Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)

A Novel MIP-based Airspace Sectorization for TMAs

Tobias Andersson Granberg, Tatiana Polishchuk, Valentin Polishchuk, Christiane Schmidt

Communications and Transport Systems, ITN
Linköping University
Norrköping, Sweden
{tobias.andersson.granberg, tatiana.polishchuk, valentin.polishchuk, christiane.schmidt}@liu.se

***Abstract*—We present a MIP-based airspace sectorization framework for Terminal Maneuvering Areas (TMAs). It incorporates an airspace complexity representation, as well as various constraints on the sectors' geometry, e.g., the requirement that points that demand increased attention from air traffic controllers should lie in the sector's interior to allow for enough time to resolve possible conflicts. In contrast to earlier integer/constraint programming approaches, which used synthesis methods with variables per elementary airspace piece that were glued together to form sectors, our IP formulation uses a variable per potential edge on the sector boundary. It is also the first step towards an integrated design of routes, the resulting complexity, and a sectorization. We present results for Stockholm TMA.**

Keywords—Air Traffic Management; Sectorization; Integer Programming; Balanced Taskload

I. 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 [10] 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 resulting congestion is particularly concentrated on Terminal Maneuvering Areas (TMAs), that is, the area surrounding one or more neighboring aerodromes, as traffic converges towards a point near the runway. This results in significant delays and operation expenses. A possibility to increase capacity is to use the existing resources more efficiently. Thus, an optimized design of the TMA control sectors is essential for coping with ever increasing numbers of aircraft movements. The constructed sectorization partitions the airspace into a certain number of sectors, where each sector is assigned to an air traffic controller.

A major challenge is posed by the central role of these humans-in-the-loop: each sector is monitored by an air traffic controller (ATCO) who guarantees safe separation of aircraft in the sector at all times. The mental workload associated with working in such a complex system gives rise to the major constraints of an airspace sectorization: the workload should be balanced and not exceed thresholds for every single ATCO. Moreover, any design must be valid w.r.t. the sector shape and how the sector boundaries interact with standard flows and critical points. The problem of accessing the workload associated with a sector and deciding when it exceeds a safe threshold is a complex question that has been investigated by several researchers, see Loft et al. [20] for an overview.

Today, the airspace layout at most airports is done manually, based on expert opinions. However, this will generally not yield an optimal sectorization for given criteria. Many papers presented suggestions for automatic design methods for airspace sectorization, see Section I-B for a detailed review, but the vast majority of approaches concentrates on en-route airspace.

In this paper, we present a general framework for integrating a complexity representation in a TMA's sectorization: we give a mathematical programming approach to compute optimal TMA sectorizations. The formulation as a mixed integer program (MIP) is by far not new—many earlier works employed integer or constraint programming to produce sectors, see Section I-B. However, all such prior work used synthesis methods: the IPs employed a variable per elementary airspace piece, and these pieces were glued together to constitute sectors. Our IP formulation, by contrast, uses a variable per potential edge of sector boundary. As input, we use the coordinates of the TMA, a numerical representation of the complexity, and constraints on the resulting sectorization. Our method is exact. Apart from computing optimal sectorizations based on a given complexity representation, this is also the first step towards an integrated design of routes, the resulting complexity, and a sectorization. Our approach is particular fitting for a TMA, as we—on the one hand—deal with a limited number of sectors, which makes solving our IP computationally feasible, but—on the other hand—we want to be able to have a fine-grained approach that encompasses a multitude of constraints representing the ATCO's operational requirements, and has polygonal sectors (in contrast to elongated sectors with highly parallel traffic

This research is funded by the grant 2014-03476 (ODESTA: Optimal Design of Terminal Airspace) from Sweden's innovation agency VINNOVA and in-kind participation of LFV.

patterns where aircraft enter/leave the sector from a common direction).

A. Roadmap

In the remainder of this section we review related work. In Section II we discuss taskload and workload, and formally introduce our problem in Section III. Section IV presents our main tool: a grid-based mixed integer programming (MIP) formulation of the sectorization problem with various constraints. In Section V we apply the MIP to Stockholm TMA using several constraint sets. We conclude in Section VI.

B. Related work

Various papers considered automated airspace sectorization, for an extensive survey see Flener and Pearson [12]. 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 [17] 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. [23] 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. [13] 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 [25] design extends a pure Voronoi diagram computation. Brinton et al. [5] 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 [16] for a comparison of four different dynamic density metrics. Conker et al. [6] 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. [19] 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 [4] and Kulkarni et al. [18] propose approximate dynamic programming methods. Drew [9] 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 [15]: 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. [24]. We refer to the survey of Allignol et al. [2] 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. [7] uses 3D Voronoi diagrams in postprocessing to construct the sector boundaries. Martinez et al. [21] assign spatial cells to the nodes of a flow network, and partition the flow network subsequently until all sub-graphs comply with an upper bound on the workload (measured as a peak traffic count).

Zelinski and Li [27] 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 centres (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 integrates various constraints, and is flexible as it can be based on different complexity representations.

II. Taskload/Workload

ATCOs must first and foremost ensure safe separation of 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, intervene to resolve them, communicate with pilots and ATCOs of neighboring sectors for handover, and perform various other tasks that contribute to the airspace's complexity and drive an ATCO's mental workload. Both workload and taskload reflect the demand of the air traffic controller's monitoring task: the taskload measures the objective demands, while the workload reflects the subjective demand experienced during that task. Loft et al. [20] give an overview on different methods that study the elaborate problem of determining the workload associated with a sector. Meckiff et al. [22] describe workload

as a function of the air traffic's geometry, the operational procedures and practices used to handle the air traffic, and the characteristics of individual ATCOs. Yousefi et al. [26] and Kopardekar and Magyarits [16] compare various air traffic management (ATM) modeling metrics and dynamic density metrics, respectively.

Recently, Zohrevandi et al. [29], [28] presented a novel model for relating the controller's taskload to the airspace complexity, represented by eight complexity factors. The authors use ATCO's clicks on the radar screen as a measure to quantify the taskload. All tasks are classified into four types that require different time periods to perform (from 2 seconds for background tasks to 50 seconds for control tasks), and the taskload is defined as a weighted combination of these clicks. The authors compared their taskload measure of (weighted) clicks with the model by Djokic et al. [8], who used controller pilot data link communication (CPDLC) and controller-pilot voice communications. Using a linear regression Zohrevandi et al. were able to explain airspace complexity, given by the eight complexity factors, about 40% better than the model by Djokic et al.; for terminal airspace they achieved an improvement of about 70% (regression analysis factor $R^2 = 0.84$). Thus, the weighted radar screen clicks are a very good model for terminal airspace complexity. The authors presented heat maps that visualize the density of weighted clicks. We use these heat maps as an input for our sectorization.

The radar screen clicks are closely related to aircraft trajectories in the considered airspace, thus, when we base our sectorization on the heat maps, we integrate the interaction with routes.

Note, that our model does not depend on these specific heat 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.

In the remainder of this paper, we will refer to both work- and taskload as taskload.

III. Notation and Preliminaries

A *simple polygon* P is given by a set of n vertices $v_1, v_2, \ldots, v_n$ and n edges $v_1v_2, v_2v_3, \ldots, v_{n-1}v_n, v_nv_1$ such that v_iv_j and v_kv_l for $j \neq k, i \neq l$ do not share a point. The polygon is the closed finite region bounded by the vertices and edges. The edges and vertices form the boundary of P. A *sectorization* is a partition of the polygon P into k disjoint subpolygons $S_1 \ldots S_k$, with $S_i \cap S_j = \emptyset \ \ \forall i \neq j$, 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 $|\mathcal{S}|$, and a set $\mathcal{C}$ of constraints on the resulting sectors.

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

IV. Grid-based MIP formulation for the sectorization problem

We discretize the search space by laying out a square grid in the TMA. Every grid node is connected to its 8 neighbors.

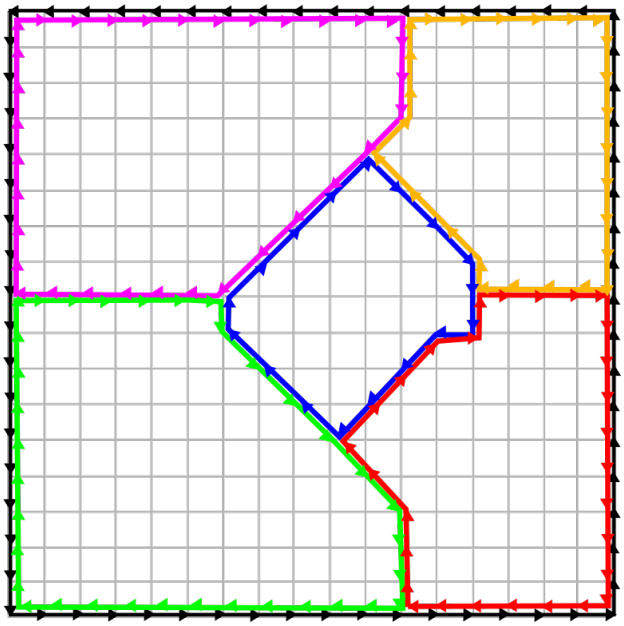


Fig. 1. The artificial sector S_0 is shown in black. A sectorization with five sectors is given. Edges are slightly offset to enhance visibility.

The resulting graph $G = (V, E)$ is bi-directed, 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 edges. That is, we use sectors in $\mathcal{S}^* = \mathcal{S} \cup S_0$ with $\mathcal{S} = \{S_1 \ldots S_k\}$. For all edges (i, j) used for sector boundary of any sector, we then 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.

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 the following constraints:

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

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

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

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

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

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

Equation (1) ensures that all counterclockwise oriented boundary edges belong to the artificial sector 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. In combination with

Equation (2) it ensures that if an edge (i, j) is used for sector S_ℓ, the edge (j, i) has to be used by some sector $S_k \neq S_\ell$. Equation (4) enforces that one edge (i, j) cannot participate in two sectors. Equation (5) enforces a minimum size for all sectors: each sector consists of at least 3 edges. Moreover, we add constraints on the degree of vertices on sector boundaries:

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

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

Equation (7) ensures balance in all nodes, that is, all nodes have the same number of ingoing and outgoing edges. Equation (8) enforces that for each sector a node can have at most one ingoing edge.

A. Sectorization Constraints

The constraints (1)-(8) guarantee that the union of the $|\mathcal{S}|$ sectors completely covers the TMA, that the sectors are pairwise disjoint, and that each of them has a non-zero area. Of course, there are various other constraints for a sectorization, see for example the survey article of Flener and Pearson [12]. The constraints we consider can roughly be split in two categories: **geometric** and **balancing** constraints. The former class could also incorporate interaction with routes.

1) Balancing Constraints: Balancing constraints are related to two factors: size/area and taskload. We consider the following constraints:

- a) *Balanced size*: The area of each sector, thus, the area that must be monitored by a single air traffic controller must be balanced out with the area of other sectors.
- b) *Bounded taskload*: There is an upper bound of movements that an air traffic controller can handle per time unit (hour), which might differ for air traffic controllers with varying experience. The taskload of each sector may not exceed this upper bound.
- c) *Balanced taskload*: The taskload of each sector, and, thus, of each air traffic controller, must be balanced out with the taskload of other sectors.

For constraint a, balanced size, we need to be able to associate an area with 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. [11]): 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 the polygon: clockwise (cw) triangles contribute positive, counterclockwise (ccw) triangles contribute negative. Let $f_{i,j}$ denote the signed area of the triangle formed by edge (i, j) and r.

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

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

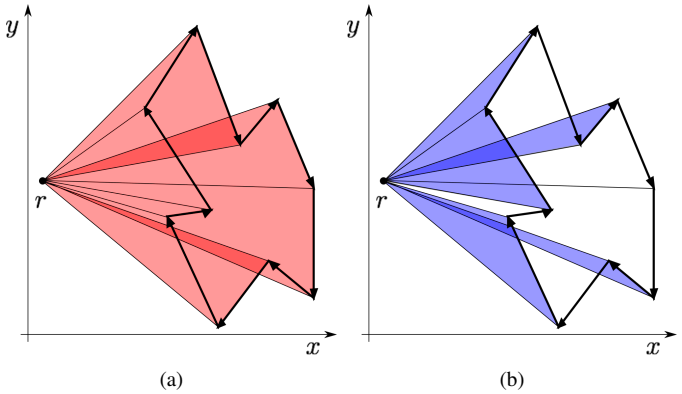


Fig. 2. Area of a polygon P (in bold): each (directed) edge of P forms an oriented triangle with a reference point r. Clockwise triangles contribute positive (a), counterclockwise triangles negative (b).

Constraint (9) assigns the area of sector s to the variable a_s, constraint (10) ensures that the sum of the areas of all sectors equals the area of the complete TMA, a_0. If we want to balance the sector size, we add the following constraint to the IP:

$$a_s \geq a_{LB} \quad \forall s \in \mathcal{S} \quad (11)$$

With this constraint we introduce a lower bound on the size of each sector, this could be a constant, or we can choose $a_{LB} = c_1 \cdot a_0/|\mathcal{S}|$, where we can choose c_1, e.g., $c_1 = 0.9$.

For constraints b and c we need to be able to associate a taskload with a sector. For this section we assume that a heatmap representing the controller's taskload is given, see Section II. 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 for example Figure 3(d): we sum up the heat values for all grid points within the triangle ($h_q \forall q \in \Delta(i, j, r)$). 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(d) is oriented clockwise (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 constraint (12), which assigns each sector s a taskload t_s. In analogy to the balanced size, we add constraint (13) to achieve a balanced taskload. Here, $t_{LB} = c_2 \cdot t_0/|\mathcal{S} \setminus S_0|$ with, e.g., $c_2 = 0.9$. For a bounded taskload we add constraint (14), with some fixed value t_{UB} giving the upper bound for the taskload in any sector. (In case the upper bounds for different sectors differ because of controllers with varying experience we can use sector-specific upper bounds t^s_{UB} for all $s \in \mathcal{S} \setminus S_0$.)

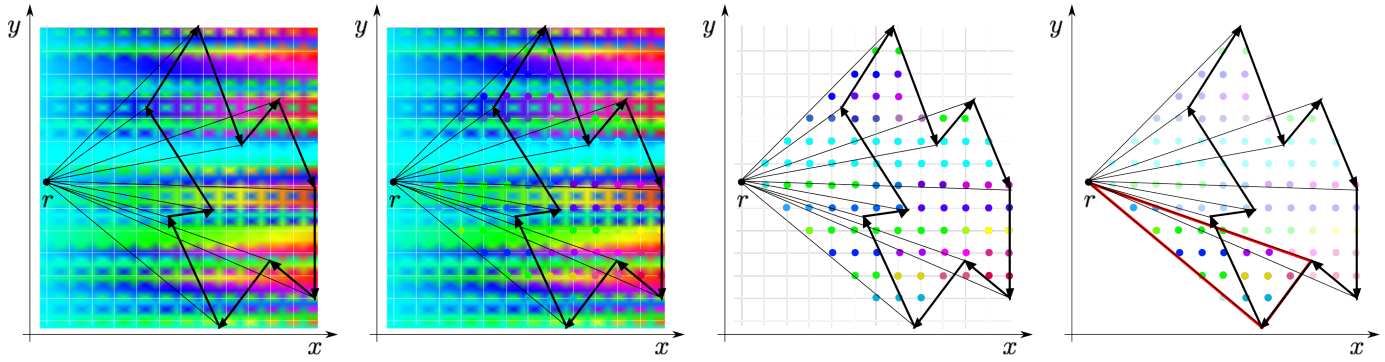


Fig. 3. (a) (Artificial) Heat map overlaid with a grid shown in white, (b) heat values extracted at grid points. (c) Shows the discretized heat map for the area of interest for P: the heat values at grid points for all grid points within some triangle of an edge e of P and the reference point r. (d)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

$$\sum_{(i,j)\in E} h_{i,j}\, y_{i,j,s} - t_s = \quad 0 \quad \forall s \in \mathcal{S} \tag{12}$$

$$t_s \geq t_{LB} \quad \forall s \in \mathcal{S} \tag{13}$$

$$t_s \leq t_{UB} \quad \forall s \in \mathcal{S} \tag{14}$$

2) Geometric Constraints:

- $\mathfrak{d}$) *Connected sectors*: A sector must be a connected portion of airspace, 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. Thus, a sector should not be fragmented into a union of several unconnected airspace units, see Flener and Pearson [12].
- $\mathfrak{e}$) *Nice shape*: A sector's boundary should not be jagged. The sector's geometric shape should be easy to remember, see Flener and Pearson [12].
- $\mathfrak{f}$) *Interior Conflict points:* Points that require increased attention from air traffic controllers should lie in the sector's interior, that is within a certain threshold from the sector boundary.
- $\mathfrak{g}$) *Convex sectors*: The sectors should be convex. Where convexity can, for example, 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, that is, no route enters the same sector more than once, see Flener and Pearson [12]. As mentioned in the introduction, we can easily integrate this constraint, and will present our results in a forthcoming publication.

For constraint $\mathfrak{d}$ we chose to use the length of the sector boundary as an objective function instead of using subtour elimination constraints, see Section IV-B.

Constraint $\mathfrak{f}$ essentially asks to have points with higher complexity to be located within the sector's interior. In particular, this is in relation to the complexity of other points and we cannot consider an absolute threshold value for the complexity of points on the sector boundary. Rather, we like to enforce points of (relatively) high complexity to be in the interior, and treat it as a softer constraint. We again make use of the objective function, see Section IV-B.

We take care of constraint $\mathfrak{e}$ in postprocessing: Given our constraint set $\mathcal{C} \subseteq \{\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}, \mathfrak{g}\}$, we solve the IP with $\mathcal{C} \setminus \{\mathfrak{e}\}$ and then use shortcuts by removing vertices as long as the constraints in $\mathcal{C} \setminus \{\mathfrak{e}\}$ are not violated.

B. Objective Function

As opposed to most mathematical programming approaches, in our case, it is not obvious what kind of objective function should be used. Cost functions used in literature are, e.g., taskload imbalance (which we use as a constraint), number of sectors (which we consider as input), and costs resulting from the interaction between the controllers of different sectors.

Because no obvious objective function exists, we consider different functions, all of which integrate constraint $\mathfrak{d}$.

1) Basic Objective Function: Our basic objective function is:

$$\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} \ell_{i,j} y_{i,j,s} \tag{15}$$

If for the balancing constraints we have $\mathfrak{a} \in \mathcal{C}, \mathfrak{b}, \mathfrak{c} \notin \mathcal{C}$, that is, we want to balance the area of the sectors, but are not interested in the sector taskload, objective function (15) ensures that sectors are connected, that is, we take care of constraint $\mathfrak{d}$, see Figure 4.

The objective function resembles the Traveling Salesman Problem (TSP), another resemblance is the search for a connected tour for the TSP. But we make use of this objective function exactly to yield the latter: connectedness. We do not need to cover specific points with our sector boundaries, but the sectors must completely cover the TMA. Thus, despite the superficial resemblance, we do not need subtour elimination constraints for all nonempty subsets to ensure connected sectors.

If we consider taskload, objective function (15) only yields connected sectors if c_2 in $t_{LB} = c_2 \cdot t_0/|\mathcal{S} \setminus S_0|$ of constraint (13) allows it: for example $c_2 = 0.9$ may not allow a "c_2-balanced" sectorization with connected sectors, but if

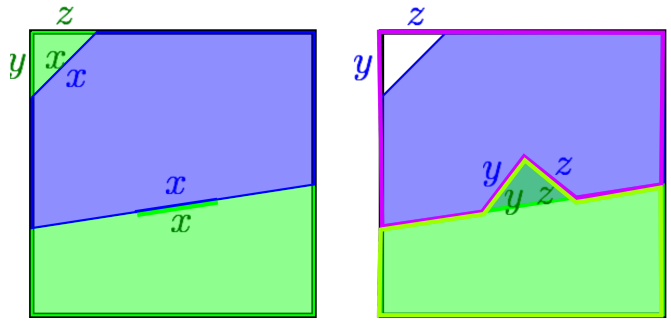


Fig. 4. Disconnected sectors are not optimal for the objective function of minimizing the perimeter of all sectors, (15) . The sectors must completely cover the TMA. Assume there is a disconnected sector, like the green sector in the left, we can merge it and decrease the total perimeter, we have: $(y+z+2x)$+$(2x)$≤ $(y+z)$ +$(2y+2z)$ by triangle inequality.

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.

2) Integration of Constraint $\mathfrak{f}$*:* In Section IV-A we described that the objective function also takes care of constraint $\mathfrak{f}$: interior conflict points. If $\mathfrak{f} \in \mathcal{C}$ we use the following objective function (an extension of the basic objective function (15)):

$$\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j})\, y_{i,j,s}, \quad 0 \le \gamma < 1 \quad (16)$$

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

(I) $w_{i,j} = h_i + h_j$
(II) $w_{i,j} = \sum_{k\in N(i)} h_k + \sum_{l\in N(j)} h_l$, with $N(i)$ denoting the neighbors of i in G (including i itself)

The former ensures that relatively large heat values are not located on the sector boundary, the latter pushes larger values further into the interior. An alternative to using objective function (16) instead of objective function (15) is to push this into a constraint with an upper bound, W_{UB}, on the (sum of the) $w_{i,j}$ of sector boundary edges. This again shows that the resulting sectors must not necessarily be connected anymore, apart from c_2 the value of W_{UB} may make a connected solution infeasible. Thus, we obtain an optimal connected solution, if, given c_2 and W_{UB}, there exists a feasible connected solution.

C. The Complete MIP

To enhance readability, we present the complete MIP in this section. From the balancing constraints we present here the constraints for $\mathfrak{c}$, balanced taskload.

$$\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} \ell_{i,j} y_{i,j,s} \quad (15)$$

s.t.

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

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

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

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

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

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

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

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

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

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

$$\sum_{(i,j)\in E} h_{i,j}\, y_{i,j,s} - t_s = 0 \qquad \forall s \in \mathcal{S} \quad (12)$$

$$t_s \ge t_{LB} \qquad \forall s \in \mathcal{S} \quad (13)$$

In case we want to have interior conflict points, we substitute the objective function (15), by (16):

$$\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j})\, y_{i,j,s}, \quad 0 \le \gamma < 1 \quad (16)$$

V. Experimental Study: Arlanda Airport

We consider $\mathcal{C}_1 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}\}$ and $\mathcal{C}_2 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$, with $w_{i,j} = h_i + h_j$. If not mentioned otherwise we use $c_2 = 0.9$.

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, or for a maximum of one CPU-hour. No instance finished with an optimality gap of more than 6%.

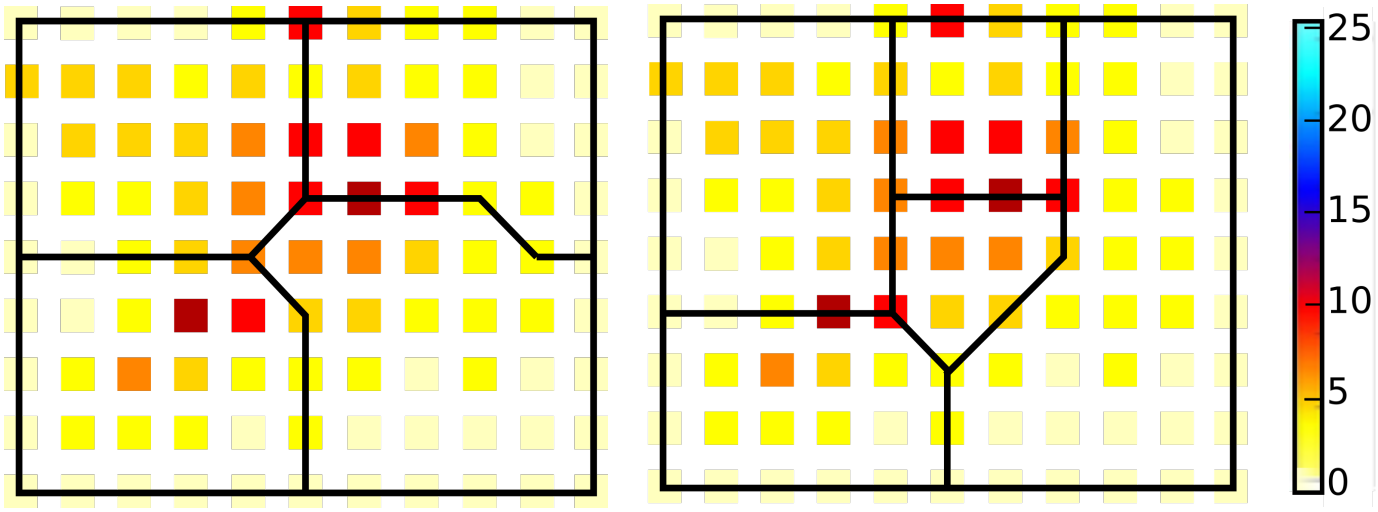
Figure 5 depicts solutions for $\mathcal{C}_1 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}\}$, that is, objective function (15). In Figure 6 we present our sectorizations for $|\mathcal{S}| = 4$, and $\mathcal{C}_2 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$, that is, objective function (16) with $w_{i,j} = h_i + h_j$, for different values of γ.

Comparing Figure 6(a) with Figure 5(a) we can observe that the objective to have interior conflict points avoids the heat value of "10" in the center; for $\gamma = 0.5$, Figure 6(b), both hotspots are avoided by sector boundaries, that is, we yield a sectorization with interior conflict points.

Figure 7(a) shows that this instance does not have a "c_2-balanced" solution for $c_2 = 0.9$. On the other hand, if we substitute constraint $\mathfrak{c}$ by $\mathfrak{a}$, we can ensure connected sectors, see Figure 7(b).

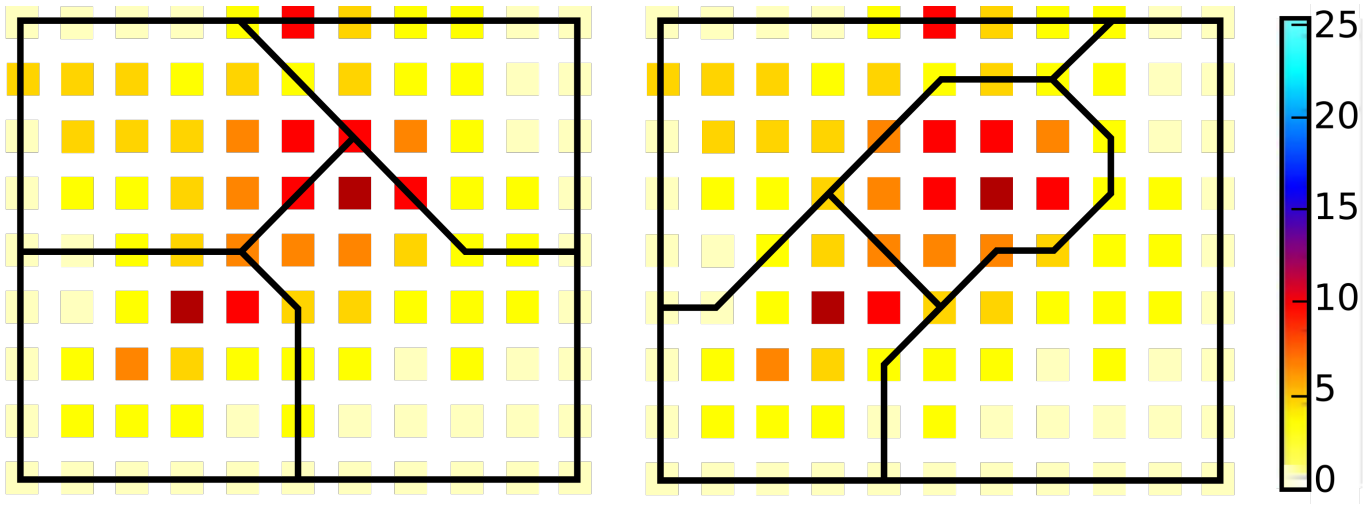
A. Influence of choosing $w_{i,j}$

In this subsection, we present an instance that is not connected to Stockholm TMA, to highlight the influence of the



(a) (b) (c)

Fig. 5. $\mathcal{C}_1 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}\}$, that is, objective function (15). (a) $|\mathcal{S}| = 4$, and (b) $|\mathcal{S}| = 5$. (c) Color scale for heat values.



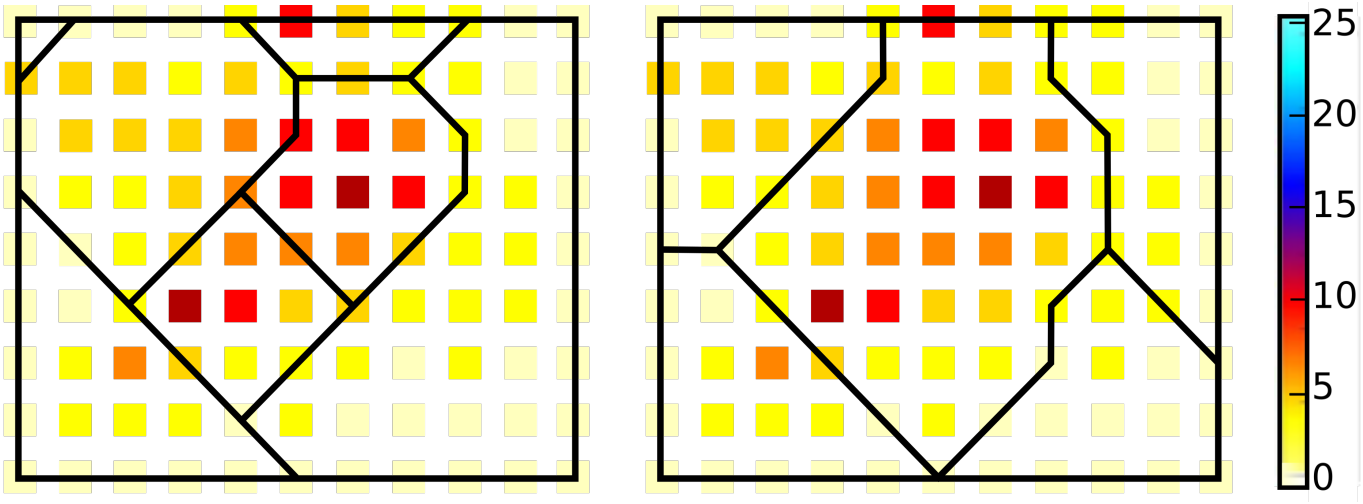
(a) (b) (c)

Fig. 6. $|\mathcal{S}| = 4$, and $\mathcal{C}_2 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$, that is, objective function (16) ($\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j})\, y_{i,j,s}, \quad 0 \leq \gamma < 1$), with $w_{i,j} = h_i + h_j$. In (a) we use $\gamma = 0.9$, in (b) $\gamma = 0.5$. Thus, in (b) the objective of interior conflict points/hotspot avoidance has a higher weight than in (a). (c) Color scale for heat values.

two definitions for the weight $w_{i,j}$. As this example is chosen to show that relation, we present results for $|\mathcal{S}| = 2$ to ease perception. Consequently, we pick one cut through the rectangle representing the square as the boundary between the two sectors. We consider $\mathcal{C}_2 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$, and apply both $w_{i,j} = h_i + h_j$ and $w_{i,j} = \sum_{k\in N(i)} h_k + \sum_{l\in N(j)} h_l$ for $\mathfrak{f} \in \mathcal{C}_2$. That is, we use $\min \sum_{s\in\mathcal{S}} \sum_{(i,j)\in E} (\gamma \ell_{i,j} + (1-\gamma) w_{i,j})\, y_{i,j,s}, \quad 0 \leq \gamma < 1$ as objective function. In Figure 8 (a) we use $\gamma = 1$, i.e., we do not try to avoid sector boundaries running through hotspots. Consequently, the cut runs along the shortest connection that balances workload. In Figure 8 (b) and (c) we use $\gamma = 0,5$. So, we use a linear combination of length and hotspot avoidance as our objective function. In (b) we use weight function (I), and in (c) weight function (II). That is, in (b) we want to avoid that the sector boundary runs through hotspots; in the solution, we see that the low heat points of value 2 are chosen. In the center the cut avoids the (turquoise) 25, and runs through one of the (inevitable) 20's (blue) on the central vertical line (as in the objective value we have $\frac{1}{2} \cdot 2 \cdot \sqrt{2} < \frac{1}{2} \cdot 5$). In (c), we also account for the weight of the neighbors of vertices on the cut. Thus, the cut that was optimal for (I) has a high weight, due to the neighboring 20's and 25's. The optimal solution now runs through the areas of low complexity, with heat values of 5 (orange), and avoids the hotspots.

VI. Conclusion and Discussion

In this paper we presented a sectorization method that balances or bounds the taskload for the sectors, based on a complexity representation. Moreover, we show how to inte-



(a) (b) (c)

Fig. 7. $|\mathcal{S}| = 5$: (a) For $c_2 = 0.9$, $\gamma = 0.5$ and $w_{i,j} = h_i + h_j$, ($\mathcal{C}_2 = \{\mathfrak{c}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$) no "$c_2$-balanced" sectorization with connected sectors exists. (b) In contrast, if we look at area balance, $\mathcal{C}_3 = \{\mathfrak{a}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$, $c_1 = 0.7$, we obtain connected sectors.(c) Color scale for heat values.

grate further geometric constraints on the resulting sectors: connectedness, nice boundary, and interior conflict points. We apply our techniques to sectorize Stockholm Terminal Maneuvering Area (TMA), and highlight the options for interior conflict points with a tailored instance. Our 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. Note that our approach can easily integrate SUA: we simply force (the edges of) an additional sector, the SUA plus a threshold, and exclude this sector from the workload balancing. In a forthcoming paper [3], we will show that we naturally can integrate a convexity constraint, and we plan to limit the deviation from at most 180° interior angle at all sector vertices (the rationale behind the latter being that a sectorization as shown in Figure 1 with a few reflex vertices is not problematic). Moreover, we plan to get a detailed feedback on the quality of the obtained solution by ATCOs. Another possible direction for future work is to adapt the approach to 3D, in which case we would no longer deal with area of triangles, but voume of pyramids. Also interesting is a comparison of our sectorization method with other approaches, for example, methods presented in the survey by Flener and Pearson [12].

Acknowledgment: 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.

References

[1] IATA air passenger forecast shows dip in long-term demand. http://www.iata.org/pressroom/pr/Pages/2015-11-26-01.aspx, November 2015. accessed on June 2, 2016.

[2] C. Allignol, N. Barnier, P. Flener, and J. Pearson. Constraint programming for air traffic management : a survey. *Knowledge Engineering Review*, 27(3):pp 361–392, Sept. 2012. http://www.sherpa.ac.uk/romeo/issn/0269-8889/.

[3] T. Andersson Granberg, T. Polishchuk, V. Polishchuk, and C. Schmidt. Convex sectorization—a novel integer prgramming appriach. In *17th Integrated Communications, Navigation and Surveillance Systems (ICNS) Conference, 2017*, 2017.

[4] M. Bloen and P. Gupta. Configuring airspace sectors with approximate dynamic programming. 2010.

[5] C. R. Brinton, K. Leiden, and J. Henkey. Airspace sectorization by dynamic density. American Institute of Aeronautics and Astronautics, 2009.

[6] R. S. Conker, D. A. Moch-Mooney, W. P. Niedringhaus, and B. T. Simmons. New process for "clean" sheet airspace design and evaluation. 2007.

[7] D. Delahaye, M. Schoenauer, and J.-M. Alliot. Airspace sectoring by evolutionary computation. In *IEEE 1998, International Conference on Evolutionary Computation*, pages pp 218 – 223, Anchorage, United States, May 1998.

[8] J. Djokic, B. Lorenz, and H. Fricke. Air traffic control complexity as workload driver. *Transportation Research Part C: Emerging Technologies*, 18(6):930 – 936, 2010. Special issue on Transportation SimulationAdvances in Air Transportation Research.

[9] M. C. Drew. A method of optimally combining sectors. American Institute of Aeronautics and Astronautics, 2009.

[10] EUROCONTROL. EUROCONTROL Long-Term Forecast EUROCONTROL Flight Movements 2010 - 2030, 2010.

[11] 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*, pages 133–136, 2015.

[12] P. Flener and J. Pearson. Automatic airspace sectorisation: A survey. *CoRR*, abs/1311.0653, 2013.

[13] I. Gerdes, A. Temme, and M. Schultz. Dynamic airspace sectorization using controller task load. In *Sixth SESAR Innovation Days*, 2016.

[14] D. Gianazza. Forecasting workload and airspace configuration with neural networks and tree search methods. *Artificial Intelligence*, 174(7):530 – 549, 2010.

[15] P. Jägare. Airspace sectorisation using constraint programming, 2011. Report IT 11 021, Facutly of Sceince and Technology.

[16] P. Kopardekar and S. Magyarits. Dynamic density: Measuring and predicting sector complexity. In *Proceedings of the 21st Digitial Avionics Systems Conference (DASC)*, volume 1, pages 2.C.4–1–2.C.4–9, 2002.

[17] I. Kostitsyna. *Balanced Partitioning of Polygonal Domains*. PhD thesis, Stony Brook University, August 2013.

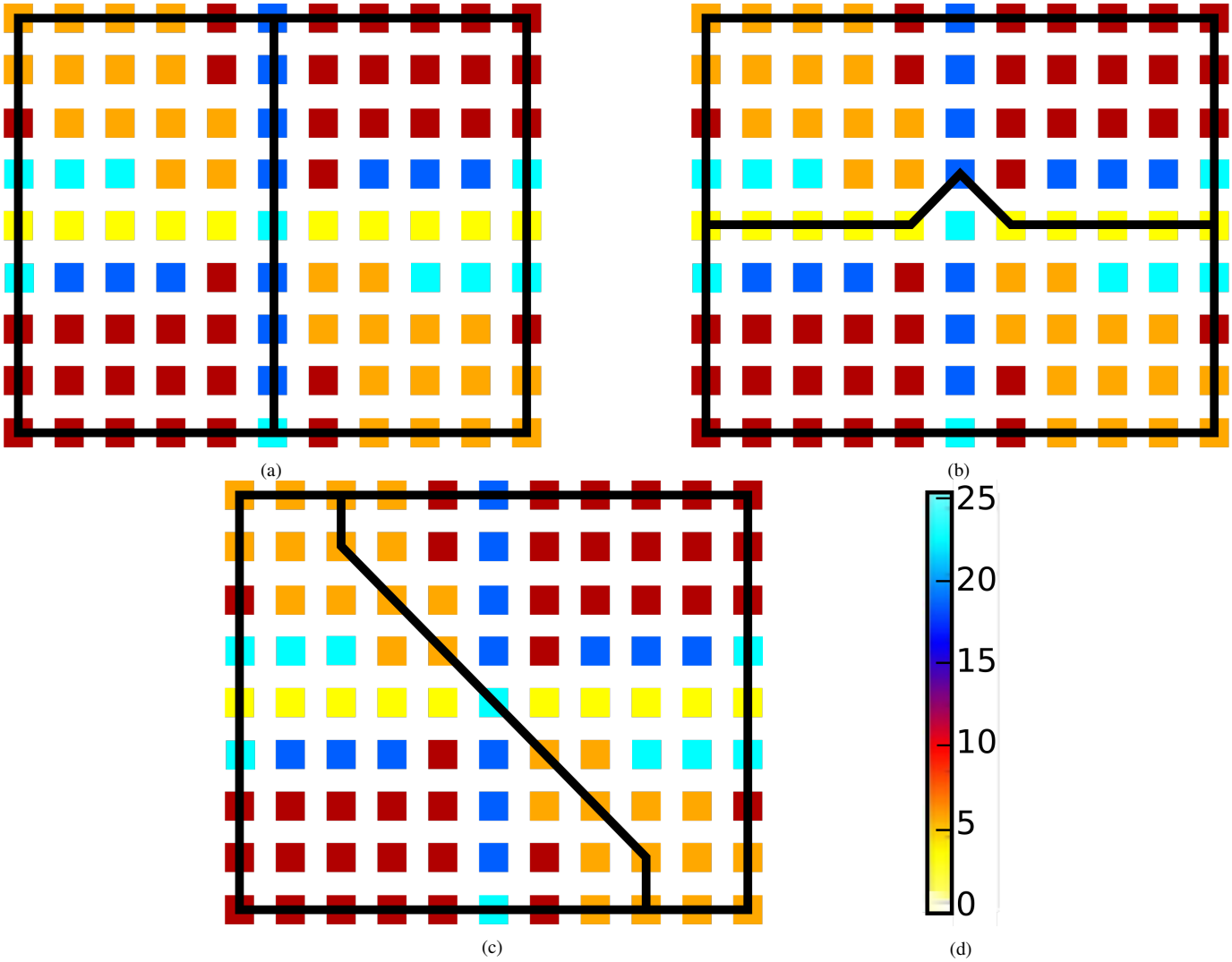


Fig. 8. Example to show the influence of choosing $w_{i,j}$.

[18] S. Kulkarni, R. Ganesan, and L. Sherry. Static sectorization approach to dynamic airspace configuration using approximate dynamic programming. In *Integrated Communications, Navigation and Surveilance Conference (ICNS), 2011*, pages J2–1–J2–9, May 2011.

[19] K. Leiden, S. Peters, and S. Quesada. Flight level-based dynamic airspace configuration. American Institute of Aeronautics and Astronautics, 2009.

[20] 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*, 49(3):376–399, 2007.

[21] S. A. Martinez, G. B. Chatterji, D. Sun, and A. M. Bayen. A weighted-graph approach for dynamic airspace configuration. American Institute of Aeronautics and Astronautics, 2007.

[22] C. Meckiff, R. Chone, and J.-P. Nicolaon. The tactical load smoother for multi-sector planning. In *Proceedings of ATM 1998, the 2nd USA/ Europe Air Traffic Management Research and Developpment Seminar*, 1998.

[23] G. R. Sabhnani, A. Yousefi, and J. S. Mitchell. Flow conforming operational airspace sector design. American Institute of Aeronautics and Astronautics, 2010.

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

[25] M. Xue. Three dimensional sector design with optimal number of sectors. American Institute for Aeronautics and Astronautics, 2010.

[26] A. Yousefi, G. L. Donohue, and K. M. Qureshi. Investigation of en route metrics for model validation and airspace design using the total airport and airspace modeler (TAAM). 2003.

[27] S. Zelinski and C. F. Lai. Comparing methods for dynamic airspace configuration. 2011.

[28] E. Zohrevandi. Effects of air traffic complexity factors on controllers workload in Stockholm terminal area, 2016.

[29] 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 *Sixth SESAR Innovation Days*, 2016.

BIOGRAPHIES

Dr. **Tobias Andersson Granberg** has a PhD in Infrainformatics (2005) and is an Associate Professor at the Division for Communication and Transport Systems at Linköping University in Sweden. His current research is within logistics and transportation, especially air transportation and emergency response.

Dr. **Tatiana Polishchuk** received a PhD (2013) in Computer Science from Helsinki University and a MSc (2007) in Applied Mathematics and Statistics from the State University of New York at Stony Brook, USA. She joined the ITN department at

Linköping University, Sweden, in 2016 as a research engineer. Her research interests include network optimization and airport resource management.

Dr. **Valentin Polishchuk** received a MSc degree (2002) in Operations Research and a PhD (2007) in Applied Mathematics and Statistics from Stony Brook University. He is an Associate Professor in the Department of Science and Technology of Linköping University, working as a Senior Lecturer in Air Transportation within Air Transport and Logistics program at Communications and Transport Systems division; he is also an Adjunct Professor in Computer Science at University of Helsinki, affiliated with the New Paradigms in Computing group of Helsinki Institute for Information Technology.

Dr. **Christiane Schmidt** holds a diploma (2006) in Mathematics for Finance and Industry and a PhD (2011) in Computer Science from Braunschweig University of Technology, Germany. After working as a postdoctoral researcher at State University of New York at Stony Brook, USA, and the Hebrew University of Jerusalem, Israel, she joined Linköping University, Sweden, in 2015. Dr. Schmidt research interests include Algorithms, Computational Geometry and Mathematical Optimization. Since her move to Linköping University she applies this knowledge to ATM research.