Tactical Prediction of the Number of Control Positions with Softmax Regression and Tree Search

Judical Bedouet, Thomas Dubot
ONERA / DTIS, Université de Toulouse
F-31055 Toulouse - France
Email: (Judicael.Bedouet|Thomas.Dubot)@onera.fr

***Abstract*—In the tactical phase, ATC supervisors and Flow Management Positions (FMP) foresee the number of control positions to be deployed in the next few hours through the analysis of traffic counts, the update of the daily sector configuration plan, and their capacity to anticipate the probable evolution of the traffic demand. We present in this paper a method based on softmax regression and tree search to predict the number of control positions based on the latest traffic count previsions for a given time period. This method is inspired by some of the current operational techniques: first a multinomial logistic regression is applied to compute an approximate value by analyzing the traffic count values of the whole Area Control Center (ACC). Then a tree search algorithm identifies all the sector configurations that could be deployed during this period, without generating an overloaded sector. The configuration with the minimal number of sectors is finally used to estimate the minimum number of control positions to be deployed. Results obtained on the French ACC of Bordeaux show that the predictions are close to the results obtained by operational expert assessment and could hence be used as a decision support tool to alleviate FMP and ATC supervisor workload.**

***Keywords*— control positions; sector configuration; softmax regression; tree search**

I. Introduction

The European airspace is controlled by Air Traffic Management units called Area Control Centers (ACC), in charge of providing Air Traffic Control (ATC) services to aircraft within their area of jurisdiction. Each ACC airspace is subdivided into elementary sectors that can be combined to build control sectors, operated by a team of air traffic controllers forming a control position. This process is dynamic: the subdivision of the ACC into control sectors varies throughout the day, depending on the incoming traffic and the number of available controllers. Basically, sectors are split when controllers workload increases and merged when it decreases. At any moment, we call *sector configuration* the set of control sectors deployed to fulfill the ACCs role.

During the strategic phase, the Flow Management Position (FMP) is in charge of building the sector configuration plan, i.e. the description of the different configurations deployed throughout the day. The sector configuration plan is updated during the execution phase, by taking into account the estimated traffic demand. The FMP compares predicted traffic counts with sector capacities to choose the most relevant sector configuration at each time period of the day. Once it has been updated with the latest data, this sector configuration plan is used by the FMP and the ATC supervisor to anticipate the number of control positions that should be deployed in the next few hours and to manage adequately the ATC roster.

A decision-support tool capable of automatically computing (from the same input data) an estimation of the required number of control positions would alleviate the FMP workload and ease the decision-making process of the ATC supervisor. In this paper, we present a method based on softmax regression and tree search that could be implemented in such a tool.

This paper is organized as follows: Section II summarizes the working methods currently used to predict the number of control positions. Section III presents an overview of related works. Section IV describes the use of a softmax regression method as a first heuristic and Section V presents the tree search method to eventually assess the minimum number of control positions. Finally results are presented in Section VI and conclusions and perspectives in Section VII.

II. Current working methods

A. Analysis of traffic counts

The occupancy count (OCC) is often used to evaluate if a sector must be split because of an unmanageable controllers' workload. The OCC for a given sector is defined as the number of flights inside the sector during a selected time period. This period is defined in terms of step and duration values, the step defining the time difference between the start time of two consecutive time periods and the duration the time difference between start and end time of each period [1]. The purpose of this paper is not to discuss about the relevance of the OCC

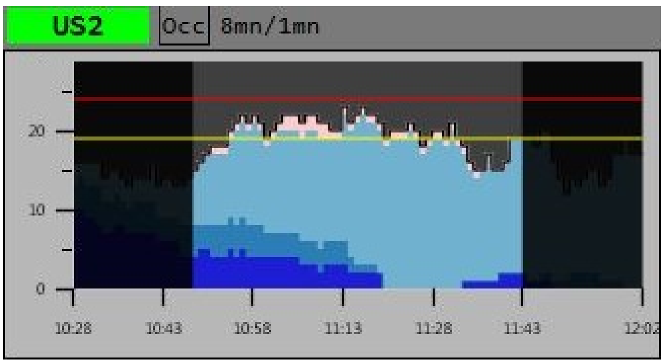


Figure 1. Occupancy traffic count with peak (red) and sustain (yellow) thresholds

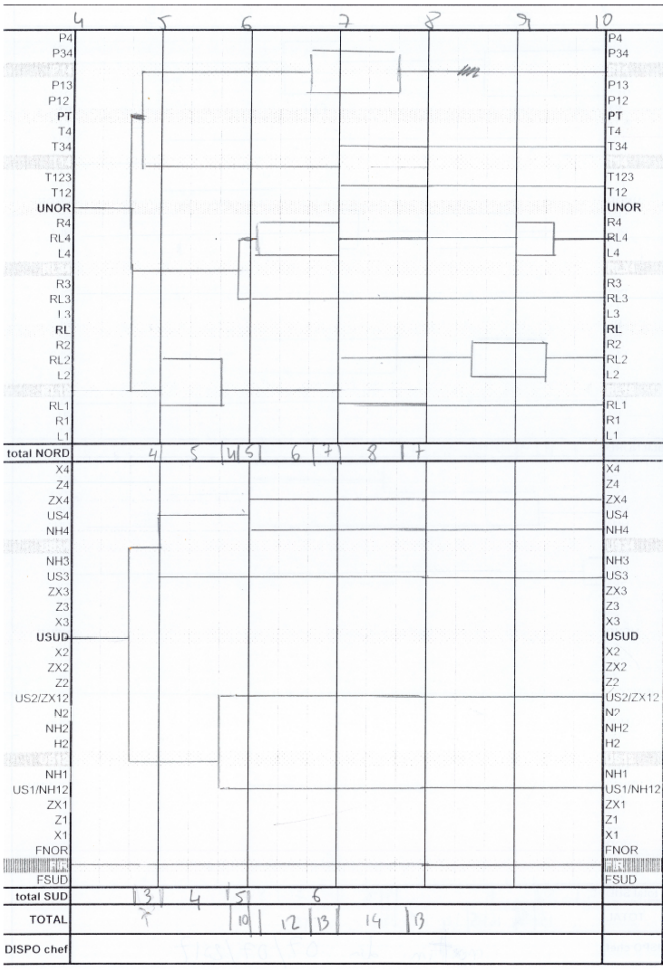


Figure 2. Example of tactical sector configuration plan in Bordeaux ACC

metric, nor the need to correctly assess controllers' workload, with more advanced ATC complexity metrics [2], [3], [4]. The method presented in this paper is intended to be relevant with input data currently used in Bordeaux ACC, i.e. essentially the occupancy traffic counts.

For each sector, sustain and peak occupancy values are defined by expert controllers as thresholds used to monitor the level of traffic demand. Figure 1 gives an example of occupancy values of sector US2, foreseen to be between the sustain threshold (in yellow) and the peak threshold (in red) within the time period selected. The chart contains vertical bars representing the number of flights. Each vertical bar is composed of a series of stacked coloured blocks based on seven different flight states (ATC activated, TACT activated with no FSA expected, RPL, PFD, IFPL, TACT activated with FSA expected, Suspended) as described in [5]. Through their operational experience, FMP and ATC supervisors are able to foresee from the different values associated to the flight states (colors) the evolution of the occupancy traffic count of the sector and hence assess if a splitting will be required.

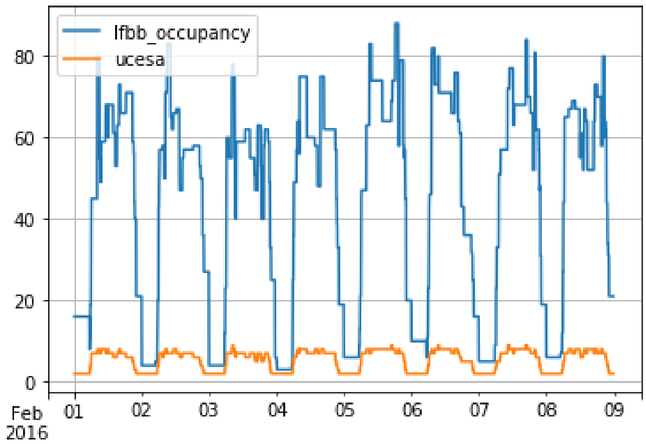


Figure 3. ACC occupancy versus number of positions during a week

B. Building a sector configuration plan

The sector configuration plan is regularly updated with the latest traffic data in the tactical phase. More precisely, the FMP analyzes occupancy traffic counts of the current sector configuration. He identifies when it will be necessary to split or merge sectors by anticipating potential overloads/underloads. He then assesses the traffic counts of the sectors that could be opened by checking whether it could cause an overload, or to choose the best division, for example in terms of smooth transitions or workload balance.

The result of this analysis is a sector configuration path for the next hours, as illustrated by Figure 2. This plan is used by the FMP and the ATC supervisor to assess the number of control positions that will probably be required in the next hours.

C. Empirical method to assess the number of control positions

Other methods are used to rapidly estimate the number of sectors to be opened. One of them consists in a heuristic based on the traffic counts of the ACC (all sectors). More precisely, this method is based on the observation that there is a clear relationship between the number of sectors and the total number of flights in the ACC, as illustrated by Figure 3. Considering the data of a week in February 2016, this figure shows that we can observe approximately a ratio of ten between the number of positions (in orange) and the number

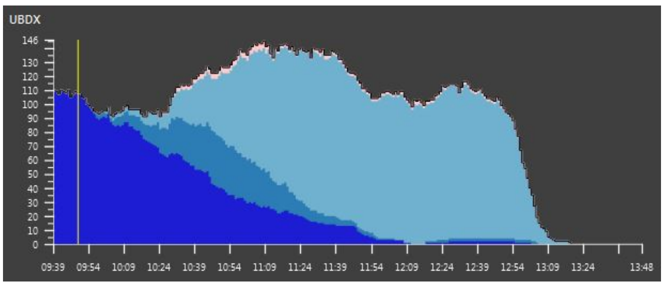


Figure 4. Occupancy traffic count of the ACC center in tactical horizon

of flights in the ACC (in blue). Nevertheless, as illustrated by Figure 4, when the FMP analyzes the occupancy traffic counts of the ACC, he has access to traffic previsions of the seven different flight states. He therefore needs to estimate the total number of flights from the number in each flight state (colors). Depending on the sector, the characteristics of flows and the past evolution of similar traffic situations, the FMP is able to predict a value based on his own traffic uncertainty model.

III. Previous related works

Many optimization methods have been applied in the past to solve the airspace sectorization problem [6]. Most of these methods take into account the number of sectors in their cost function, i.e. the sector configurations optimized for a given time period often to minimize the number of control positions. Such methods could thus be used in our context but they present two main limitations. First they often rely on stochastic models, such as genetic algorithms [7] and hence cannot guarantee reproducible results. Besides they are mostly based on the optimal association of elementary airspace volumes [8], that often leads to unusual control sectors, far from those usually deployed. Instead, we present here a determinist method based on the exploration of operational sector configurations.

In [9], [10], Gianazza presents a method based on the operational behavior of the FMP and ATC supervisors. A tree search method is used to explore all possible combinations of elementary airspace modules. Each sector configuration is kept as long as it remains acceptable, i.e. with a workload predicted as normal. The consecutive sector configurations obtained give a good estimation of the number of control sectors opened. The main difference with our method is that this method relies on the forecast of the ATC workload by a neural network trained with historical data instead of using OCC traffic counts and associated peak/sustain thresholds. Besides the tree search is realized with a branch and bound algorithm to cut branches based on the cost of each node. The method presented in this paper is closer to the current operational practise, as the workload assessment is performed with occupancy data and the algorithm performs an exhaustive search within a catalogue of operational configurations.

IV. Using softmax regression to approximate the number of control positions

A. Evaluation of the empirical method

The first step of our method is directly inspired by the empirical method described in II.C. In order to evaluate the relationship between the number of control positions and the ACC occupancy, we collected the overall data from 2016 and analyze the occupancy values at transition times from one configuration to another. Figure 5 shows all the samples (more than 13000) with blue crosses. As observed by the operational experts, the result is a linear relationship between the occupancy and the positions deployed. If we look at the straight line obtained via a simple linear regression using the ordinary least squares method (in red), we notice that this line

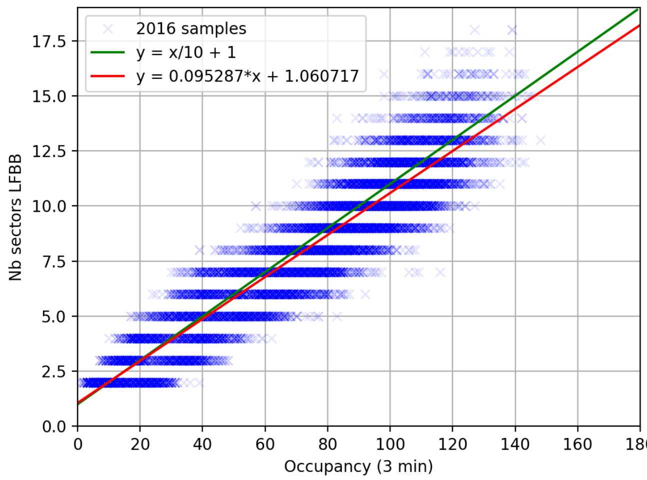


Figure 5. Linear regression of the number of positions versus the occupancy

is very close to the formula used by operational experts (in green).

B. Description of the softmax regression model

If the use of this formula is accurate enough for operational experts, the method can be improved with the use of a real classification model, returning directly one of the 17 classes (from 2 to 18) instead of a real number, and the minimization of the cost function with the cross entropy.

The softmax regression, or multinomial logistic regression, is a classification method that generalizes logistic regression to multiclass problems. For each instance, the softmax regression model computes a score $s_k(\mathbf{x})$ for each class k, then estimates the probability of each class by applying the softmax function (or normalized exponential).

$$\hat{p}_k = \exp\left(s_k(\mathbf{x})\right) \Bigg/ \sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right) \tag{1}$$

Parameters (or weights) Θ are obtained by minimizing the following cost function:

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right) \tag{2}$$

$y_k^{(i)}$ is equal to 1 if the target class for the i^{th} instance is k. Otherwise $y_k^{(i)}$ is equal to 0. K is equal to 17 and m is equal to 8928 as we used as training data a month of data (January 2016) sampled in periods of five minutes.

The optimization is realized using the Limited-memory BFGS (L-BFGS) solver [11], which belongs to quasi-Newton methods.

The classifier was used to predict the results on another week of data (not used in the training phase). The accuracy obtained is above 65% (95% with a tolerance of one control position).

C. Limitations of the regression approach

Such results could lead to the conclusion that our objective could be fulfilled with an algorithm based on the current empirical method described in II.C, i.e. by using a regression model based on the occupancy traffic counts of the ACC. Nevertheless several limitations must be noted. First the results presented in IV.B are obtained in the winter season, which is more predictable than the summer season with higher traffic loads. Besides the results are obtained by using exact occupancy values at each time step, and not traffic predictions. To be totally in line with the empirical method based on available input data, this model should use the traffic counts per flight type (see colors on Fig. 5). The prediction would be hence based on logical rules expressing how to estimate the total number of flights from these traffic counts. Nevertheless the softmax regression model can be used as an heuristic to improve the efficiency of the tree search model presented in the next section.

V. Tree search method to assess the minimum number of control positions

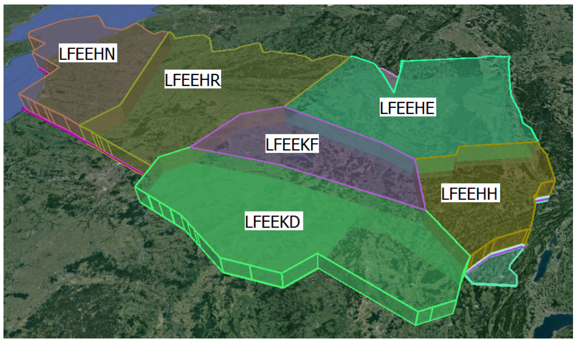
The objective of the tree search method is to rapidly build and explore an exhaustive catalogue of the different possible sector configurations for each number of control positions in the range determined by the softmax regression model presented in IV. Even if the general sectorization problem can be seen as a graph partition problem, which is known to be NP-difficult [12], we observe that, given the complexity of the current ACCs, it is possible to enumerate all the different sector configurations once considering operational constraints.

A. Enumerating the different control sectors

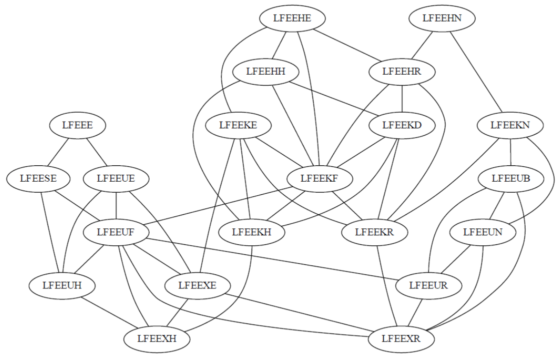
Each ACC declares a limited number of elementary and control sectors. In some cases, we may have a higher number of elementary sectors than average but these sectors can not be individually controlled at the same time, because of radio frequency interference. Some of them are even declared uncontrollable but they can be exchanged between different control sectors. Nowadays, even if we can refine some elementary sectors, radio communications limit their size and their use. The number of control sectors is then naturally limited.

Another major operational constraint to define a new control sector is compactness [13]. A control sector should not contain any balcony, or at least a limited number of balconies when they can not be avoided. The ideal control sector should have the same shape from top to bottom to avoid any misinterpretation when it is controlled. This constraint limits the different combinations of elementary sectors to form a control sector.

Training considerations and ATCO licensing also limit the control sectors that can be declared. In this paper, we consider declared control sectors and possibly new control sectors that are operationally acceptable. Given a list of control sectors, the objective of the tree search model is to efficiently enumerate all the sector configurations composed of sectors from this list.



(a) 21 Reims ACC blocks (some are covered by top blocks)



(b)

Figure 6. Reims ACC building blocks (a) vs. their graph representation (b)

B. The clustered tree search model

The airspace controlled by an ACC can be modeled as a graph where each elementary sector is mapped to a node of the graph $G = (V, E)$ (see Figure 6). V, the set of vertices, is the set of elementary sectors and E is the set of edges such that (u, v) belongs to E only if the two sectors u and v are adjacent [13].

A sector configuration with k controllable sectors can be modeled as a partition of G in k subsets such as $P_k = S_1, ..., S_k$. As a partition, the following constraints must be satisfied:

1) each subset S_i must be non-empty ;
2) each subset S_i must be disjoint from the other subsets ;
3) each vertex inside a subset S_i must be directly connected to another vertex of the same subset ;
4) the union of all the subsets must be equal to V.

The main idea of the tree search algorithm is to rapidly explore with a k-ary tree the different combinations of control sectors by cutting some branches when a configuration does not respect the four constraints previously stated.

To initialize the algorithm, we consider an ordered sequence of the different controllable sectors and create a k-ary tree for each controllable sector. Specifically, each of the sectors of the sequence is the root of a k-ary tree. The children of a node n (position in the sequence) in this tree corresponds to all the sectors listed in the sequence after the sector n. Therefore, we ensure to have a combination of the different control sectors, i.e. a selection of control sectors where the order of selection does not matter. A sector configuration corresponds to a path between a root node and a leaf where all the constraints are satisfied. While exploring the tree, the algorithm cuts branches and stops the exploration of a node n when:

- constraint 4) is satisfied. The set of elementary sectors corresponding to the different parent nodes and the node n itself is equal to the set V. The path from the root node to the node n corresponds to a sector configuration. This configuration is registered in a catalogue of available configurations.
- constraint 4) is not satisfied and there are no more nodes to be explored.
- constraint 2) is not satisfied. The node n has elementary sectors that are already contained by the parent nodes. It also ensures that we cannot have more elementary sectors than $\#V$, the total number of elementary sectors.

It has to be noted that constraints 1) and 3) are ensured by construction. A controllable sector is composed of at least one elementary sector and the different elementary sectors of a control sector are connected.

An important point to optimize the cutting process is the way the initial sequence is ordered. To explain this point, we define the *elementary size* of a sector as the number of elementary sectors that it contains. If we sort (in descending order) the initial sequence according to the elementary size of the different controllable sectors, we can increase the cutting efficiency. Indeed, a branch can be cut when the sum of the elementary size of the parent nodes added to the sum of the maximum elementary size of the nodes to explore is strictly smaller than $\#V$. At any step of the exploration, we verify there is a sufficient number of elementary sectors to satisfy constraint 4). Sorting the sequence according to their elementary size helps to have a good estimation of this sum, since the maximum elementary size is the elementary size of the current node. This is particularly efficient when nodes to be explored are small and plentiful, and thus the number of possible combinations is high.

An incomplete example of exploration is given in Figure 7 for a cluster that would only be composed of the four elementary and controllable sectors R1, R2, L1 and L2. Five control sectors are declared: RL12, RL1, RL2, R12 and L12. They are compact and satisfy the connectivity constraint. The control sector RL12 is the sector that entirely covers the cluster. The four other sectors contain two of the four elementary sectors. The initial sequence is ordered according to the elementary size of the different controllable sectors, i.e. {RL12, RL1, RL2, R12, L12, R1, R2, L1, L2}. As RL12 entirely covers the cluster, the exploration of RL12 is immediately stopped from the root, feeding the catalogue with the sector configuration {RL12}. The next sector in the sequence, RL1, is explored. Its exploration produces the two sector configurations {RL1, RL2} and {RL1, R2, L2}. Some branches are cut because the constraints 2) and 4) are not satisfied. In the exploration of the node R1, the sequence {R1, L1} is cut because we know that the sum of the elementary size of the remaining nodes to be explored is strictly smaller than $\#V$.

Sectors of an ACC are often organized into clusters. A cluster corresponds to a subset of sectors that are not connected to a sector of another cluster. The catalogue of all the sector configurations for an ACC is then the aggregation of the catalogues of the different clusters.

Finally, the size of a sector configuration is limited by the number of available controllers. Each ACC declares, in the European Network Operations Plan, the maximal number of sectors to be opened in the following years [14]. This constraint limits the depth of the tree and so the nodes to be explored.

C. Numerical considerations

If the connectivity constraint is not considered, the number of possible configurations is given by the second Stirling number $S(n,k)$:

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n,$$

where n is the number of elementary sectors,
k is the number of positions to be opened,
$\binom{k}{j}$ is the binomial coefficient $\dfrac{k!}{j!(k-j)!}$

We can compare this number to the nodes really explored by the algorithm to determine its cutting performance. Let us define

m the number of controllable sectors,
N_{conf} the total number of valid configurations,
N_{expl} the total number of explored nodes,
$Perf$ the cutting performance $N_{expl} \Big/ \sum_{i=1}^{n} S(n,i)$.

Table I shows the total number of valid configurations for the current clusters in the Bordeaux ACC. It also presents the performance of the branch cutting. The higher the number of elementary sectors n, the less the percentage of explored nodes is, which is interesting because the combinatorial complexity explodes with n.

TABLE I
PERFORMANCE OF THE ALGORITHM FOR THE BORDEAUX ACC

Cluster	n	m	N_{conf}	N_{expl}	$\sum_{i=1}^{n} S(n,i)$	$Perf$
East	12	26	165	1.8e4	4.2e6	0,4%
North	10	29	174	1.4e4	1.0e5	14%
South	16	64	6185	1.4e6	1.0e10	0,01%

Table II presents the result for two nonexistent clusters. In both clusters, we consider all the possible compact sectors that may be defined. The cluster $North_{new}$ contains all the

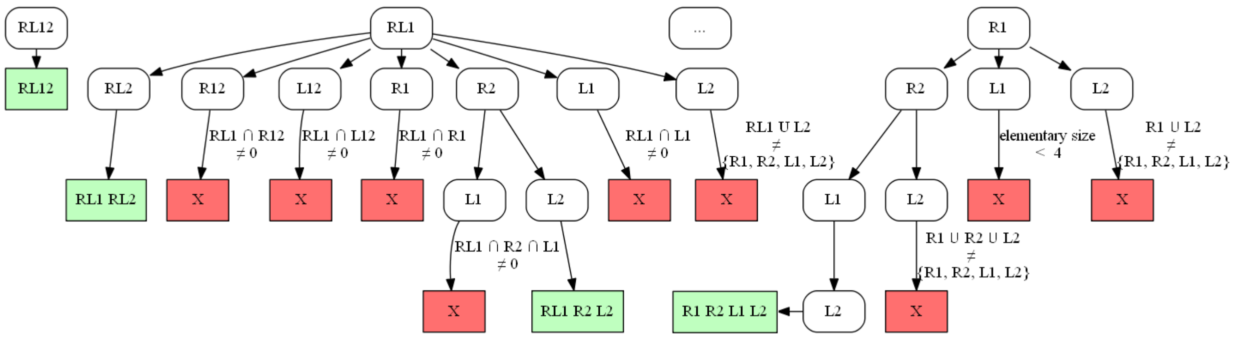


Figure 7. Example of tree search exploration for a cluster called RL12

elementary sectors from the eastern and southern clusters except the FIR sectors[1].

TABLE II
PERFORMANCE OF THE ALGORITHM FOR BORDEAUX EXPLORATORY MODE

Cluster	n	m	N_{conf}	N_{expl}	$\sum_{i=1}^{n} S(n,i)$	$Perf$
$North_{new}$	18	121	47439	3.1e7	6.8e11	0,004%
$South_{new}$	16	140	126856	3.3e7	1.0e10	0,3%

It has to be noted that the Bordeaux ACC is representative of the most complex ACC in Europe. Thus, this approach should be applicable to most European ACCs. A simpler version (without cluster and cut optimization) had already been applied to the Reims ACC [13]. An extended approach would consist in dynamically exploring the tree while cutting branches where sectors are overloaded.

D. Exploration of the catalogue

At the end of the tree exploration, we have a catalogue of sector configurations that can be deployed to fulfill the ACC role. The next phase is to determine, for a given time period, which configurations among this catalogue would be the most relevant, and hence give a good estimation of the number of control positions to be deployed.

An optimal sector configuration must not contain any overloaded sector. Nevertheless, the notion of an overloaded sector is not easy to define. It depends on many factors related to real traffic complexity or human factors. To share with FMP and ATC supervisors a common understanding of the way an overload is triggered, we define the following generic parameters:

- avg_{min} average number of occupancy counts above peak before considering an overload ;
- δ_{peak} maximal difference between peak and the occupancy value before considering an overload ;
- δt_{min} minimum duration of an overload to trigger it.

[1]The FIR cluster forms a very small cluster with only four elementary sectors inside the current eastern cluster. It is a special operational case that can be ignored to assess the performance.

Finally, we browse the catalogue to find the optimal size N_{optim} such as

$$\exists c \in C \ \text{size}(c) = N_{optim} \implies \forall \text{sector} \in c \ !\text{overloaded}(\text{sector})$$
$$\forall c \in C \ \text{size}(c) < N_{optim} \implies \exists \text{sector} \in c \ \text{overloaded}(\text{sector})$$

where C is a catalogue of sector configurations,
size gives the number of sectors of a configuration,
overloaded indicates if a sector is overloaded or not.

VI. RESULTS OF THE METHOD AND DISCUSSION

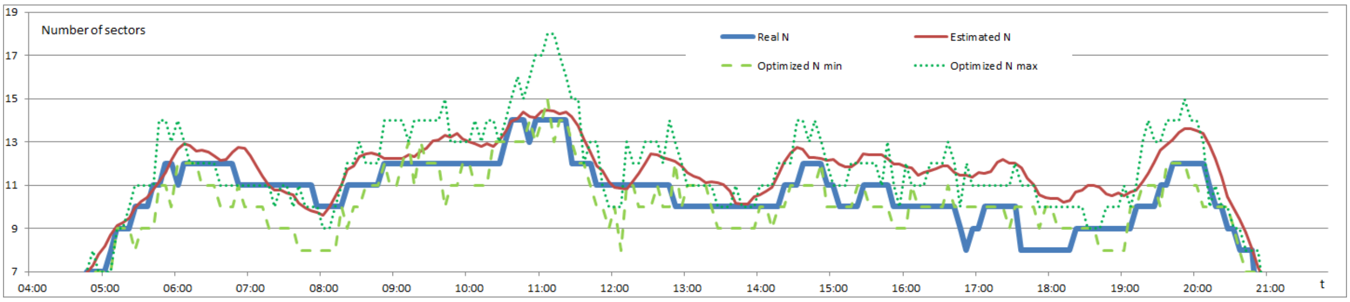
A. Results of the method

The aforementioned approaches were applied to the Bordeaux ACC. We chose a busy day to compare our prediction with the real sector configurations. On Friday August 3^{rd}, 2018, we recorded every five minutes the current sector configuration and the current elementary flight lists (lists of the flights through Bordeaux elementary sectors) for the next five hours and computed occupancy counts for each controllable sector. Flight lists are obtained via the European Network Manager (NM) B2B Web Services [15].

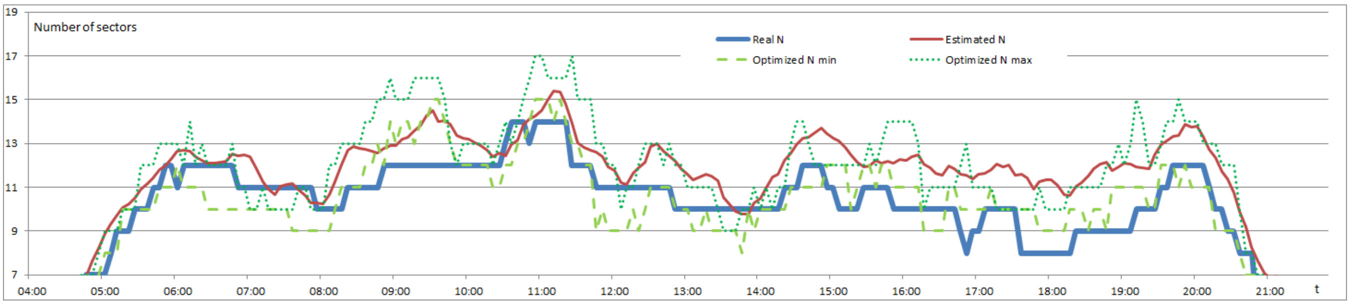
We define two profiles to detect overloads on controllable sectors: a safe profile and an daring profile. The safe profile parameters are chosen so that sectors are rapidly considered as overloaded (depending on the peak and sustain thresholds) and splittings are rapidly proposed (avg_{min} = 0.5 flight/min, δ_{peak} = 3 flights, δt_{min} = 5 min). Conversely, the daring profile defines parameters so that when a sector is clearly overloaded, the splitting is not automatically triggered (avg_{min} = 2.5 flight/min, δ_{peak} = 6 flights, δt_{min} = 10 min).

Every five minutes, we estimate which value N_{optim} could take with the regression model presented in section IV-B. We browse the catalogue, presented in section V-D, to refine the estimated value and find the N_{optim} sector configurations corresponding to the two previous profiles. The computation is done for the periods [$t0$ + 20 minutes ; $t0$ + 40 minutes] and [$t0$ + 120 minutes ; $t0$ + 140 minutes] where $t0$ is the current instant of time.

To select sector configurations close to the current configuration, we must define a distance between two configurations. There are two categories of distance functions: the geometric-based and the shared-cell distances [16]. The geometric functions can, for example, be based on Hausdorff distance [17].



(a) Occupancy counts at H-20 minutes



(b) Occupancy counts at H-2 hours

Figure 8. Bordeaux results for Friday August 3rd, 2018 versus the real number of sectors (in blue)

The cell-based functions are based on the elementary sectors that are shared or not by the different sectors of two sector configurations [13]. For this study, we introduce a shared-cell function to estimate the distance between two configurations : distance $(P_k, P'_{k'}) = \#\ P_k \triangle P'_{k'}/3$, based on the symmetric difference of two partitions. This distance can be interpreted as the number of transitions to go from the source to the target configuration when the only differences between two configurations are due to merging or splitting operations. When merging or splitting, we exchange one sector with two other sectors. The cardinality of the symmetric difference is then 3 and the distance is equal to 1, which measures one transition to go from the source to the target configuration.

Figure 8 presents the size of the real configuration (blue thick continuous curve), the value given by the regression model (red continuous curve), N_{max}, N_{optim} for the safe profile and N_{min}, N_{optim} for the daring profile (dashed curves).

Figure 8a shows that the real configuration is surrounded with the two profiles. The corresponding solutions can be interpreted in three ways :

- if a solution, found with the daring profile, is close to the current configuration, it identifies the sectors that will surely be opened ;
- if a solution, found with the safe profile, is close to the current configuration, it identifies the sectors that may be split or not ;
- if a solution is not close to the current configuration, another strategy to open the room is identified. This strategy can be applied if the control room supervisors think that it may be interesting to better balance workload between the different positions and then save some resources when the traffic demand becomes very high.

Figure 8b shows the results of the same model for the period [$t0$ + 120 minutes ; $t0$ + 140 minutes]. In this figure, we can compare the real configuration with the prediction at H-2. Even if we can find a model that can predict the sectors to be opened with a relative uncertainty, the model is unable to correctly predict the sectors to be opened in two hours because of the high volatility of traffic demand. This is typical of what is experienced in a control room. Traffic demand must be continuously monitored to adapt the sector configuration plan.

B. Short-time prediction results

In this paragraph, we discuss some results of Figure 8a.

At 8 o'clock, the model finds configurations with a smaller size than the real one. The current configuration, the min and max predictions twenty minutes before and the real configuration are presented just after this paragraph. For readability reasons, we remove sectors common to the two configurations. The model estimates that RL3 resp. ZX4 can be merged with RL45 resp. NH4. There is an uncertainty on the US12 merge. In reality, ZX4 and NH4 were merged but not RL3 and RL45. As demand decreases during a short period of time, the transition may be too short to implement RL345.

```
Current    NH12 ZX12    ZX4 NH4    RL3 RL45
Min -20      US12         US4        RL345
Max -20    NH12 ZX12      US4        RL345
Real       NH12 ZX12      US4      RL3 RL45
```

Around 11 o'clock, the uncertainty is maximal. The found configurations are quite identical except that three sectors might be split. Two sectors could be merged into RL45. In reality, P14 was split between P12 and P34. RL2 occupancy is very high but traffic complexity is probably not. Occupancy is a rather good metric to estimate workload but sometimes complexity may be lower than usual.

Current	ZX12	P14	RL2	RL4 RL5
Min -20	ZX12	P14	R2 L2	RL45
Max -20	Z12 X12	P1 P23 P4	R2 L2	RL4 RL5
Real	ZX12	P12 P34	RL2	RL4 RL5

VII. Conclusions

In order to provide FMP and ATC supervisors with a decision-support tool capable of automatically computing from occupancy traffic counts an estimation of the number of control positions to be deployed, we developed a two-steps method inspired by the current empirical techniques used by operational experts.

First a softmax regression model is used to compute as an heuristic a range of possible values, based on the analysis of traffic counts of the overall ACC. Then a tree search algorithm is used to build a catalogue of valid sector configurations. For any given time period in the close future, these configurations are filtered to only keep configurations without any overloaded sector. The configurations with a minimum number of control sectors can then be used to estimate the minimal number of control positions to be deployed.

Nevertheless, as the overload of a sector is difficult to assess, we defined with operational experts two sets of parameters to build two models: a safe model triggering a splitting as soon as a traffic counts exceed predetermined thresholds, and one unsafe with a higher tolerance. The final algorithm can hence be used to give a range of potential values corresponding to the number of control positions, in the same way as some FMP can answer when their ATC supervisor requests an estimation.

Results obtained on the French ACC of Bordeaux show that this model gives good predictions in the tactical phase (20-40 minutes). Nevertheless, with larger time horizons, the model is limited by the uncertainty of the total traffic counts from the different values associated to the seven different flight states in the occupancy traffic counts, as detailed in II.A. As reported by operational experts, the sum of these values can be significantly different from the real final occupancy value, depending on the sector, the type of traffic and the targeted time period. To compensate this lack of expertise, a next step could be to complement our algorithm with a traffic count prediction model based on machine learning methods, such as LSTM networks [18].

Another possible improvement is to continuously make evolve the current sector configuration. The objective would be to ensure that the sector configurations proposed at a given time period are formed by sectors without overloads on this period, but also that they are reachable from the current sector with a smooth path. This implies to determine a sequence of sector configurations, with acceptable sector configuration transitions [19], where each intermediate sector configuration is not overloaded.

Acknowledgments

Authors would like to thank operational experts from Bordeaux ACC for their numerous feedbacks and DSNA/ONERA colleagues that provided insight and expertise on this sector configuration problem. Authors would like to express their gratitude to Luis Basora, Antoine Joulia, Valerie Neyns and SIDs anonymous reviewers, for their valuable and constructive suggestions before and during the reviewing of this paper.

References

[1] M. Dalichampt and C. Plusquellec, “Hourly entry counts versus occupancy count–relationship, definitions and indicators,” 2007.

[2] B. Sridhar, K. S. Sheth, and S. Grabbe, “Airspace complexity and its application in air traffic management,” in *2nd USA/Europe Air Traffic Management R&D Seminar*, 1998, pp. 1–6.

[3] D. Delahaye and S. Puechmorel, “Air traffic complexity: towards intrinsic metrics,” in *Proceedings of the third USA/Europe Air Traffic Management R & D Seminar*, 2000.

[4] D. Gianazza and K. Guittet, “Evaluation of air traffic complexity metrics using neural networks and sector status,” in *ICRAT 2006, 2nd International Conference on Research in Air Transportation*, 2006.

[5] EUROCONTROL. CFMU Human Machine Interface (CHMI) ATFCM reference guide interoperability. [Online]. Available: https://www.eurocontrol.int/sites/default/files/publication/files/chmi-atfcm-reference-guide-current.pdf

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

[7] M. Sergeeva, D. Delahaye, C. Mancel, L. Zerrouki, and N. Schede, “3d sectors design by genetic algorithm towards automated sectorisation,” in *5th SESAR Innovation days*, 2015.

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

[9] D. Gianazza and J.-M. Alliot, “Optimization of air traffic control sector configurations using tree search methods and genetic algorithms,” in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, vol. 1. IEEE, 2002, pp. 2A5–2A5.

[10] D. Gianazza, “Forecasting workload and airspace configuration with neural networks and tree search methods,” *Artificial intelligence*, vol. 174, no. 7-8, pp. pp–530, 2010.

[11] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[12] C.-E. Bichot, “Élaboration d’une nouvelle métaheuristique pour le partitionnement de graphe: la méthode de fusion-fission. application au découpage de l’espace aérien,” Ph.D. dissertation, Institut National Polytechnique de Toulouse, 2007.

[13] J. Bedouet, T. Dubot, and L. Basora, “Towards an operational sectorisation based on deterministic and stochastic partitioning algorithms,” in *The Sixth SESAR Innovation Days*, 2016.

[14] EUROCONTROL, “European Network Operations Plan 2018-2019/22,” 2018.

[15] ——. NM B2B Web Services Web interface for system-to-system interoperability. [Online]. Available: https://www.eurocontrol.int/sites/default/files/publication/files/nm-b2b-factsheet-2017-update-1512.pdf

[16] A. Yousefi, R. Hoffman, M. Lowther, B. Khorrami, and H. Hackney, “Trigger metrics for dynamic airspace configuration,” in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009, p. 7103.

[17] N. Gregoire and M. Bouillot, “Hausdorff distance between convex polygons,” *Computational Geometry Web project*, 1998.

[18] F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to time series predictable through time-window approaches,” in *Neural Nets WIRN Vietri-01*. Springer, 2002, pp. 193–200.

[19] T. Dubot, “Predicting sector configuration transitions with autoencoder-based anomaly detection,” in *Proceedings of the International Conference for Research in Air Transportation*, 2018.