Optimal Dynamic Airspace Configuration (DAC) based on State-Task Networks (STN)

M. Florencia Lema-Esposto, Manuel Ángel Amaro-Carmona, Natividad Valle-Fernández,
Enrique Iglesias-Martínez and Adrián Fabio-Bracero
Centro de Referencia I+D+i en ATM (CRIDA)
Madrid, Spain
mflema@e-crida.enaire.es, maamaro@e-crida.enaire.es, nvallef@e-crida.enaire.es,
eiglesiasm@e-crida.enaire.es, afabio@e-crida.enaire.es

***Abstract*— The use of a more efficient allocation for meeting user needs is an essential element of the on-going transformation of the Air Traffic Management System (ATMS). Dynamic Airspace Configuration (DAC) presents several advantages with respect to traditional airspace management. DAC promises more flexible sector configurations capable to adapt to air traffic demand, complexity, and weather conditions. This is achieved by replacing static sector boundaries with a large number of airspace building blocks that could be merged depending upon the traffic conditions and resulting in a more dynamic airspace allocation. The use of flexible boundaries as a capacity management technique leads to more efficient flights and requires less site-specific training. In contrast, a flexible airspace allocation, as proposed by the DAC concept, allows more variables to be considered while continually adjusting the capacity to accommodate air traffic demand.**

The interchangeability of DAC building blocks (each airspace volume is merged with other(s)) is a large-scale optimization challenge. Hence, to obtain dynamic capacity, it is required to determine more efficient airspace allocation considering the distribution of air traffic demand and complexity among all building blocks.

This paper presents a novel approach, in which the problem of merging and interchangeability of Dynamic Airspace Configuration is modelled using a single-layer State-Task Network (STN). The approach led to developing an optimization framework capable of efficiently allocating dynamic airspace volumes depending on the traffic demand and complexity. Subsequently, a use case containing 60 airspace building blocks using the DAC concept is defined over the Madrid ACC and solved using the developed framework and a Mixed Integer Programming (MIP) optimizer.

Keywords – Air Traffic Management System ATMS; Dynamic Airspace Configuration DAC; Demand Capacity Balancing DCB; Optimization; State-Task Network STN

I. INTRODUCTION

Following the COVID-19 pandemic, levels of air traffic dropped significantly. However, during the last months, traffic demand has growth and this increase is expected to accelerate to pre-pandemic levels (Eurocontrol, 2021). Consequently, pressure on air traffic capacity will also rise with an associated risk of fast increases in delays. To support the foreseen growth and reduce traffic congestion, it is important to develop mechanisms to improve the management of limited airspace capacity. In this context, Dynamic Airspace Configuration (DAC) promises a more flexible sectorization capable of adapting to the expected air traffic demand. A more dynamic airspace allocation results in more efficient demand distribution and a better use of limited resources. This is achieved by redesigning the airspace based on a large number of building blocks, resulting in a more flexible airspace. However, a challenge associated with DAC, is the process of generating more efficient sector configurations among a wider number of combinations.

A problem of this process is to adjust these combinations to efficiently manage the expected traffic demand whilst meeting constraints. Obtaining this flexibility by combining airspace building blocks could be considered a large-scale optimization problem, in which each airspace volume could be merged with another to adapt to the continuous fluctuating traffic demand and complexity.

This paper presents a sector configuration framework that aims to address this problem. Airspace volume and traffic demand are modelled using State-Task Networks (STN), a multi-purpose batch processing technique introduced by Kondili (Kondili *et al.*, 1993) formulation of a Mixed Integer Programming (MIP) problem and it is solved using an off-the-shelf optimizer. We defined a Free Route airspace over Madrid ACC composed of 60 airspace building blocks to test and validate the developed framework. The process obtains a ranking of optimal airspace configurations according to the available airspace resources (e.g., number of available air traffic controllers) and relevant traffic conditions.

This paper is structured into six sections. The first section reviews the current state-of-the-art. Next, we explain the approach of adapting STN to model the DAC problem that leads to the development of our framework. Section IV, presents the application of DAC concept to Madrid ACC Route 1, in which the problem of merging building blocks is modelled and solved with the developed framework. Results are detailed and analyzed in the Section V. Finally, Section VI summarizes the work and evaluates possible future improvements.

II. STATE OF THE ART

Optimization in airspace management has been widely discussed by several authors, such as Bertsimas, Lulli and Odoni (2011) or Xu, Prats and Delahaye (2018). In this context, Dynamic Airspace Configuration (DAC), as a new paradigm, gives rise to further optimization of airspace management. This section presents a literature review on DAC and optimization in airspace allocation.

A. *Dynamic Airspace Configuration*

Dynamic Airspace Configuration aims to create a more flexible airspace design and configuration process, increasing the capability of the airspace to adapt to the fluctuating traffic demand. The concept switches from the current prevailing technique of fixed airspace structures to the use of non-defined operating sectors made of a larger amount of building blocks and/or flexible boundaries.

DAC is included within the Demand and Capacity Balance (DCB) process throughout its time horizon. In the strategic planning (months before the day of operation), when designing more flexible airspace structures; in the pre-tactical phase (from hours up to minutes before the day of operation), when the configuration plan is created and changed dynamically (e.g. changing the opened configuration every 20 min, if necessary) to optimize the use of the available capacity and balance the ATCOs workload; and in the execution phase, when the final configuration is implemented by the Air Traffic Control Center.

Since DAC is a relatively new and not fully implemented concept, several SESAR projects have tackled this topic, such as COTTON (Zhang Zheng, Puntero Parla and Cidoncha Sánchez, 2021) (SESAR JU, 2019a), W1-PJ08 (SESAR JU, 2019b), W1-PJ09 (SESAR JU, 2019c) and W3-PJ32 (SESAR JU, 2021). Some of those, have also approached to the concept integrating it in a Free Route Airspace (FRA) environment. This is due to the fact that DAC implementation in the ECAC area is expected to start on 2025 (EUROCONTROL, 2019), when Free Route operations will also be deployed.

B. *Optimization for airspace allocation*

Optimization methods have been widely used to solve a variety of problems. Over the past decade, optimization techniques have been applied to airspace allocation by authors that approached the DAC issue and airspace modelling through different ways.

Researchers first confronted the airspace sectorization by the combination of small volumes. In this way, Jagare *et al.* (2013) proposed a free-form static airspace sectorization starting from a regular mesh of cells. From this point on, they designed a constraint-based local search (CBLS) to actively use constraints in the sectorization process. Chen & Zhang (2014) described the airspace configuration as a weighted graph partitioning problem, where vertices represented key points such as airports and waypoints, and the edges represented air routes. They solved the problem with a combination of a graph partitioning, an optimal dynamic load balancing and heuristic algorithms. This analysis of an airspace as a weighted graph is widely used in DAC research, Sergeeva *et al*. also implemented it in their work. In 2015, they combined the graph technique with the use of functional airspace blocks (Shareable Airspace Modules SAMs and Sectors Building Blocks SBBs) to represent the airspace. Then a stochastic optimization algorithm is applied to generate a sequence of sector configurations for one day of operation minimizing Air Traffic Controller (ATCO) workload. A few years later, Sergeeva *et al.* (2017) experimented with starting from 3D airspace blocks and creating different configurations also by graph partitioning, but finally solved the optimization problem with a genetic algorithm.

Other authors have presented a multi-objective approach to solve the sectorization problem. Wong *et al.* (2017) created airspace sectors with a Voronoi diagram and proposed a multi-objective formulation to obtain a range of solutions with a varying trade-off among the objectives. Also, the number of available ATCOs has been an important issue for sectorization. Treimuth *et al.* (2016) applied a branch-and-price method to a unique weighted objective function to optimally reducing the number of controllers. A recent trend in the ATM field is to move towards flight-centric operations. In this regard, Gerdes *et al.* (2018) developed the idea of AutoSec (automatic sectorization), which is a combination of fuzzy logic for clustering traffic flows, Voronoi diagrams for creating new sectors and evolutionary algorithms to find the optimal sectorization. This technique combined multi-criteria optimization and flight-centric operations. One of the most recent works in DAC, also presented by Wong *et al.* (2020), takes up the idea of changing the shape of the sectors but, this time, based on future traffic demand. However, this innovative approach must necessarily be developed in parallel with the training of controllers, which is closely dependent on the shape of the sectors.

From the different optimization techniques that have been reviewed, this paper proposes the use of State-Task Networks as an innovative perspective to address Dynamic Airspace Configuration for modelling the relationship between different airspace structures (Building Blocks and Configured Sectors) with the predicted traffic demand. The process starts with a predefined set of building blocks that are combined forming controllable sectors to create the airspace configuration and uses a Coin-or Branch and Cut optimizer (CBC) (Forrest, 2000) to solve the optimization problem.

III. METHODOLOGY

A. *State-Task Networks*

State-Task Networks (STNs) are a modelling technique that is commonly used for representing batch process. STNs were introduced by Kondili *et al*. (1993) as a general algorithm for scheduling multipurpose batch operations in a chemical plant. STN represent a batch model based on “recipe networks”, a generalized version of a flowsheet representation of continuous plants which has been modified to include two different kinds of nodes. “State” nodes represent inputs and products, and “Task” nodes represent the actions (or processing operations) to transform the material from one or more input states. Figure 1 represents a general State-Task Network representation of a chemical process.

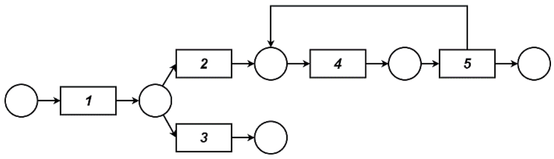

Figure 1: State-task network representation. Source (Kondili, Pantelides and Sargent, 1993)

The main advantage of STNs is that are free of ambiguities associated to recipe networks because the products and tasks performed to transform these products are clearly represented by single interconnected nodes (Kondili, Pantelides and Sargent, 1993). STNs have been widely used for different applications such as manufacturing (Lee *et al*., 2016; Lin & Floudas, 2001; Bose & Bhattacharya, 2009; Vanzetti *et al*., 2021) and equipment maintenance scheduling (Hazaras *et al*., 2012). As a general modelling technique, STNs are mostly associated to optimization process, therefore most formulations result in Mixed Integer Programs (MIP) problems involving many binary variables representing the scheduling of equipment/tasks to transform relevant states.

STN represents an ideal methodology to model the relationships among elements associated to the DAC interchangeability problem. A single feed layer of "*states*" (representing air traffic demand) interconnected to a layer of "*tasks*" (representing configured sectors) leading to a unique final state. The problem is formulated to obtain the optimal *"unit"* (representing building blocks) allocation to each configured sector while passing the traffic demand to a final state. In contrast to multipurpose batch process, in which several sequential tasks are considered, we created a particular model for representing the problem. The following section explains this approach in more detail.

B. Model

The model developed for this study seeks to represent the air traffic demand and the sectorization to be processed by the algorithm. For this purpose, each of the three (3) elements presented in STN methodology is linked to a particular concept. Hence, we have the following associations:

- "States", that represent the air traffic demand.
- "Tasks", corresponding to the configured sectors.
- "Units", that are the building blocks that compose the configured sectors.

The mapping of each element of the problem (building blocks, configure sectors and traffic demand) to STN elements (units, tasks and states, respectively) allows for structuring data in an orderly manner and establishing unambiguously which building blocks compose which configured sectors (Figure 2). The process to apply this technique to a sectorization problem consists of starting from an initial state (initial traffic demand D_n) and, by the use of all the units (building blocks), implementing an optimization algorithm to find the most appropriate tasks (configured sectors which have been optimized in terms of cost) to reach the final state (controlled traffic demand D_c) (Figure 3).

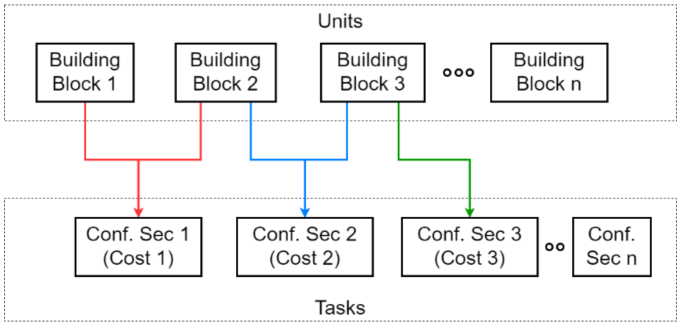

Figure 2: Building Blocks (Units) - Configured Sectors (Tasks) modelling

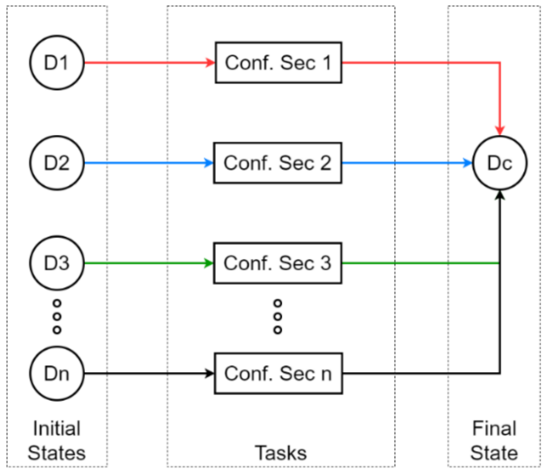

Figure 3: Demand (States) - Configured Sectors (Tasks) modelling

The next step consists of creating an optimization problem composed by a model, a cost function, and a set of constraints. We used the Pyomo (Bynum *et al.*, 2021) framework for this purpose. Finally, it is solved using a MIP optimizer, in this case we used the Coin-or Branch and Cut Optimizer (CBC).

The model includes all the variables to be considered and the constraints that apply to the problem solution. The variables (air traffic demand, building blocks, configured sectors) are obtained from the structure defined in the previous step, including all the relations among them.

C. Constraints

The following constrains are considered to solve the optimization problem for a previously fixed time slot of interest.

- **Building blocks simultaneity (eq. 3.1, eq. 3.2 and eq. 3.3).** Ensures that the same building block is not allocated to two different configured sectors at the same time, which would result in overlaps**.**
- **Building blocks allocation (eq. 3.4).** Guarantees that all building blocks are assigned to at least one configured sector, ensuring the absence of gaps in the airspace.
- **Maximum number of configured sectors (eq. 3.5).** Allows considering whether there is a limited number of ATCO available or not, bringing the problem closer to a real situation, in which these types of limitations are frequent.
- **Capacity threshold limits (eq. 3.6).** It only allows the use of configured sectors with entry counts (number of flights entering the sector during a time interval) lower than the

capacity threshold. We consider this constraint as "*optional*" if there is no possible solution with all configured sectors complying it. This possibility represents a more realistic operative environment, where configured sectors may exceed their capacity at some point under determined conditions agreed upon by the ATCO.

$$W_{(i,j)} - W_{(i,k)} = 0 \quad \forall\, m_i = 1, i \in O,\ j \neq k, (j,k) \in E_i \tag{3.1}$$

$$\sum_i^{|O|} W_{(i,j)} \begin{cases} = 1 & if\ m_i = 1 \\ \leq 1 & if\ m_i \neq 1 \end{cases} \quad \forall\, j \in B \tag{3.2, 3.3}$$

$$\sum_i^{|O|} \sum_j^{|B|} (W_{(i,j)}) = |B| \tag{3.4}$$

$$\sum_i^{|O|} \sum_j^{|B|} (W_{(i,j)}/|E_i|) \leq \mathrm{ATCO}_{max} \tag{3.5}$$

$$(W_{(i,j)} C_{i_{max}}) - (W_{ij} HEC_i) \geq 0 \quad \forall\, i \in O,\ j \in B \tag{3.6}$$

where,

$W(i,j)$: its value is 1 if building block j is assigned to configured sector i. Otherwise, it equals 0.

B: set of building blocks.

O: set of configured sectors.

E_i: set of building blocks forming configured sector i.

m_i: its value is 1 if a configured sector i contains more than one building block. Otherwise, it equals 0.

$ATCO_{max}$: max. number of Controller Working Positions.

$C_{i_{max}}$: max. capacity (in terms of entry counts) for configured sector i.

HEC_i: entry counts for configured sector i.

D. *Cost Calculation*

A sector configuration is obtained by assigning each building block to a configured sector. The objective is to obtain an optimal configuration formed by the configured sectors with minimal cost. In this respect, occupancy counts are the variables considered to compute the cost of each configured sector, since they are directly related to ATCO workload. Occupancy counts correspond to *the number of flights that are inside a defined location at a precise time and correspond to the flights that are (or will be) worked by ATC at that time* (EUROCONTROL, 2021). In this case, we consider the occupancy counts inside a configured sector in time periods of 5 minutes duration and an interval of 1 minute between each of them.

$$C = W_{OVLD} A_{OVLD} + W_{UNLD} A_{UNLD} \tag{1}$$

Cost (C), to minimize, is defined as the weighted sum of two terms. The first term *Overload Area* (A_{OVLD}) represents, for a given configured sector and time interval, the total occupancy counts above an upper sustained threshold, reflecting an excess of ATCO workload. The second term *Underload Area* (A_{UNLD}) represents the total difference between low occupancy counts and a lower sustained threshold, which corresponds to ATCO work underload. Each term is weighted by W_{OVLD} and W_{UNLD}, respectively. Finally, the total cost of a configuration is the sum of the costs of each of its configured sectors. Figure 4 shows a graphic representation of overload and underload areas concept.

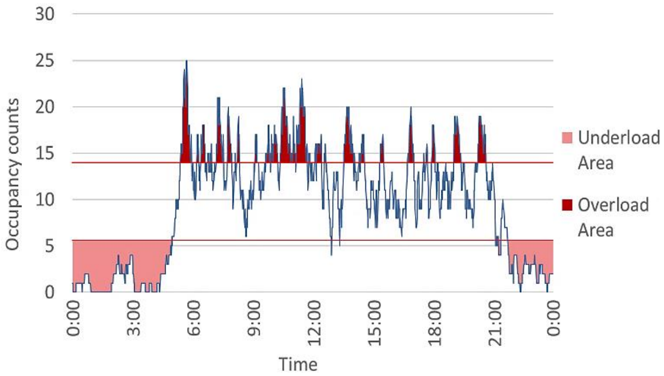

Figure 4: Overload and Underload areas representation

IV. DAC INTERCHANGEABILITY PROBEM IN MADRID ACC

A. *Airspace Building Blocks*

For sector design and configuration DAC considers different airspace elements with which to shape a flexible operating sector (SESAR, 2015). Despite being different, all those elements share the same underlying principle: the creation of multiple airspace volumes, smaller and better adapted to the traffic flows, to use as the building blocks to configure the workable sectors.

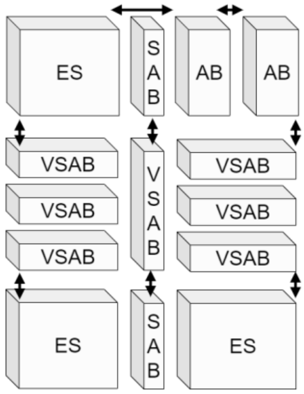

Figure 5: Example of DAC Airspace design elements

Some of these airspace design elements, in particular those that have been used on this specific study, are shown in Figure 5 and described below:

- **Elementary Sector** (ES), an ATC workable 3D airspace that cannot be split into a controllable sector.
- **Airspace Block** (AB), a primary volume that needs to be merged with other AB to conform a workable sector.
- **Shareable Airspace Block** (SAB), a non-controllable sector that needs to be attached to any adjacent ES or AB to build an operating sector.
- **Vertical Shareable Airspace Block** (VSAB), this non workable sector is the same concept as a SAB, but in the vertical plane. It splits the space vertically and typically cover 1.000 to 4.000 fts.

B. *DAC in Madrid ACC*

Having significant levels of traffic demand, the Madrid ACC represents a highly complex environment (SESAR, 2017). Nonetheless, in this airspace only the concept of Elementary Sectors is currently being used, with the minor exception of a

VSAB defined in the Zamora sector (ZMM) (Figure 6). Hence, the design is not very flexible and there is room for improving traffic allocation and its overall capacity using the DAC concept.

In order to redesign the Madrid ACC following DAC principles, the current sectors were split horizontally and vertically into *smaller building blocks* (Shareable Airspace Blocks SABs and Vertical Shareable Airspace Blocks VSABs) and into two Elementary Sectors (Upper and Lower), where size allowed it. The horizontal boundaries of these sectors were also slightly modified to follow the main traffic flows of the new traffic samples used. These samples were generated from AIRAC 1908 historical traffic samples from FL245 by adapting them into FRA samples using the EUROCONTROL tool NEST (Eurocontrol, 2020) to increase the consistency with the DAC concept.

Also using the NEST tool, the previous sectors modifications were implemented. The design was further validated with operational staff from ENAIRE, whose background in ATC of Madrid ACC guarantees the feasibility of the design.

The new design is shown in Figure 6 and Figure 7. In 2D the most important changes were especially made in the center of the ACC. Meanwhile, the 3D changes include the creation of three VSAB to allow four different cut levels and the possibility of intermediate sectors (Figure 7). It contrasts with today's vertical cuts, which are fixed in FL345 or FL325, depending on the area, and except for the "Zamora slice". Since Madrid ACC comprises a large area, the studied use case will contain only a portion of the Madrid ACC sectors, which are outlined in red in Figure 6, and are the most interesting sectors due to their high traffic density and higher complexity peaks. For defining the configuration plan, this airspace is considered as an independent Air Traffic Service Unit (ATSU).

C. *STN-DAC Optimization Framework*

The developed framework can propose a set of ranked optimal configurations for a given airspace in a specific time interval. These configurations are obtained by merging building blocks (ES, SAB and VSAB) into configured sectors and logically determining all possible combinations with them. Then, the framework derives the cost of each configured sector to obtain the configuration with the lowest possible cost while meeting the defined constraints.

We provide the optimization framework with inputs structured in several sets. First, a list of building blocks is provided. These need to be univocal and will form the configured sectors. Then, a set of configured sectors is defined along with the characteristics of each of them. These characteristics include the list of building blocks involved, a capacity threshold, a number of entry counts for the given time interval and a cost associated to that configured sector, which has been previously computed as described in Section III.D. The last set of inputs includes a maximum number of configured sectors and a quantity for the ranking of solutions. The framework gives the possibility to obtain a ranking of solutions ordered according to their cost. This makes possible to evaluate the best solutions and decide among them based on aspects other than cost.

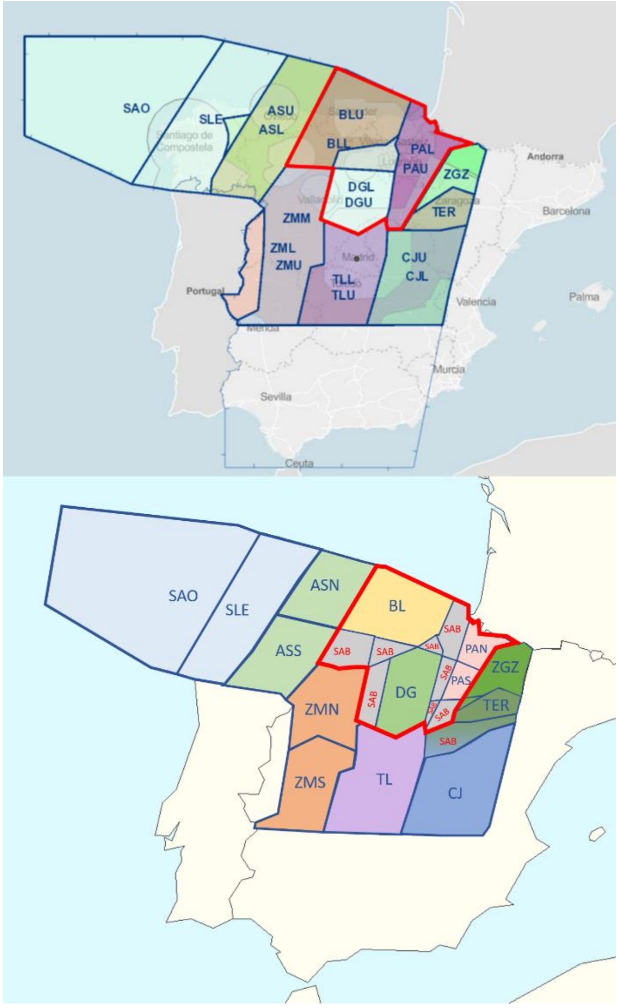

Figure 6: 2D map of current Madrid ACC vs Madrid ACC modified using DAC concept

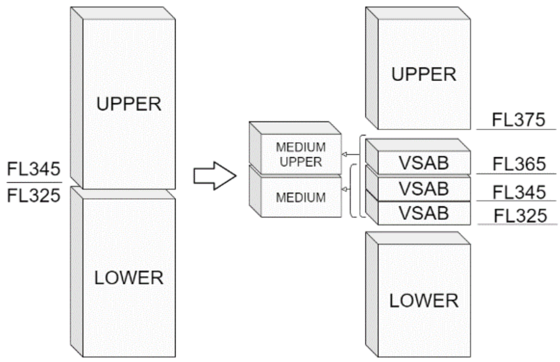

Figure 7: Vertical profile of the current Madrid ACC vs Madrid ACC modified using DAC concept

D. *Use Case*

The presented use case assesses the benefits of the DAC approach for airspace management of some sectors of Madrid ACC Route 1 (Figure 6). This is achieved comparing the configuration plan obtained by our framework with the configuration plan obtained for a reference scenario. This scenario has a similar sectorization as the used nowadays, but slightly adapted to the traffic flows generated by Free Route traffic. This adaptation was carried out to guarantee a fair comparison between scenarios and mainly consists in modifying the sectors boundaries to avoid short-crossings and re-entries that arise with the new FRA traffic samples. The reference configuration plan is obtained with the EUROCONTROL tool NEST and its optimization algorithm ICO at the day of study. The selected day was the 21st of July 2019 since it is a busy day in terms of traffic demand and complexity. The solution scenario corresponds to the same day of study and traffic demand, with the sectorization being the only difference. The solution scenario represents the same airspace design but with the DAC concept integrated (Figure 7). There are 60 airspace building blocks (4 upper ES and 4 lower ES, 8 upper SABs and 8 lower SABs, 36 VSABs), whose logical and spatially ordered associations result in a set of 290 configured sectors. The framework we have developed evaluates the complexity of each configured sector, obtaining a configuration plan as a final solution. Results of trails utilizing this concept are presented and compared with the reference scenario in the next section.

V. Results

We executed the framework for each time interval of one hour (a total of 24 runs). The overall execution time was ≈ 65s using a 3.2Ghz CPU and 16GB of RAM. Tests correspond to the configuration plan for the entire day of July 21st, 2019. However, the results presented in this section focus on the optimal configuration obtained for a time interval characterized by complex traffic demand in terms of occupancy counts, which is from 6am to 9am (UTC) (Figure 8). This period concentrates the highest peak of traffic followed by a significant decrease.

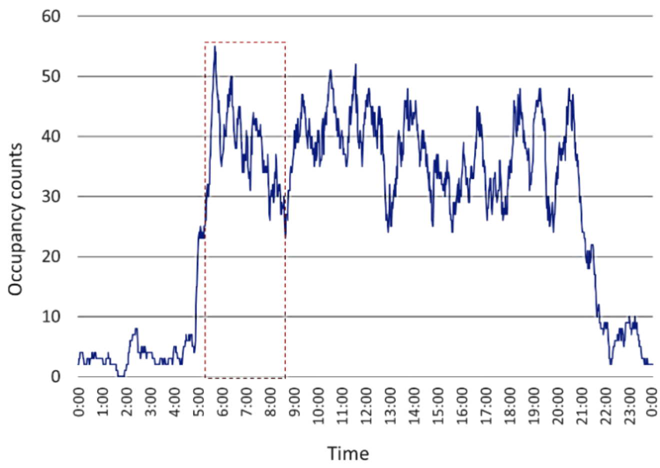

Figure 8: Occupancy counts in Madrid ACC Route 1 for the 21st July 2019

The configuration plan for the reference scenario (reference configuration) is represented in Figure 9. It has two different configurations, one formed by six (6) configured sectors from 6am to 7am and a second one of five (5) configured sectors from 7am to 9am.

The configuration plan for the reference scenario (reference configuration) is represented in Figure 9. It has two different configurations, one formed by six (6) configured sectors from 6am to 7am and a second one of five (5) configured sectors from 7am to 9am.

6am-7am	7am-8am	8am-9am
LECMBLL	LECMBLU	LECMBLU
LECMBLU	LECMDGU	LECMDGU
LECMDGL	LECMPAL	LECMPAL
LECMDGU	LECMPAU	LECMPAU
LECMPAL	LECMBDL	LECMBDL
LECMPAU		
6A	5A	5A

Figure 9: Reference configuration plan for Madrid ACC Route 1 (21st July 2019)

The configuration plan obtained with our framework for the same time interval is represented in Figure 10. In this case, there are three different configurations. In terms of number of configured sectors, the first and second hour have six (6) and five (5) configured sectors respectively, as in the reference configuration.

6am-7am	7am-8am	8am-9am
LECMBLSLMU	LECMBDPLL	LECMBLSI
LECMBLSUU	LECMBLSUML	LECMDGSLMU
LECMDGSPLM	LECMDGSPMMU	LECMDPUU
LECMDGSU	LECMDGSPUU	LECMPALMU
LECMPAU	LECMPAWUML	
LECMPAWLM		
6A	5A	4A

Figure 10: Solution configuration plan for Madrid ACC Route 1 (21st July 2019)

However, the main difference with the reference configuration is that the number of configured sectors decreases along with the traffic demand to the point of having a configuration consisting of four (4) configured sectors from 8am to 9am. For other time intervals of the day of study, results also adhered to the traffic demand.

To compare reference and solution configuration plans in terms of complexity we analyzed the occupancy counts. Figure 11 illustrates the occupancy counts distribution for each scenario from 6m to 7am by configured sector.

The highest peak traffic for the solution configuration is 17 occupancy counts, in contrast with the 23 registered for sector LECMBLU of the reference scenario around 6:45am. Despite having a very similar average occupancy counts for the same time interval, the solution configuration balances the ATCO workload among the configured sectors and smooths the peaks of traffic. This workload balance is reflected by the standard deviation (σ) of the occupancy counts. Table 1 presents the standard deviation for reference and solution scenarios from 6am to 9am.

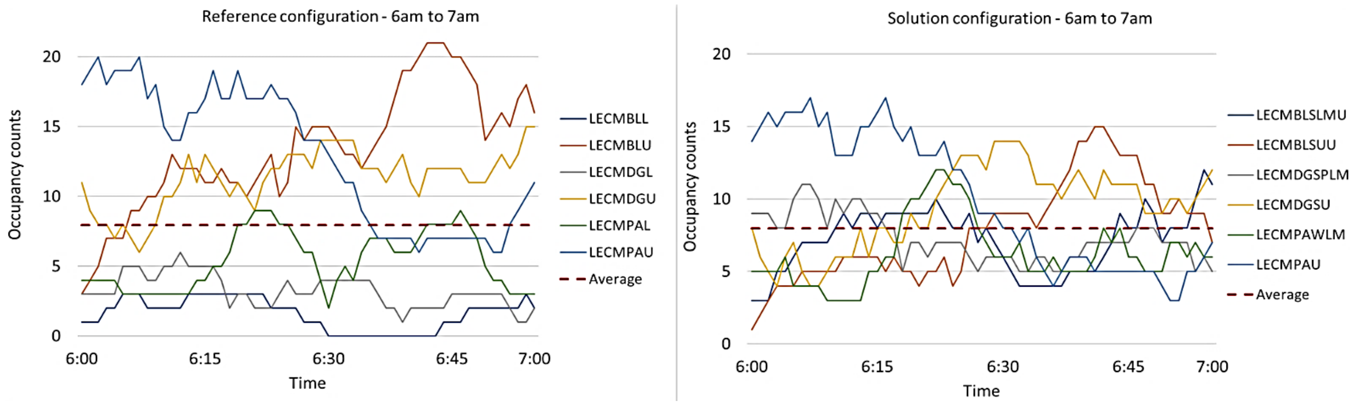

Figure 11: Reference and Solution configurations from 6am to 7am

Time Interval	Standard deviation (σ)	
	Reference scenario	Solution scenario
6am-7am	5.6	3.3
7am-8am	4.2	3.8
8am-9am	3.3	2.1

Table 1: Standard deviation of occupancy counts

Results show that the dispersion of the occupancy counts for the reference configuration is greater than the solution scenario. A similar trend was obtained for the rest of the day.

For a better understanding of the proposed configuration plan, the changes from one configuration to another are also presented graphically in Figure 12. Each configured sector is highlighted in a distinctive color and their respective flight levels involved for the vertical cuts are detailed under their denominations.

During the framework development phase, we performed other tests varying the traffic demand and time intervals. In few cases, we have observed recurrent changes in the proposed configuration (e.g., 6A-5A-6A). This occurs due to the presence of recurrent traffic peaks observed in short periods of time, to the adherence of the solution to the traffic demand and the fact that each optimal solution is obtained independent of previous time intervals. To tackle this problem, further improvements of the framework will be focused on including further information about previous time intervals, so that we could provide additional weights that penalizes changes of current configurations.

VI. Conclusions and Future Work

We have adopted the State-Task Networks methodology to model the DAC concept, creating an optimization problem with defined constraints. Then, we used a MIP solver to find an optimal solution. The framework can define a ranked list of sector configurations with the minimal cost. It is obtained by removing the best solution of the problem and running the algorithm again.

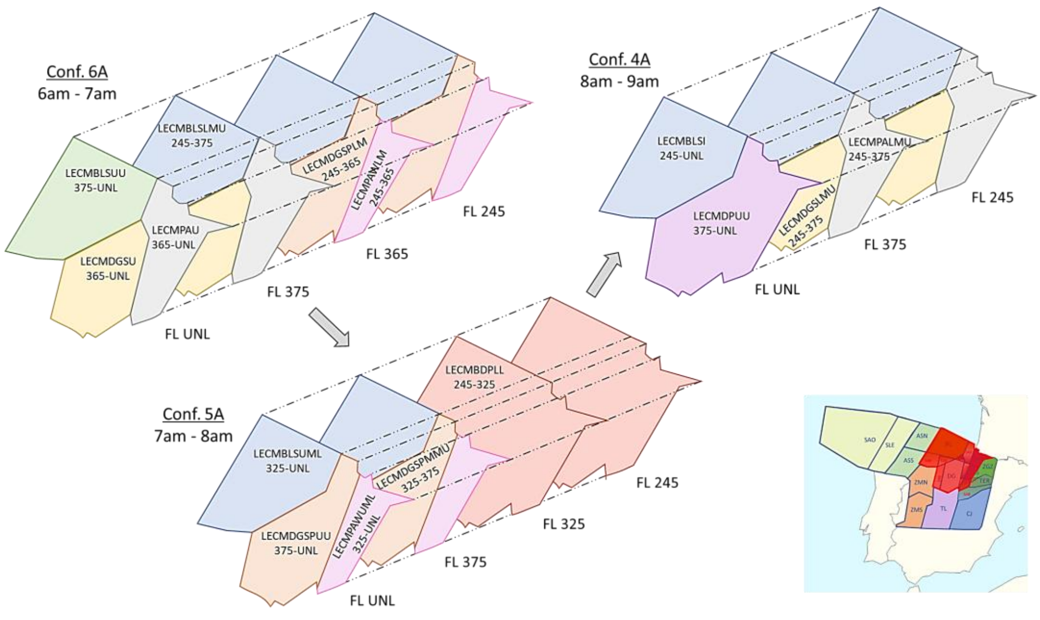

Figure 12: Graphical representation of the configuration plan for solution scenario

This allows a configuration plan to be established by evaluating similar configurations (in terms of cost), which may be discarded for more complex reasons to model (e.g., cost of switching to another configuration or staff availability).

To validate our approach, the framework was tested by simulation. We selected a use case in Madrid ACC Route 1 airspace to define a set of alternative configuration plans for the historical traffic of July 21st 2019, which was a representative busy traffic day. For this purpose, we performed a complexity assessment based on measuring the overloaded and underloaded areas from the distribution of occupancy counts and the assignment of a cost value to each configured sector. Finally, the building block merging problem was solved by logically evaluating different combinations.

Results of the use case show that the number of configured sectors used by both reference and solution scenarios was similar. Nevertheless, the solution configuration had a more balanced ATCO workload distribution among the configured sectors and their traffic peaks were lower.

Some cases involving recurrent traffic peaks in short periods of time produced solutions that included unnecessary configuration changes. We associate these inaccuracies to the lack of information about previous time intervals (memory of the system). Hence, future development could be focused on tackling this problem by including the current and previous states of the airspace configuration in the modelling process. Moreover, complexity is based on traffic counts, consequently future versions could include other variables (such as traffic density or uniformity), increasing the computational complexity of the problem.

ACKNOWLEDGMENT

This project has received funding from SESAR Joint Undertaking (JU) under grant agreement No 874463. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the SESAR JU members other than the Union. Also, we thank ENAIRE's operational staff for their support in the design and use case validation.

REFERENCES

[1] Bertsimas, D., Lulli, G. and Odoni, A. (2011) 'An integer optimization approach to large-scale air traffic flow management', Operations Research, 59(1), pp. 211–227. doi: 10.1287/opre.1100.0899.

[2] Bose, S. K. and Bhattacharya, S. (2009) 'A State Task Network model for scheduling operations in cascaded continuous processing units', Computers and Chemical Engineering, 33(1), pp. 287–295. doi: 10.1016/j.compchemeng.2008.09.023.

[3] Bynum, M. L. et al. (2021) Pyomo – Optimization Modeling in Python. Third Edit. Springer.

[4] Chen, Y. and Zhang, D. (2014) 'Dynamic airspace configuration method based on a weighted graph model', Chinese Journal of Aeronautics, 27(4), pp. 903–912. doi: 10.1016/j.cja.2014.06.009.

[5] Eurocontrol (2020) NEST tool EUROCONTROL. Available at: https://www.eurocontrol.int/model/network-strategic-modelling-tool.

[6] Eurocontrol (2021) 'EUROCONTROL Forecast Update 2021-2024 European Flight Movements and Service Units (May2021)', (May).

[7] EUROCONTROL (2019) 'SESAR Concept of Operations (CONOPS)', p. 146. Available at: http://www.sesarju.eu/sites/default/files/documents/highlight/SESAR_ConOps_Document_Step_1.pdf.

[8] EUROCONTROL (2021) 'ATFCM OPERATIONS MANUAL Network Manager', pp. 0–287. Available at: https://www.eurocontrol.int/sites/default/files/2021-03/eurocontrol-atfcm-operations-manual-25-26032021.pdf.

[9] Forrest, J. (2000) 'Coin-or Branch and Cut Solver'. Available at: https://projects.coin-or.org/Cbc.

[10] Gerdes, I., Temme, A. and Schultz, M. (2018) 'Dynamic airspace sectorisation for flight-centric operations', Transportation Research Part C: Emerging Technologies, 95(July), pp. 460–480. doi: 10.1016/j.trc.2018.07.032.

[11] Hazaras, M. J., Swartz, C. L. E. and Marlin, T. E. (2012) 'Flexible maintenance within a continuous-time state-task network framework', Computers and Chemical Engineering, 46, pp. 167–177. doi: 10.1016/j.compchemeng.2012.06.028.

[12] Jagare, P., Flener, P. and Pearson, J. (2013) 'Airspace sectorisation using constraint-based local search', Proceedings of the 10th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2013.

[13] Kondili, E., Pantelides, C. C. and Sargent, R. W. H. (1993) 'A general algorithm for short-term scheduling of batch operations-I. MILP formulation', Computers and Chemical Engineering, 17(2), pp. 211–227. doi: 10.1016/0098-1354(93)80015-F.

[14] Lee, J. U. et al. (2016) 'Optimal design of the multi-bed storage system in fusion fuel cycle under periodic demand based on the state-task network representation', International Journal of Hydrogen Energy, 41(38), pp. 17008–17021. doi: 10.1016/j.ijhydene.2016.07.078.

[15] Lin, X. and Floudas, C. A. (2001) 'Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation', Computers and Chemical Engineering, 25(4–6), pp. 665–674. doi: 10.1016/S0098-1354(01)00663-9.

[16] Sergeeva, M. et al. (2015) 'Dynamic airspace configurations generated by evolutionary algorithms', AIAA/IEEE Digital Avionics Systems Conference - Proceedings, pp. 1F21-1F215. doi: 10.1109/DASC.2015.7311352.

[17] Sergeeva, M. et al. (2017) 'Dynamic airspace configuration by genetic algorithm', Journal of Traffic and Transportation Engineering (English Edition), 4(3), pp. 300–314. doi: 10.1016/j.jtte.2017.05.002.

[18] SESAR (2015) 'SESAR Solution PJ09 SPR INTEROP OSED - Part I', (March 2019).

[19] SESAR (2017) 'Methodology for the Performance Planning and Master Plan Maintenance', (Ed 0.13).

[20] SESAR JU (2019a) COTTON - Capacity Optimisation in TrajecTory-based OperatioNs. Available at: https://cordis.europa.eu/project/id/783222/es.

[21] SESAR JU (2019b) PJ08 Advanced Airspace Management. Available at: https://cordis.europa.eu/project/id/731796/es.

[23] SESAR JU (2019c) PJ09 Advanced DCB. Available at: https://cordis.europa.eu/project/id/731730/es.

[24] SESAR JU (2021) 'PJ32 Virtual Centre'. Available at: https://cordis.europa.eu/project/id/101017587/es.

[25] Treimuth, T., Delahaye, D. and Ngueveu, S. U. (2016) 'A branch-and-price algorithm for dynamic sector configuration', ICAOR 2016, 8th International Conference on Applied Operational Research, 8, pp. 47–53.

[26] Vanzetti, N., Corsano, G. and Montagna, J. M. (2021) 'Integrated scheduling of the drying process in a sawmill', Computers and Chemical Engineering, 153. doi: 10.1016/j.compchemeng.2021.107407.

[27] Wong, C. S. ., Venugopalan, T. K. and Suresh, S. (2017) 'A multi-objective approach for 3D airspace sectorization: A study on Singapore regional airspace', 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016. doi: 10.1109/SSCI.2016.7850098.

[28] Wong, C. S. Y., Suresh, S. and Sundararajan, N. (2020) 'A rolling horizon optimization approach for dynamic airspace sectorization', IFAC Journal of Systems and Control, 11, p. 100076. doi: 10.1016/j.ifacsc.2020.100076.

[29] Xu, Y., Prats, X. and Delahaye, D. (2018) 'Synchronization of Traffic Flow and Sector Opening for Collaborative Demand and Capacity Balancing', in 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). IEEE, pp. 1–10. doi: 10.1109/DASC.2018.8569789.

[30] Zhang Zheng, D., Puntero Parla, E. and Cidoncha Sánchez, M. (2021) 'Probabilistic Complexity in support of Airspace Capacity Management Optimisation', ATM Seminar.