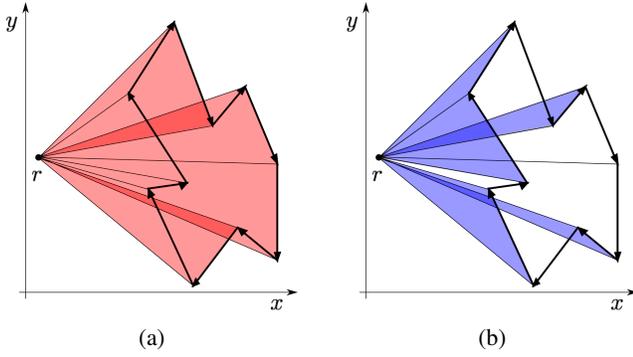
Constraints (1)-(8) guarantee that the union of the  $|\mathcal{S}|$  pairwise disjoint sectors completely covers the TMA. Of course, there are various other constraints for a sectorization, see for example the survey article of Flener and Pearson [3]. The constraints we consider can roughly be split in two categories: balancing and geometric.

### Balancing Constraints

Balancing constraints can be related to two factors: size/area and taskload. We consider:

- a) *Balanced taskload*: The taskload of each sector, and, thus, of each ATCO, must be balanced out with the taskload of other sectors.

To integrate constraint a, we need to be able to associate a taskload with a sector. As a first step,



**Figure 2. Area of polygon  $P$  (bold):** each edge of  $P$  forms an oriented triangle with a reference point  $r$ . Cw triangles contribute positive (a), ccw triangles negative (b).

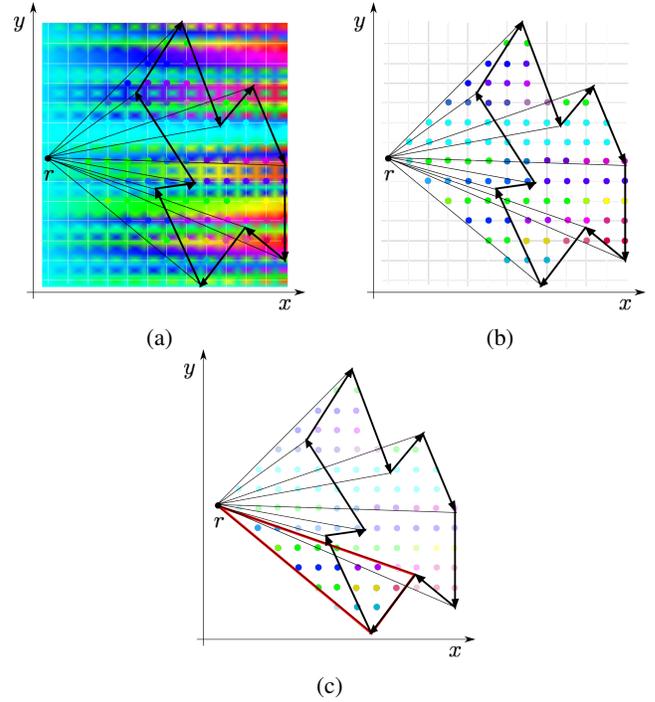
we need to assign an area to the sector selected by the boundary edges. The area of a polygon  $P$  with rational vertices is rational, and can be computed efficiently (see Fekete et al. [27]): we introduce a reference point  $r$ , and compute the area of the triangle of each directed edge  $e$  of  $P$  and  $r$ , see Figure 2. We then sum up the triangle area for all edges of  $P$ : cw and ccw triangles contribute positive and negative, respectively. Let  $f_{i,j}$  denote the signed area of the triangle formed by  $(i,j)$  and  $r$ .

$$\sum_{(i,j) \in E} f_{i,j} y_{i,j,s} - a_s = 0 \quad \forall s \in \mathcal{S}^* \quad (9)$$

$$\sum_{s \in \mathcal{S}} a_s = a_0 \quad (10)$$

Equation (9) assigns the area of sector  $s$  to the variable  $a_s$ , Equation (10) ensures that the sum of the  $a_s$ 's equals the area of the complete TMA.

Now we are ready to associate a taskload with a sector. Here, we assume that a heatmap representing the controller's taskload is given. Given this heatmap we overlay it with a grid, see Figure 3(a), extract the value at the grid points, see Figure 3(b), and use this discretized heatmap, see Figure 3(c), for further computations. We associate each discrete heatmap point,  $q$ , with a "heat value",  $h_q$ . Again, we consider triangles for each directed edge  $(i,j)$  of  $P$  and the reference point  $r$ , see, e.g., Figure 3(c): we sum up the heat values for all grid points within the triangle. The sign of the heat value for a triangle is determined by the sign of  $f_{i,j}$ , denoted by  $p_{i,j}$ , e.g., the triangle highlighted in Figure 3(c) is oriented



**Figure 3. Heat value extraction for a triangle:** (a) (Artificial) Heat map overlaid with a grid, (b) heat values extracted at grid points. (c) Shows the discretized heat map for the area of interest for  $P$ : the heat values at grid points for all grid points within some triangle of an edge  $e$  of  $P$  and the reference point  $r$ . The highlighted triangle is cw, thus, also its heat value is positive.

ccw (indicated by the red boundary), its heat value is positive ( $p_{i,j} = +1$ ). Let  $h_{i,j}$  denote the signed heat value of the triangle formed by edge  $(i,j)$  and  $r$ , that is:

$$h_{i,j} = p_{i,j} \sum_{q \in \Delta(i,j,r)} h_q.$$

If the taskload is of interest, we add Equation (11), which assigns each sector  $s$  a taskload  $t_s$ . We add Equation (12) to achieve a balanced taskload.  $t_{LB}$  can be chosen as a constant. Here, we use  $t_{LB} = c_2 \cdot t_0 / |\mathcal{S}|$  with, e.g.,  $c_2 = 0.9$ .

$$\sum_{(i,j) \in E} h_{i,j} y_{i,j,s} - t_s = 0 \quad \forall s \in \mathcal{S}^* \quad (11)$$

$$t_s \geq t_{LB} \quad \forall s \in \mathcal{S} \quad (12)$$

### Geometric Constraints

- b) *Connected sectors:* A sector must be a connected portion of airspace, see Flener and Pearson [3],

that is, from each point in a sector each other point in a sector must be reachable via a path that runs only in the same sector.

- c) *Nice shape*: A sector should have a smooth boundary and an easily memorable shape, see [3].
- d) *Convex sectors*: The sectors should be convex: convexity can be defined either geometrically, that is, for any pair of points in the sector the straight line connection between these points is also fully contained in the sector, or trajectory-based, i.e., no route enters the same sector more than once, see Flener and Pearson [3]. This was not included in our original IP: We present the necessary IP constraints in the next section.
- e) *Interior conflict points*: Points that require increased attention from ATCOs should lie in the sector's interior.

For **b** we include the length of the sector boundary in the objective function. For constraint **e** we cannot use an absolute threshold heat value for the complexity of points on the sector boundary: we like to enforce points of relatively high airspace complexity, represented by heat values, to be in the interior. Again, we use the objective function.

We take care of constraint **c** in postprocessing: Given constraint set  $\mathcal{C}$ ,  $c \in \mathcal{C}$ , we solve the IP with  $\mathcal{C} \setminus \{c\}$  and then use shortcuts by removing vertices as long as the constraints in  $\mathcal{C} \setminus \{c\}$  are not violated.

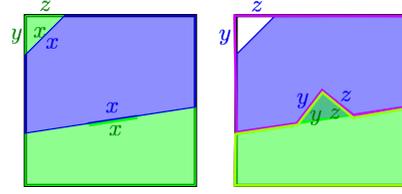
### Objective Function $\mathcal{F}$

As opposed to most optimization approaches, in our case, it is not obvious what kind of objective function should be used. Cost functions used in literature, cp. [3], are, e.g., taskload imbalance (constraint **a**), and number of sectors (which we consider as input). Because no obvious objective function exists, we consider different functions, all of which integrate constraint **b**. Our basic objective function is:

$$\min \sum_{s \in \mathcal{S}} \sum_{(i,j) \in E} \ell_{i,j} y_{i,j,s} \quad (13)$$

If we have  $a \notin \mathcal{C}$ , that is, we are not interested in the sector taskload, objective function (13) ensures that sectors are connected, that is, we take care of constraint **b**, see Figure 4.

If we consider taskload, objective function (13) only yields connected sectors if  $c_2$  in  $t_{LB} = c_2 \cdot t_0 / |\mathcal{S}|$



**Figure 4. Disconnected sectors are not optimal for (13). The sectors must completely cover the TMA. Assume there is a disconnected sector, like the green sector in the left, we can merge and decrease the total perimeter, we have:  $(y+z+2x)+(2x) \leq (y+z) + (2y+2z)$  by triangle inequality.**

of constraint (12) allows it: for example  $c_2 = 0.9$  may not allow a “ $c_2$ -balanced” sectorization with connected sectors, but if we allow for larger disparities between sectors, making a connected solution feasible by lowering the parameter, e.g.,  $c_2 = 0.7$ , we again obtain connected sectors. Essentially, this translates to: given the current complexity map a user must allow larger imbalances between controller’s taskload, if having connected sectors is a necessary condition.

*Integration of Constraint **e***: If  $e \in \mathcal{C}$  we use the following objective function (an extension of the basic objective function (13)):

$$\min \sum_{s \in \mathcal{S}} \sum_{(i,j) \in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j}) y_{i,j,s}, \quad 0 \leq \gamma < 1 \quad (14)$$

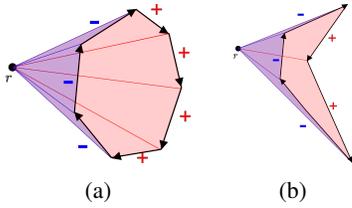
Where  $w_{i,j}$  represents an edge weight that depends on the heat-values of its endpoints. We choose one of:

- (I)  $w_{i,j} = h_i + h_j$ .
- (II)  $w_{i,j} = \sum_{k \in N(i)} h_k + \sum_{l \in N(j)} h_l$ .

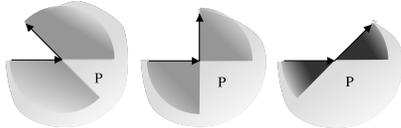
(I) ensures that relatively large heat-values are not located on the sector boundary, (II) pushes larger values further into the interior. An alternative to using objective function (14) instead of objective function (13) is to use a constraint with an upper bound  $W$ . This shows that we obtain an optimal connected solution, if, given  $c_2$  and  $W$ , there exists a feasible connected solution.

### Constraint $\delta$ : Convex Sectors

We can easily integrate convexity in our approach—a feature many other optimization approaches lack. For a convex sector, constraint  $\delta$ , there

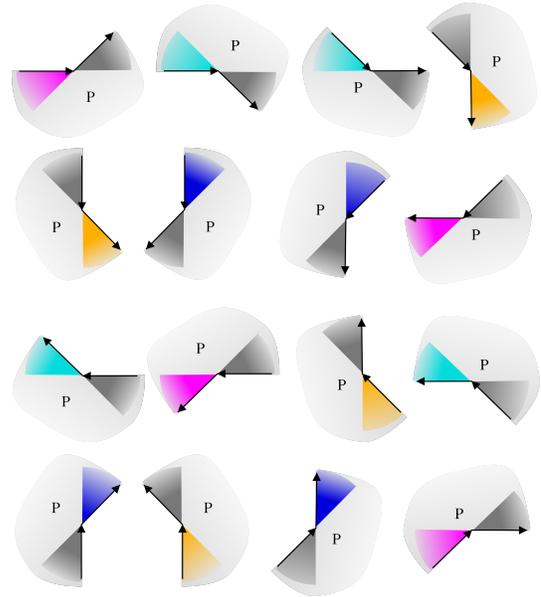


**Figure 5. A convex (a) and a non-convex (b) polygon.**

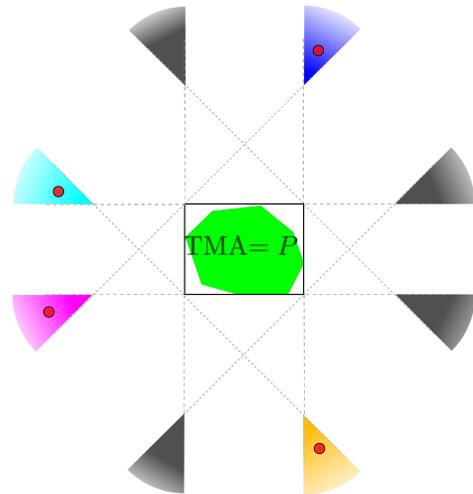


**Figure 6. Three outgoing edge directions yield a non-convex polygon (interior of  $P$  below ingoing edge).**

exist only one connected chain of edges with cw triangles, and one connected chain of edges with ccw triangles, see Figure 5(a). Unfortunately, the only-if part of that statement is not true, see Figure 5(b): there are non-convex polygons with a single chain of edges with ccw triangles and a single chain of edges with cw triangles. We can make use of the fact that we have only eight edge directions. For every direction of an incoming edge, there are three directions of outgoing edges that are forbidden in a convex polygon, see Figure 6: there exist two open cones (indicated in gray) in which a reference point must be located to detect the switch. Thus, any reference point located in the dark gray cones, both the intersection of three cones, yields a switch in the triangle orientation. If we consider all possible edge directions (eight incoming edge directions with two outgoing edge directions each for the dark gray cones from Figure 6), the cones for the necessary directions overlap, see Figure 7. Thus, we only need four points located in the intersection of these cones for all points of the grid, see Figure 8: we place one reference point in each of the four colored cones in Figure 8, and denote them by  $r_1, \dots, r_4$  ( $r = r_m$ , for some  $m \in \mathcal{M} = \{1, \dots, 4\}$ ). At least one of the  $r_m$  will result in a cw/ccw switch for non-convex polygons. Let  $p_{i,j,m}$  denote the sign of the triangle of the edge  $(i, j)$  and reference point  $r_m$ ,  $m \in \mathcal{M}$ . We add:



**Figure 7. Four (colored: pink, turquoise, blue, and orange) cones can be used for all edge directions.**



**Figure 8. Four red reference points for detecting non-convexity. The TMA, the polygon, is shown in green, the black box depicts the polygon's axis-aligned bounding box.**

$$q_{j,m}^s = \frac{1}{2} \left( \sum_{i:(i,j) \in E} p_{i,j,m} y_{i,j,s} - \sum_{l:(j,l) \in E} p_{j,l,m} y_{j,l,s} \right) \quad (15)$$

$\forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M}$

This constraint assigns, for each sector, a value of -1,0,1 to each vertex. An interior vertex of either a chain of cw or ccw triangles has  $q_{j,m}^s = 0$ ; if at  $j$  a chain with ccw (cw) triangles switches to a chain of cw (ccw) triangles  $q_{j,m}^s = -1$  ( $q_{j,m}^s = 1$ ). For a convex sector  $s$ , the sum over the  $|q_{j,m}^s|$  for all sector vertices  $j$  is 2 for all reference points  $r_m$ ; for non-convex sectors this value is larger than 2 for at least one reference point  $r_m$ . Equations (16), (17) define the absolute values. To enforce convexity (18) must hold. But, as two variables are multiplied in Equation (18), we cannot add it to the IP. Instead, we use Equations (19)-(22) to define variables  $z_{i,j,m}^s = y_{i,j,s} \cdot qabs_{j,m}^s \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M}$ , and add Equation (23), a reformulation of Equation (18).

$$qabs_{j,m}^s \geq q_{j,m}^s \quad \forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M} \quad (16)$$

$$qabs_{j,m}^s \geq -q_{j,m}^s \quad \forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M} \quad (17)$$

$$\sum_{i \in V} \sum_{j \in V} y_{i,j,s} \cdot qabs_{j,m}^s = 2 \quad \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (18)$$

$$z_{i,j,m}^s \geq 0 \quad \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (19)$$

$$z_{i,j,m}^s \leq qabs_{j,m}^s \quad \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (20)$$

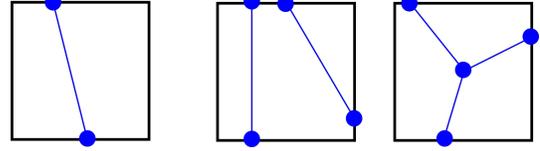
$$z_{i,j,m}^s \leq y_{i,j,s} \quad \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (21)$$

$$z_{i,j,m}^s \geq y_{i,j,s} - 1 + qabs_{j,m}^s \quad \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (22)$$

$$\sum_{i \in V} \sum_{j \in V} z_{i,j,m}^s = 2 \quad \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (23)$$

## The Complete MIP

To enhance readability, we present the complete MIP in the end of this paper, see Figure 12.



**Figure 9. Different topologies. Left:**  $|\mathcal{S}| = 2$ . **Middle, Right:**  $|\mathcal{S}| = 3$ .

## Topologies

In a TMA only few sectors are needed, and only a limited number of topologies must be checked for a given number of convex sectors. Because adding the convexity constraint to the IP is computationally expensive, we compare our IP results for the convex sectorization of Stockholm TMA to those obtained by computing workload balanced convex sectorizations by enumeration. That is, for each topology we compute the best balanced solution that fulfills all constraints in  $\mathcal{C}_0 = \{\alpha, \beta, \gamma, \delta\}$ . In particular, this approach does not have the constraint of limited (grid) edge directions, on the other hand the number of topologies to check rapidly increases with  $|\mathcal{S}|$ . Moreover, because we aim for perfect balance, the solutions may be worse than the IP solutions w.r.t. constraint  $\epsilon$ .

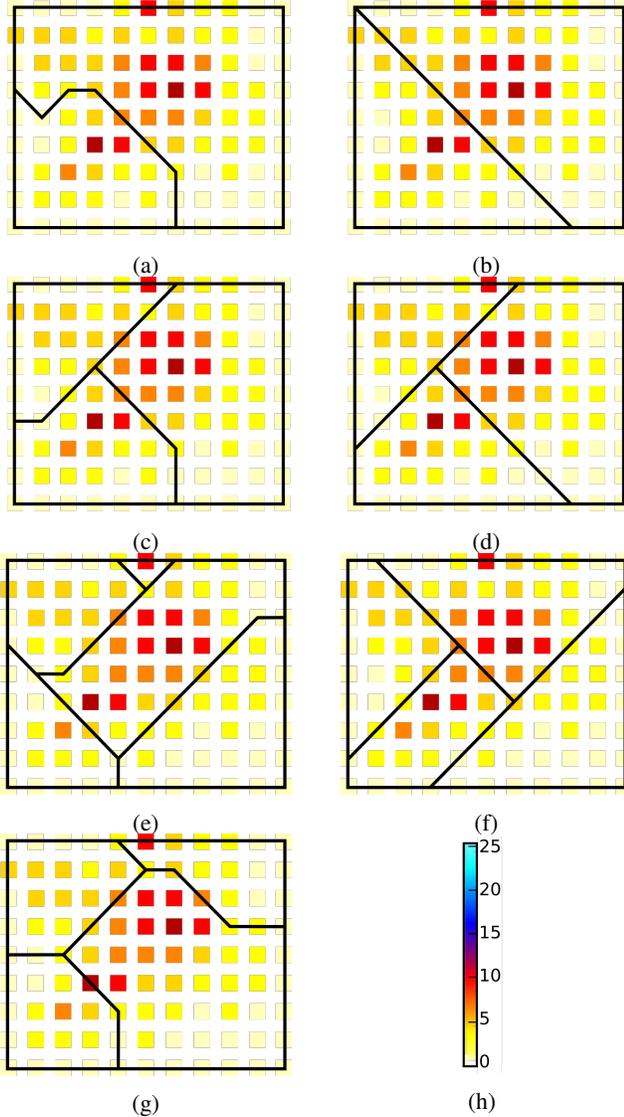
For  $|\mathcal{S}| = 2$  we search for the best chord that connects any two points on the TMA boundary, for  $|\mathcal{S}| = 3$  we need to determine the location of four points, either all of them on the boundary (for two chords), or one center point connecting to the sector boundary in a Y-shape, see Figure 9.

## Experimental Study: Arlanda Airport

The model was solved using AMPL and CPLEX 12.6 on a single server with 24GB RAM and four kernels running on Linux. Each instance was run until a solution with less than 1% gap had been found. The computation times varied from a few seconds up to several days. More sectors, and the convexity constraints made the problem harder to solve.

### A. Sectorizations with and without Convexity Constraint

We first compare sectorizations without the convexity constraints to sectorizations with the convexity constraints. Figure 10, left column, shows sectorizations without the convexity constraints, while the



**Figure 10. Sectorizations obtained by IP. Left column: Sectorizations without convexity constraints. Right column: Convex sectorizations. Left column:  $\mathcal{C}_2 = \{a, b, c, e\}$ . Right column:  $\mathcal{C}_3 = \{a, b, c, d, e\}$ . All with  $c_2 = 0.6$  and  $w_{i,j} = h_i + h_j$ . (a)-(f):  $\gamma = 0.2$ , (g):  $\gamma = 0.8$ . (a), (b):  $|\mathcal{S}| = 2$ , (c), (d):  $|\mathcal{S}| = 3$ , (e)-(g):  $|\mathcal{S}| = 4$ . The sectorization in (e) has a disconnected sector. (h) Color scale for heat values.**

right column shows sectorizations with the convexity constraints. That is, the right column depicts sectorizations obtained by adding constraints (16), (17), and (19)-(23) to the IP (that is, using  $\mathcal{C}_3 = \{a, b, c, d, e\}$ ). We can observe, that for  $|\mathcal{S}| = 4$ ,  $\gamma = 0.2$ , and  $w_{i,j} = h_i + h_j$ , that is, Figure 10(e), the sectorization does not have connected sectors, because of the extended objective function as explained earlier, compare Figure 4. If we increase  $\gamma$  to 0.8 in Figure 10(g), we obtain a connected sectorization with four sectors.

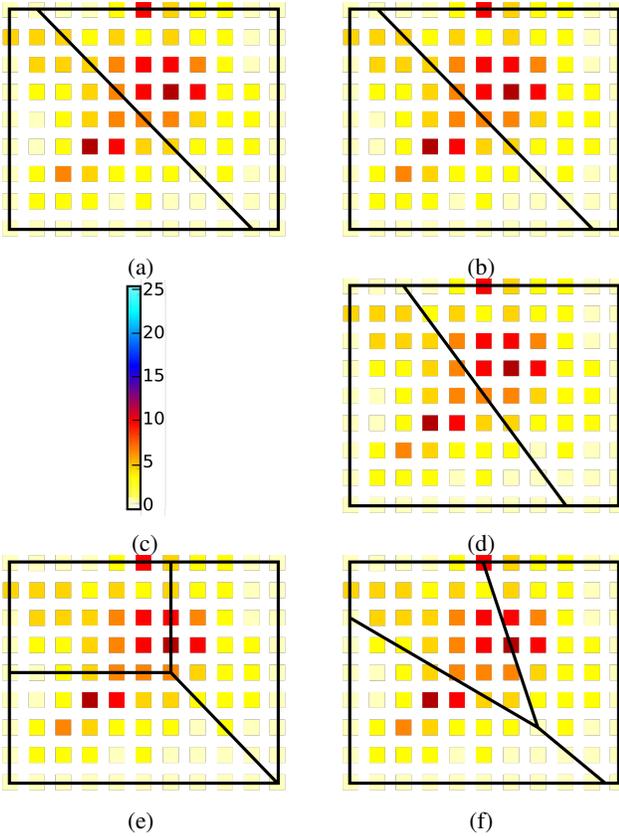
### B. Convex Sectorizations: IP and Topology Enumeration

Figure 11, left column, shows convex sectorizations obtained by adding the convexity constraints to the IP. Figure 11, right column, shows sectorizations with convex sectors obtained by enumerating all solutions for the possible topologies ( $\mathcal{C}_0 = \{a, b, c, d\}$ ).

The convex sectorizations in the right column of Figure 11 yield a perfect workload balance for two (with two equally good solutions) and three sectors. The comparison of Figure 11(a) and Figure 11(b) shows that the IP results in the perfectly balanced solution without any edge directions restriction. For larger values of  $|\mathcal{S}|$ , this restriction no longer gives the perfectly balanced solution, but for 3 sectors we present a solution with  $c_2 = 0.95$ , that is, the deviation between the sectors is very small. Moreover, the perfectly balanced solutions may be less attractive for constraint  $\epsilon$ , that is, the location of hot spots.

## Conclusion and Discussion

In this paper we extended our sectorization method that balances sector taskload (based on a complexity representation): we enforce convex sectors. We apply our techniques to sectorize Stockholm Terminal Maneuvering Area (TMA). As in a TMA only a few sectors are needed, and adding the convexity constraint is computationally expensive, we compare our results to convex sectorizations obtained by enumerating all possible topologies for a given number of sectors. Our IP method is highly flexible, and allows a fine-grained view on the TMA. It also constitutes the first step towards an integrated design of aircraft routes, the induced complexity, and the sectors. In future work, we aim to show that we can limit the angle deviation from only convex vertices (the



**Figure 11. Convex sectorizations. Left column: Convex sectorizations obtained by IP ( $\mathcal{C}_3 = \{a, b, c, d, e\}$ ). Right column: Convex sectorizations obtained by enumerating the topologies ( $\mathcal{C}_0 = \{a, b, c, d\}$ ). (a), (b), (d):  $|S| = 2$ ; (e),(f):  $|S| = 3$ . (a), (e):  $\gamma = 0.5$  and  $w_{i,j} = h_i + h_j$ , (a)  $c_2 = 0.9$ , (3)  $c_2 = 0.95$ . All sectorizations obtained by enumeration with perfect workload balance. (c) Color scale for heat values.**

rationale behind the latter being that an exemplary sectorization as shown in Figure 1 is acceptable). That is, the IP formulation will allow us not only to strictly enforce the convexity of the sectors, but also to allow having a limited angle deviation from only convex vertices per sector. Thus, we will be able to easily enforce any of the options (a) general sectors, (b) sectors with an upper bound on angle deviation from only convex vertices, and (c) convex sectors.

## Acknowledgments

This research is funded by grant 2014-03476 (ODESTA: Optimal Design of Terminal Airspace)

from Sweden’s innovation agency VINNOVA and in-kind participation of LFV. We thank Billy Josefsson (project co-PI, LFV) and the members of the project reference group—Eric Hoffman (Eurocontrol), Håkan Svensson (LFV/NUAC), Anne-Marie Ragnarsson (Transportstyrelsen, the Swedish Traffic Administration), Anette Näs (Swedavia, the Swedish Airports Operator), Patrik Bergviken (LFV), Håkan Fahlgren (LFV)—for discussions of sectorization structure and design flexibility.

## Email Addresses

tobias.andersson.granberg@liu.se,  
tatiana.polishchuk@liu.se,  
valentin.polishchuk@liu.se,  
christiane.schmidt@liu.se

## References

- [1] IATA air passenger forecast shows dip in long-term demand, <http://www.iata.org/pressroom/pr/Pages/2015-11-26-01.aspx>, accessed on June 2, 2016.
- [2] “EUROCONTROL long-term forecast EUROCONTROL flight movements 2010 - 2030,” 2010.
- [3] P. Flener and J. Pearson, “Automatic airspace sectorisation: A survey,” *CoRR*, vol. abs/1311.0653, 2013.
- [4] G. R. Sabhnani, A. Yousefi, and J. S. Mitchell, “Flow conforming operational airspace sector design,” in *Proceedings of the 10th AIAA Aviation Technology, Integration and Operations (ATIO) Forum*, American Institute of Aeronautics and Astronautics, 2010.
- [5] H. Trandac, P. Baptiste, and V. Duong, “Airspace sectorization with constraints,” *RAIRO - Operations Research*, vol. 39, no. 2, pp. 105–122, 2005.
- [6] T. A. Granberg, T. Polishchuk, V. Polishchuk, and C. Schmidt, “A novel MIP-based airspace sectorization for TMAs,” Submitted for Publication, 2017.
- [7] T. A. Granberg, T. Polishchuk, and C. Schmidt, “A novel MIP-based airspace sectorization for TMAs,” in *EuroCG 2017*, 2017.
- [8] I. Gerdes, A. Temme, and M. Schultz, “Dynamic airspace sectorization using controller task load,” in *Sixth SESAR Innovation Days*, 2016.
- [9] I. Kostitsyna, “Balanced partitioning of polygonal domains,” PhD thesis, Stony Brook University, 2013.

$$\begin{aligned}
\min \sum_{s \in \mathcal{S}} \sum_{(i,j) \in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j}) y_{i,j,s} \quad & 0 \leq \gamma \leq 1 & (13)/(14) \\
\text{s.t.} & \\
\sum_{s \in \mathcal{S}^*} y_{i,j,s} - \sum_{s \in \mathcal{S}^*} y_{j,i,s} = & 0 & \forall (i,j) \in E \quad (2) \\
y_{i,j,s} + y_{j,i,s} \leq & 1 & \forall (i,j) \in E, \forall s \in \mathcal{S}^* \quad (3) \\
\sum_{s \in \mathcal{S}} y_{i,j,s} \leq & 1 & \forall (i,j) \in E \quad (4) \\
\sum_{(i,j) \in E} y_{i,j,s} \geq & 3 & \forall s \in \mathcal{S}^* \quad (5) \\
\sum_{l \in V: (l,i) \in E} y_{l,i,s} - \sum_{j \in V: (i,j) \in E} y_{i,j,s} = & 0 & \forall i \in V, \forall s \in \mathcal{S}^* \quad (7) \\
\sum_{l \in V: (l,i) \in E} y_{l,i,s} \leq & 1 & \forall i \in V, \forall s \in \mathcal{S}^* \quad (8) \\
\sum_{(i,j) \in E} f_{i,j} y_{i,j,s} - a_s = & 0 & \forall s \in \mathcal{S}^* \quad (9) \\
\sum_{s \in \mathcal{S}} a_s = & a_0 & (10) \\
\sum_{(i,j) \in E} h_{i,j} y_{i,j,s} - t_s = & 0 & \forall s \in \mathcal{S} \quad (11) \\
t_s \geq & t_{LB} & \forall s \in \mathcal{S} \quad (12) \\
p_{j,m}^s = & \frac{1}{2} \left( \sum_{(i,j) \in E} p_{i,j,m} y_{i,j,s} - \sum_{(j,l) \in E} p_{j,l,m} y_{j,l,s} \right) & \forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M} \quad (15) \\
qabs_{j,m}^s \geq & q_{j,m}^s & \forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M} \quad (16) \\
qabs_{j,m}^s \geq & -q_{j,m}^s & \forall s \in \mathcal{S}, \forall j \in V, \forall m \in \mathcal{M} \quad (17) \\
z_{i,j,m}^s \leq & qabs_{j,m}^s & \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (20) \\
z_{i,j,m}^s \leq & y_{i,j,s} & \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (21) \\
z_{i,j,m}^s \geq & y_{i,j,s} - 1 + qabs_{j,m}^s & \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (22) \\
\sum_{i \in V} \sum_{j \in V} z_{i,j,m}^s = & 2 & \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (23) \\
y_{i,j,0} = & 1 & \forall (i,j) \in \mathcal{S}_0 \quad (1) \\
y_{i,j,s} \in & \{0, 1\} & \forall (i,j) \in E, \forall s \in \mathcal{S}^* \quad (6) \\
z_{i,j,m}^s \geq & 0 & \forall i, j \in V \forall s \in \mathcal{S}, \forall m \in \mathcal{M} \quad (19)
\end{aligned}$$

**Figure 12. Complete MIP.**

[10] M. Xue, “Three dimensional sector design with optimal number of sectors,” in *Proceedings of the AIAA Conference on Guidance, Navigation, and Control (GNC) and Modeling and Simulation Technologies (MST)*, American Institute for Aeronautics and Astronautics, 2010.

[11] C. R. Brinton, K. Leiden, and J. Henkey, “Airspace sectorization by dynamic density,” in *Proceedings of the 9th AIAA Aviation Technology, Integration and Operations (ATIO) Forum*, American Institute of Aeronautics and Astronautics, 2009.

[12] P. Kopardekar and S. Magyarits, “Dynamic density: Measuring and predicting sector complexity,” in *Proceedings of the 21st Digital Avionics Systems Conference (DASC)*, vol. 1, 2002, pp. 2.C.4–1–2.C.4–9.

[13] R. S. Conker, D. A. Moch-Mooney, W. P. Niedringhaus, and B. T. Simmons, “New process for “clean” sheet airspace design and evaluation,” in *Proceedings of ATM 2007, the 7th USA/ Europe Air Traffic Management Research and Development Seminar*, 2007.

[14] D. Gianazza, “Forecasting workload and airspace configuration with neural networks and tree search methods,” *Artificial Intelligence*, vol. 174, no. 7, pp. 530–549, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370210000275>.

[15] K. Leiden, S. Peters, and S. Quesada, “Flight level-based dynamic airspace configuration,” in *Proceedings of the 9th AIAA Aviation Technology, Integration and Operations (ATIO) Forum*, American Institute of Aeronautics and Astronautics, 2009.

- [16] M. Bloen and P. Gupta, "Configuring airspace sectors with approximate dynamic programming," in *Proceedings of the 27th International Congress of the aeronautical sciences*, 2010.
- [17] S. Kulkarni, R. Ganesan, and L. Sherry, "Static sectorization approach to dynamic airspace configuration using approximate dynamic programming," in *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2011*, 2011, J2-1-J2-9.
- [18] M. C. Drew, "A method of optimally combining sectors," in *Proceedings of the 9th AIAA Aviation Technology, Integration and Operations (ATIO) Forum*, American Institute of Aeronautics and Astronautics, 2009.
- [19] P. Jägare, *Airspace sectorisation using constraint programming*, Report IT 11 021, Faculty of Science and Technology, 2011.
- [20] C. Allignol, N. Barnier, P. Flener, and J. Pearson, "Constraint programming for air traffic management: a survey," *Knowledge Engineering Review*, vol. 27, no. 3, pp. 361-392, Sep. 2012, <http://www.sherpa.ac.uk/romeo/issn/0269-8889/>. DOI: 10.1017/S0269888912000215. [Online]. Available: <https://hal-enac.archives-ouvertes.fr/hal-00934670>.
- [21] D. Delahaye, M. Schoenauer, and J.-M. Alliot, "Airspace sectoring by evolutionary computation," in *IEEE 1998, International Conference on Evolutionary Computation*, Anchorage, United States, May 1998, pp. 218-223. DOI: 10.1109/ICEC.1998.699504. [Online]. Available: <https://hal-enac.archives-ouvertes.fr/hal-00937715>.
- [22] S. A. Martinez, G. B. Chatterji, D. Sun, and A. M. Bayen, "A weighted-graph approach for dynamic airspace configuration," in *Proceedings of the AIAA Aviation Technology, Integration and Operations (ATIO) Forum*, American Institute of Aeronautics and Astronautics, 2007.
- [23] S. Zelinski and C. F. Lai, "Comparing methods for dynamic airspace configuration," in *Proceedings of the 30th Digital Avionics Systems Conference (DASC)*, 2011.
- [24] S. Loft, P. Sanderson, A. Neal, and M. Mooij, "Modeling and predicting mental workload in en route air traffic control: Critical review and broader implications," *Human Factors*, vol. 49, no. 3, pp. 376-399, 2007, ISSN: 0018-7208. DOI: 10.1518/001872007X197017.
- [25] E. Zohrevandi, V. Polishchuk, J. Lundberg, Å. Svensson, J. Johansson, and B. Josefsson, "Modeling and analysis of controller's taskload in different predictability conditions," in *6th SESAR Innovation Days*, 2016.
- [26] J. Djokic, B. Lorenz, and H. Fricke, "Air traffic control complexity as workload driver," *Transp. Res. Part C: Emerging Technologies*, vol. 18, no. 6, pp. 930-936, 2010, ISSN: 0968-090X. DOI: <http://dx.doi.org/10.1016/j.trc.2010.03.005>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0968090X10000318>.
- [27] S. P. Fekete, S. Friedrichs, M. Hemmer, M. Papenberg, A. Schmidt, and J. Troegel, "Area- and boundary-optimal polygonalization of planar point sets," in *EuroCG 2015*, 2015, pp. 133-136.

*2017 Integrated Communications Navigation  
and Surveillance (ICNS) Conference  
April 18-20, 2017*