This is a preprint of the following article, which will be available from personal website/university archive.
Go Nam Lui, Luigi De Giovanni, Martina Galeazzo, Guglielmo Lulli, Iciar Garcia-Ovies Carro, and Rebeca Llorente Martinez. "Robust Dynamic Airspace Configuration: An Optimization Approach." Journal of Air Transportation (2026): 1-12. https://arc.aiaa.org/doi/10.2514/1.D0587.
The published article may differ from this preprint, and is available at the DOI above.

Robust Dynamic Airspace Configuration: An Optimization Approach

Go Nam Lui*[1], Luigi De Giovanni†[2], Martina Galeazzo‡[3], Guglielmo Lulli§[4], Iciar Garcia-Ovies Carro¶[5], and Rebeca Llorente Martinez‖[6]

[1]City University of Hong Kong, Hong Kong SAR
[2,3]Università di Padova, Via VIII Febbraio, 2, 35122 Padova PD, Italy
[4]Lancaster University, Lancaster, LA1 4YW, United Kingdom
[5,6]CRIDA, C/ de Campezo, 1 San Blas-Canillejas 28022 Madrid, Spain

Abstract The Air Traffic Management (ATM) system is under considerable pressure, with congestion occurring almost daily, particularly during summer. Looking ahead, the system is expected to confront even greater complexity driven by sustained traffic growth, new airspace user categories, and more extreme weather events. In this context, efficiently managing limited airspace capacity while accounting for uncertainty in traffic demand has become critical. Dynamic Airspace Configuration provides a framework to maximize efficiency by adapting capacity to evolving traffic patterns, thereby reducing overflow, regulations, and delays. Given a predefined set of configurations, we aim to determine an optimal robust configuration plan that absorbs air traffic under demand uncertainty. To achieve this, we propose a robust optimization formulation that explicitly captures uncertainties in traffic demand and sector capacities, allowing solutions with adjustable protection levels against potential demand increases. To ensure computational efficiency, we design an algorithm that uses a constrained shortest-path procedure as its core component. We evaluate our approach on Madrid ACC using available configurations and traffic data from August 2024. Results explore trade-offs between minimizing overflow and increasing robustness, demonstrating that even moderate conservatism can substantially influence excess traffic and associated delays, underscoring the importance of computing optimal robust solutions.

1 Introduction

Air traffic controllers (ATCO) face increasing challenges due to the growing complexity of airspace management, driven by rising air traffic demand and adverse weather conditions. In the near

*Research Assistant Professor, Department of Mechanical Engineering. Former senior research associate at Lancaster University
†Associate Professor, Dip. di Matematica "T. Levi-Civita".
‡PhD, Dip. di Matematica "T. Levi-Civita".
§Professor, Centre for Transport and Logistics, g.lulli@lancaster.ac.uk, Corresponding Author.
¶Professor, Centre for Transport and Logistics, g.lulli@lancaster.ac.uk, Corresponding Author.
‖R&D Engineer, Centro de Referencia I+D+i en ATM (CRIDA).

future, the incorporation of emerging airspace users, including hybrid-electric regional aircraft and supersonic or near-supersonic commercial jets, into the Air Traffic Management system is likely to intensify these challenges. The Single European Sky ATM Research (SESAR) proposes Dynamic Airspace Configuration (DAC) as the primary method to accommodate demand and reduce delays. As a key element of the Airspace Architecture Study SESAR Joint Undertaking (2019), DAC is also being leveraged by the SMARTS project to deliver capacity with maximum efficiency Horizon Europe (2023). DAC relies on an airspace model based on the concept of sectors - i.e., the units (portions) of the airspace managed by one or more ATCO. Each sector has an associated capacity, which indicates the volume of air traffic that can be safely managed within it. The union of a subset of non-overlapping sectors that covers the whole considered airspace is an airspace configuration. DAC allows optimizing airspace capacity provision by adopting different airspace configurations over time (resulting in a sequence of configurations, or configuration plan) to adapt to traffic variations and prevent uneven workloads. Dynamically rearranging airspace during a given time frame offers many advantages, such as more efficient capacity management and a balanced overload distribution. Hence, the focus of the study is not on the configuration itself, but rather on the sequence of configurations (configuration plan) to be deployed. More specifically, we address the Dynamic Airspace Configuration problem under uncertainty by proposing a robust optimization framework capable of producing configuration plans that remain feasible across diverse traffic scenarios. To mitigate excessive conservatism, the framework incorporates a tunable parameter, Γ, which limits the number of uncertain parameters (traffic demand and sector capacity) allowed to deviate from their nominal values simultaneously. By adjusting the value of Γ, decision-makers can explicitly balance robustness against optimality.

For the practical deployment of the proposed framework, it is essential that the robust formulation compute optimal configuration plans within short computational times. To enhance computational performance, we adopt a graph-based modelling approach that captures the temporal evolution of airspace configurations together with all operational constraints, including configuration-change limits and compatibility requirements. In this formulation, the DAC problem becomes a constrained shortest-path problem on the constructed graph, where each path represents a feasible sequence of configurations. The associated cost of the shortest path corresponds to the robust cost of the plan, which we formally define as the total cost evaluated under the worst-case scenario, assuming that up to Γ uncertain parameters deviate from their nominal values to their maximum limits. This graph-based representation enables the use of specialized shortest-path algorithms, allowing robust optimal solutions to be computed efficiently and substantially reducing solution times.

It is important to observe that the proposed approach and formulations hold for any criterion of performance (objective function), provided that it is configuration and air traffic specific. For instance, several metrics have been used to assess both traffic pattern complexity and reconfiguration complexity Zelinski and Lai (2011) in DAC operations. Traffic pattern complexity metrics include the percentage of flight tracks within two nautical miles of sector boundaries, as flights in close proximity to boundaries increase controller awareness workload Brinton and Cook (2008); Kopardekar et al. (2009). Additional metrics capture the average distance between traffic flow intersections and sector boundaries, as controllers require sufficient time to familiarize themselves with flights before they approach major intersections. The number flights with short flight time within the sector (less than two minutes) also contributes to workload, as these situations often require direct handoffs without controllers taking ownership Jung et al. (2010). Reconfiguration complexity is measured through volume transfer complexity and aircraft transfer complexity, with studies identifying percent airspace volume transferred and number of aircraft transferred as primary contributors to controller workload during reconfiguration events Lee et al. (2010); Jung et al.

(2010). As mentioned above any of these metrics could be potentially used within our approach.

The geometric and kinematic complexity metrics discussed above illustrate the flexibility of our approach. To accurately assess controller workload and demonstrate our model's practical capabilities, we use both the excess of demand and a complexity measure calculated using the COMETA tool (Cognitive Model for the assessment of the workload of ATCO), a computerized model developed by CRIDA for ENAIRE. More specifically, we evaluate our robust optimization framework on the Madrid Area Control Center (ACC) using real configuration data and traffic demand from August 2024. Our computational study systematically explores the trade-offs between traffic overflow minimization and robustness levels, demonstrating that even moderate conservatism (small Γ values) can substantially impact traffic excess and resulting delays. These findings underscore the practical importance of computing optimal robust solutions that explicitly balance operational efficiency with reliability under uncertainty.

The paper is structured as follows: Section 2 provides a literature review on dynamic airspace configuration and sectorization. The mathematical model is presented in Section 3, while the corresponding graph-based solution approach is described in Section 4. Section 5 describes our case study on Madrid ACC and presents the complexity measure used in our experiments. Section 6 presents the computational results and their analysis and discussion. Finally, Section 7 concludes this study and outlines potential future works.

2 Literature review

Literature in this field is extensive and rich, and, among the variety of contributions, two main lines of work can be outlined: Dynamic Airspace Sectorization (DAS) and Dynamic Airspace Configuration (DAC). Although this specific terminology is generally agreed upon, the definitions of DAS and DAC can sometimes overlap; therefore, we begin by clarifying the definitions that will be used from now on, following those provided in Gerdes et al. (2018). In this light, DAS relies on the concept of an unstructured airspace that can be freely partitioned to form different configurations, whereas in DAC, predefined portions of the airspace (airspace blocks) are periodically rearranged into sectors to obtain the configuration plan. For instance, one of the theoretical tools used in DAS are Voronoi diagrams (see, e.g., Gerdes et al. (2018); Wong et al. (2020)), in which a set of points (seeds) is used to partition the plane into regions containing all the points with the same closest seed. As for DAC approaches, we define a further division into two categories, based on the permitted airspace blocks combinations: the first category takes into account only information on the airspace blocks and, therefore, allows for a higher degree of freedom in their aggregation with respect to the second category, in which only predefined sectors and/or configurations can be considered.

Many techniques have been proposed to tackle both categories; as for the first, in Sergeeva et al. (2017) and Sergeeva et al. (2015) DAC is modeled as a graph partitioning problem on a graph whose nodes correspond to airspace blocks and arcs represent the adjacency relation between blocks. Airspace blocks are divided into two groups: non-shareable blocks (that can be chosen as the center of sectors) and shareable blocks. In both cases, the objective function consists of several terms, taking into account workload imbalances between sectors and traffic complexity, and the problem is solved via genetic algorithms. Reference Treimuth et al. (2016) considers the same graph, on which graph partitioning is performed to produce a time-dependent set of available configurations. An Integer Linear Programming (ILP) model provides the configuration plan by choosing one configuration for each time period, with the goal of minimizing the costs associated to workload and its balance, and to the change rate between configurations. Due to its exponential number of variables, the model is solved with column generation and a Branch and Price algorithm. In

Bloem and Kopardekar (2008), a recursive greedy algorithm is proposed; in every time period, it considers the current airspace sectors and combines the under-utilized ones in light of a measure on the predicted traffic excess, where traffic is given by the maximum aircraft count in a 15-minute time interval. As a bridge between the two categories, reference Bedouet et al. (2016) explores the possibility of combining airspace blocks into both commonly used sectors and new sectors with admissible shapes.

Regarding the second category, reference Lema-Esposto et al. (2021) presents an ILP model, in which each airspace block is assigned to a sector from a predefined set, with constraints ensuring that no block is left unassigned or assigned to multiple sectors and that the upper limit on the number of configured sectors is not exceeded. In Bloem and Gupta (2010), the goal of providing a configuration plan that minimizes workload cost (considering the excess workload in each sector and each time period, where workload is given by the peak traffic count in every period) is met by employing three algorithms: a myopic heuristic, an exact dynamic programming algorithm, and a rollout approximate dynamic programming algorithm. Among the tools that have been put into play, we also mention machine learning techniques: in Gianazza (2010), for instance, Neural Networks are used to evaluate workload probabilities, and airspace blocks are combined through a Branch and Bound algorithm. As for generating the configuration plan, the current configuration is evaluated at each time step and reconfiguration is triggered whenever workload is too close to given upper or lower bounds. A recent study Galeazzo et al. (2024) contributes an ILP model that generates optimal configuration plans minimizing traffic overload, with validation performed on historical data from Madrid ACC under various time discretizations and traffic increment scenarios.

While uncertainty is a well-studied element in broader Air Traffic Flow Management (ATFM) problems, its integration into the DAC problem remains largely unexplored. To the best of our knowledge, this work represents the first application of a Γ-robust optimization framework to the DAC problem, building upon the deterministic foundations established in Galeazzo et al. (2024). In the wider ATM literature, alternative paradigms such as stochastic programming Solak et al. (2018); Wang and Jacquillat (2020) – sometimes formulated via chance-constrained optimization Yan et al. (2025) – have gained attention for their ability to satisfy constraints across probabilistic futures, offering an intuitive link to operational risk. However, such approaches typically require precise knowledge of probability distributions for traffic demand and capacity, or the generation of extensive scenario trees. This data is often difficult to characterize accurately in volatile tactical environments. By contrast, the Γ-robust approach requires only the definition of uncertainty intervals (bounds), making it more resilient to distributional errors. Furthermore, while stochastic and scenario-based models often suffer from significant computational complexity (e.g., the curse of dimensionality), the proposed graph-based robust formulation maintains computational tractability, allowing it to be integrated directly into tactical-phase decision support.

3 Mathematical model

In this section, we introduce the robust dynamic airspace configuration mathematical programming formulation. Our objective is to compute a configuration plan, i.e., a sequence of airspace configurations over time that allows absorbing the expected air traffic demand as much as possible, or in other words, that allows reducing air traffic control regulations and consequently air traffic delays. Specifically, we suppose that a family of configurations $\mathcal{C}$ from which to choose the active (implemented) one at each time interval is given, together with the air traffic demand and the capacity for each sector of the airspace during a time horizon $[0, T]$. The time horizon is discretized into n time periods, and the related set is denoted by $\mathcal{T} = 1 \dots n$.

For the sake of clarity and to facilitate understanding, we first present the deterministic formulation of the problem, originally introduced in Galeazzo et al. (2024), before introducing its robust counterpart. In the formulations below, we use the excess of air traffic demand of configuration c at any time period $t \in \mathcal{T}$ – here denoted $e_c(t)$ – as a criterion of performance. The benefit of using the excess of air traffic demand as performance measure is that any fluctuation in demand or capacity is naturally captured as a change in the excess itself. However, as mentioned in the Introduction, other performance criterion (i.e., workload metrics) could be potentially be used within our approach.

3.1 Deterministic DAC formulation

The deterministic DAC problem is formulated as an integer linear program, using the following decision variables, one for each $c \in \mathcal{C}$ and $t \in \mathcal{T}$:

$$x_t^c = \begin{cases} 1, & \text{if configuration } c \text{ is implemented at time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

To model the fact that some configurations cannot be operated at specific time periods, we denote by $\mathcal{C}^t \subseteq \mathcal{C}$ the set of feasible configurations at time t, and limit the definition of decision variables to these subsets.

The objective is to minimize the total excess of demand:

$$\min \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}^t} e_c(t) \cdot x_t^c. \tag{1}$$

Decision variables are subject to the following conditions:

$$\sum_{c \in \mathcal{C}^t} x_t^c = 1 \quad \forall t \in \mathcal{T} \tag{2}$$

$$x_t^c - \sum_{c' \in \mathcal{C}_c^{t+1}} x_{t+1}^{c'} \leq 0 \quad \forall t \in \mathcal{T},\, c \in \mathcal{C}^t \tag{3}$$

$$x_t^c - x_{t-1}^c \leq x_{t+\delta}^c \quad \forall t \in \mathcal{T},\, c \in \mathcal{C}^t,\, \delta = 1, \ldots, t_p - 1 \tag{4}$$

$$x_t^c \in \{0, 1\} \quad \forall t \in \mathcal{T},\, c \in \mathcal{C}^t. \tag{5}$$

Constraints (2) impose that exactly one feasible configuration is active (implemented) at each time period. Constraints (3) and (4) impose that the vector of decision variables belongs to the set of feasible "dynamics", i.e., that the transition between configurations over time can be actually implemented by ACCs. When the configuration is modified, only smooth transitions are allowed. This aspect is modelled by constraints (3), where $C_c^{t+1} \subseteq \mathcal{C}^{t+1}$ is the set of configurations that can be implemented in $t+1$ if c is implemented in t. To overcome the challenge of frequent configuration changes in response to even small demand fluctuations, constraints (4) ensure that any implemented configuration remains active for at least t_p consecutive time periods, that will be referred to as *permanence* interval. Finally, constraints (5) state the decision variables' domain.

3.2 Robust DAC formulation

Due to several unforeseen events, traffic demand and sector capacities are subject to uncertainty. In particular, we consider that $e_c(t)$ can take any value between a minimum $\underline{e}_c(t)$ and a maximum $\overline{e}_c(t)$, and seek for a robust configuration plan, i.e., the plan that minimizes the total excess of demand in the worst case. Formally, denoting by x, e, $\underline{e}$ and $\overline{e}$ the vectors of the corresponding elements, we want to determine a configuration plan that satisfies constraints (2) to (5), and optimizes the following objective function:

$$\min_x \max_{\underline{e} \le e \le \overline{e}} \sum_{t\in\mathcal{T}} \sum_{c\in\mathcal{C}^t} e_c(t) \cdot x_t^c = \min_x \sum_{t\in\mathcal{T}} \sum_{c\in\mathcal{C}^t} \overline{e}_c(t) \cdot x_t^c \tag{6}$$

The solution tends to be over-conservative, since it is unlikely that all the excess coefficients take their maximum value for every sector and time period. We thus resort to the approach proposed by Bertsimas and Sim (2003) to control the degree of conservatism. In particular, we define a parameter Γ that specifies the maximum number of coefficients that can deviate from their nominal values in our uncertainty set. Denoting by N the set of all the indices of the excess coefficients, i.e., $N = \{(c,t) \mid t \in \mathcal{T}, c \in \mathcal{C}^t\}$, the objective function becomes:

$$\min_x \left(\sum_{t\in\mathcal{T}} \sum_{c\in\mathcal{C}^t} \underline{e}_c(t) \cdot x_t^c + \max_{S\subseteq N, |S|\le\Gamma} \sum_{(c,t)\in S} [\overline{e}_c(t) - \underline{e}_c(t)] \cdot x_t^c \right) \tag{7}$$

In this formulation, the "nominal value" specifically refers to the minimum expected excess demand, denoted as $\underline{e}_c(t)$, which serves as the baseline from which the worst-case deviations are measured. The Γ-*robust DAC problem* corresponds to finding the configuration plan that optimizes (7) under constraints (2) to (5), i.e., the configuration plan that minimizes the worst-case total excess of demand assuming that up to Γ excess coefficients are allowed to deviate from their nominal value. The parameter Γ represents the degree of conservatism in the robust solution: a higher Γ corresponds to a greater level of protection in the configuration plan against potential traffic deviations from the nominal value. By varying Γ, the approach enables systematic trade-off analysis between nominal performance and robustness against demand uncertainty, from the more optimistic scenario, where all excess coefficients take their minimum value ($\Gamma = 0$), to the more pessimistic one, where all coefficients are allowed to change ($\Gamma = |N|$) thus resorting to (6). We observe that, because of constraint (2), the total excess of any solution to the DAC problem is determined by $|\mathcal{T}|$ excess coefficients. All the scenarios with $\Gamma \ge |\mathcal{T}|$ are therefore equivalent from a robustness perspective, as, although more than $|\mathcal{T}|$ coefficients are allowed to change, only $|\mathcal{T}|$ of them influence the value of the objective function. As a consequence, a $|\mathcal{T}|$-robust optimal solution is also a Γ-robust optimal solution for every $\Gamma \ge |\mathcal{T}|$, and the trade-off analysis can be restricted to $0 \le \Gamma \le |\mathcal{T}|$.

3.3 Solving the Γ-robust formulation

The formulation for the Γ-robust DAC problem presented above is not suitable for direct implementation with off-the-shelf mathematical programming solvers due to its objective function. However, reference Bertsimas and Sim (2003) provides a convenient solution procedure based on addressing a limited number of standard integer linear programming (ILP) formulations.

At each iteration l of the solution procedure, we solve a specific deterministic ILP problem based on a threshold value d_l. To determine this threshold, we first calculate the maximum possible deviation, $\overline{e}_c(t) - \underline{e}_c(t)$, for all configuration-time pairs in N, and sort these deviation values in descending order. We define d_l as the l-th value in this sorted list (the l-th largest deviation). At each iterations, the threshold d_l decreases, which defines a subset L of configuration-time pairs that take adjusted worst-case values $\overline{e}_c(t) - d_l$ in that specific ILP solve, while all other configurations

Algorithm 1: Optimal Γ-robust configuration plan

Input: Γ, N, and $\mathcal{C}^t$, $\mathcal{C}_c^t$, $\underline{e}_c(t)$, $\overline{e}_c(t)$, $\forall\,(c,t) \in N$
Output: the optimal Γ-robust configuration plan x^*
Sort N by decreasing $\overline{e}_c(t) - \underline{e}_c(t)$ and let $N[l]$ denote the element of N in position l;
Initialize $\mathcal{D}^* \leftarrow +\infty$, $L \leftarrow \emptyset$;
for $l = 0 \dots |N|$ **do**
  if $l = 0$ **then**
    Let $d_l = 0$;
  else
    Let $d_l = \overline{e}_c(t) - \underline{e}_c(t)$ where $(c,t) = N[l]$;
    $L \leftarrow L \cup \{N[l]\}$;
  if $d_l \neq d_{l-1}$ *or* $l = 0$ **then**
    Solve the ILP model (8)–(12) with deviation level d_l and let $\bar{x}$ be the optimal solution with objective value $\mathcal{D}_l$;
    if $\mathcal{D}^* > \Gamma \cdot d_l + \mathcal{D}_l$ **then**
      Update $x^* \leftarrow \bar{x}$ and $\mathcal{D}^* \leftarrow \Gamma \cdot d_l + \mathcal{D}_l$;

remain at their nominal values $\underline{e}_c(t)$. This l-th ILP problem takes the form:

$$\mathcal{D}_l = \min \quad \sum_{(c,t)\in L} [\overline{e}_c(t) - d_l] \cdot x_t^c + \sum_{(c,t)\in N\setminus L} \underline{e}_c(t) \cdot x_t^c \tag{8}$$

$$\text{s.t.} \quad \sum_{c\in\mathcal{C}^t} x_t^c = 1 \quad \forall\, t \in \mathcal{T} \tag{9}$$

$$x_t^c - \sum_{c'\in\mathcal{C}_c^{t+1}} x_{t+1}^{c'} \leq 0 \quad \forall\, t \in \mathcal{T},\, c \in \mathcal{C}^t \tag{10}$$

$$x_t^c - x_{t-1}^c \leq x_{t+\delta}^c \quad \forall\, t \in \mathcal{T},\, c \in \mathcal{C}^t,\, \delta = 1, \dots, t_p - 1 \tag{11}$$

$$x_t^c \in \{0,1\} \quad \forall\, t \in \mathcal{T},\, c \in \mathcal{C}^t, \tag{12}$$

where L denotes the set of (c,t) pairs corresponding to the l largest deviations and d_l is the smallest deviation within this set. Constraint (9) to (12) are defined in Section 3.1. As proven in Bertsimas and Sim (2003), the optimal Γ-robust solution is obtained by evaluating the robust objective $\Gamma \cdot d_l + \mathcal{D}_l$ across all distinct deviation levels and selecting the configuration plan that minimizes this value. Algorithm 1 formalizes this procedure, exploiting the fact that only distinct deviation values need to be considered, thereby avoiding redundant ILP solves.

We remark that Equation (8) corresponds to objective function (1) after replacing the excess coefficients $e_c(t)$ by the maximum excess $\overline{e}_c(t)$ reduced by d_l for a selected subset L of configuration-time pairs (both L and d_l depend on the iteration), and by the minimum excess $\underline{e}_c(t)$ for the remaining pairs. The subsets L are obtained by sorting the configuration-time pairs by decreasing difference between the maximum and the minimum excess, starting from the pair having the largest difference, and by adding, at each iteration, the next pair in the defined order.

4 Graph-based solution approach

The ILP formulation for DAC presented in Section 3 leads to a computationally challenging problem due to the combination of configuration transition constraints and minimum duration requirements.

Moreover, the airspace configuration process requires the analysis of different trade-offs between optimistic nominal performance and degree of conservatism against demand uncertainty. To this end, the Γ-robust approach presented in Section 3.3 calls for the solution of a relevant number of ILP models; in principle, one per time period and per feasible configuration. Since the airspace configuration process has to be performed during the tactical phase of air traffic management, a computationally efficient alternative to solving the ILP formulation for DAC by off-the-shelf mathematical programming solvers is practically needed. To this end, we propose a graph-based representation that captures both the temporal evolution of configurations and the associated constraints. This approach transforms the integer programming problem into a constrained shortest path problem on a directed graph.

4.1 Augmented configuration-transition graph

Let $G = (V, A)$ be a directed acyclic graph (as Figure 1 illustrates) where:

- $V = \{(c,t) \mid t \in \mathcal{T}, c \in \mathcal{C}^t\} \cup \{(\alpha,0), (\omega, |\mathcal{T}|+1)\}$ is the set of nodes representing configuration-time pairs and artificial source and sink nodes (i.e., nodes $(\alpha, 0)$ and $(\omega, |\mathcal{T}|+1)$ respectively);
- $A = A^P \cup A^T \cup A^\alpha \cup A^\omega$ is the set of arcs comprising permanence arcs (A^P), transition arcs (A^T) and artificial arcs (A^α and A^ω), representing potential transitions between configurations over time.

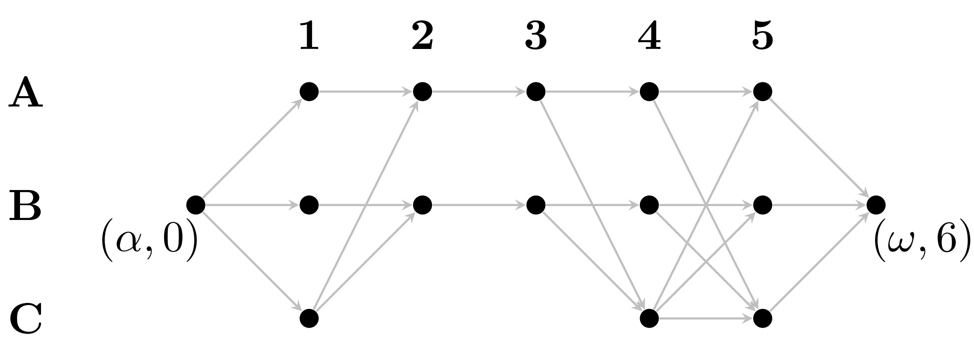

Figure 1: The directed acyclic graph structure illustration of configuration plan, where integers represent time period and capitalized letters represent available configurations. The example shows that Configuration C is not available at time periods 2 and 3.

In particular, the set A^P of *permanence arcs* is defined as:

$$A^P = \{((c,t),(c,t+1)) \mid t \in \mathcal{T}, c \in \mathcal{C}^t \cap \mathcal{C}^{t+1}\}. \tag{13}$$

Each permanence arc represents the possibility of keeping the same configuration active across two consecutive time periods. The set A^T of *transition arcs* is:

$$A^T = \{((c,t),(d,t+1)) \mid t \in \mathcal{T}, c \in \mathcal{C}^t, d \in \mathcal{C}_c^{t+1} \setminus \{c\}\} \tag{14}$$

representing valid transitions between different configurations. The sets of dummy arcs connect the source and the sink nodes to other nodes, and are defined as:

$$A^\alpha = \{((\alpha,0),(c,1)) \mid c \in \mathcal{C}^1\}, \tag{15}$$

$$A^\omega = \left\{((c,|\mathcal{T}|),(\omega,|\mathcal{T}|+1)) \mid c \in \mathcal{C}^{|\mathcal{T}|}\right\}. \tag{16}$$

Each arc $a \in A$ from node $(b, t-1)$ to node (c, t) is assigned a weight $w(a) = e_c(t)$, i.e., the excess of air traffic demand associated to the "head" configuration c at time t. For all arcs $a \in A^\omega$, the weight is set to zero, i.e., $w(a) = 0$. We recall that the proposed approach remains valid for any performance measure that is specific to the configuration and the air traffic. For example, alternative workload metrics may be used without compromising the applicability of the method. It is also important to note that the graph-based approach can naturally capture the workload associated with configuration transitions by assigning an additional weight to the arcs $a \in A^T$.

It is easy to see that, if we neglect the permanence constraints (4), the DAC problem can be solved by finding a shortest path from $(\alpha, 0)$ to $(\omega, |\mathcal{T}|+1)$ according to the metric w. Similarly, we can solve a shortest path problem on G to obtain the optimal solution of the ILP model required at each iteration l of **??** 1 by setting:

$$w(a) = \begin{cases} \overline{e}_c(t) - d_l & \text{if } (c,t) \in L, \\ \underline{e}_c(t) & \text{if } (c,t) \in N \setminus L. \end{cases} \tag{17}$$

For this purpose, efficient procedures, such as, e.g., Dijkstra's algorithm Dijkstra (1959), are available.

To enforce permanence constraints in the configuration dynamics, we augment the graph's state space with a remaining duration counter associated to each node $v = (c, t) \in V$. Each state in the search algorithm is thus represented as a triplet:

$$s = (c, t, \tau)$$

where c is the configuration at time t, and τ denotes the remaining number of time periods that must elapse before allowing a transition to a different configuration. The state s evolves to a new state s' when an arc $a = ((c,t),(c',t+1))$ is crossed. The new state will be:

$$s' = (c', t+1, \tau')$$

where the number of remaining time periods τ' depends on the type of the arc a used to reach $(c', t+1)$, namely:

$$\tau' = \begin{cases} \max\{0, \tau - 1\} & \text{if } c = c', \text{ i.e., } a \in A^P, \\ t_p - 1 & \text{if } c \neq c', \text{ i.e., } a \in A^T \cup A^\alpha \cup A^\omega. \end{cases}$$

Descriptively, the number of remaining time periods required to satisfy the permanence constraints is reset to $t_p - 1$ each time a transition to a new configuration occurs, and decreases by one, reaching a minimum of 0, each time the same active configuration is maintained. A configuration change is permitted when $\tau = 0$, indicating that the permanence constraints are satisfied. The *augmented configuration-transition graph* defined above can be conveniently exploited to design a label setting algorithm that solves DAC as a shortest path problem while guaranteeing that permanence constraints are met.

4.2 Permanence-constrained shortest path algorithm

Our permanence-constrained shortest path algorithm extends Dijkstra's algorithm Dijkstra (1959) to manage airspace configuration transitions while enforcing minimum permanence times. The algorithm operates on the configuration-transition graph G defined in Section 4.1, where edge weights encode excess demand costs. Moreover, its key innovation lies in the integration of a duration-aware state space with efficient First In First Out (FIFO) queue management to ensure both optimality and constraint satisfaction. **??** 2 proceeds as follows:

Algorithm 2: Permanence-Constrained Shortest Path

Input: Graph $G = (V, A)$
Output: Optimal path P^* or $\emptyset$
Initialize FIFO queue $Q \leftarrow \{(\alpha, 0, 0)\}$;
Initialize distance map $\Delta(\alpha, 0, 0) \leftarrow 0$;
Initialize path map $\Pi(\alpha, 0, 0) \leftarrow [(\alpha, 0)]$;
repeat
    Extract state $\mathbf{s} = (c, t, \tau)$ from Q
    if $(c, t) = (\omega, |\mathcal{T}| + 1)$ **then**
        return $P^* \leftarrow \Pi(\mathbf{s})$;
    foreach arc $a = ((c, t), (c', t')) \in A$ **do**
        if $a \in A^T$ **and** $\tau > 0$ **then**
            continue;
        else
            $\tau' \leftarrow \begin{cases} \max\{0, \tau - 1\} & \text{if } a \in A^P \\ t_p - 1 & \text{if } a \in A^T \end{cases}$;
        $\mathbf{s}' \leftarrow (c', t', \tau')$;
        $\gamma \leftarrow \Delta(\mathbf{s}) + w(a)$;
        if $\gamma < \Delta(\mathbf{s}')$ **then**
            $\Delta(\mathbf{s}') \leftarrow \gamma$;
            $\Pi(\mathbf{s}') \leftarrow \Pi(\mathbf{s}) \oplus (c', t')$;
            $Q \leftarrow Q \cup \{\mathbf{s}'\}$;
until Q *is empty*;
return $\emptyset$

1. **Initialization**: Create a FIFO queue Q with the source state $(\alpha, 0, 0)$, a distance map Δ tracking minimum path costs, and path map Π recording node sequences;

2. **State Expansion**:
   - Extract the state $\mathbf{s} = (c, t, \tau)$ from Q;
   - Terminate if (c, t) matches target $(\omega, |\mathcal{T}| + 1)$;
   - For each arc $(c', t+1)$ stemming from node $v = (c, t)$, create a new state $\mathbf{s} = (c', t+1, \tau')$, where τ' is determined by the evolution rule described in Section 4.1;

3. **Cost Update**: For each expanded state $\mathbf{s}'$, update $\Delta(\mathbf{s}')$ and $\Pi(\mathbf{s}')$ and queue $\mathbf{s}'$ into Q, if a cheaper path is found.

The correctness of **??** 2 is guaranteed by the evolution rule that prevents the exploration of unfeasible states related to permanence constraints violation or invalid transitions. We observe that the configuration-transition graph is, by construction, acyclic; therefore, each arc is visited at most once per value of τ. Due to the fact that the graph is also layered, FIFO queuing implies a topological order visit of the graph nodes, which guarantees that the final state for node ω reports an optimal path, with no need to dequeue minimal distance states, differently from the standard Dijkstra algorithm. As a consequence, **??** 2 is correct and has time complexity $O(|A| \cdot t_p)$.

5 Case study

In this section, we will introduce our case study at Madrid Area Control Center (ACC).

5.1 Data description

This study relies primarily on two essential data sources: flight trajectory data and airspace sector information. The flight trajectory data consists of Automatic Dependent Surveillance–Broadcast (ADS-B) data collected through the `OpenSky Network` Schäfer et al. (2014), encompassing aircraft movements over Madrid ACC during August 2024. The dataset comprises 105,854 discrete flight trajectories, and our analysis focuses specifically on aircraft operating between flight levels 245 and 550, representing the upper airspace where most commercial traffic operates.

Figure 2 demonstrates the complex interplay of flight trajectories across the Madrid ACC's controlled airspace, where the black polygon represents its official boundaries. The density distribution of trajectories, represented by red lines, indicates significant traffic convergence along northeastern corridors and coastal approaches.

The airspace sector analysis in this study draws from Solution 44 Lui et al. (2024), developed by the Centro de Referencia I+D+i ATM (CRIDA). The airspace structure consists of 281 basic volumes that are combined to form 119 elementary sectors. These elementary sectors can be further merged into 969 collapsed sectors, which serve as the fundamental components of 285 distinct airspace configurations. Throughout this study, we will refer to the collapsed sectors simply as "sectors", as they represent the operational building blocks from which the set of airspace configurations is constructed. The hourly capacity for each sector is also provided in CRIDA Solution 44. The capacity defines the maximum number of entries in a one-hour time interval.

5.2 Data preprocessing

5.2.1 Traffic demand

For each flight, we mapped its trajectory pattern to the corresponding airspace sectors using predefined sector-flow pattern relationships. We processed the data at five-minute intervals, examining

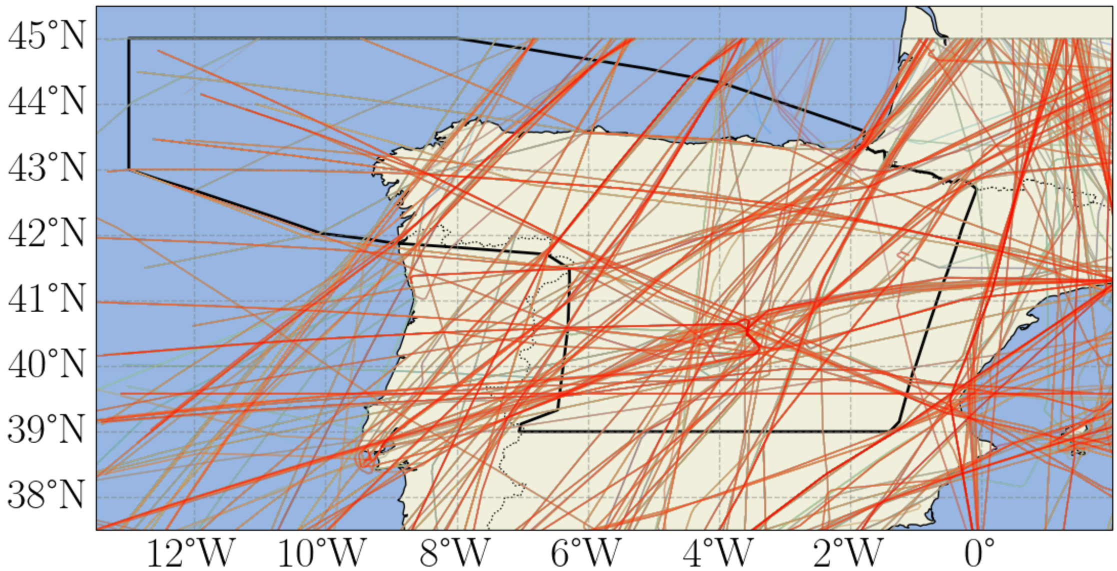

Figure 2: Flight trajectory visualization for August 15, 2024; the black polygon represents the official boundaries of Madrid ACC's controlled airspace.

traffic demand for the upcoming hour at each interval (e.g., looking ahead from 10:00 to 11:00, then from 10:05 to 11:05, etc.). At each interval, we counted the number of flights projected to enter each sector within the next hour. The preprocessing covered all flights operating between August 1 and August 31, 2024, producing rolling one-hour entry demand measurements at five-minute steps that serve as the foundation for our subsequent analysis.

Traffic demand plays a key role in determining the parameters of the objective function. As mentioned in Section 3, the proposed approach can accommodate different performance criteria, provided they satisfy the additivity property over time. In the following, we show how traffic demand is used both to compute excess demand and to feed the complexity calculation, as detailed in the next subsection.

As far as the excess of air traffic demand of configuration c at any time period $t \in \mathcal{T}$ – here denoted $e_c(t)$ – it can be easily computed as:

$$e_c(t) = \sum_{s \in c} \max\{d_s(t) - c_s(t); 0\}, \tag{18}$$

where $d_s(t)$ and $c_s(t)$ are the demand and capacity of sector s at time period t respectively. Within this framework, excess demand ($e_c(t)$) serves as a computational proxy for operational overflow; consequently, minimizing this metric directly correlates with improved ATFM performance indicators, specifically reduced average delays per flight. However, it is important to note that this baseline excess demand quantification does not inherently account for spatial and temporal distribution, which is critical for understanding full operational impacts. Indeed, concentrated overloads increase the need for regulations, leading to larger delays. To mitigate these effects, excess demand parameters can be adjusted, e.g., by using superlinear coefficients to take distribution effects into account.

5.2.2 Traffic complexity for practical validation

While Equation (1) utilizes excess demand for the baseline mathematical formulation, actual air traffic control operations require more sophisticated measures to accurately assess controller workload. To address this and demonstrate the real-world applicability of our framework, we also evaluated the system using a practical sector complexity metric. This metric, calculated using the COMETA tool, allows us to validate the DAC framework in a production-ready environment. The successful integration of this complexity measure into the graph-based DAC approach for practical operational usage has been recently demonstrated in Romero et al. (2025). Within the framework of Air Traffic Complexity Management (ATCM), complexity is understood as a multidimensional concept that goes beyond the mere number of aircraft in a sector. It reflects how challenging it is to maintain safe and efficient operations under varying conditions. Consequently, sector complexity provides a more accurate estimate for assessing controller workload than simple aircraft counts.

The complexity metric implemented in our case study, taken from the SMARTS research project Horizon Europe (2023), is calculated using the COMETA tool (Cognitive Model for the assessment of the workload of ATCO), a computerized model developed by CRIDA for ENAIRE. COMETA has been widely applied in the SESAR program Ibáñez-Gijón et al. (2023), in air traffic complexity management Suarez et al. (2014), and in human factors research Cañas et al. (2018).

This model incorporates the eCOMMET algorithm, which evaluates traffic demand by identifying and quantifying complexity factors. A key concept introduced by eCOMMET is the *equivalent flight*, which distinguishes the contribution of each flight to overall complexity. Rather than simply counting aircraft, the algorithm considers operational factors such as level changes, interactions with other traffic, maneuvers, and trajectory adjustments. This approach ensures that sector occupancy reflects the actual impact of flights on controller workload. eCOMMET Sector complexity is formally defined as

$$CPLX_{total} = CPLX_{base} + a \cdot NSF + b \cdot EVOL + c \cdot SFI + d \cdot MIL + e \cdot OCE$$

where

- $CPLX_{base}$: Initial complexity value, serving as the baseline;
- *NSF (Non-Standard Flow)*: Complexity from flights outside standard routing flows;
- *EVOL*: Complexity associated with evolving or transitioning flights;
- *SFI (Standard Flow Interaction)*: Complexity from interactions within standard flows; the core component of the model;
- *MIL*: Complexity due to military flights;
- *OCE*: Complexity related to oceanic or transatlantic operations.

Each factor is weighted by parameters (a, b, c, d, e), calibrated using operational data to reflect their relative impact.

With the use of COMETA, cognitive complexity over the traffic sample of study is calculated for all the possible sectors contained in the airspace designed, using a 5-minute calculation window with a 1-minute shift. The resulting complexity metric is expressed in aircraft count units, consistent with the occupancy metric and the main input to the sector configuration algorithm.

Once the complexity of each sector has been computed, the next step involves defining capacity thresholds to determine when traffic demand exceeds manageable limits. In particular, the metric requires the establishment of two operational thresholds: the *Peak Capacity Threshold* and the

Sustained Capacity Threshold. The Peak Capacity Threshold refers to a traffic demand level that is considered operationally unacceptable due to its excessive complexity or volume, which may compromise safety and controller performance. The Sustained Capacity Threshold represents a level of traffic complexity that, while not immediately critical, exceeds what an ATCO can safely manage over a prolonged period without degradation in performance or increased risk. For a sector i, the calculation of the peak capacity threshold $CPLX_i^P$ and the sustain capacity threshold $CPLX_i^S$ within SMARTS followed the Warning Density Area method derived from the Hourly Entry Count (HEC) versus Occupancy Count (OCC) relationship. Using calibrated abacuses, two critical occupancy values were defined: a Sustain Occupancy, representing the maximum level of traffic that can be managed safely and consistently, and a Peak Occupancy, indicating the upper limit before sector overload requires regulatory measures.

The corresponding HEC values were then extracted from traffic data in the Madrid ACC sectors, with expert-defined thresholds mapped to HEC values. Building on this, an automatic process — developed according to the methodology described in EUROCONTROL (2007) — was applied, using empirical analysis of HEC→OCC distributions to derive linear trends of maximal and standard occupancy for OCC values as shown in Figure 3. The interval between these thresholds defines the

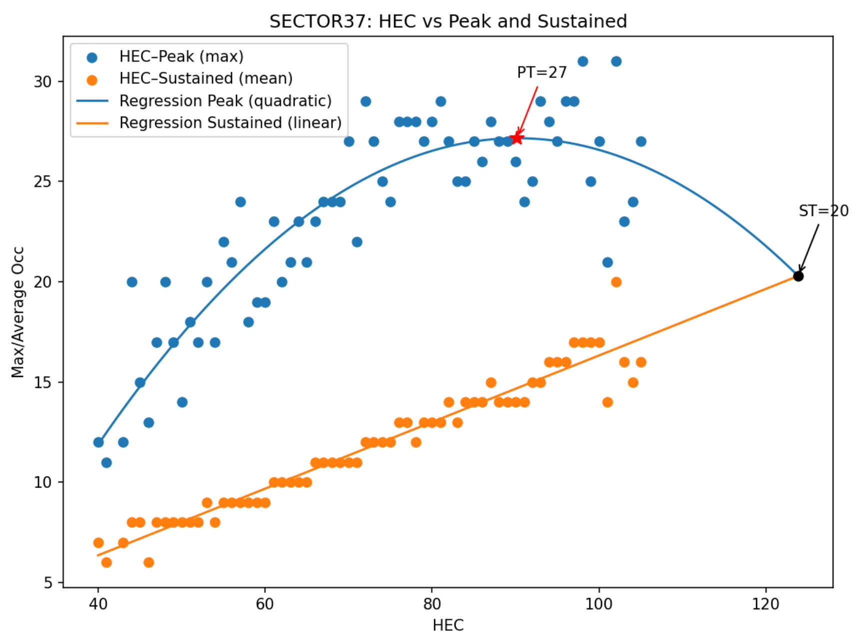

Figure 3: Using the peak and sustain thresholds calculation method for a specific sector of the solution scenario Horizon Europe (2023).

Warning Density Area, where traffic monitoring and tactical interventions ensure sector stability while maintaining a quantifiable link between strategic capacity planning and real-time ATCM regulation.

5.2.3 Valid configuration set

In the DAC formulation (objective function (1), constraints (2) to (5)) and its robust counterpart – as well as in the graph representation of the problem – we used sets $\mathcal{C}^t$ and $\mathcal{C}_c^t$ to denote the set of (valid) configurations available at time period t and the set of configurations that can be reached from configuration c at time period t, respectively. The former set allows us to identify and use only the configurations that are consistent with the rostering of ATCO, while the latter models feasible configuration transitions. Using the set of 285 configurations, we populate sets $\mathcal{C}^t$ and $\mathcal{C}_c^t$ according to the following rules, which mirror the operational needs that arise in practice. For $\mathcal{C}^t$, we establish time-dependent constraints on the number of sectors in valid configurations. To this end, let $\mathcal{S}(c)$ denote the set of sectors that are active in configuration c, $\mathcal{C}^t$ is defined as follows:

$$\begin{aligned} \mathcal{C}^t = \{c \in \mathcal{C} \mid |\mathcal{S}(c)| \leq 7\} \quad & \text{from 6:00 to 7:00} \\ & \text{and from 22:00 to 24:00} \\ \mathcal{C}^t = \mathcal{C} \quad & \text{otherwise} \end{aligned}$$

This filter imposes a maximum limit of 7 sectors during early morning (06:00-07:00) and late night (22:00-24:00) periods, reflecting the reduced traffic demand during these hours. During peak hours (07:00-22:00), no restrictions on the number of sectors are applied, allowing more flexible airspace management based on actual traffic demand.

As for the configuration transitions, given two arbitrary configurations $c_i, c_j \in \mathcal{C}$, a transition from c_i to c_j is valid if either of the following conditions are met:

$$|\mathcal{S}(c_i)| \leq 4 \text{ and } |\mathcal{S}(c_j)| \leq 4$$
$$|\mathcal{S}(c_i) \cap \mathcal{S}(c_j)| \geq 0.5 \cdot |\mathcal{S}(c_i)| \text{ and } ||\mathcal{S}(c_i)| - |\mathcal{S}(c_j)|| \leq 3$$

The transition rule incorporates two key operational constraints: condition 1 states that configurations with 4 or fewer sectors can transition freely among themselves, reflecting greater flexibility in managing smaller configurations. Condition 2 applies to larger configurations. A transition between two configurations is feasible if they share at least 50% of the sectors, promoting operational continuity, and limiting the difference in the number of active sectors between the two configurations to at most 3 sectors, thus preventing abrupt changes in staffing requirements. These constraints are designed to match real operational scenarios while maintaining efficient airspace management. In light of this definition, given configuration c active at time t, $\mathcal{C}_c^{t+1}$ can be defined as follows:

$$\mathcal{C}_c^{t+1} = \{c' \in \mathcal{C}^{t+1} \mid \text{transition from } c \text{ to } c' \text{ is valid}\}. \tag{19}$$

5.3 Experiment setup

We conducted our robust optimization experiments using historical air traffic data from August 2024, incorporating the airspace specifications defined in Solution 44. Our comprehensive analysis covered the entire month to evaluate the framework performance under various operational conditions. Additionally, we performed detailed case studies on three days identified by CRIDA (August 3$^{\text{rd}}$, 17$^{\text{th}}$ and 24$^{\text{th}}$) when extensive regulations were implemented, providing deeper insights into the framework's effectiveness during periods of heightened operational complexity. All computational experiments were performed on a workstation equipped with an AMD Ryzen Threadripper 3990X 64-Core Processor operating at 2.9 GHz base frequency (4.3 GHz boost) with 256 GB DDR4-3200 RAM. The implementation framework, including data processing and Algorithm 1, was developed in Python. For performance-critical operations related to graph modeling, namely the representation of the temporal configuration network structure and the implementation of Algorithm 2, we used C++.

Our analysis covered the operational period from 6:00 to 24:00, discretized into 5-minute intervals as described in Section 5.2.1, resulting in 216 time steps. For robustness considerations, we incorporated uncertainty in traffic demand by setting a threshold of 20% to account for potential demand increases in each sector within the configuration. This threshold was chosen based on historical traffic variation patterns and operational requirements for maintaining safe buffer capacities.

Before discussing the results in details, we here provide a brief analysis on the impact of the conditions on valid configuration sets. We recall that the proposed criteria are a proxy of (sometimes implicit) criteria and rules underpinning the configuration process. Indeed, there might be many factors that affect the airspace configurations decision process. The temporal graph structure is determined by the combination of available configurations (285) and time intervals. Before applying operational constraints, the graph consisted of 61,562 nodes (including artificial source and sink nodes, α and ω, respectively) and approximately 17.46 million arcs representing potential configuration transitions. After implementing the valid configuration constraints defined in Section 5.2.3, we achieved a significant dimension reduction: the number of nodes decreased by 4.2% and edges by 94.1%, leaving 1,026,192 valid transitions in the final graph structure. Therefore, these conditions not only enhance the realism of the proposed model but also offer a significant advantage by reducing the graph size, which in turn improves computational efficiency.

6 Results and discussion

This section presents the computational performance and validation results of our graph-based robust configuration framework. We first demonstrate the framework's viability for tactical-phase operations by analyzing computational times for both deterministic and robust optimization approaches (Table 1), followed by validation results from production deployment.

Table 1: Computational performance of the graph-based airspace configuration framework.

Performance Metric	**Value**	**Unit**
Deterministic Approach		
Average instance generation time	10.42	seconds
Average solving time	0.28	seconds
Average total computational time	**10.70**	**seconds**
Standard deviation for computational time	**1.70**	**seconds**
Robust Optimization Approach		
Average set ($\mathcal{D}$) generation time	29.48	seconds
Average time to retrieve the robust plan per each Γ	0.84	seconds
Average total computational time per each Γ	**40.73**	**seconds**
Configuration plan horizon	18	hours
Time-discretization interval	5	minutes

The deterministic framework achieves an average total computational time of 10.70 seconds (standard deviation: 1.70 seconds), most of which is spent for instance generation (10.42 seconds), and nearly negligible time to solve it (0.28 seconds), thanks to the low computational complexity of Algorithm 2. The low variability indicates consistent performance across planning instances.

Using Algorithm 2 represents a significant advantage over mathematical programming approaches based on directly solving the ILP formulation for DAC through state-of-the-art off-the-shelf solvers, which requires minute-scale computation. The advantage becomes determinant for robust optimization, where an instance of DAC has to be solved at each iteration of Algorithm 1 to generate the set $\mathcal{D}$ of values $\mathcal{D}_l$, one for each distinct deviation d_l of excess parameters (see Equation 8). In principle, as observed in Section 4, up to $|N| + 1$ values may need to be evaluated (one per time period and feasible configuration), which highlights the importance of a fast solution procedure for deterministic DAC.

Based on our experiments (that considered all the days of August 2024 and Γ equal to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200 and 216) computing a robust configuration plan for a given value of Γ takes 40.73 seconds on average, including the time needed for instance generation (about 10 seconds), the time (29.48 seconds) needed to compute all the required values of $\mathcal{D}_l$ (about 90 distinct values per day, on average), and the additional time (0.84 seconds) needed by the last step of Algorithm 1 to select the $\mathcal{D}_l$ leading to the optimal Γ-robust plan and retrieve it.

We observe that the airspace configuration process, as well as the analysis proposed in the remainder of this section, requires evaluating trade-offs between various degrees of conservatism, and thus computing robust plans for multiple values of Γ. In this respect, we note that the only step of Algorithm 1 that depends on the specific value of Γ is the final one. The computation of the set $\mathcal{D}$, the most computationally demanding task, does not depend on Γ and therefore needs to be performed only once. As a consequence, generating the information required by the planning process takes on the order of one minute (about 40 seconds plus less than one second per value of Γ), confirming the framework's suitability for tactical operations and near real-time applications.

Beyond numerical analysis based on excess demand (Equation (18)), we validated the framework in a production-ready environment using practical sector complexity metrics (Section 5.2.2) Romero et al. (2025). The system is built with FastAPI Ramírez (2018) and integrated with CRIDA's PUZZLE tool. In the PUZZLE interface, the user has the flexibility to fix the following parameters:

1. **Overload penalty weight** (o): Applied when configuration complexity exceeds the sustained threshold. For configurations satisfying $CPLX_i > CPLX_i^S$, the objective penalty associated to sector i is computed as:

$$P_{\text{overload}} = o \times (CPLX_i - CPLX_i^S); \tag{20}$$

2. **Underload penalty weight** (u): Applied to configurations operating significantly below capacity. When $CPLX_i \leq 0.4 \cdot CPLX_i^P$, the objective penalty becomes:

$$P_{\text{underload}} = u \times (0.4 \cdot CPLX_i^P - CPLX_i) \tag{21}$$

The sum of overload and underload penalty weights equals 1:

$$u + o = 1; \tag{22}$$

3. **Maximum sectors constraint** ($N_{\max}$): Defines the upper bound on the number of sectors permitted within any configuration plan;

4. **Permanence duration** (t_p): Specifies the minimum activation period for any configuration following transition, ensuring operational stability.

The system consistently delivered effective solutions with 12-16 sector configurations. Flight Management Position (FMP) operators confirmed that the traffic distribution and sector balancing were generally effective, demonstrating the framework's practical viability beyond experimental settings.

6.1 Trade-off analysis in the Γ-robust framework

The analysis presented in this section investigates the trade-off between efficiency and robustness of the configuration plans provided by the Γ-robust optimization framework. To this end, we focus on three selected days in the dataset, namely, August 3rd, 17th and 24th, which are characterized by a relevant operational complexity. For these three cases, Figure 4 displays the value of the worst-case total excess, which is the *Robust Cost* minimized by **??** 1. We recall that the robust cost represents the optimal daily demand excess when accounting for potential demand fluctuations at protection level Γ. It corresponds to the cost of the optimal Γ-robust configuration plan in the worst-case traffic-demand realization among the ones where up to Γ excess parameters deviate from the nominal value. This means that any other configuration plan would perform worse than the robust one in at least one of the traffic realizations allowed by the same protection level. We first

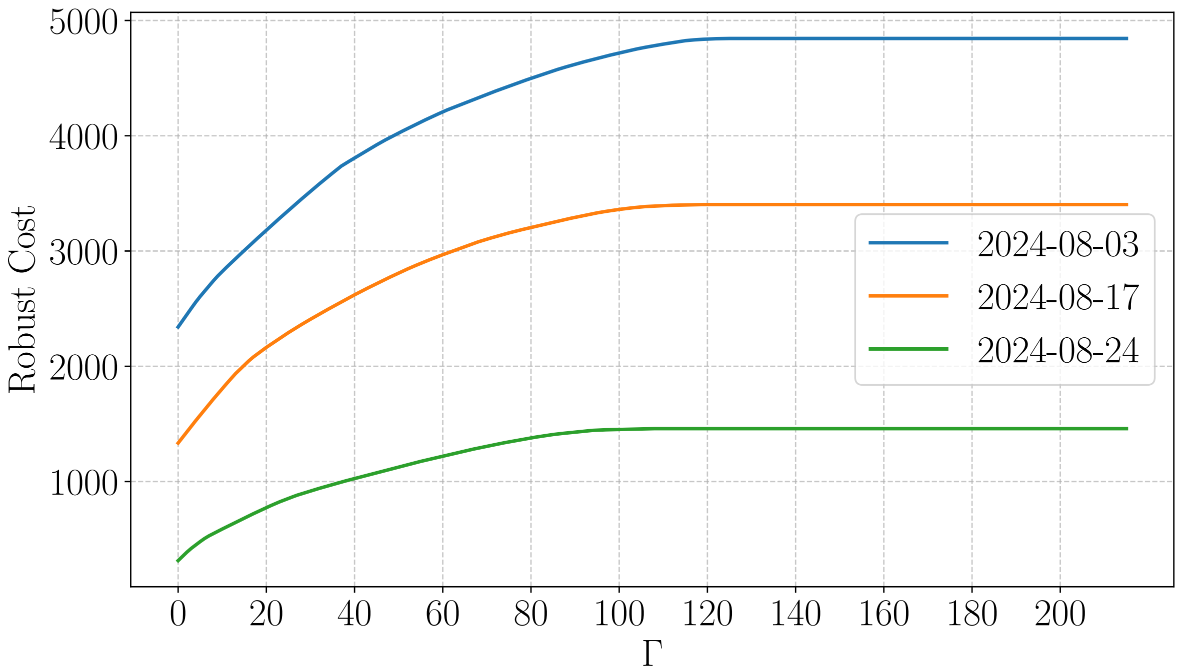

Figure 4: Robust cost v.s. Γ, August 3, 17 and 24, 2024

observe that, for $\Gamma = 0$, representing the optimistic nominal case where non excess deviates from the minimum value, the robust cost is between 314 and 2341. Although this latter value might appear high, it translates to an average of approximately 11 units of excess entries per hour, taking into account that the total excess sums up 216 five-minute interval within a configuration plan. For a configuration consisting of 7 sectors, this means that each sector experiences an excess of at most 2 aircraft per hour - a remarkably efficient outcome given the complexity of airspace management especially if we consider extensive and prolonged congestion phenomena occurred on August 3^{rd}, which affect our nominal traffic data.

The curves reveal three distinct phases in the system's response to increasing protection levels. In a first phase, the robust cost rapidly increases from the nominal value, meaning that even for relatively optimistic scenarios that consider low protection level ($\Gamma \leq 20$), the system, although optimally configured, may experience a significant increase in the total excess. With $\Gamma = 10$, the cost of the worst-case scenario increases by 21% in August 3^{rd}, by 35% in August 17^{th} and by 87% in August 24^{th} (in this case, the nominal excess is fairly small), and by additional 15%, 27% and 59% (respectively) with $\Gamma = 20$. Provided that non-optimal configuration plans have even more critical worst-case performance, we observe that implementing the right robust solution is crucial to avoid huge loss in operational airspace efficiency and, consequently, very large amount of assigned delays.

In a second phase, for moderate level of conservatism up to about $\Gamma = 100$, the marginal increment of the robust cost is mitigated to figures between 20% and 5% (August 3$^{\text{rd}}$ and 17$^{\text{th}}$) or between 45% and 10% (August 24$^{\text{th}}$) per 10-units of Γ. In this phase, by accepting an additional 50% deterioration in the worst-case performance (from 3181 to 4718), the protection level increases fivefold (from 20 to 100). On August 17$^{\text{th}}$ and 24$^{\text{th}}$, the same improvement in robustness derives from additional, respectively, 55% and 87% robust cost.

Finally, the system enters a saturation phase beyond $\Gamma = 100$, where the Γ-robust cost converges to the fully robust cost (which, we recall, is equivalent to $\Gamma = |\mathcal{T}|$, as observed in Section 3.2). This means that no further protection is needed to optimally face uncertainty, and even in the more pessimistic case, where all excess parameters take their maximum value, the worst-case cost is not going to significantly improve. For example, on August 3$^{\text{rd}}$, the convergence is reached for $\Gamma = 124$, therefore the 124-robust configuration plan is able to optimally absorb any level of uncertainty in traffic demand, and provides a worst-case cost of 4884 even in the more pessimistic scenario.

6.2 Simulation of Γ-robust configurations plans

From an operational perspective, the proposed trade-off analysis helps airspace managers understand the efficiency implications of varying levels of preparedness for demand uncertainty. Implementing a configuration with a low protection level tends to perform better in nominal or near-nominal scenarios. However, the risk of very poor performance at unprotected uncertainty levels is significant, with potentially severe delays. On the other hand, more robust optimal configuration plans provide some performance guarantees, even in pessimistic scenarios with high levels of uncertainty, although their performance may deteriorate under more optimistic uncertainty realizations. For example, Figure 4 suggests the airspace managers the importance of considering a protection level of at least $\Gamma = 20$, as the worst-case performance of less conservative optimal solutions (e.g., $\Gamma = 10$) quickly deteriorates already with increasing uncertainty levels. However, optimal Γ-robust configuration plans are designed to provide the best worst-case performance under level of protection Γ. In addition to compare the cost-robustness trade-off, further insights into the performance of specific Γ-robust solutions in different uncertainty scenarios may offer additional guidance to airspace managers.

With reference to the case-study of August 3$^{\text{rd}}$, Figure 5 illustrates the performance of the Γ-robust configuration plans output by **??** 1, one for each conservatism level Γ. The line chart plots the total excess realized by the associated Γ-robust plan in three different uncertainty scenarios: (i) the nominal scenario where all excess cost take the minimum value, (ii) the more pessimistic scenario where maximum excess is considered and (iii) the worst-case scenario associated to Γ (in this case, the total excess is the robust cost). We remark that the excess in nominal (respectively more pessimistic) scenario is increasing (respectively decreasing) with Γ, whereas the robust cost goes from the nominal minimum excess ($\Gamma = 0$) to the fully robust cost ($\Gamma = |\mathcal{T}|$ according to trend analysed in Section 6.1). In particular, each Γ-robust plan may realize a larger excess than the Γ-robust cost if the realized traffic demand affects more than Γ excess parameters.

In order to have further insight into the performance of an optimal Γ-robust configuration plan $\Pi^*(\Gamma)$ in intermediate scenarios, we propose a Monte Carlo simulation approach. For each repetition: we uniformly select a subset of the 216 configuration-time pairs of $\Pi^*(\Gamma)$; we assign the maximum excess to the selected pairs and the nominal one to the remaining ones; we compute the total excess of $\Pi^*(\Gamma)$. Figure 6 presents the cumulative distribution functions (CDF) of the total daily excess demand

First, we observe that no single plan dominates the others across various levels of total daily excess demand. For lower values of total daily excess demand (below approximately 3500, corre-

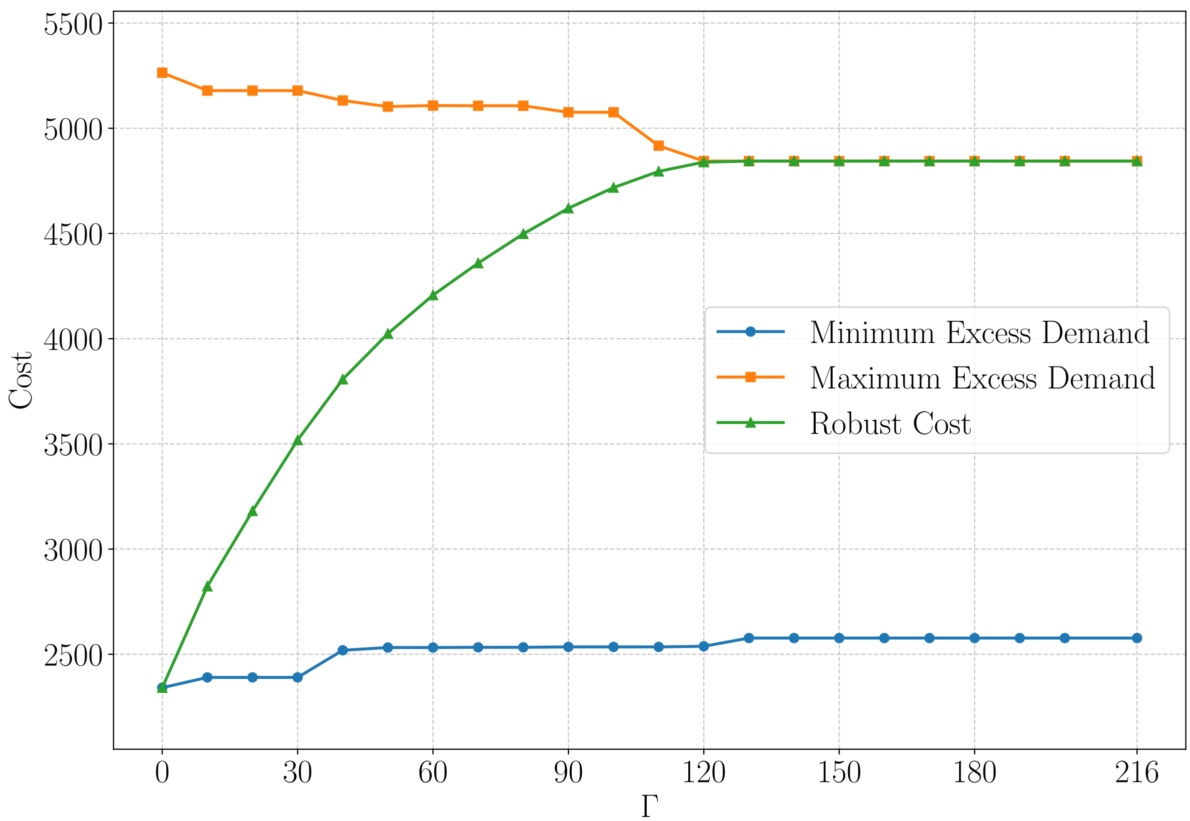

Figure 5: Protection level Γ v.s. efficiency: minimum excess demand scenario, maximum excess demand scenario, and robust cost.

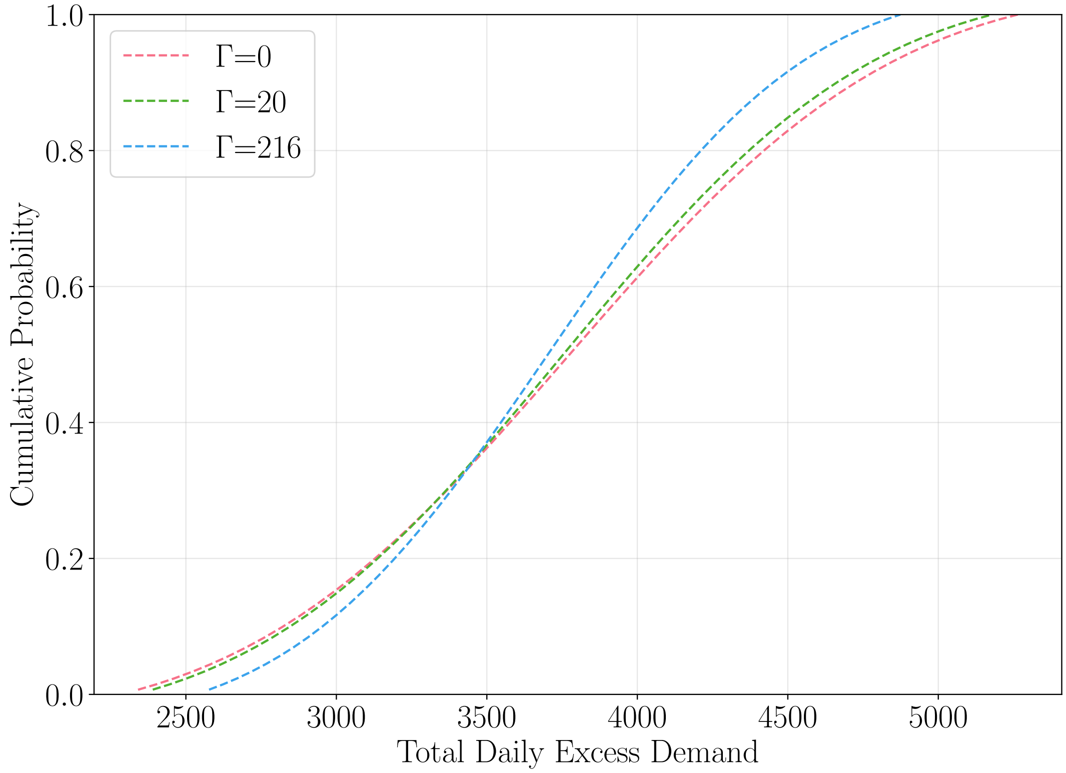

Figure 6: The cumulative distribution function based on Monte Carlo Simulation for $\Gamma = 0$, 20, and 216 ($|\mathcal{T}|$).

sponding to a cumulative probability of 0.37), the nominal solution ($\Gamma = 0$) performs slightly better than the 20-robust plan, which, in turn, is more efficient than the fully robust solution ($\Gamma = 216$). For larger values of total daily excess demand (above 3500), this dominance relationship reverses, making the fully robust solution the best choice. Furthermore, the analysis reinforces $\Gamma = 20$ as a well-balanced trade-off. Its performance closely matches the nominal scenario for lower total daily excess demand values while providing better protection against higher excess demand values, as shown by how its CDF curve falls between the nominal and fully robust solutions. Collectively, these findings establish $\Gamma = 20$ as the practical threshold for maintaining nominal efficiency, whereas pushing conservatism beyond the $\Gamma = 100$ saturation point (identified in Section VI.A) significantly degrades nominal performance without yielding proportional worst-case benefits.

6.3 Configurations utilization pattern

The scatter plot (Figure 7) presents a comprehensive analysis of configuration utilization patterns derived from our optimization results with August 2024 data. Each point represents a unique configuration, with the x-axis indicating total occurrences and the y-axis showing average duration in minutes. The occurrences of each configuration are counted based on consecutive assignments in the solution sequence - for example, in a sequence A-A-B-B-A, configuration A would register 2 occurrences (one for each continuous segment) and B would register 1 occurrence. The plot includes results across different robustness parameters (Γ), ranging from 0 (nominal solution) to 216 ($|\mathcal{T}|$), allowing us to examine how uncertainty protection affects utilization patterns. Three key configurations emerge in our analysis: CSS11G (11 sectors), CSS7Z1 (7 sectors), and CNS7T2 (7 sectors), each of which demonstrates distinct utilization characteristics that provide information on the relationship between sector count and operational deployment.

In particular, configuration CSS11G demonstrates a strategically relevant pattern in the context of protection-level requirements. At lower protection levels ($\Gamma = 0$, 20), it maintains a baseline presence with around 35 occurrences, while at higher protection levels ($\Gamma = 100$, 216), its usage increases to approximately 42 occurrences. Throughout all protection levels, it maintains consistent average durations around 100 minutes. This pattern indicates that CSS11G not only serves as a stable primary configuration but is preferentially selected when higher levels of protection against uncertainties are required. The increased utilization under higher Γ values, while maintaining consistent duration metrics, suggests that larger sector configurations can be used to provide enhanced robustness without compromising operational efficiency.

As for configuration CSS7Z1, a 7-sector configuration, it shows notably different characteristics, appearing in the low-occurrence range (0-5) but with the highest average duration (approximately 345 minutes). Similarly, configuration CNS7T2 is associated with an average duration of around 250 minutes, but is rarely used. This inverse relationship between occurrence frequency and duration for the 7-sector configurations suggests they might be strategically deployed for longer, specialized operations rather than routine assignments. The pattern remains relatively stable across different values of Γ, indicating that this sector-specific behavior is inherent to the configuration rather than an artifact of uncertainty protection.

6.4 Comparison with the actual scenarios

Based on the actual operational data, we identify three days with the highest accumulated air traffic control capacity regulation delay in August 2024, which are August 3$^{\text{rd}}$, 17$^{\text{th}}$, and 24$^{\text{th}}$. On each of these days, different sectors of the Madrid ACC experienced regulations throughout the day, indicating a more complicated operation scenario. Since the current sector implementation in Madrid ACC is different from Solution 44, we could not compare the excess demand per sector.

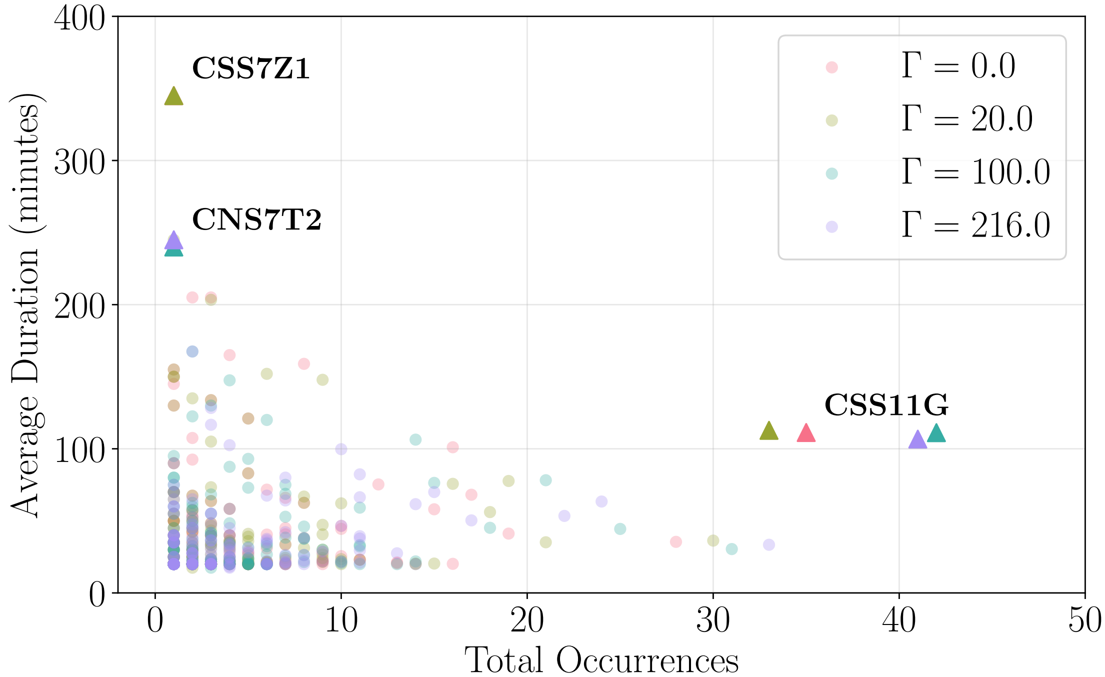

Figure 7: The utilization of configurations, x-axis denotes the total occurrence for each configuration while y-axis represents the average duration for each occurrence.

Instead, we extract from the NEST output the number of transitions (from 6:00 to 24:00) and compare them.

The comparison between the actual number of transitions and our simulation results with different values of Γ is presented in Figure 8. The actual transition counts are 22, 24 and 27 on August on August 3$^{\text{rd}}$, 17$^{\text{th}}$ and 24$^{\text{th}}$ respectively.

It is important to notice that our optimization model shows varying performance in terms of transition counts compared to actual operations. Although our model generally achieves fewer transitions, an interesting trade-off can be observed on August 3$^{\text{rd}}$, where all six Γ scenarios show higher transition counts than actual operations. This increased number of transitions likely results from our algorithm's primary objective of minimizing excess demand, suggesting that, in some cases, more frequent reconfigurations are needed to better manage airspace capacity and demand.

For the subsequent dates (August 17$^{\text{th}}$ and 24$^{\text{th}}$), our model shows a smaller number of transitions with respect to actual plans, with the only exception being $\Gamma = 100$ for August 24$^{\text{th}}$.

7 Concluding remarks

This paper proposes a robust optimization framework for dynamic airspace configuration that minimizes total excess traffic demand over extended operational periods—potentially spanning an entire day—through effective configuration plan management. To address the complexities of airspace configuration transitions and traffic demand uncertainties, our methodology integrates integer linear programming with a graph-based approach. Specifically, we designed and implemented a Permanence-Constrained Shortest Path algorithm to incorporate operational requirements for configuration transitions into robust configuration plan computation.

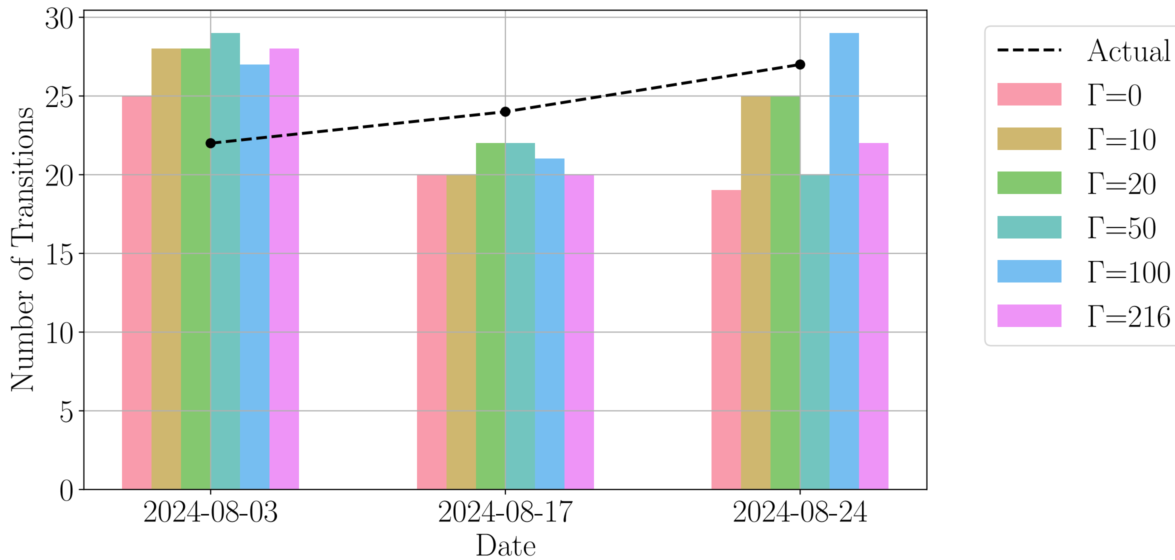

Figure 8: Comparison of the number of transitions, robust optimization results v.s. the actual operation scenarios.

We demonstrated the framework's effectiveness through a case study of Madrid ACC using real flight trajectory data and airspace sector information. The results show that configuration plans robust to demand fluctuations can be generated, effectively reducing excess demand and potential delays, particularly in highly congested scenarios. Analysis of the computational results reveals that robust cost variance increases with Γ, necessitating careful calibration of protection levels specific to each operational day.

To apply this framework practically, an Air Navigation Service Provider (ANSP) should interpret Γ not merely as a mathematical limit, but as a strategic risk-tolerance threshold reflecting the anticipated volatility of the operational day. In practice, we recommend a sensitivity-based approach for selecting Γ. On days with stable weather and highly predictable scheduled traffic, an ANSP might select a lower protection level (e.g., $\Gamma = 20$), which our analysis shows preserves nominal efficiency while offering a well-balanced buffer against moderate uncertainty. Conversely, during periods prone to severe disruptions – such as convective weather events or significant military operations – the ANSP should shift to a higher protection level (e.g., approaching the saturation point of $\Gamma = 100$). By utilizing empirical tools, such as the cumulative distribution functions of excess demand demonstrated in this study, ANSPs can dynamically set Γ by weighing the marginal cost of airspace reconfiguration against the probability of critical traffic overloads.

While robust solutions may not be optimal under optimistic scenarios with low demand excess, they provide significant advantages during congestion. From a practical standpoint, accepting moderate excess demand in low-congestion periods is preferable if it substantially reduces congestion during critical periods, ultimately benefiting both ATCO workload and ATFM delays.

The results presented consider an excess demand definition that does not account for spatial concentration effects, which, as noted in Section 5.2.2, impact required air traffic regulations. Further analysis of robust configuration plans regarding spatial and temporal peaks in demand overload remains for future study. Future research may also investigate adjustments to the excess demand parameters or the mathematical (ILP or graph) formulation to directly incorporate concentration effects. However, the proposed approach remains valid when alternative metrics—such as complexity metrics—are employed, provided initial proofs for this framework's generality.

Additional directions for future work include refining robustness parameters and uncertainty sets to enhance model adaptability and performance. While the current framework utilizes a budgeted uncertainty set controlled by Γ, it assumes bounded independent deviations. In practice, traffic demand uncertainty often exhibits temporal and spatial correlations. A limitation of the current model is that it may under-represent these structured patterns, leading to more conservative solutions. Future research could extend this approach toward correlated uncertainty models to better capture the systemic spatiotemporal nature of airspace disruptions. Extending the framework to incorporate machine learning techniques for real-time demand forecasting could further evaluate the model's practical capabilities. Though we validate the viability of practical complexity metric from COMETA (described in Section 5.2.2), additional practical complexity metrics – such as traffic pattern complexity measures (percentage of flight tracks near boundaries, distance between flow intersections and boundaries, and number of short dwell time flights) and reconfiguration complexity metrics (volume transfer and aircraft transfer complexity) – would provide a more comprehensive assessment of controller workload and operational feasibility. These metrics, validated in previous DAC studies, could further align theoretical optimization models with real-world implementation. Overall, this study contributes to ongoing efforts to optimize airspace configuration plans and improve the efficiency and robustness of air traffic operations.

Nomenclature

$\mathcal{C}$	=	set of all airspace configurations
$\mathcal{C}_c^{t+1}$	=	set of configurations that can be implemented at time $t+1$ if configuration c is active at time t
$\mathcal{T}$	=	set of time periods, $\mathcal{T} = \{1, 2, \ldots, n\}$
$\mathcal{S}(c)$	=	set of sectors active in configuration c
N	=	set of all configuration-time pair indices
x_t^c	=	binary decision variable: 1 if configuration c is implemented at time t, 0 otherwise
$d_s(t)$	=	traffic demand for sector s at time period t
$c_s(t)$	=	capacity of sector s at time period t
$e_c(t)$	=	excess traffic demand for configuration c at time period t
t_p	=	permanence interval: minimum number of consecutive time periods
Γ	=	robustness parameter: maximum number of excess coefficients allowed to deviate
$\mathcal{D}$	=	set of objective function deviations in robust optimization
$G = (V, A)$	=	directed acyclic graph representing configuration transitions
$w(a)$	=	weight (excess demand) associated with arc a
τ	=	remaining number of time periods before configuration change is allowed
$s = (c, t, \tau)$	=	state in augmented graph: configuration c at time t with remaining duration τ
$CPLX$	=	sector complexity
Q	=	FIFO queue in shortest path algorithm
$\Delta(s)$	=	distance map: minimum path cost to reach state s
$\Pi(s)$	=	path map: sequence of nodes to reach state s
$\Pi^*(\Gamma)$	=	optimal Γ-robust configuration plan
ACC	=	Area Control Center
ATCO	=	Air Traffic Controller
ATM	=	Air Traffic Management
ATFM	=	Air Traffic Flow Management

DAC	=	Dynamic Airspace Configuration
HEC	=	Hourly Entry Count
OCC	=	Occupancy Count

Funding sources

This paper is an output 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

Bedouet, J., Dubot, T., Basora, L.. Towards an operational sectorisation based on deterministic and stochastic partitioning algorithms. In: SESAR Innovation Days. 2016. URL: `https://www.sesarju.eu/sites/default/files/documents/sid/2016/SIDs_2016_paper_28.pdf`.

Bertsimas, D., Sim, M.. Robust discrete optimization and network flows. Mathematical programming 2003;98(1):49–71. doi:doi:10.1007/s10107-003-0396-4.

Bloem, M., Gupta, P.. Configuring airspace sectors with approximate dynamic programming. In: 27$^{\text{th}}$ International Congress of the Aeronautical Sciences (ICAS). 2010. URL: `https://ntrs.nasa.gov/citations/20100036788`.

Bloem, M., Kopardekar, P.. Combining airspace sectors for the efficient use of air traffic control resources. In: AIAA Guidance, Navigation and Control Conference and Exhibit. 2008. doi:doi:10.2514/6.2008-7222.

Brinton, C., Cook, L.. Analysis of current airspace operations and implications for dynamic airspace configuration. In: AIAA Guidance, Navigation and Control Conference and Exhibit. 2008. p. 7224. doi:doi:10.2514/6.2008-7224.

Cañas, J.J., Ferreira, P., de Frutos, P.L., Puntero, E., López, E., Gómez-Comendador, F., De Crescenzio, F., Lucchi, F., Netjasov, F., Mirkovic, B.. Mental workload in the explanation of automation effects on atc performance. In: International Symposium on Human Mental Workload: Models and Applications. Springer; 2018. p. 202–221. doi:doi:10.1007/978-3-030-14273-5_12.

Dijkstra, E.W.. A note on two problems in connexion with graphs. Numerische Math 1959;:269–271doi:doi:10.1007/BF01386390.

EUROCONTROL, . Hourly Entry Count versus Occupancy Count Relationship – Robustness, Calibration, Application (II). Technical Report EEC Note No. 16/07; Eurocontrol Experimental Centre; Brétigny-sur-Orge, France; 2007. URL: `https://www.eurocontrol.int/publication/hourly-entry-count-versus-occupancy-count-relationship-robustness-calibration`.

Galeazzo, M., De Giovanni, L., Lema-Esposto, M.F., Lulli, G.. An integer programming approach to dynamic airspace configuration. In: ICRAT. 2024. URL: https://research.lancaster-university.uk/en/publications/an-integer-programming-approach-to-dynamic-airspace-configuration/.

Gerdes, I., Temme, A., Schultz, M.. Dynamic airspace sectorisation for flight-centric operations. Transportation Research Part C: Emerging Technologies 2018;95:460–480. doi:doi:10.1016/j.trc.2018.07.032.

Gianazza, D.. Forecasting workload and airspace configuration with neural networks and tree search methods. Artificial Intelligence 2010;174(7):530–549. doi:doi:10.1016/j.artint.2010.03.001.

Horizon Europe, . Smart sectors: Optimising airspace design to meet air traffic demand efficiently. 2023. doi:doi:10.3030/101114686.

Ibáñez-Gijón, J., Travieso, D., Navia, J.A., Montes, A., Jacobs, D.M., Frutos, P.L.. Experimental validation of cometa model of mental workload in air traffic control. Journal of Air Transport Management 2023;108:102378. doi:doi:10.1016/j.jairtraman.2023.102378.

Jung, J., Lee, P., Kessell, A., Homola, J., Zelinski, S.. Effect of dynamic sector boundary changes on air traffic controllers. In: AIAA Guidance, Navigation, and Control Conference. 2010. p. 8289. doi:doi:10.2514/6.2010-8289.

Kopardekar, P.H., Schwartz, A., Magyarits, S., Rhodes, J.. Airspace complexity measurement: An air traffic control simulation analysis. International Journal of Industrial Engineering: Theory, Applications and Practice 2009;16(1):61–70. doi:doi:10.23055/ijietap.2009.16.1.18.

Lee, P.U., Prevot, T., Homola, J., Lee, H., Kessell, A., Brasil, C., Smith, N.. Impact of airspace reconfiguration on controller workload and task performance. In: 3rd International Conference on Applied Human Factors and Ergonomics. 2010. URL: https://human-factors.arc.nasa.gov/publications/Lee_etal_%20AHFE2010.pdf.

Lema-Esposto, M.F., Amaro-Carmona, M.Á., Valle-Fernández, N., Iglesias-Martínez, E., Fabio-Bracero, A.. Optimal dynamic airspace configuration (DAC) based on state-task networks (STN). In: SESAR Innovation Days. 2021. URL: https://sesar.eu/sites/default/files/documents/sid/2021/papers/SIDs_2021_paper_48.pdf.

Lui, G.N., Lulli, G., Lema-Esposto, M.F., Martinez, R.L.. Airspace sector design: an optimization approach. In: SESAR Innovation Days. 2024. URL: https://www.sesarju.eu/sites/default/files/documents/sid/2024/papers/SIDs_2024_paper_069%20final.pdf.

Ramírez, S.. FastAPI. 2018. URL: https://github.com/fastapi/fastapi.

Romero, A.B., de Urrengoecha, T., Delahaye, D., Guitart, A., Lema-Esposto, F., Lui, G.N., Lulli, G., Sanaei, R., Solzer, J.. Smarts solution for airspace design and configuration. In: SESAR Innovation Days. 2025. URL: https://www.sesarju.eu/sites/default/files/documents/sid/2025/papers/SIDs_2025_paper_121-final.pdf.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., Wilhelm, M.. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks. IEEE; 2014. p. 83–94. doi:doi:10.1109/IPSN.2014.6846743.

Sergeeva, M., Delahaye, D., Mancel, C., Vidosavljevic, A.. Dynamic airspace configuration by genetic algorithm. Journal of Traffic and Transportation Engineering 2017;4(3):300–314. doi:doi:10.1016/j.jtte.2017.05.002.

Sergeeva, M., Delahaye, D., Zerrouki, L., Schede, N.. Dynamic airspace configurations generated by evolutionary algorithms. In: 2015 IEEE/AIAA 34$^{\text{th}}$ Digital Avionics Systems Conference (DASC). 2015. p. 1F2–1–1F2–15. doi:doi:10.1109/DASC.2015.7311520.

SESAR Joint Undertaking, . A proposal for the future architecture of the European airspace. Publications Office of the European Union 2019;doi:doi:10.2829/309090.

Solak, S., Solveling, G., Clarke, J.P.B., Johnson, E.L.. Stochastic runway scheduling. Transportation Science 2018;52(4):917–940. doi:doi:10.1287/trsc.2017.0784.

Suarez, N., López, P., Puntero, E., Rodriguez, S.. Quantifying air traffic controller mental workload. In: Fourth SESAR Innovation Days. volume 220; 2014. URL: `https://www.sesarju.eu/sites/default/files/SID_2014-03.pdf`.

Treimuth, T., Delahaye, D., Ulrich Ngueveu, S.. A branch-and-price algorithm for dynamic sector configuration. In: 8th International Conference on Applied Operational Research. 2016. p. 47–53. URL: `https://laas.hal.science/hal-01481466v1/document`.

Wang, K., Jacquillat, A.. A stochastic integer programming approach to air traffic scheduling and operations. Operations Research 2020;68(5):1375–1402. doi:doi:10.1287/opre.2020.1985.

Wong, C.S.Y., Suresh, S., Sundararajan, N.. A rolling horizon optimization approach for dynamic airspace sectorization. IFAC Journal of Systems and Control 2020;11. doi:doi:10.1016/j.ifacsc.2020.100076.

Yan, J., Hu, H., Wang, Y., Ma, X., Hu, M., Delahaye, D., Alam, S.. Robust predeparture scheduling for a nation-wide air traffic flow management. Chinese Journal of Aeronautics 2025;38(4):103223. doi:doi:https://doi.org/10.1016/j.cja.2024.08.054.

Zelinski, S., Lai, C.F.. Comparing methods for dynamic airspace configuration. In: 2011 IEEE/AIAA 30$^{\text{th}}$ Digital Avionics Systems Conference. 2011. p. 3A1–1–3A1–13. doi:doi:10.1109/DASC.2011.6096043.