Airspace Sector Design: An Optimization Approach

Go Nam Lui, Guglielmo Lulli
Management School
Lancaster University
Lancaster, United Kingdom
{g.n.lui, g.lulli}@lancaster.ac.uk

M. Florencia Lema-Esposto, Rebeca Llorente Martinez
Centro de Referencia I+D+i en ATM (CRIDA)
Madrid, Spain
{mflema, rllorente}@e-crida.enaire.es

***Abstract*—This paper presents a Mixed Integer Programming model for optimal airspace sector design based on basic volume aggregation, focusing on workload balance and air traffic flow convexity. To overcome potential computational limitations, we develop a simple two-stage heuristic approach. The heuristic approach are evaluated using real-world traffic data from the Madrid Area Control Center, with the MIP as a benchmark. Our key contributions include: (1) the first rigorous mathematical formulation for this problem, (2) a fast heuristic achieving near-optimal solutions in under one second, and (3) a comprehensive assessment across various traffic scenarios. Results show that our MIP model generates operationally relevant sector designs, and the heuristic could provide good-quality solutions with exceptional efficiency. This research advances airspace management techniques, offering both theoretical insights and practical tools for optimizing air traffic control.**

***Keywords*—Airspace sector design; mixed-integer programming (MIP); heuristic approach**

I. Introduction

Air traffic controllers (ATCOs) face mounting challenges as they navigate the increasing complexity of airspace, further exacerbated by adverse weather conditions [1] and the integration of emerging airspace users [2]. To enhance the flexibility and capacity of airspace, the concept of Dynamic Airspace Configuration (DAC) has been proposed. The SESAR Concept of Operations recommends DAC as the first means to resolve demand capacity imbalances, and it is also envisioned as a major cornerstone for the future architecture of the European airspace described in the Airspace Architecture Study. Based on the predicted traffic demand and weather conditions, airspace sectors are dynamically configured (i.e., are open for certain period of time during the day) to avoid workload imbalance and maximize the airspace capacity.

In line with the strategic vision provided by the European ATM Master Plan, we aim to make the airspace design and configuration process more efficient taking full advantage of the airspace potential. Our ambition is to develop sector design capabilities that allow the definition of a catalogue of sectors that serves as the "ground set" for configurable sectors thus enabling DAC operations. The catalogue should include all the sectors to handle all the traffic scenarios that can occur.

Traditionally, sector design process has been based on Air Traffic Services (ATS) fixed routes structure and it is manually performed by a group of experts. Considerations on main flows have been progressively introduced, but automation of sector creation is not widespread. Modern approaches favor grouping "basic volumes" (i.e., portions of the airspace) over conventional direct partitioning approaches [3, 4], in the attempt of ensuring a higher degree of operational flexibility.

In this paper, we present a methodology for the sector design, which is based on grouping basic volumes. At each execution of the proposed approach, we design multiple sectors simultaneously. The motivation stems from the challenges that may arise in a single sector design approach. Indeed, it may create an extremely large number of sectors, which might fail to account for system-wide implications and inter-sector relationships. There is no guarantee that feasible configurations can be retrieved from the corresponding sector catalog.

Our proposed approach primarily focuses on two critical operational requirements for sector design: workload balance and air traffic flow convexity. Workload balance refers to the equitable distribution of aircraft occupancy counts within designated sectors, ensuring a fair allocation of air traffic management responsibilities. Air traffic flow convexity - which can be envisioned as an approximation of the sector convexity property - focuses on aligning sector boundaries parallel to the main traffic flows within each sector. This alignment offers several advantages:

1) It reduces the frequency of aircraft crossing sector boundaries, allowing flights to remain within a single sector for extended periods. This minimizes the number of handoffs between controllers, thereby reducing workload and communication complexity.
2) It minimizes short crossings and re-entries, lowering the inter-sector traffic flow. This reduction decreases the likelihood of aircraft briefly entering a sector only to quickly exit or re-enter shortly after, improving overall airspace efficiency and safety.

Although these aspects have been considered in previous research efforts, they have been often used for evaluation rather than in the design/sectorization process [5]. In this work, our contribution is three-fold. First, we present a Mixed Integer Programming (MIP) model for airspace sector design based on basic volumes aggregation, introducing the first rigorous mathematical formulation for this problem to the best of our knowledge. Second, to address the potential computational challenges, we develop a simple two-stage heuristic approach that achieves near-optimal solutions with significantly reduced processing time. Lastly, using the MIP solution as a benchmark, our comprehensive assessment evaluates the heuristic's performance focusing on critical operational metrics such as workload balance and air traffic flow convexity.

This paper is structured as followed: Section II provides a literature review on airspace sector design. The model implementation is presented in Section III. Section IV describes the case study while Section V presents the results and discussion. Section VI concludes this work and discusses the future extension.

II. Literature review

The problem of airspace sector design has been considered as an airspace sectorization problem. Flener and Pearson provided a comprehensive survey to classify the automatic airspace sectorization methods, based on criteria like modeling approach, constraints, and solution techniques [5]. Conventional airspace sectorization is a network partitioning problem. That is, the air traffic transportation network - modeled as a weighted planar graph - is partitioned into K sub graphs (each representing a sector), all with similar values of the total weight (workload). The computational complexity of such discrete graph partitioning problem is NP-hard. Indeed, the difficulty to solve the problem is further exacerbated by the presence of topological constraints, e.g., convex sectors (partitions).

Delahaye *et al.* [3] pioneered the use of genetic algorithms for airspace sectorization. The application of Voronoi diagram guarantees that the designed sectors have a convex shape. In a subsequent study [4], they refined this method by applying route convexity instead of shape convexity and generating the Voronoi diagram on the underlying air route network directly rather than on the airspace's boundaries.

To consider a series of more specific operational constraints such as minimum sector crossing time, Trandac *et al.* [6] proposed a constraint programming approach. Their two-phase method first finds an initial solution for the constraint programming using a Restricted Kernighan-Lin heuristic, then applies random local re-optimization to address the fragmentation of the sectors.

In 2007, R.Ehrmanntraut and S.McMillan [7] studied the airspace design process for dynamic configurations, highlighting the importance of mathematical optimization and fast capacity-simulation for dynamic traffic demand. J.Mitchell *et al.* [8] used a recursive partitioning algorithm including three local graph partitioning cuts to divide airspace into sectors on the National Aviation System (NAS) air route structure.

Brinton *et al.* [9] developed an airspace sectorization algorithm using Dynamic Density as the objective function to minimize sectors while maintaining dynamic density below a threshold. Dynamic Density refers to the complexity or difficulty of an air traffic situation, which includes the number of aircraft occupancy and other complexity metrics such as separation criticality index in a weighted combination [10]. Kulkarni *et al.* [11] explores the potential of Approximate Dynamic Programming in static airspace sectorization, to create a benchmark for potential dynamic re-sectorization in DAC. J. Tang *et al.* [12] proposed a multi-objective optimization for Dynamic Airspace Sectorization, which demonstrated the challenge in balancing workload fairness, sector shape convexity, and average flight time inside individual sector. Zou *et al.* [13] proposed an approach for airspace sectorization using constrained evolutionary algorithms on an undirected graph model, which propose a concave hull-based method to automatically depict sector boundaries.

III. Methodologies

In this section, we formulate a MIP model for sector design problem. We assume the designed sectors span all the upper airspace. This approach allows for efficient sector design in airspaces with consistent vertical traffic distribution. In the real-world air traffic network, airways are composed of multiple routes that share the same ground projection but differ in altitude based on their direction (following the semicircular rule) [3]. Thus, the assumption of 2D projection is appropriate.

The airspace topology is modeled as an undirected graph $G = (V, E)$, where V represents airspace volumes and E denotes shared borders between volumes. Each vertex $v_i \in V$ corresponds to a basic volume's geographic center. The set V comprises three classes: Elementary Sectors (ES), Airspace Blocks (AB), and Shareable Airspace Blocks (SAB), such that $V = ES \cup AB \cup SAB$. ES are independently controllable sectors, which can be used as controllable sectors if needed. AB must be attached to another AB or ES to form a controllable sector, while SAB are non-workable volumes that need dynamic attachment to any ES or AB to create an operational sector.

To formulate the model, we introduce the following notation:

- $\overline{W}$: maximum capacity (workload) of a sector;
- c_{ij}: inter-block flow between adjacent volumes i & j;
- w_i: workload/occupancy counts in volume i;

The decision variables are

$$x_{ij} = \begin{cases} 1, & \text{if volume } i \text{ is grouped with volume } j\ , \\ 0, & \text{otherwise.} \end{cases}$$

and

$$W = \text{the minimum workload across sectors.}$$

For the sake of clarity, the index j of variable x_{ij} identifies the sector. Sector j is defined as the set of all volumes i for which $x_{ij} = 1$.

A. The MIP formulation

In this section, we present the sector design model. The objective function of the proposed model consists of two components with a balancing parameter α:

- *Workload Balancing*: The first term maximizes the minimum workload across all sectors, which is captured by the decision variable W. This approach, coupled with a predefined upper bound on workload, ensures a balanced distribution of air traffic management responsibilities.

- *Internal Sector Traffic Flow*: The second term maximizes the sum of flow c_{ij} between basic volume i and j within the same sector. This is equivalent to minimizing the inter-sector traffic flow, as the total exchanged traffic flow is constant for each historical traffic scenario.

The model formulation is listed as follows:

$$\max \quad \alpha W + (1-\alpha) \sum_{(i,j)\in E, k\in V} c_{ij}(x_{ik} \cdot x_{jk}) \tag{1}$$

$st.$

$$\sum_{j\in V} x_{ij} = 1, \quad \forall i \in V \tag{2}$$

$$\sum_{i\in V} w_i \cdot x_{ij} \leq \overline{W}, \quad \forall j \in V. \tag{3}$$

$$W \cdot x_{jj} \leq \sum_{i\in V} w_i \cdot x_{ij}, \quad \forall j \in V. \tag{4}$$

$$\sum_{i\in ES \bigcup AB\backslash\{j\}} x_{ij} \geq x_{jj}, \quad \forall j \in AB. \tag{5}$$

$$x_{jj} = 0, \quad \forall j \in SAB \tag{6}$$

$$x_{ij} \leq x_{jj}, \quad \forall ij \in V \tag{7}$$

$$x_{ij} \in \{01\}, \quad \forall ij \in V \tag{8}$$

Constraint (2) requires each basic volume i to be uniquely allocated to one sector. Constraint (3) imposes that the workload of each sector does not exceed the maximum value. Constraint (4) is used to fix the value of variable W to the minimum workload. We aim to maximize the minimum value W in our objective function to ensure equitable workload distribution. This approach correspond to a *maxmin* approach that is quite common in the equity (or fairness) literature, e.g., Young [14]. Constraint (5) imposes that if AB is the anchor of a sector it is attached to at least one other AB and/or ES, while constraint (6) implies that SAB cannot be the root of a sector. Finally, the coupling constraint designates volume j as the root of a sector if any other block i is assigned membership to the sector j.

To maintain contiguity between basic volumes, we implemented connectivity constraints using a flow-based formulation which has been applied in other application domains, e.g., political districting [15–17]. To formulate these constraints, we introduce the variable:

$$f_{ij}^{v} = \text{the amount of flow from } v\text{, passed edge}_{ij} \tag{9}$$

Using $f^j(S)$ for $S \subseteq A$ as the shorthand for $\sum_{(u,v)\in S} f_{uv}^{j}$, we will have the following constraints:

$$f^j\left(\delta^-(i)\right) - f^j\left(\delta^+(i)\right) = x_{ij}, \quad \forall i \in V\backslash\{j\}, \forall j \in V \tag{10}$$

$$f^j\left(\delta^-(i)\right) \leq (n-1)x_{ij}, \quad \forall i \in V\backslash\{j\}, \forall j \in V \tag{11}$$

$$f^j\left(\delta^-(j)\right) = 0, \quad \forall j \in V \tag{12}$$

$$f_{ij}^{v} \geq 0, \forall (i,j) \in A, \quad \forall v \in V \tag{13}$$

Vertex i will only use one unit of flow of type j if it is allocated to the root of sector j, as per constraint (9); otherwise, it will use no flow. Vertex i can only receive the flow of type j if it is assigned as the root of sector j, according to constraint (10). Constraint (11) prevents the circulation of flow.

B. Formulation enhancements

To improve the formulation and its computational performance, we here present three optimization "tricks".

1) Lower bound of sectors: It is easy to obtain a lower bound on the number of sectors K, where $K = \frac{\sum w_i}{\overline{W}}$. If we provide the additional information of the constraint to K:

$$\sum_{j\in V} x_{jj} \geq K, \tag{14}$$

the optimization solver will take advantage of this additional constraint (14), which allows computing the optimum solution in a shorter computational time.

2) Re-index adjustment: The formulation (1)-(13) has "symmetries", i.e., it has multiple equivalent solutions that make the search space redundant. This is due to the fact that one sector can be represented in the formulation by two or more different sets of variables, e.g., one grouped with j (i.e., all $x_{ij} = 1$), and the other grouped with j' (i.e., all $x_{ij'} = 1$). To overcome this issue - which deteriorates the efficiency of solution algorithms -, we implement a re-indexing method of the basic volumes based on geographic coordinates. This allows us to identify the sector with the volume with the largest index and impose the following set of constraints:

$$x_{ij} = 0, \forall i > j \in V \tag{15}$$

By eliminating the need to consider lower-indexed volumes as potential sector identifier, the constraint significantly reduces the solution space, leading to faster convergence times in optimization algorithms. Moreover, it allows us to halve the number of decision variables.

3) Variable fixing strategy: To reduce the computational time of the MIP model, we also implement a pre-processing strategy which fixes the variables to zero if the summation of basic volumes' workload exceed the allowed maximum value of workload $\overline{W}$.

The "Variable Fixing Strategy" (Algorithm 1) operates on the graph G, with binary variable, and the upper bound workload $\overline{W}$. For each pair of nodes (i, j) in the graph, the algorithm calculates the shortest path s between them using Dijkstra's algorithm [18], which utilizes the average node weight of edge (i, j). The algorithm then computes the total weight Ω by summing up the weights of all nodes in s. If this total weight is greater than or equal to the threshold $\overline{W}$, the binary variable x_{ij} is set to 0. This process is repeated for all pairs of nodes in the graph.

Algorithm 1 Variable fixing strategy

procedure VFS($G, x, \overline{W}$) ▷ graph $G(V, E)$, binary variable x_{ij} for $(i, j) \in E$, threshold $\overline{W}$
  for each node i in graph G **do**
    for each node j in graph G **do**
      Compute the shortest path (s) between node i and j using Dijkstra's algorithm using the average node weight
      Sum up the node weight $\Omega = \sum_{n \in s} w_n$
      If $\Omega \geq \overline{W}$: $x_{ij} = 0$
    end for
  end for
end procedure

C. The heuristic approach

The MIP model provides exact solutions to the sector design problem, however it can be computationally intensive for large-scale instances. As the number of basic volumes and potential sectors increases, the computational cost for the MIP model grows exponentially. To address this limitation, we propose a two-stage heuristic (Algorithm 2), which is a more scalable approach.

The first stage aims to quickly identify high traffic regions within the airspace, which generate a initial sector list that is convex to air traffic flow and connected. The heuristic approach leverages the graph representation of the airspace used in the MIP model. It focuses on the inter-block traffic flow c_{ij}, to prioritize the most significant traffic patterns, and balances the workloads for the unassigned basic volumes. This aligns with the MIP model's objective of optimizing sector design based on traffic flow patterns. Worth mentioning, the input parameter K is not necessary for the model, the implementation is here just for constructing a comparable assessment with the MIP model results as a benchmark.

The first stage of the heuristic approach consists of the following key steps:

- Sort the edges of the graph in descending order based on inter-block traffic flow c_{ij}.
- Iterate through the sorted edges if $|S| < K$:
  - If both vertices u, v of the edge are unassigned, create a new sector containing these vertices and mark them as assigned;
  - If one vertex is assigned and the other is unassigned, add the unassigned vertex to the existing sector;
  - If both vertices are already assigned, skip the edge.
- After processing all the edges, assign any remaining unassigned individual vertices to sectors have most connections with them (best sector B). If there are more than one best sector for the unvisited node, we use the average distance from v to all nodes in each sector as a tiebreaker, to maintain connectivity.
- Return the set of identified sectors.

Following the initial sector design produced by the heuristic, a local search algorithm (Algorithm 3) is applied to further refine the solution. This local search focuses on three primary objectives: maximizing the internal traffic flow within sectors, balancing the workload across sectors, and maintaining connectivity.

Algorithm 2 Two-Stage Airspace Sector Design Heuristic

Require: Graph $G(V, E)$, edge weights c_{ij}, node weights w_i, number of sector K, balance parameter α
Ensure: Set of sectors $S = \{S_1, S_2, ..., S_k\}$
$E_{sorted} \leftarrow$ Sort(E, key $= c_{ij}$, order = descending)
$S \leftarrow \{\}$ ▷ Initialize empty set of sectors
$A \leftarrow \{\}$ ▷ Initialize set of assigned nodes
for $e_t = (u, v) \in E_{sorted}$ **do**
  if $|S| \geq K$ **then**
    Stop iteration
  end if
  if $u \notin A$ and $v \notin A$ **then**
    Create new sector with u and v
  else if exactly one of u or v is in A **then**
    Add unassigned node to sector of assigned node
  end if
  Update A
end for
$U \leftarrow V \setminus A$ ▷ Set of unassigned nodes
for $v \in U$ **do**
  $B \leftarrow argmax_j |\{(v, u) \in E : u \in S_j\}|$ ▷ Best sectors
  if $|B| > 1$ **then**
    $i \leftarrow argmin_{j \in B} \frac{1}{|S_j|} \sum_{u \in S_j} dist(v, u)$ ▷ Tiebreaker
  else
    $i \leftarrow$ the single element in B
  end if
  $S_i \leftarrow S_i \cup \{v\}$ ▷ Assign v to chosen sector
end for
  return S_{init}
LocalSearch(S_{init})
return S
where:
$\mathcal{W}$ ▷ The list of total workload $\forall S_k \in S$
$\mathrm{C}(S_i) = 1 \iff \forall u, v \in S_i, \exists$ a path $s_{uv} \subseteq S_i$
$P_c = \sum_{i=1}^{k} |S_i| \cdot [1 - \mathrm{C}(S_i)]$ ▷ The disconnection penalty
Overall score =
$\alpha \min(\mathcal{W}) + (1 - \alpha) \sum_k \sum_{i,j \in S_k} c_{ij} - \max(\mathcal{W}) - P_c$

The algorithm iteratively considers moving nodes between adjacent sectors, accepting moves that improve a composite score function based on the MIP's objective:

$$\alpha \min(\mathcal{W}) + (1 - \alpha) \sum_{k} \sum_{i,j \in S_k} c_{ij} - \max(\mathcal{W}) - P_c$$

where: $\mathcal{W}$ is the list of workload for each sector, $\alpha \in [0, 1]$ is a balancing parameter, $C(S_i)$ is the discriminator for sector connection, and P_c is a penalty for disconnection.

Algorithm 3 LocalSearch(S)

for $i \leftarrow 1$ to $max\_iterations$ **do**
  for all $S_i \in S, v \in S_i, S'_i \in N_G(v) \cap (S \setminus \{S_i\})$ **do**
    $S' \leftarrow (S \setminus \{S_i, S'_i\}) \cup \{S_i \setminus \{v\}, S'_i \cup \{v\}\}$
    if $\mathrm{C}(S_i) = 1 \wedge Score(S') > Score(S)$ **then**
      return LOCALSEARCH(S')
    end if
  end for
end for
return S

The first two terms for the score function align with the objective function of MIP, which it seeks to maximize. The third term represents maximum value of $\mathcal{W}$, and the last term is the disconnection penalty. The local search proceeds until no improving moves are found or a maximum number of iterations is reached. This approach allows for fine-tuning of the sector boundaries, potentially improving upon the initial heuristic solution by finding a better balance between strong internal traffic flows and equitable workload distribution.

IV. CASE STUDY

To test the viability of the proposed approach, we use the observed traffic data of Madrid Area Control Center (ACC) upper airspace (above 37,500 feet) as a case study. The underlying assumption is that sectors span all the flight level at upper airspace. Data analysis from Eurocontrol [19] indicates requested cruise levels have risen over the past 25 years; by September 2021, 40% of flights requested to fly above FL350 compared to just 6% in September 1995. The Madrid ACC above FL375 is structured into 50 basic volumes (Fig. 1), representing a planar projection of three-dimensional basic volumes at this uniform high altitude. In this study, we analyzed one month (July 31 - August 31, 2020) of 52,325 ADS-B flight data from FlightRadar24 [20], over the Madrid ACC.

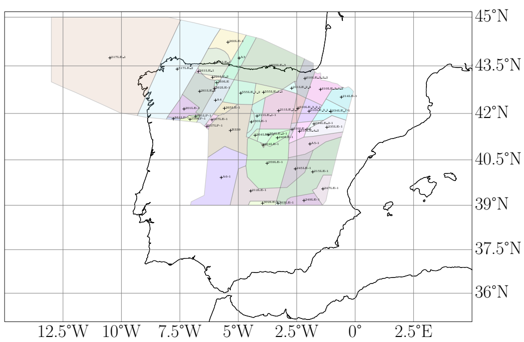


Figure 1. Madrid ACC above FL375, with the names and the geographical centers of basic volumes.

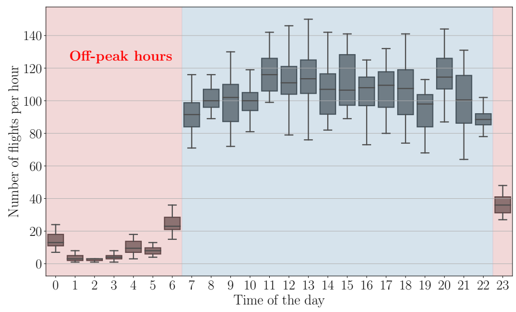


Figure 2. Hourly distribution of one month flight data (52,325 flights) over Madrid ACC.

Fig. 2 illustrates the hourly distribution of flights during this month. The data exhibits a distinct pattern with significantly higher flight volumes between 07:00 and 22:00 as expected.

During this peak period, the mean number of flights per hour is approximately 99. Notably, this peak period also exhibits higher variance, as evidenced by significant fluctuations in flight numbers from hour to hour, ranging from fewer than 70 to more than 140 flights. In contrast, the off-peak hours (23:00-06:00) show a markedly lower mean of about 10 flights per hour, with lower absolute variance, although the relative variance may be higher due to occasional small spikes in activity. The mean aircraft count during peak hours is approximately 10 times that of off-peak hours. In this study, we utilize the accumulated hourly flight data for one month during the peak period (7:00-22:00) as our experimental instances.

In this paper, because the value of $\overline{W}$ is not applicable, we use a predefined range of K as an alternative approach. The input parameters are set as follows: $\alpha = 0.5$, and $K \in [5, 15]$. For this preliminary analysis we assume $SAB = \emptyset$. All experiments are conducted on an AMD Ryzen Threadripper 3990X 64-Core Processor. The implementation utilizes `Gurobi` Optimizer [21] for solving the MIP problem, and `NetworkX` [22] for graph modeling. The mixed-integer programming (MIP) optimality gap tolerance is set to zero, and a maximum time limit of 3,600 seconds is imposed.

V. RESULTS AND DISCUSSION

In this section, we present our results and discussion. First, we evaluate the capability of the proposed heuristic approach by comparing it with the MIP optimal solution, alongside with a discussion on MIP's computational efficiency. Secondly, we examine the overall workload balance and air traffic flow convexity across traffic scenarios for both MIP and heuristic. The last part discusses the quality of generated shape of the sectors.

A. Computational performance

Fig. 3 presents the results of optimal solutions and computational times. Notably, our improved formulations for MIP (detailed in Sections III-B1, III-B2, III-B3) enable the solving

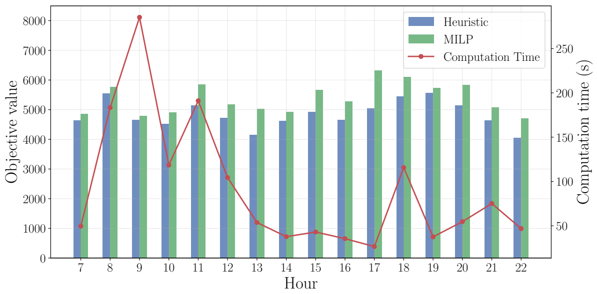


Figure 3. The comparison of optimal MIP solution and heuristic performance (left y-axis), with the computational time of MIP (right y-axis).

of all instances within reasonable computational times, representing a significant advancement in problem tractability. Using the MIP model as a benchmark, we evaluate our heuristic approach's performance for airspace sector design. Results show that the heuristic achieves solutions within an average 11% gap of the optimal MIP values (Fig. 3). Given that the proposed heuristic solves problems nearly instantaneously, it provides a high-quality approximation of the exact solution while requiring minimal computational resources.

B. Workload balance and air traffic convexity

This section examines two key components of our objective function: workload balance and air traffic flow convexity. We assess workload balance by analyzing the standard deviation of workload distribution across sectors; a lower standard deviation indicates a more equitable distribution among controllers. Air traffic flow convexity is evaluated by measuring inter-sector flow; lower inter-sector flow values suggest improved flow-convex conditions for all sectors. These metrics allow us to quantitatively compare the effectiveness of different sector designs in achieving balanced workloads and optimizing traffic flow patterns.

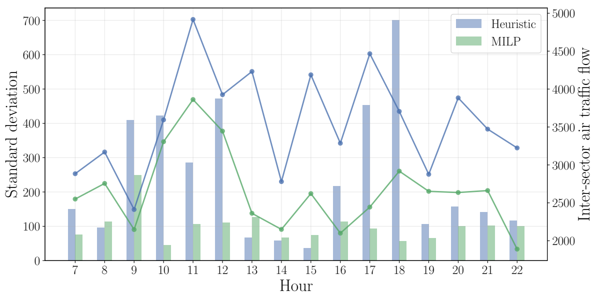


Figure 4. The comparison of sector workload standard deviation (left y-axis, bars) and inter-sector traffic flow (right y-axis, lines) between MIP and heuristic.

The results reveal that the heuristic approach sometimes outperforms the MIP model in terms of workload standard deviation (Fig. 4). The performance of both approaches varies throughout the traffic scenarios, with the heuristic showing lower standard deviations in several instances (8, $13-15$). In contrast, the MIP approach consistently achieves lower inter-sector air traffic flow compared to the heuristic approach, with a average difference of 907.12 flows.

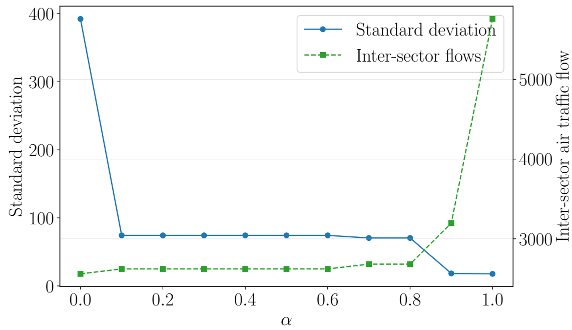


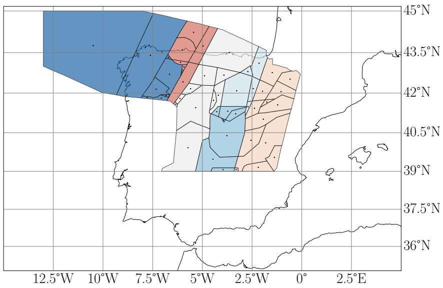
Figure 5. Sensitivity analysis of α between standard deviation (left y-axis) and inter-sector traffic flow (right y-axis), with $H = 15$ & MIP case study.

With $H = 15$ and MIP as a case study, we also perform the sensitivity analysis of α. Fig. 5 demonstrates the trade-off between workload balance (measured by standard deviation) and inter-sector traffic flows as α varies from 0 to 1. When α approaches 0, the objective function prioritizes minimizing inter-sector flows, resulting in lower inter-sector traffic flow values but higher workload standard deviation, indicating less balanced workload distribution across sectors. Conversely, as α increases toward 1, the focus shifts to workload balancing, leading to lower standard deviation but higher inter-sector flows. There appears to be a notable inflection point around $\alpha = 0.1$ to 0.6, where both metrics show reasonable performance, suggesting this range might offer a good compromise between workload balance and traffic flow optimization. This analysis explains why the study chose $\alpha = 0.5$ as the default parameter value, as it provides a balanced consideration of both objectives.

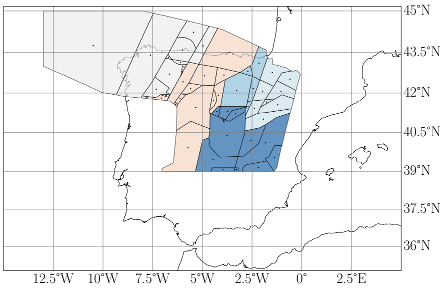
A potential reason influencing the counter-intuitive results of workload balance is the weighting of components in the objective function. With weights α set to 0.5, the flows and workload components are theoretically balanced. However, the bi-directional nature of flows tends to inflate their component's value relative to the workload component. This inherent characteristic of the problem formulation may lead the MIP model to the maximization of internal sector flow in its pursuit of global optimality, potentially at the expense of workload balance in some scenarios. The heuristic, less constrained by this global optimization approach, may inadvertently achieve better workload distribution in certain cases.

C. Sector shape evaluation

To provide a comprehensive analysis of sector generation, we present two evaluation cases. First, we compare the sectors generated by MIP and heuristic for a single traffic scenario ($H = 9$), chosen for its computational intensity in the MIP approach. Fig. 6a illustrates the MIP-generated sector design, while Fig. 6b shows the heuristic result. Second, we examine how sector designs change across different traffic scenarios using the MIP results. Fig. 7 depicts the MIP results for $H = 7$, $H = 17$, and $H = 22$ respectively, allowing us to analyze the adaptability of sector designs to varying traffic patterns.



(a) Six sectors generated by the MIP model.



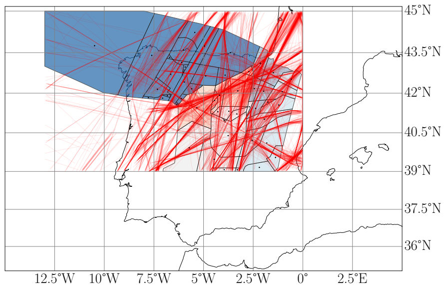
(b) Five sectors generated by the heuristic.

Figure 6. MIP & heuristic sector design based on $H = 9$ traffic scenario.

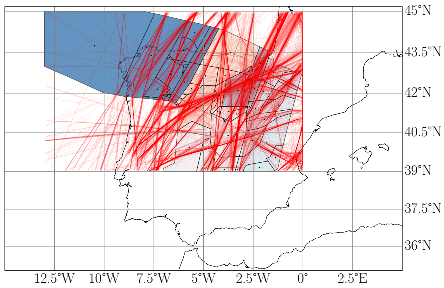
It is important to note that the range of possible sectors, K, is constrained to $[5, 15]$ for MIP. The objective function of our MIP inherently favors a lower number of sectors, as it aims to maximize internal traffic flow and minimize workload. In an non-peak case that has a very low total workload across the airspace, this would lead to a single sector ($K = 1$). The extended computational time observed for the MIP method in this scenario can be attributed to its inability to find an optimal solution when $K = 5$ for $H = 9$.

Despite the differences in approach and computational characteristics, it's important to note the significant similarities in the outcomes of both MIP and heuristic. Remarkably, there is a substantial overlap in the sector boundaries produced by the two methods. A notable example is the sector located in the west and the middle region of the airspace.

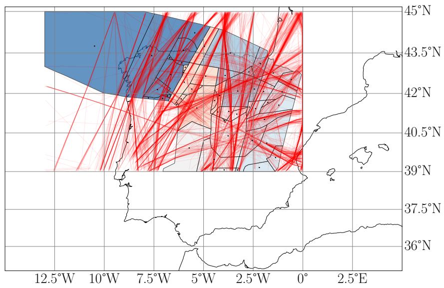
It is also reflected in Fig. 7 how flows are aligned with sector borders, maximizing flights time inside the same sector and, consequently, reducing the coordination among sectors. Even though main traffic patterns are similar in the hours explored, some small adjustments are done to accommodate the little differences. The sector design adapts to specific traffic patterns, as evidenced at $H = 7$. During this hour, there is a notable increase in west-to-east traffic flow through the north of Spain. In response, the western sector is designed with a wider boundary. This expanded shape allows aircraft following this traffic pattern to remain within a single sector for a



(a) Sector design at $H = 7$.



(b) Sector design at $H = 17$.



(c) Sector design at $H = 22$.

Figure 7. MIP sector design comparison for $H = 7$, $H = 17$, $H = 22$ traffic scenarios, with the corresponding flight traffic projection.

longer duration, potentially reducing the number of handoffs and improving overall efficiency. Similarly, we evaluated the heuristic approach under the same varying traffic conditions, which also shows a certain level of adaptability.

VI. Concluding Remarks

This study introduces a Mixed Integer Programming (MIP) model for airspace sector design, use the Madrid Area Control Center as a case study. Key findings include:

1) The MIP model and the heuristic generate operationally relevant sector designs that effectively align with major traffic flows and adapt to temporal variations in traffic patterns.
2) Our proposed heuristic approach achieves good solutions with exceptional computational efficiency, requiring less

than one second of processing time.

3) Interestingly, the heuristic sometimes suggests solutions with improved workload balance, attributed to the nature of bi-directional traffic flows in the problem.

These results open several promising avenues for future research. First, development of a hybrid approach, utilizing the heuristic to provide initial feasible solutions for the MIP model, potentially enhancing overall efficiency for the model. Second, extension of the model to address multi-layer sector design, increasing its applicability to more complex airspace and traffic structures. Lastly, investigation of sector design robustness under various operational conditions, ensuring adaptability to real-world scenarios such as adverse weather condition.

ACKNOWLEDGMENT

This paper is part of the SMARTS project. SMARTS has received funding from the SESAR 3 Joint Undertaking (JU) under grant agreement No.101114686. The JU receives support from the European Union's Horizon Europe research and innovation programme and the SESAR 3 JU members other than the Union. UK participants in SMARTS receive funding from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant numbers 10086651 (Lancaster University) and 10091277 (NATS)]. Opinions expressed in this work reflect the authors views only, and the SESAR 3 JU and UKRI are not responsible for any use that may be made of the information contained herein.

REFERENCES

[1] EUROCONTROL, *Summer 2023: High weather impacts on the network*, https://www.eurocontrol.int/publication/summer-2023-high-weather-impacts-network, Accessed December 31st, 2023.

[2] B. Pang, W. Dai, T. Ra, and K. H. Low, "A concept of airspace configuration and operational rules for UAS in current airspace," in *2020 AIAA/IEEE* 39^{th} *Digital Avionics Systems Conference (DASC)*, IEEE, 2020, pp. 1–9.

[3] 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*, IEEE, 1994, pp. 291–297.

[4] D. Delahaye, M. Schoenauer, and J.-M. Alliot, "Airspace sectoring by evolutionary computation," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, IEEE, 1998, pp. 218–223.

[5] P. Flener and J. Pearson, "Automatic airspace sectorisation: A survey," *arXiv preprint arXiv:1311.0653*, 2013.

[6] H. Trandac, P. Baptiste, and V. Duong, "Airspace sectorization by constraint programming.," in *RIVF*, Citeseer, 2003, pp. 49–58.

[7] R. Ehrmanntraut and S. McMillan, "Airspace design process for dynamic sectorisation," in *2007 IEEE/AIAA* 26^{th} *Digital Avionics Systems Conference*, IEEE, 2007, pp. 3–D.

[8] J. Mitchell, G. Sabhnani, R. Hoffman, J. Krozel, and A. Yousefi, "Dynamic airspace configuration management based on computational geometry techniques," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, p. 7225.

[9] C. Brinton, J. Hinkey, and K. Leiden, "Airspace sectorization by dynamic density," in 9^{th} *AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009, p. 7102.

[10] P. Kopardekar and S. Magyarits, "Dynamic density: measuring and predicting sector complexity [ATC]," in *Proceedings. The* 21^{st} *Digital Avionics Systems Conference*, IEEE, vol. 1, 2002, pp. 2C4–2C4.

[11] S. Kulkarni, R. Ganesan, and L. Sherry, "Static sectorization approach to dynamic airspace configuration using approximate dynamic programming," in *2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings*, IEEE, 2011, J2–1.

[12] J. Tang, S. Alam, C. Lokan, and H. A. Abbass, "A multi-objective approach for dynamic airspace sectorization using agent based and geometric models," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 89–121, 2012.

[13] X. Zou, P. Cheng, B. An, J. Song, *et al.*, "Sectorization and configuration transition in airspace design," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[14] H. P. Young, *Equity: in theory and practice*. Princeton University Press, 1995.

[15] J. Oehrlein and J.-H. Haunert, "A cutting-plane method for contiguity-constrained spatial aggregation," *Journal of Spatial Information Science*, no. 15, pp. 89–120, 2017.

[16] T. Shirabe, "Districting modeling with exact contiguity constraints," *Environment and Planning B: Planning and Design*, vol. 36, no. 6, pp. 1053–1066, 2009.

[17] H. Validi, A. Buchanan, and E. Lykhovyd, "Imposing contiguity constraints in political districting models," *Operations Research*, vol. 70, no. 2, pp. 867–892, 2022.

[18] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math*, pp. 269–271, 1959.

[19] EUROCONTROL, *Data Snapshot: Aircraft keep flying higher, creating extra airspace capacity and burning less fuel, saving 1 million tonnes of* CO_2 *per year*, https://www.eurocontrol.int/sites/default/files/2021-11/eurocontrol-data-snapshot-21-flying-higher.pdf, Accessed December 31st, 2021.

[20] Flightradar24, *Data*, https://www.flightradar24.com/data.com, Accessed July 3rd, 2023.

[21] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2024. [Online]. Available: https://www.gurobi.com.

[22] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.