

# Forecasting workload and airspace configuration with neural networks and tree search methods

David Gianazza

*Planning, Optimisation, and Modelling team  
DSNA/DTI/R&D  
7 avenue E. Belin, 31055 Toulouse, France*

*Parallel Algorithms and Optimisation Group  
Institut de Recherche en Informatique de Toulouse (IRIT)*

---

## Abstract

The aim of the research presented in this paper is to forecast air traffic controller workload and required airspace configuration changes with enough lead time and with a good degree of realism. For this purpose, tree search methods were combined with a neural network.

The neural network takes relevant air traffic complexity metrics as input and provides a workload indication (high, normal, or low) for any given air traffic control (ATC) sector. It was trained on historical data, i.e. archived sector operations, considering that ATC sectors made up of several airspace modules are usually split into several smaller sectors when the workload is excessive, or merged with other sectors when the workload is low. The input metrics are computed from the sector geometry and from simulated or real aircraft trajectories.

The tree search methods explore all possible combinations of elementary airspace modules in order to build an optimal airspace partition where the workload is balanced as well as possible across the ATC sectors. The results are compared both to the real airspace configurations and to the forecast made by flow management operators in a French *en-route* air traffic control

---

*Email addresses:* [gianazza\[at\]tls.cena.fr](mailto:gianazza[at]tls.cena.fr) (David Gianazza)

*URL:* <http://pom.tls.cena.fr> (David Gianazza)

DSNA (Direction des Services de la Navigation Aérienne) is the French Air Navigation Services Provider.

centre.

*Key words:* Neural network, tree search, *Branch & Bound*, airspace configuration, air traffic complexity

---

## 1. Introduction

Air traffic is currently controlled by human operators (air traffic controllers) who monitor aircraft trajectories and give instructions to pilots so as to avoid mid-air collisions and dangerous situations. The airspace is partitioned into managerial units, air traffic control centres (ATCC), which are themselves partitioned into *elementary airspace modules*<sup>2</sup>. These basic airspace modules may be combined together so as to form *air traffic control (ATC) sectors* each operated by a small team of 2-3 controllers<sup>3</sup>.

In many centres, the *airspace configuration*, i.e. the partitioning of the ATCC airspace into *ATC sectors*, may change during the day, depending on the incoming traffic and controller workload. Sectors may be split<sup>4</sup> when the workload increases, or merged (or collapsed) when the workload decreases. More complex recombinations may sometimes be decided by the control room manager, so as to balance the workload across all the control sectors.

The ultimate objective of the research presented in this article is to forecast airspace configurations with a good degree of realism, using a reliable workload forecast grounded on relevant air traffic complexity metrics. The basic idea is to learn from the current control sector operations in order to propose efficient forecasting models and algorithms.

Assessing the controller workload and predicting when this workload will exceed safe limits are difficult problems involving human factors, which have

---

<sup>2</sup>The usual term to denote these elementary airspace modules is *sector*, which may be confusing as it can either denote a single airspace module, or a control sector made up of several modules. So, in the rest of this paper, the elementary geographic sectors will be referred to as *modules*, in order to avoid confusion with *ATC (Air Traffic Control) sectors*.

<sup>3</sup>A team operating a control sector is composed of a *radar controller* who gives instructions to the pilots (direction or flight level changes), and a *planning controller*. The planning controller monitors the incoming traffic a few minutes before it enters the sector and is in charge of pre-detecting any potential conflicts, as well as coordination with the adjacent sectors. In some countries, an additional controller may assist these two controllers when the traffic is heavy.

<sup>4</sup>Splitting an ATC sector requires that it is composed of at least two modules.

been the subject of many studies (see [1] for an overview). Some studies focus on the relationship between air traffic complexity metrics and controller workload. A general statement is that workload depends largely on the number of flights, but also on the traffic complexity: aircraft flying parallel tracks at constant flight levels are less difficult to handle than climbing/descending flights on various converging trajectories, for example. However, no universal workload metric has been agreed on so far, and one can only hope to select a set of relevant air traffic complexity metrics that fits the chosen context and application. Our main contribution on this subject ([2] and [3]) was to use the sector status (*merged*, *normal*, or *split*) as a dependant variable, trying to find the subset of metrics that was best correlated to this sector status. The basic assumption is that the decision to reconfigure the ATC sectors is somewhat related to the controller’s actual workload.

Airspace design and management have also been the subjects of many studies, using several methods: mixed integer programming techniques ([4]), evolutionary algorithms ([5], [6], [7]), seed growth methods inspired by crystal growth ([8]), constraint programming ([9], [10], [11]), computational geometry ([12]), graph partitioning methods or a new meta-heuristic inspired by nuclear fusion and fission ([13], [14]). These studies address a variety of operational contexts: strategic airspace design, pre-tactical planning, and tactical airspace management. Some background on air traffic management and more details on related works on airspace design and management are provided in section 2. Let us just say that many of these studies ([7], [9], [10], [13], [14], [8]) addressed a highly difficult airspace partitioning problem, only slightly reduced by the introduction of connectivity<sup>5</sup> and convexity<sup>6</sup> constraints on sectorisation. Among these studies, most of them used only mock-up sectors, whereas very few used real sectors and real traffic. Some other studies ([4]) were made in a highly realistic context but used metrics that are not sufficient to model the actual workload (see subsection 2.2), and a fairly reduced subset of pre-defined airspace configurations. Others ([11]) used more relevant air traffic complexity metrics to assess the controller workload, but did not propose airspace reconfigurations.

Our contribution aims at improving the predictability and flexibility of

---

<sup>5</sup>An air traffic control sector should not be made up of several disconnected volumes of airspace.

<sup>6</sup>Flights following standard routes should not exit and re-enter the same control sector.

today’s airspace management in Europe, in a pre-tactical or tactical context<sup>7</sup>. The initial difficulty of the airspace partitioning problem is highly reduced by considering only sector recombinations within each managerial unit, using only operationally valid ATC sectors listed in the en-route<sup>8</sup> air traffic centre’s database. A realistic workload prediction model is proposed, relying on relevant air traffic complexity metrics and using a neural network trained on historical data. Our previous works dealt with the selection of the relevant metrics (see section 3). An initial version of our airspace configuration algorithm was given in [36], using a simple exhaustive tree search method for local sector recombinations. In this article, a *branch & bound* algorithm was used to reconfigure the whole airspace when necessary. Some preliminary results were presented in [37]. The current article gives full details of the model and algorithm hybridizing the neural network for workload prediction and the *branch & bound* that computes airspace partitions of minimum cost, balancing the workload as well as possible across the ATC sectors. In addition, we have introduced the possibility of posting constraints on the maximum number of ATC sectors in an airspace partition.

The rest of the paper is organized as follows: section 2 describes the air traffic management context and related works. Section 3 gives a short description of the methods we used to select the most relevant complexity metrics. Section 4 details the neural network used for workload prediction. The tree search algorithms that build optimal combinations of airspace modules are introduced in section 5, and the iterative algorithm that builds the full opening schedule for a whole day of traffic is presented in section 6. The experimental setup is described in section 7. Some results are shown in section 8, and section 9 concludes and gives the perspectives of future research and potential applications in the operational field. A glossary of terms can be found in the appendix.

---

<sup>7</sup>In the European air traffic flow management process, *pre-tactical* operations usually take place the previous day (sector opening schedules, preliminary flow regulations plan, etc.), and *tactical* operations take place in real time, or up to a few hours before flights actually enter the airspace.

<sup>8</sup>*En-route* air traffic management deals with traffic that follows routes between its origin and its destination, as opposed to *terminal* air traffic management, which deals with traffic near airports.

## 2. Background and related works

### 2.1. *The air traffic management and airspace configuration context*

Let us start with a short description of the current air traffic management system in Europe. We have seen in the introduction how elementary airspace modules can be combined so as to form control sectors. The airspace configuration may change during the day, depending on the controller workload. However, it is not always possible to split an ATC sector into several smaller sectors when the workload increases, either because the initial sector is composed of a single module or because there are not enough air traffic controllers on duty. This may lead to potentially dangerous situations when the traffic within a control sector is too heavy and too complex to be safely handled by the human operators.

Such situations are avoided by re-routing or assigning departure delays to aircraft that plan to enter the sector in the overloaded time interval. In Europe, the departure slots are allocated by the Central Flow Management Unit (CFMU). The Flow Management Position (FMP) operators of each air traffic control centre try to anticipate future overloads in their airspace. When necessary, flow regulations, which are used as input to the CFMU slot allocation algorithms assigning ground delays to aircraft, are enforced.

In some air traffic control centres, like the five French en-route centres, pre-tactical sector opening schedules are built by the FMP operators one or two days ahead. In France, the current method to build such schedules is fairly simple. A set of the usual airspace configurations is filed in a database. The FMP operator chooses among them the ones he (or she) thinks are the most adequate for each time period. The day is divided into fixed periods of usually one hour and no sliding time window is used. Candidate configurations are empirically assessed by counting the flights entering each ATC sector and comparing that count to pre-defined threshold values (*sector capacities*).

These pre-tactical schedules are unrealistic, partly because they rely on an estimated traffic demand, but also for other reasons: first, counting entering flights is not sufficient to model the controllers' actual workload, and second only a small subset of pre-defined configurations is considered. The current method directly results from former procedures, when flights were counted by hand.

As a consequence, the FMP schedules are not actually used to forecast future overloads. Instead, the FMP and CFMU operators rely on their past

experience of similar traffic situations to enforce flow regulations on specific airspace boundaries, entry points, or airspace volumes<sup>9</sup>. The causal relationship between the slot allocation based on these regulations and the actual workload experienced by the controllers in real time is not clearly established. A more accurate assessment of future workload and a better forecast of future airspace configurations could certainly improve the predictability of the European air traffic management system.

So far, we have only described the French and European air traffic management, which is the context of our study. In the United States, the context is largely similar, although air traffic management is more concerned with convective weather problems. Many studies focus on the dynamic adjustment of the airspace structure to traffic flow reroutings caused by severe weather conditions. It is expected that more flexible boundaries would allow a more efficient use of airspace and increase overall capacity. In [30], pre-defined scenarios of airspace sectorisations associated with traffic rerouting scenarios are proposed as a short-term improvement to the current practices. A more dynamic re-sectorisation with flexible boundaries is envisioned in future operational concepts ([31], [32], and some SESAR<sup>10</sup> *Operational Improvement* steps). It is expected that moving the sector boundaries in real-time to adapt to the traffic demand would increase the capacity and efficiency of the ATM (Air Traffic Management) system. The actual capabilities and potential benefits of this new operational paradigm are still largely unknown at this early stage, however. There is also some concern that unlimited flexibility in the sectors boundaries would lead to a loss of situational awareness by air traffic controllers (see discussion and literature review in [33]).

In the rest of this paper, we stick to the current context where the airspace is divided into pre-defined elementary modules.

## 2.2. Related works on airspace design and management

Current research on airspace configuration is manifold and may deal with strategic airspace partitioning, pre-tactical sector opening schedules, or tactical airspace management. Several methods have been tried out, with fairly

---

<sup>9</sup>For example: no more than N flights per hour over point P for flights originating from geographic area Z

<sup>10</sup>SESAR: Single European Sky ATM Research Programme, where ATM stands for Air Traffic Management.

different definitions for "workload". In [12], Basu et al. used geometric algorithms to design sector boundaries so as to balance a macroscopic workload (i.e. cumulated traffic over a whole day) across the sectors. The sectorisation problem was addressed by Delahaye et al. as a graph partitioning problem ([7]), and solved with evolutionary algorithms. Workload was an aggregation of conflict workload, coordination workload, and monitoring workload. A graph representing the air routes was partitioned so as to minimize coordinations between sectors, under various constraints (connectivity, convexity). Klein ([8]) designed airspace boundaries for the American air route traffic control centres by growing hexagonal cells from initial seed locations, using an equalized traffic mass metric. Bichot et al. ([13]) worked on today's actual sectors and tried to partition the European airspace into optimally balanced airspace blocks (air traffic control centres), considering inter-sector traffic flows and using graph partitioning methods (standard methods or a new fusion-fission meta-heuristic). These studies are focused on strategic airspace design.

In [6], Delahaye et al. addressed the dynamic airspace sectorisation problem for tactical operation, although only with mock-up sectors. The graph model and workload definition are the same as in their other works described above. This same problem was addressed by Tran Dac, Baptiste, and Duong ([10], [9]) using a Kernighan-Lin heuristic combined with Constraint Programming techniques. Note that the airspace partitioning problem is highly combinatorial in its initial formulation: the number of partitions of a set of  $n$  elements into  $k$  subsets is the Stirling number of the second kind. The introduction of connectivity constraints (an ATC sector should be a single geographic area, not composed of several connected components), convexity constraints (a flight following a route should not enter the same sector twice), or operational constraints (choose from only identified operational sectors) reduces this difficulty.

A more realistic operational context was considered by Verlhac and Manchon ([4]) of Eurocontrol, who applied mix integer programming techniques to improve the pre-tactical planning of sector configurations in Europe, using actual sectors. Only a small subset of pre-defined usual configurations was considered. The traffic load was assessed by counting the flights entering the sector in a one-hour time window, like in the current FMP schedule. This is clearly not sufficient to model the actual controller workload: for example, a flow of 40 flights per hour may well be below the sector capacity (maximum value allowed for this flow), but may not be acceptable in terms of actual

workload if all the flights entered the sector in the first 15 minutes of the hour.

In [34], we proposed several algorithms exploring all combinations of airspace modules in a realistic context, using the same variables (number of flights entering the sector in a given time window) and thresholds (sector capacities), as well as the same constraints (number of controllers on the duty schedule) as the French Flow Management Positions (also used in [4]). Standard tree search methods proved efficient when considering only operationally valid ATC sectors, i.e. those defined in the air traffic control centre database. An evolutionary algorithm was also proposed as an alternative, in case a wider range of airspace modules and ATC sectors was to be considered, and possibly larger geographic areas. The results were not realistic enough (see [35]), however, as the flow variables do not model the workload with enough accuracy. The conclusion was that a better assessment of the controller workload was needed.

Much research has been undertaken on traffic complexity metrics (see next section) with the aim of capturing the main features explaining the air traffic controller workload. Such metrics were used by Flener et al. to model controller workload in [11]. Tactical reroutings and flight profile modifications were used to balance traffic complexity across several sectors, for tactical multi-sector planning purposes. However, no airspace reconfiguration was proposed.

We propose to forecast the controller workload in each ATC sector, as well as the airspace configuration of the whole centre airspace, with the aim of improving the current pre-tactical or tactical air traffic management operations. Currently, each controller is qualified only for operations in a given geographic area (qualification zone<sup>11</sup>). As a consequence, the ATC sectors of a qualification zone cannot be combined with sectors from another qualification zone. In addition, we only consider operationally valid ATC sectors, listed in the en-route air traffic centre’s database. With these restrictions and in the current context, the airspace partitioning problem is much less combinatorial and standard tree search methods can be used to find optimal combinations of elementary airspace modules, balancing the controller workload as well as possible across the ATC sectors.

The next section details how the relevant air traffic complexity metrics

---

<sup>11</sup>A qualification zone is usually the whole airspace of an air traffic centre, or half of it.



that model the controller’s actual workload were selected.

### 3. Selection of the relevant complexity metrics

The selection process that allowed identification of the most relevant complexity metrics has already been described in previous publications ([2], [3]). Let us just recall the main steps of this process.

A set of relevant variables is usually selected by maximizing the correlation of the chosen metrics with a quantifiable dependant variable assumed to represent the actual controller workload. Many proposals have been made in past studies on the choice of dependent variables: physical activity ([23]), physiological indicators ([24], [25]), simulation models of the controller’s tasks ([26], [27], [20]), or subjective ratings ([18], [21], [19]).

For our problem, we made the basic assumption that past decisions to reconfigure the airspace by splitting or merging the ATC sectors were statistically related to the controllers’ actual workload<sup>12</sup>: ATC sectors can be split into smaller sectors when the workload is too heavy, or merged with other sectors when the workload is too light. They can be operated normally otherwise. Other, more complex, recombinations are also possible. The sector status (*split*, *normal*<sup>13</sup>, or *merged*) was chosen as the dependent variable quantifying the actual workload, when the sector was observed to be in one of these three states<sup>14</sup>. This variable has the advantage of being directly related to our problem. In addition, it does not require a heavy experimental setup to collect the data: it is already available in large quantities from the air traffic control centres archives.

The candidate explanatory variables were complexity metrics chosen from

---

<sup>12</sup>Note that this may not always be the case for all Air Navigation Service Providers. Some control centres are far less flexible in the way they merge or split sectors, and strictly follow a pre-defined duty roster. However, in the chosen context (France) and in many countries, the airspace configuration is highly flexible and is adapted in real-time.

<sup>13</sup>Some other terms (such as *operational*, *manned*, or *armed*) may be found in previous papers for this status which denotes that the sector is, or could be, operated in its normal workload domain.

<sup>14</sup>At a given time of the day an ATC sector, defined as a group of airspace modules, may be split, merged, normally operated, or be in a nondescript state where, for example, some of its modules are part of a larger sector, and the rest are operated normally, or split, or merged into another sector. In such cases, the observed state cannot be interpreted as an indication of the actual controller workload.

the literature (see [15], [16] and [17] for a review), and included aircraft count, proximity-related and conflict-related<sup>15</sup> metrics, flow metrics, track or speed disorder metrics, etc. Several methods have been used in past studies to correlate such metrics to chosen dependent variables: linear ([18]) or logistic ([19]) regression, cross-sectional time series analysis ([20]), and neural networks ([21]). We used a neural network, although not as a regression method but as a classification method (see details in the next section).

For workload modelling purposes, the metrics were computed from recorded aircraft trajectories ("radar tracks") and from the sector geometry. Later on in this paper, when trying to forecast the workload, we use trajectories simulated from flight plans<sup>16</sup>. Flight planning errors (cancellations, missing flight plans, short-notice departures, etc.), as well as navigation errors, and controllers' actions – for example vectoring instructions to avoid conflicts with other trajectories – may introduce forecasting errors and biases on the metrics values. These issues are further discussed in section 7 and in the conclusion. Let us just say that we can expect that the biases due to the controllers' actions on the traffic should remain relatively small. The influence of the differences between planned traffic and actual traffic on the quality of the airspace configuration forecast is covered in subsection 8.2.

The metrics selection process started with a principal component analysis that reduced the dimensionality of the initial set of candidate metrics. The most relevant components were selected using Akaike's AIC<sup>17</sup> ([2]). Schwartz's Bayesian Information Criterion (BIC) was also used as an alternative in [3] when selecting the best individual metrics in each relevant principal component. The use of such information criteria avoids the bias due to the increasing number of model parameters (the neural network weights) when considering subsets of input metrics of increasing sizes.

Among the initial 28 explanatory variables chosen from [18], [21], [28], [29] and other sources, the 6 most relevant variables were the sector volume  $V$ , the number of aircraft within the sector  $Nb$ , the average vertical speed

---

<sup>15</sup>In the French upper airspace, and in many other European countries, the horizontal separation between two aircraft should be at least 5 nautical miles, unless their vertical separation is at least 1000 feet. A "conflict" occurs when it is expected that the future relative positions will not achieve this separation.

<sup>16</sup>A flight plan is sent by the airline operator before take-off, describing the planned route, requested flight levels, and other basic flight intentions.

<sup>17</sup>AIC: "An Information Criterion"

*avg\_vs*, the incoming flows with time horizons of 15 minutes and 60 minutes ( $F_{15}$ ,  $F_{60}$ ), and the number of potential trajectory crossings with an angle greater than 20 degrees (*inter\_hori*<sup>18</sup>).

Let us note that these metrics are fairly basic, compared to others that can be found in the initial set of candidate metrics. This does not mean that all these other metrics were not relevant. Many of them were redundant with the basic metrics that were found to be most correlated to the sector status. This is probably due to the fact that our observation of the actual workload is relatively macroscopic, as we only have three levels of workload: "too low" when the sector is merged, "normal" when it is operated, or "too high" when it is split into smaller sectors operated separately. This does not allow very small variations of the actual workload, within the normal domain of operation, to be captured with more sophisticated metrics. This is enough, however, for our purpose of predicting airspace configurations.

In [38], we improved this workload prediction model by smoothing the input metrics: the raw metrics showed high variations in time, causing too frequent airspace reconfigurations (see [36]). Another promising approach, which is left for further work, would be to consider the input metrics as time series.

To conclude on the metrics selection, let us say that our final model with its 6 variables could probably be improved by considering the sector volume that is actually available at every moment of the day to air traffic controllers, taking account of convective weather or military activity, instead of the overall volume. Other metrics related to the sector complexity (see [41]) could also be tried, as well as other traffic complexity metrics that have not yet been implemented in our libraries.

#### 4. A neural network for workload prediction

Let us now detail the model that is used for workload prediction, based on the assumption that previous decisions to split or merge ATC sectors in the past were statistically related to the controllers' actual workload.

---

<sup>18</sup>This *inter\_hori* metric with its 20-degree threshold comes from a metric defined by the Eurocontrol Performance Review Unit with the aim of comparing the performance of ATC centres. It accounts for the fact that traffic following parallel routes is less complex to handle than traffic with crossing trajectories.

Considering the nature of the observed data, we chose a "1 of 3" encoding for the desired target representing the actual sector status:

- $d = (1, 0, 0)^T$  when the sector is *merged* with other sectors into a larger ATC sector (low workload),
- $d = (0, 1, 0)^T$  when the sector is in its *normal* domain of operation (acceptable workload),
- $d = (0, 0, 1)^T$  when the sector is *split* into several smaller control sectors (excessive workload).

Observed patterns where the sector is in a state other than one of the above<sup>19</sup> were discarded.

The explanatory variables are the relevant complexity metrics  $\{V, Nb, avg\_vs, F_{15}, F_{60}, inter\_horiz\}$  (see section 3), normalized by subtracting the mean value and dividing by the standard deviation. Let us denote  $x = (x_1, \dots, x_i, \dots, x_6)^T$  this vector of input variables, where  $T$  denotes the transpose operator.

Our aim is to determine from any given measure of  $x$  if the workload in the considered ATC sector is low, normal, or too high. This is typically a classification problem, that we chose to address with a simple feed-forward network with one hidden layer, as a first approach. The reader can refer to [42] and [43] for an extensive presentation of neural networks for pattern recognition, or [44] for a shorter review. The results presented in this paper were produced with a network having 6 units on the input layer, 15 units on the single hidden layer, and 3 units in the output layer. Let us denote  $y = (y_0, y_1, y_2)^T$  the output vector of the neural network. The neural network equation is written as follows:

$$(y_0, y_1, y_2)^T = \Psi\left(\sum_{j=1}^{15} w_{jk} \Phi\left(\sum_{i=1}^6 w_{ij} x_i + w_{0j}\right) + w_{0k}\right) \quad (1)$$

where  $\Psi$  is the softmax function:

$$\Psi_k(z_k) = \frac{e^{z_k}}{\sum_{m=1}^3 e^{z_m}} \quad (2)$$

---

<sup>19</sup>For example when, for sector AB, A is merged within a sector AC and B within another sector BD.

$\Phi$  is the sigmoid logistic function:

$$\Phi(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

The error function we are trying to minimize when training the network is the cross-entropy function:

$$E(w) = - \sum_{n=1}^N \sum_{k=1}^C d_k^{(n)} \ln\left(\frac{y_k^{(n)}}{d_k^{(n)}}\right) \quad (4)$$

with  $C = 3$  classes, and considering  $N$  patterns for training, and where  $d = (d_0, d_1, d_2)^T$  is the desired target vector. The cross-entropy function is simply the log-likelihood

$$L(w) = - \sum_{n=1}^N \sum_{k=1}^C d_k^{(n)} \ln(y_k^{(n)})$$

from which was subtracted the minimum log-likelihood

$$L_{min} = - \sum_{n=1}^N \sum_{k=1}^C d_k^{(n)} \ln(d_k^{(n)})$$

reached when  $y_k = d_k$  (see [42], pp. 237-238). See also [45] for justifications on the choice of the cross-entropy penalty function for classification problems, and on the natural pairing of the softmax activation and cross-entropy penalty functions.

A standard back-propagation method ([48]) was used to compute the gradient of the error function, and a quasi-Newton method (BFGS, namely) was used to minimize the error. These methods were implemented in a library written in Objective Caml.

The output vector can be interpreted as a vector of posterior probabilities of class-membership:  $y_0$  can be seen as the probability  $p(C_{low}/x)$  that the ATC sector falls in the "merged" class (low workload) when the measured air traffic complexity vector is  $x$ , and similarly for  $y_1$  and  $y_2$ , with classes  $C_{normal}$  and  $C_{high}$  respectively. Using an abbreviated notation, we shall denote  $y = (p_{low}, p_{normal}, p_{high})^T$  the output vector in the rest of this paper, so as to clarify the nature of the neural network output.

As we are necessarily in one of the above three cases (low, normal, or excessive workload), the sum of the three probabilities  $p_{low}$ ,  $p_{normal}$ , and  $p_{high}$  is always 1, which is ensured by the use of the softmax function  $\Psi$ .

If interpreting the network output in terms of probabilities related to the controller workload is rather straightforward, this is not so easy concerning the function  $\Phi$  and the cross-entropy<sup>20</sup>. As stated by Jordan and Bishop ([44]), neural networks may best be viewed as a class of algorithms for statistical modelling and prediction. A neural network is simply a statistical model made of a mixture of adaptive functions. The functions  $\Phi$  map each a portion of the multi-dimensional space of input variables (the air traffic complexity metrics). The model is adaptive in the sense that the function's weights  $w$  are tuned during the training phase so as to fit the observed data as well as possible, minimizing the error between the computed output and the observed sector status (using the cross-entropy in our case, which is just an expression of the log-likelihood).

This combination of functions, when using a network with at least one hidden layer of sigmoidal units, has the interesting property of approximating any decision boundary to arbitrary accuracy (see [42], p. 130 for bibliographic references on this property). This provides universal non-linear discriminant functions, and allows modelisation of posterior probabilities of class-membership for classification problems. This is why this class of algorithms was preferred to linear or logistic discrimination techniques<sup>21</sup>.

This does not mean that those methods or other statistical models would not perform well. For example, an ordered logit model like the one used in [46] could be well adapted to our classification problem. However, although it would be interesting, it is not in the scope of this paper to make such a comparative study on various methods. This could be addressed in future works

## 5. Tree search algorithms for airspace partitioning

Let us now see how to partition the airspace, using the workload indications given by the neural network to assess the optimality of candidate

---

<sup>20</sup>Note that this kind of domain-related interpretation would not be easy for other classes of algorithms either, such as polynomial regression.

<sup>21</sup>Note that linear techniques can be seen as networks with linear units, and logistic discrimination as a network with sigmoidal units but with no hidden layer

configurations. In order to find an optimal airspace configuration, one needs to explore all possible partitions of elementary airspace modules, while restricting them to operationally valid configurations. As an example, let us imagine an ATCC airspace divided into 5 modules  $\{1, 2, 3, 4, 5\}$ . Let us say that the possible combinations of modules into ATC sectors are  $a = \{2, 3\}$ ,  $b = \{3, 4\}$ ,  $c = \{4, 5\}$ ,  $d = \{1, 5\}$ , and  $e = \{1, 2, 3, 4, 5\}$ , to which we may add each module operated as a single sector. A valid configuration shall be a partition of the set of airspace modules into ATC sectors: for example  $[(\{1\}, s); (\{2, 3\}, a); (\{4, 5\}, c)]$ , or  $[(\{1, 5\}, d); (\{2\}, s); (\{3, 4\}, b)]$ , where  $s$  is the generic notation for a singleton.

Finding an optimal partition requires two things: first a mechanism to build valid configurations and, second, some cost functions to evaluate and compare the candidate configurations. The next subsections detail the cost of a configuration and the tree search algorithms.

### 5.1. Cost of a configuration

To evaluate an airspace configuration, we consider the number of ATC sectors and the values of probabilities  $p_{low}(s_i, t)$ ,  $p_{normal}(s_i, t)$ , and  $p_{high}(s_i, t)$  issued by the neural network for each sector  $s_i$  in the configuration  $c = \{s_i/i \in [1, m]\}$ , where  $t$  is the time. As illustrated in our previous example, a sector  $s_i$  is a group of airspace modules labelled by a sector name.

Considering configuration  $[(\{1\}, s); (\{2, 3\}, a); (\{4, 5\}, c)]$  of this example, it would be perfectly balanced in terms of workload if we had  $(p_{low}, p_{normal}, p_{high})$  equal to  $(0, 1, 0)$  for all ATC sectors  $a, b$ , and  $s = \{1\}$ . Such ideal situations rarely occur in reality, however, where  $p_{normal}$  may be less than 1 in several sectors, and where overloads and under-loads<sup>22</sup> may occur, sometimes in the same configuration.

Before defining the cost of a configuration, let us state the objectives we are trying to reach, in decreasing order of priority. The first objective is to satisfy the constraint on the maximum number of ATC sectors, when such a constraint is posted. This constraint is useful when, for some reason, some controller working positions are not available, or when there are not enough controllers to operate more than a maximum number of sectors. The second objective is to avoid overloads, which are potentially dangerous situations.

---

<sup>22</sup>In our context, a sector will be said to be *overloaded* (resp. *under-loaded*) when the probability  $p_{high}$  (resp.  $p_{low}$ ) is higher than the two others.

The third objective, supposing there are no overloads, is to open as few sectors as possible. Finally, we would like to balance the workload as well as possible across the ATC sectors, by minimizing under-loads and selecting configurations where the  $p_{normal}$  probabilities are as close as possible to 1.

In some previous papers, the cost was expressed as a number where the leftmost digits were assigned to sub-costs with the highest priorities. Note however that there is no need to express the cost as a real number, as long as we are able to compare two configurations. In the current paper, the cost is defined as a record structure, whose attributes contain sub-costs. This removes some numerical problems that were encountered when restricting sub-cost values to a chosen number of digits.

The sub-costs assigned to a configuration  $c = \{s_i/i \in [1, m]\}$  are the following, in decreasing order of priority:

- the number of ATC sectors above the maximum allowed number, when such a constraint is defined.
- the maximum value of probability  $p_{high}(s_i, t)$  across overloaded sectors, except when this value is 1<sup>23</sup> for the two configurations we are comparing. In this case, this criterion alone does not allow highly overloaded configurations to be compared, and we will simply use the number of aircraft in the sector instead<sup>24</sup>,
- the total number of ATC sectors in the configuration,
- the maximum value of probability  $p_{low}(s_i, t)$  across under-loaded sectors,
- the maximum value of  $(1 - p_{normal}(s_i, t))$  across normally loaded sectors.

When comparing two candidate configurations, we simply compare these sub-costs in decreasing order of priority. Now let us describe the algorithms that build valid airspace configurations and select the optimal one, using the above cost comparison.

---

<sup>23</sup>or close enough to 1, in practice

<sup>24</sup>Note that such situations mainly occur when the number of ATC sectors is highly constrained.



### 5.2. Exhaustive tree search

Figure 1 illustrates a tree search that builds all valid combinations of airspace modules, on the same example as before. Here, each tree node is a list of couples where the first element is a group of modules, and the second is the list of valid ATC sectors that contain these modules, but no modules from the other groups of the node.

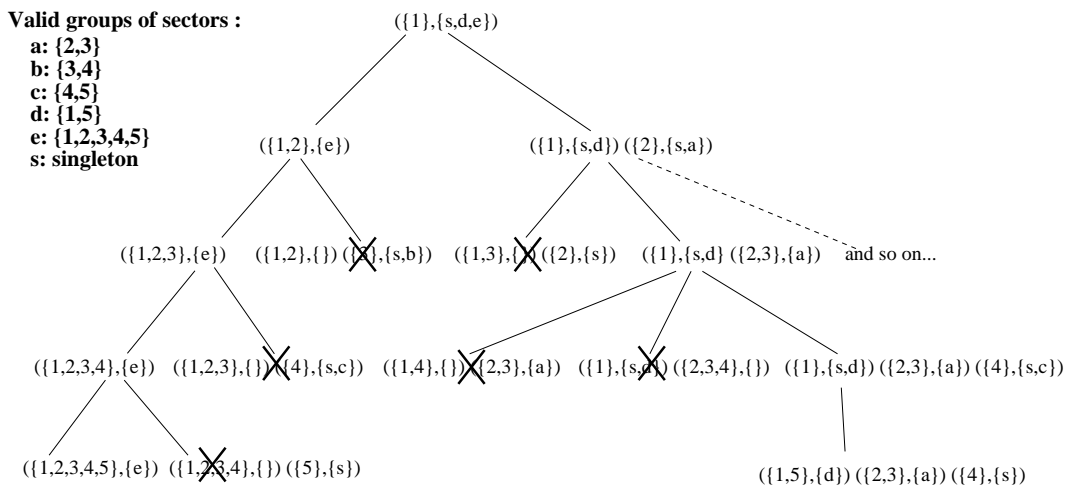


Figure 1: Exhaustive tree search algorithm for airspace partitioning.

Let us illustrate how this tree is built. The root is a node containing a single group  $\{1\}$ , made of only one module. The possible ATC sectors for  $\{1\}$  are  $s = \{1\}$ ,  $d = \{1, 5\}$ , and  $e = \{1, 2, 3, 4, 5\}$ . From this point, we may add a second module, say 2, either in the same group already containing 1 (left branch), or in a new group (right branch). For the left branch  $(\{1, 2\})$ , the only ATC sector containing both 1 and 2 is  $e$ . For the right branch  $(\{1\}$  and  $\{2\}$  as separate groups), the list of possible ATC sectors for  $\{1\}$  now contains only  $s = \{1\}$  and  $d$ . The ATC sector  $e$  is no longer a valid possibility: it contains 1, but also 2 which is already in a separate group. For this group  $\{2\}$ , the valid ATC sectors are  $s = \{2\}$  and  $a = \{2, 3\}$ .

While exploring this tree, it may happen that the list of possible ATC sectors becomes empty. This is the case for example with the node made of groups  $\{1, 2\}$  and  $\{3\}$  (when developing the left branch). There is no valid ATC sector containing both 1 and 2 which does not also contain 3. There is no need to develop further this node, as it will never lead to an operationally valid airspace partition.

The selection of the best airspace partition is made according to the cost criteria defined in section 5.1, which depend on the number of ATC sectors and the values of the probabilities  $p_{low}$ ,  $p_{normal}$ , and  $p_{high}$  issued by the neural network for each control sector of the considered airspace configuration.

The exhaustive tree search may be computationally intensive, or even not feasible in a reasonable amount of time, when the number of elementary airspace modules and possible ATC sectors is too high.

### 5.3. Branch & Bound

Let us describe the *Branch & Bound* algorithm used to build airspace configurations. The main difference with the exhaustive tree search is that, for each node being explored, a lower bound of the costs of all leaves (full configurations) that can be reached from that node is computed. If this lower bound is higher than the cost of the best configuration found so far, then there is no need to continue the search from this node.

Figure 2 shows how the tree is explored, on the same example as in the previous subsection. In the beginning (steps 1 to 4 in the figure), the search is performed like in the exhaustive search, until a first valid configuration is reached (step 5). Afterwards, the best configuration is memorized, together with its cost, and the following nodes are evaluated and compared to this best configuration.

The cost of a node is similar to the cost of a configuration (see subsection 5.1), except that we consider the best possible choice from among all the control sectors associated with each group in the node, so as to provide a lower bound of all full configurations that can be obtained from that node.

In the example shown in Figure 2, the cost function for configurations is denoted  $cost$ , whereas the cost function for a node is denoted  $cost'$ . When evaluating the cost of node  $[(\{1\}, \{s, d\}); (\{2\}, \{s, a\})]$  in step 6, we shall take the best possible choice for group  $\{1\}$ , from among singleton  $s = \{1\}$  and valid group  $d = \{1, 5\}$ , and for group  $\{2\}$ , we will choose among singleton  $s = \{2\}$  and valid group  $a = \{2, 3\}$ . Let us say that the best possible choices are  $d$  for group  $\{1\}$  and  $a$  for group  $\{2\}$ . The node cost  $cost'([( \{1\}, \{s, d\}); (\{2\}, \{s, a\})])$  is then equal to  $cost([d; a])$ , the cost of a virtual, and incomplete, configuration containing only these two ATC sectors.

Note that choosing the best ATC sector for each group in the node may lead to virtual configurations that are not airspace partitions. In step 8 for example, a virtual configuration with  $d = \{1, 5\}$ ,  $a = \{2, 3\}$  and  $c = \{4, 5\}$  is

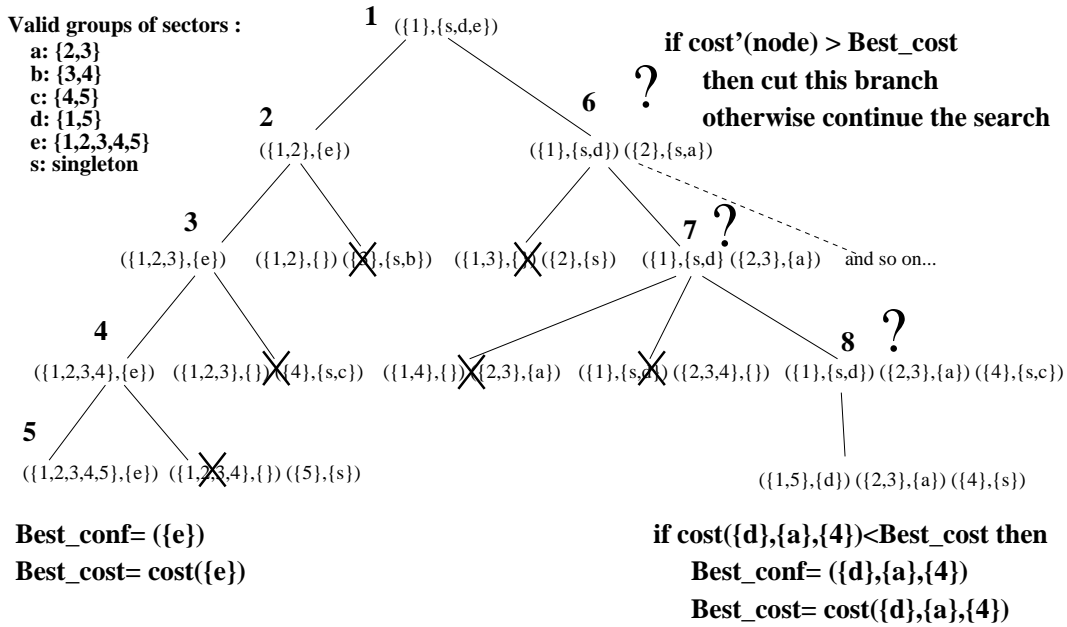


Figure 2: Branch & Bound algorithm for airspace partitioning.

not a valid airspace partition as module 5 is in both sectors  $d$  and  $c$ . However, this heuristic ensures that the cost of a node is always lower than the cost of any valid configuration that can be reached from that node.

## 6. Predicting airspace configurations throughout the day

### 6.1. Description of the algorithm

So far, we have seen how a neural network could be used to assess the controller workload for any given ATC sector, and how to partition the airspace so as to balance this workload as well as possible among all sectors, at any given time  $t$ . Now, let us see how to build an airspace configuration schedule for a whole day of traffic.

Finding an optimal airspace partition of the whole airspace at every moment of the day seems the most straightforward solution, but it would lead to a succession of drastically different configurations in short periods of time. In reality, the airspace is reconfigured around 30 times a day (for French airspace), and usually with relatively minor changes from one configuration to another. The reason is that transferring one or more airspace modules

from one controller to another must be done safely, ensuring that the receiving controller does not miss any potentially dangerous situations in the new traffic and the new airspace sector he will have to handle.

---

1. Initial configuration (t=0):  
1 ATC sector  $\leftarrow$  all elementary airspace modules
  2. At each time step (1 minute):
    - Decide if airspace must be reconfigured  
 $\rightarrow$  check workload in each ATC sector
    - If so, reconfigure airspace:
      - select modules to recombine,
      - explore all combinations,
      - select configuration with minimum cost.
- 

Figure 3: Iterative algorithm for airspace sectors opening schedule.

So it was decided to mimic the actual behaviour of control room managers as well as possible, keeping the current airspace configuration as long as it remained acceptable. Figure 3 describes the main loop of the chosen algorithm. The current airspace configuration is checked at every time step. If the workload in some sectors of the current configuration is really too low ( $p_{low}$  close enough to 1) or too high ( $p_{high}$  close enough to 1), then a reconfiguration is triggered. Depending on a user-preferred option, either only a few sectors are recombined, or the whole airspace is reconfigured.

Let us recall that the network was trained on observed sector statuses (merged, normally operated, or split). So, when a configuration change is triggered, one might interpret the network output as a recommendation to split or merge the triggering sector. This output is just a workload indication, though, and the tree search algorithm does not restrict itself to splitting or merging sectors: it recomputes a new optimal partition of the set of modules chosen for recombination.

As a result, the transition from one configuration to the next may show split sectors, merged sectors, or modified boundaries between two ATC sectors due to the transfer of one or several modules from one sector to the other.

There may be even more complex recombinations such as  $A, BC, D, E \rightarrow AB, CDE$  (where  $A, B, C, D$ , and  $E$  are airspace modules). Some of these complex recombinations may actually not be feasible in operation. In future work, we plan to optimize *sequences* of configurations while satisfying constraints on the transitions, allowing for example only one simple change per sector: split, merge, shift boundaries, or open a new sector made up of modules transferred from other sectors.

### 6.2. Decision criterion

In order to describe more formally the decision criterion triggering the configuration changes, let us consider our three probabilities ( $p_{low}, p_{normal}, p_{high}$ ) sorted by decreasing values  $p_1, p_2, p_3$ . The decision criterion is expressed as follows, depending on which probability is the highest:

- if  $p_1$  is the probability  $p_{low}$  that the workload in the sector is too low: the decision to reconfigure is taken only if  $1 - p_1 < \alpha$
- if  $p_1$  is the probability  $p_{high}$  that the workload is too high: we decide to reconfigure only if  $1 - p_1 < \beta$ , where  $\beta$  is a chosen parameter.

In the other cases, the sector does not trigger a reconfiguration. We choose different criteria for probabilities  $p_{low}$  and  $p_{high}$ , because we may need to be more reactive when the workload is increasing than when it is decreasing.

### 6.3. Choosing the sectors to recombine

When a reconfiguration is triggered, we can adopt different strategies when building the set of sectors to recombine. We can select either:

- only the triggered sectors ( $p_{high}$  or  $p_{low}$  close enough to 1),
- or the sectors for which  $p_{high}$  (or  $p_{low}$ ) is higher than the other two probabilities. This usually gives us a slightly larger set than with the previous criterion,
- or the previous set to which we add the neighbouring air traffic control sectors. This increases the chances of a better local recombination, especially for under-loaded sectors. These sectors cannot be recombined alone, whereas overloaded sectors can always be split as long as they are made up of several modules,

- or all the sectors. This allows an optimal partition for the whole airspace to be found.

The user can choose one of the above options, or a mix of local and full recombinations. In the latter case, the default behaviour is to make a local recombination (for example only sectors for which the probability  $p_{high}$  or  $p_{low}$  is the highest), except when there are more than 2 connected components (or a chosen number) in the set of ATC sectors to recombine. In this case, a full airspace reconfiguration is triggered. The rationale underlying this logic is that there are some cases when the limited recombination is not sufficient, for example when there are two under-loaded sectors that are not geographically adjacent.

Sectors can be recombined using either the exhaustive tree search for local recombinations only, or the *branch & bound* for both local and full reconfigurations. The results presented in this article are obtained with full reconfigurations with the *branch & bound*. Before presenting these results, let us detail the experimental setup.

## 7. Experimental setup

### 7.1. Discussion on the choice of input trajectories

The neural network was trained on actual aircraft trajectories (see details in the next subsection). When trying to forecast airspace configurations, the input trajectories were simulated from flight plans.

Choosing radar tracks to tune the workload model and flight planned trajectories to predict future workload may introduce a bias between modelling and forecasting, as the actual trajectories may have changed from the planned trajectories, due the controllers' actions. With radar tracks, trajectory conflicts are solved by the controllers before separation losses<sup>25</sup> actually occur, whereas this is not the case when using flight plans. So the proximity-related metrics relying on the distance between aircraft, as well as the conflict-related metrics, may be under-estimated when using radar tracks instead of flight planned trajectories.

---

<sup>25</sup>In the french upper airspace, and in many other European countries, the horizontal separation between two aircraft should be at least 5 nautical miles, unless their vertical separation is at least 1000 feet. A "conflict" occurs when it is expected that the future relative positions will not achieve this separation.

This may have slightly biased the selection of relevant metrics (see section 3), although the conflict-related metrics that were implemented in our initial set of metrics use relative directions and velocities, so potential conflicts should be correctly detected with enough anticipation, before trajectories are deviated by the controller. In any case, these biases should remain small, considering the relative influence of conflict-related metrics when compared to the aircraft count, the sector volume, and other metrics (see [2] and [3]).

So we use radar tracks for workload modelling, and flight planned trajectories for forecasting purposes, as we have no better guess on where the aircraft will be in the future. In future works, a mix of radar tracks and intended<sup>26</sup> trajectories may provide greater accuracy. After this discussion on the possible biases due to the chosen input traffic, let us return to the details of our experimental setup.

### 7.2. Training the neural network on historical data

For the results shown below, the neural network was trained on data samples from 2nd June, 2003, using recorded traffic ("radar" tracks) and sector opening archives from the five French air traffic control centres (ATCCs).

The data set used to train the neural network was built by measuring the relevant complexity metrics every minute of the day, in all ATC sectors of the five centres. The complexity metrics were computed from the aircraft trajectories and sector geometry. The training and test patterns were made from the input complexity metrics and from the observed state of each sector, *merged*, *normal* (i.e. actually operated), or *split*, coded as a state vector (see section 4). Data samples corresponding to a state where the sector is neither operated, nor split into several smaller sectors, nor collapsed within a larger sector were discarded, as we cannot deduce anything on the actual workload from such states.

In our training set, about 46% of the observed patterns fell in the *merge* class, 27% in the *normal* class, and 27% in the *split* class. Detailed results on the neural network outputs can be found in previous publications ([2], [3], [38]). Let us just say that the correct classification rates across all classes were

---

<sup>26</sup>"Intended trajectory" can be understood here as the baseline trajectory entered by the pilot in his flight management system (FMS), and from which he might deviate if instructed to do so by the controller so as to avoid other traffic. It may be different from the flight plan available in ground-based systems, or simply more accurate.

around 85% with a feed-forward neural network with 15 hidden units, when using smoothed input metrics (see [38]). Considering each class individually, we obtained 90% of correct classifications for the *merge* class, 68% for the *normal* class, and 93% for the *split* class. The results on the test set were quite consistent with those obtained on the training set. We have a feeling that these results could still be improved by considering the input metrics as time series and by using recurrent neural networks, but this remains to be done.

### 7.3. Choice of parameters

Our initial results with a straightforward decision criterion based on the highest probability, and with raw complexity metrics as input showed too frequent configuration changes during the day, although the evolution in the number of ATC sectors stayed fairly close to the actual airspace configurations.

The choice of parameters  $\alpha$  and  $\beta$ <sup>27</sup> for the decision criterion was discussed in [36]. The best compromise that was found was  $\alpha = 0.1$  and  $\beta = 0.3$ . These values mean that our algorithms are more reactive when the workload increases than when it decreases. This seems to reflect the controllers' actual behaviour.

Smoothing the input metrics over 30 minutes with a "moving average" method also improved the results for an air traffic control centre with large sectors such as Brest ATC centre<sup>28</sup>. We use these parameters in the results presented in this paper.

### 7.4. Assessing the airspace configuration on recorded trajectories

In addition to the neural network evaluation, we also need to assess the airspace configurations produced by our algorithms. A first step to validate our approach was to compute the complexity metrics from real recorded traffic ("radar tracks"), and compare the resulting configurations to the actual sector openings.

---

<sup>27</sup>An additional parameter  $\eta$  expressing a threshold on the difference between the two highest probabilities was also used in [36]. As this parameter was useless, considering the current values used for  $\alpha$  and  $\beta$ , it was abandoned.

<sup>28</sup>The smoothing strategy may vary from one centre to another: trials with centres where ATC sectors are smaller seem to show better results with a shorter time window, but no exhaustive results on this subject have been published yet.



Ideally, the computed schedules should reproduce the actual configurations recorded that day. However, there is a high variability in the decisions made by control room managers on *how* to reconfigure the airspace, which comes in addition to the variability of decisions on *when* to reconfigure. We may hope that our workload model could give an indication on when to trigger a reconfiguration and allow the tree search method to build realistic configurations, but our algorithms may not compute exactly the same configuration as in reality.

So we will mainly assess the realism of the computed schedule by comparing the number of ATC sectors in our computed configurations with the actual number of sectors open on the same day. Pearson’s correlation coefficient may give an indication of the linear correlation between the computed and the real number of sectors. However, this may not always be reliable<sup>29</sup> so we will also compute an *ad-hoc* ”dissimilarity measure”, which is the surface delimited by the two curves (computed and real number of ATC sectors operated during the day), divided by the surface between the  $x$ -axis and the curve of the real number of ATC sectors. With this measure, two identical curves have a dissimilarity 0 if they are exactly superposed. In addition, we will also consider the number of reconfigurations throughout the day, which should be close enough to the real one.

#### 7.5. Assessment of the airspace configuration forecasted from planned traffic

Radar tracks are available only *after* the aircraft have flown through the airspace. The proposed algorithms may be useful only if we are actually able to *forecast* airspace configurations *before* the aircraft enter the airspace.

For this purpose, predicted aircraft trajectories are simulated from the anticipated traffic demand (i.e. flight plans filed by the airline operators), using a fast-time simulator (see [39], [40]). The resulting airspace configuration schedule is then compared both to the prediction that was made the day before by flow management, and to the actual sector openings. Let us now see a few results, starting with the validation on historical data.

---

<sup>29</sup>The correlation coefficient between two equal variables  $x$  and  $y = x$  will be 1. Let us note however that this coefficient is not sufficient actually to measure how close we are to equality: the correlation coefficient between a variable  $x$  and another variable  $y = x + d$ , where  $d$  is a constant offset, will also be 1.

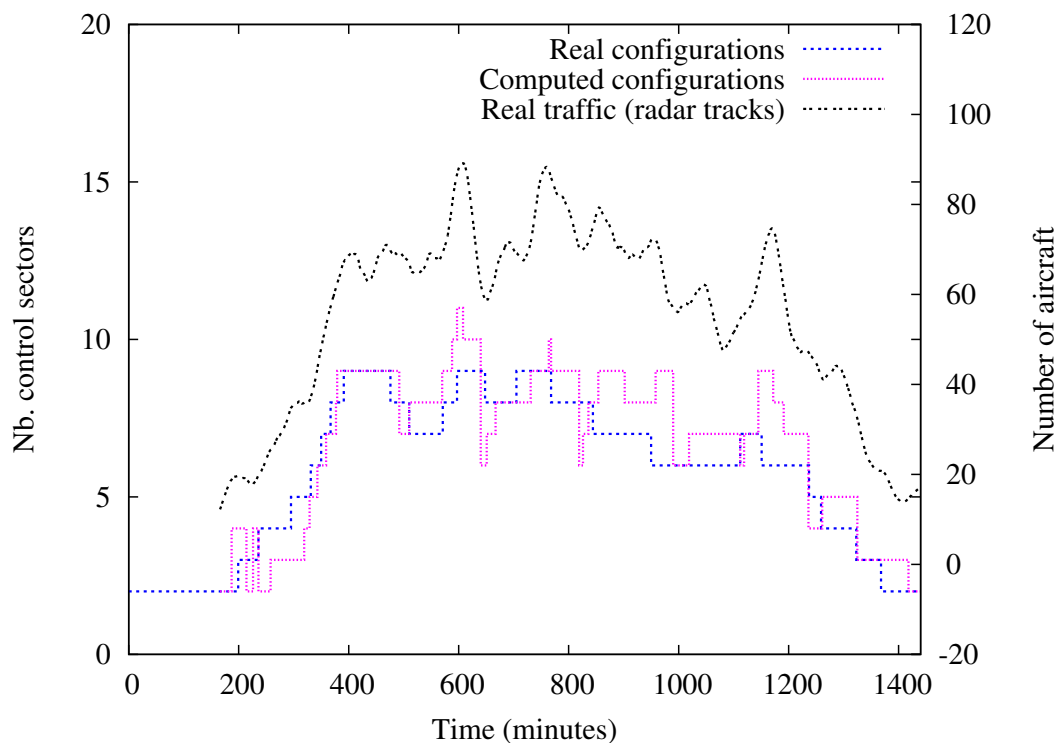


Figure 4: Computed *vs.* real configurations (Brest ACC, 6th June 2003), using real traffic as input.

## 8. Results

### 8.1. Comparison to real configurations, using recorded traffic

The model tuned on 2nd June was tested on two other days of traffic, using recorded radar tracks as input. Figure 4 shows the number of air traffic control sectors in the computed configurations, and the real number of ATC sectors that were open on 6th June 2003, in Brest air traffic control centre. The upper curve shows the number of aircraft within the centre boundaries, as an indication of the total traffic that day. Note that this curve is related to the  $y$  axis on the right, and that it has been shifted upward so as to make the figure more readable. The left  $y$  axis shows the number of sectors, and the  $x$  axis is the time, expressed in minutes after 0h00.

As shown on this figure, the computed number of sectors is fairly close to the actual number of sectors that were operated that day. The dissimilarity

measure is 0.191 and Pearson’s correlation coefficient is 0.91. Our algorithms found 46 configuration changes, whereas there were 33 in reality.

The total number of aircraft within the centre (upper curve) is given as a very crude indication of the overall workload. We can observe that the number of ATC sectors follows this curve more or less, although not always strictly. One should bear in mind that the traffic may be dispatched differently across the sectors throughout the day, and also that the number of aircraft is not the only factor of air traffic complexity and workload. Still, this traffic curve gives a good hint of the traffic variations throughout the day, and allows finer interpretations of the other curves.

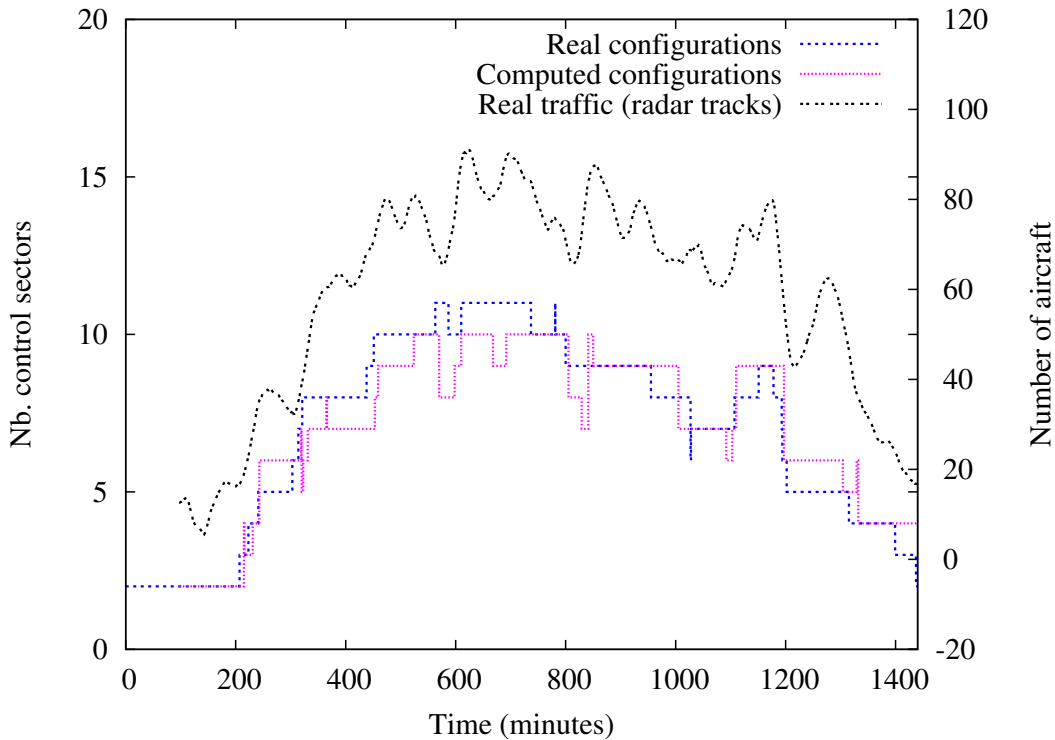


Figure 5: Computed *vs.* real configurations (Brest ACC, 7nd June 2003) using real traffic as input.

Figure 5 shows the same information, but for 7th June 2003. For that day, the algorithms show slightly better results, with a dissimilarity measure of 0.115 and a correlation coefficient of 0.94. There were 37 computed configurations, against 33 in reality.

The two figures, 4 and 5, illustrate the fact that the model only reflects the average behaviour of air traffic controllers and control room managers when they decide to split or merge sectors, or reconfigure otherwise. There are variations in this behaviour among the population. Some controllers are more experienced or more efficient than others and may consequently manage higher workloads, or there may be younger controllers completing the final period of their training under the supervision of fully qualified instructors.

### 8.2. Planned vs. real

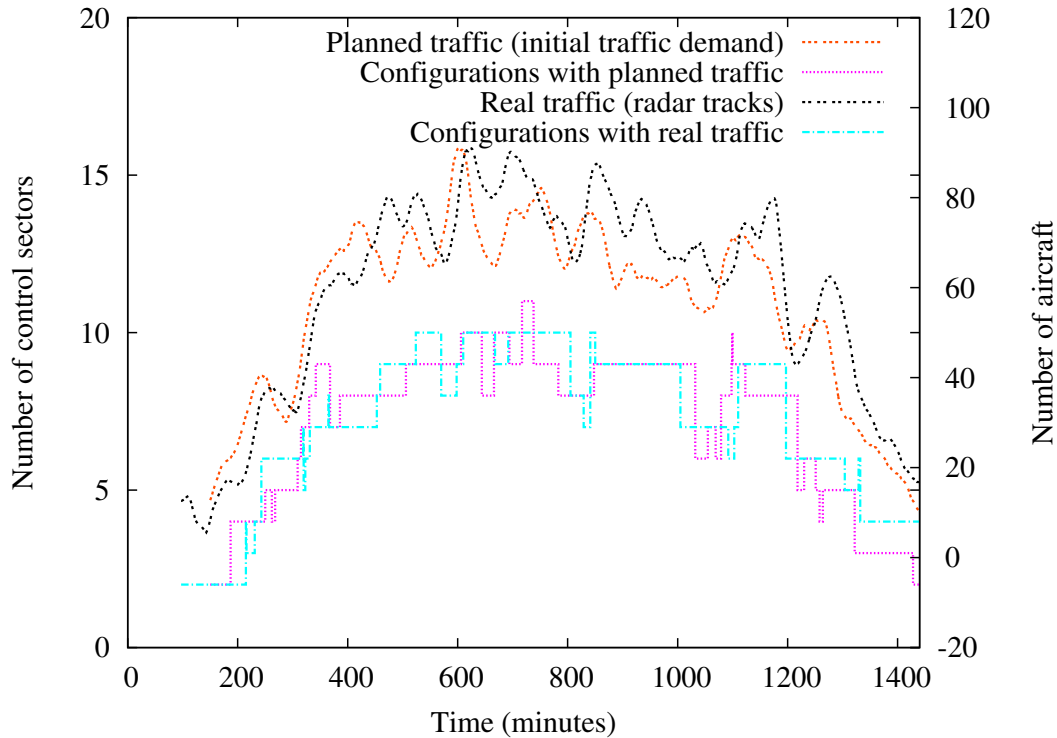


Figure 6: Initial traffic demand (flight plans) *vs.* real traffic (radar tracks), in Brest ACC on 7th June 2003.

Figure 6 shows the traffic in Brest airspace on 6th June 2003, and the number of ATC sectors in the configurations computed with either the real traffic or the simulated traffic as input. In the latter case, a fast-time<sup>30</sup> air

<sup>30</sup>The simulation of a whole day of traffic over France takes about one minute of com-

traffic simulator was used to compute the aircraft trajectories from the flight plans of the initial traffic demand. This is typically what might be done in a pre-tactical context, well before flights take off, when no radar tracks are available.

We can observe the differences between the planned traffic and real traffic by considering the two upper curves in Figure 6. Although it may be described as a statistical phenomenon ([47]), this difference between planned and real traffic is rather difficult to anticipate with great accuracy in a given sector at a given time. This leads to uncertainties when forecasting the complexity metrics used as input to the neural network.

Missing flight plans or short-notice departures may indeed lead to drastically under-estimated aircraft counts and other complexity metrics (see [22]), especially for departure sectors. Considering the results in Figure 6, this does not seem to be the case with our flight plan data and with the considered *en-route* ATCC. However, we still observe some differences that may stem from navigation uncertainties, re-routings, flight cancellations, etc. These uncertainties on planned traffic do have an influence on the quality of the forecast, although this influence is certainly somewhat lessened by the use of a moving average method to smooth the input metrics.

Flight planning uncertainties may be reduced by using a mix of radar tracks and flight intentions, if the proposed algorithms are to be used in a tactical context where complexity metrics are forecast 20-30 minutes ahead. In any case, the difference between planned traffic and actual traffic may be an important source of error. The quality of the prediction depends on the reliability of the 4D-trajectory forecast.

### 8.3. Comparison with the current forecast

Let us now compare our forecast, using planned traffic, with the actual prediction made on the same data by the flow management operators. We can see in Figure 7 that the schedule computed with planned traffic as input is much closer to the number of control sectors that were actually open that day than what was predicted by the FMP operators from the same flight plans that we used.

The dissimilarity metric measuring the difference between prediction and reality is 0.568 for the FMP schedule, whereas it is only 0.137 for our fore-

---

putation time

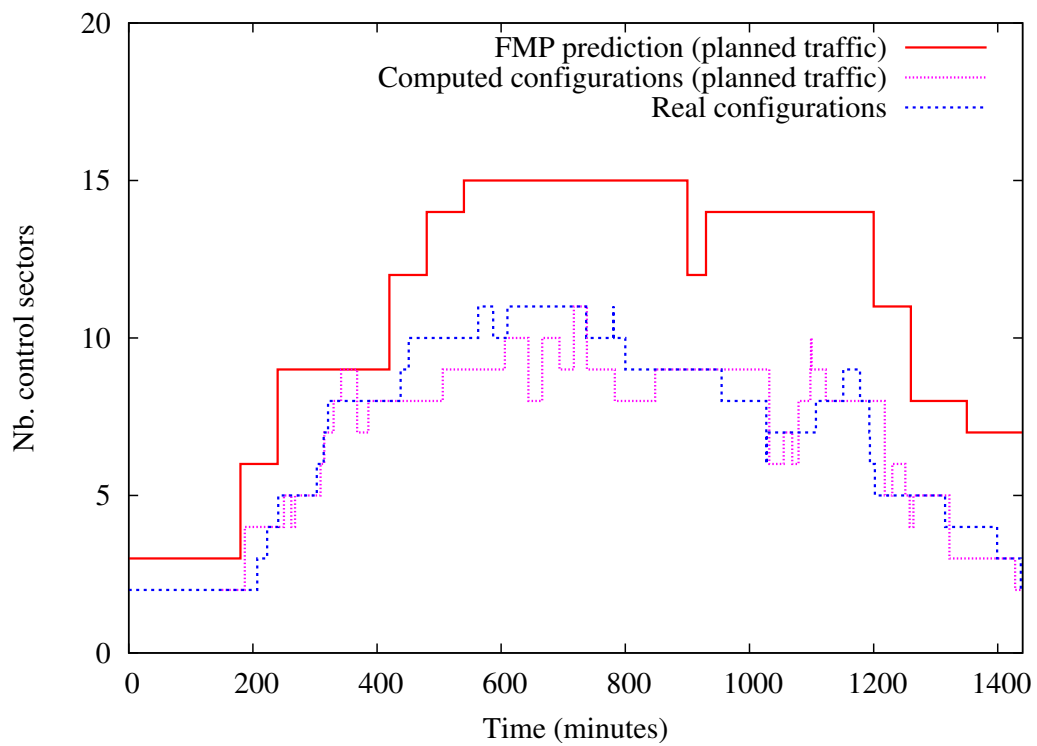


Figure 7: Airspace configuration forecast using flight plans as input: computed prediction *vs.* current FMP prediction (Brest ACC, 7th June 2003).

cast<sup>31</sup>.

#### 8.4. Results with or without constraints on the number of sectors

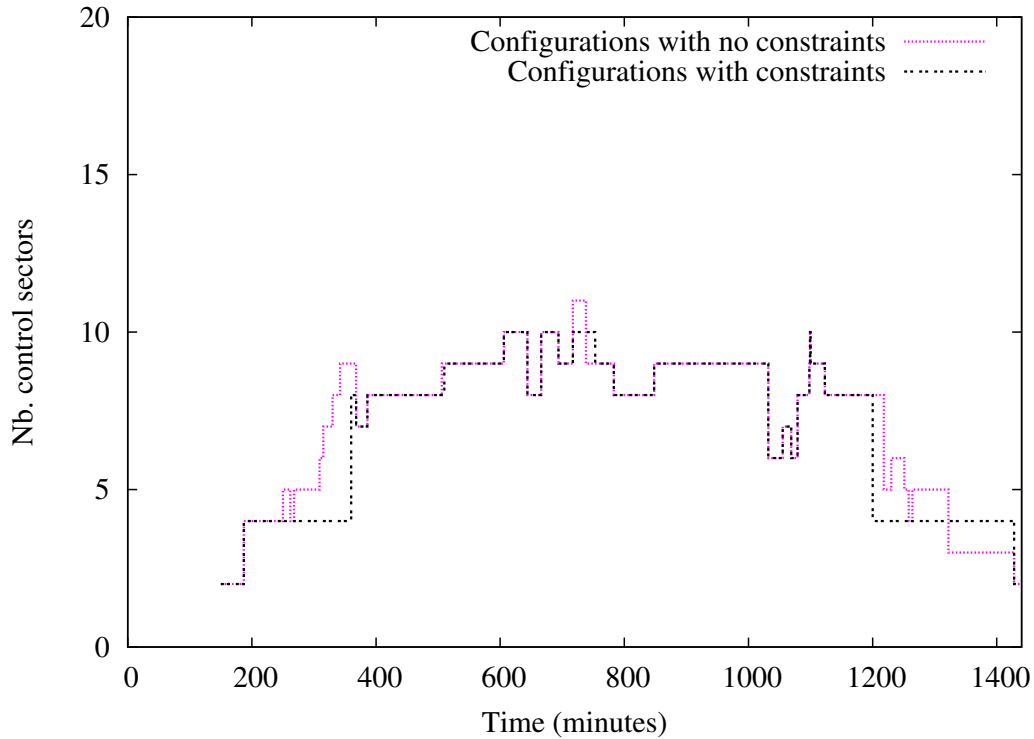


Figure 8: Airspace configuration forecast with or without constraints on the maximum number of ATC sectors (Brest ACC, 7th June 2003).

Finally, let us illustrate what may be the most useful application of the proposed algorithms. So far we have not introduced constraints on the maximum number of ATC sectors in each configuration. Running the algorithms without such constraints allows the operator to check if the duty roster is sufficient to handle the planned traffic.

Now, what happens if for some reason, there are not enough controllers, or not enough working positions available to handle the traffic? In such cases,

---

<sup>31</sup>The correlation coefficient is 0.91 for the FMP prediction and 0.93 for our prediction, which confirms that this coefficient is not usable to assess the quality of the prediction

the cost comparison described in section 5.1 allows the *branch & bound* algorithm to find configurations that satisfy such constraints while balancing the workload as well as possible. In Figure 8, we have constrained the maximum number of control sectors to 4 between 00h00 and 06h00, 8 between 06h00 and 08h30, 10 between 8h30 and 20h00, and finally 4 after 20h00. The figure shows the number of sectors for both solutions, with or without constraints.

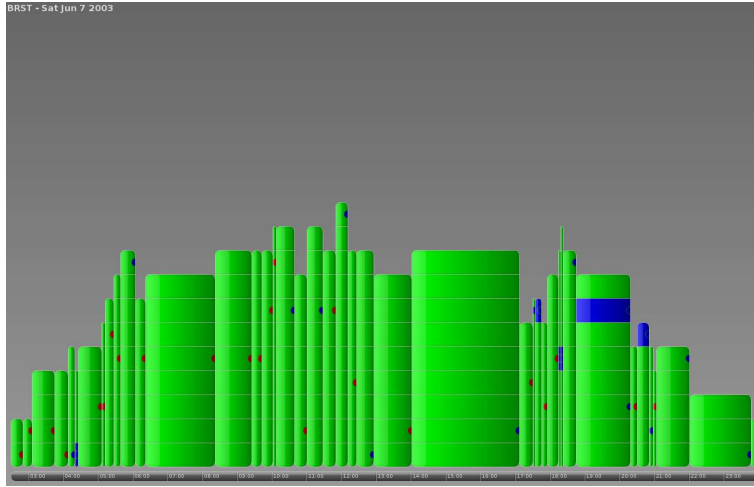


Figure 9: HMI display of the airspace configuration forecast, without constraints (Brest ACC, 7th June 2003).

An experimental human-machine interface is currently under development, with the aim of demonstrating and refining the proposed algorithms. Figure 9 shows the general view of the sector opening schedule for Brest centre, without constraints, over one day of traffic. Each column represents a partition of the airspace into ATC sectors. A pop-up window (not shown) may give more details on each configuration when browsing over the coloured boxes. The colour code is the following: blue when the sector is under-loaded, green when it is normally loaded, red when it is overloaded (although there are none here, without constraints). Coloured dots on the right of each box give an indication of which workload threshold (low or high) triggered the configuration change.

When posting no constraints on the maximum number of sectors, this display shows how many sectors would be necessary to handle a given traffic demand. This could allow the staff manager to adapt the duty roster to the input traffic. Note that without constraints, only ATC sectors made



up of a single airspace module can be overloaded: otherwise a sector composed of several modules would have been split into two smaller sectors when becoming overloaded.

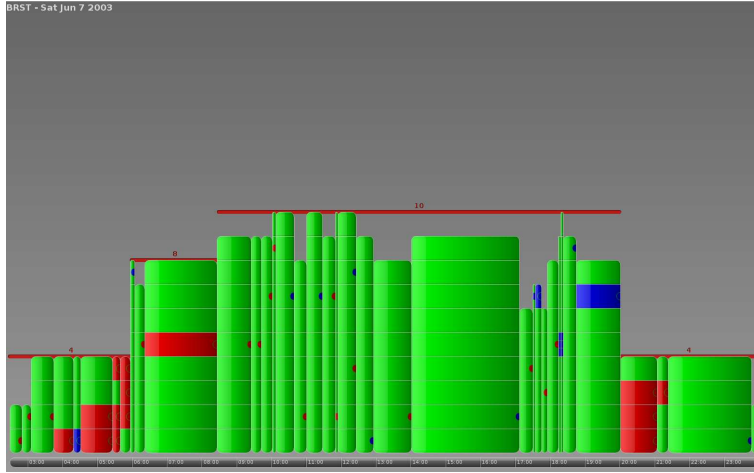


Figure 10: HMI display of the airspace configuration forecast, with constraints (Brest ACC, 7th June 2003).

Figure 10 shows the same situation, with constraints on the maximum number of sectors. We can observe the overloaded control sectors shown in red. Such a display could help flow managers to take preventive measures (departure delays, reroutings) so as to prevent such dangerously overloaded situations from actually occurring.

The user can switch to a more detailed view, as shown in Figure 11 where the workload evolution across time is displayed for each selected sector. For this purpose, each selected sector box shows the three probabilities issued by the neural network, stacked one above the other, using the same colour code: blue for  $p_{low}$ , green for  $p_{normal}$ , and red for  $p_{high}$ . The transitions between airspace configurations are also shown.

## 9. Conclusion and perspectives

### 9.1. Summary remarks

Forecasting airspace configurations requires at least two things: first a model assessing the controller workload for any given ATC sector in traffic conditions of variable complexity, and second an algorithm providing an optimal partition of the airspace into ATC sectors.

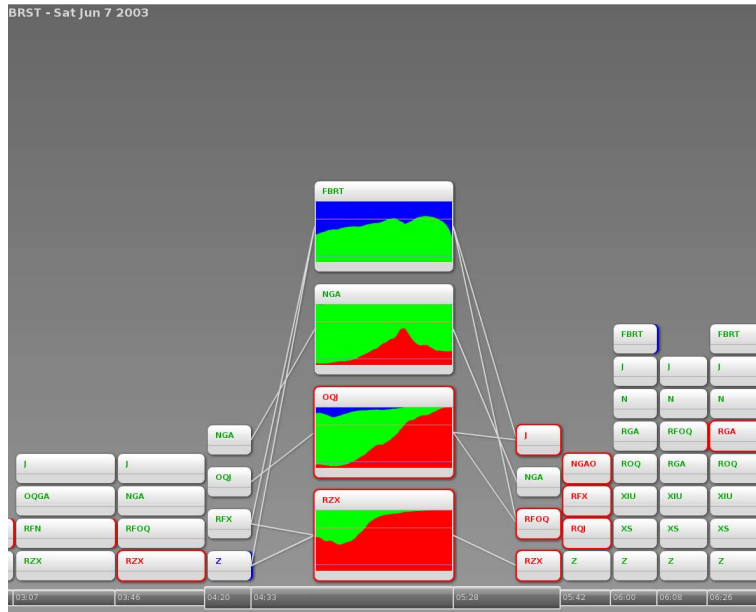


Figure 11: Details of some overloaded control sectors when enforcing constraints on the number of available working positions (Brest ACC, 7th June 2003).

The proposed workload model assumes that the observed sector operations reflect the controllers' actual workload: ATC sectors made of several modules may be recombined when the workload is either too high or too low. The model relies on a neural network to issue workload probabilities, using the sector volume and a selection of air traffic complexity metrics as input. Among several others found in the literature, these metrics appeared to be the most correlated to the observed sector status in past operations. Some others also proved relevant but were redundant with the selected variables and slightly less relevant. As the literature is fairly extensive on this subject, we have not implemented all existing metrics, and other complexity factors may be tried in the future that might improve the model.

The partitioning problem, which is highly combinatorial when considering all the possible combinations of airspace modules, has been addressed using standard tree search methods. This was made possible by restricting the tree search to partitions made up only of ATC sectors described in the ATCC databases. The neural network output probabilities were used to assess the cost of candidate configurations while exploring the tree.

This hybrid method was first tried on recorded traffic, and assessed by

comparing the results to the archived sector openings. The neural network was trained on data collected in the five French ATC centres on 2nd June 2003 and assessed on two other days (6th and 7th June 2003). Two measures were used to quantify the differences between the computed output and reality: a dissimilarity measure accounting for the differences in the number of ATC sectors, and the number of configuration changes.

The proposed method showed that the number of ATC sectors was fairly close to the actual one. The dissimilarity measure was 0.191 on 6th June and 0.115 on 7th June. However, the number of configuration changes was still above what actually occurred in reality: 46 against 33 in reality on 6th June, and 37 against 33 on 7th June.

After outlining the possible sources of bias and forecasting errors that may occur due to the difference between the planned traffic demand and actual flights, we tried our algorithms on simulated trajectories computed from flight plans. The comparison to the actual forecast made by the flow management operators shows that our prediction is far more realistic than those done currently in operations: the dissimilarity measure is 0.137 for the computed configurations, and 0.568 for the FMP forecast on 7th June.

### *9.2. Limitations of the proposed method*

In our results, the influence of flight planning and navigation uncertainties was certainly somewhat lessened by two factors. First by the use of a moving average method to smooth the input metrics (including the aircraft count), and second by the fact that the neural network trained on historical data reflects the average behaviour of controllers and control room managers over the observed past airspace operations.

Concerning the first factor, one must be aware that there is a trade-off between smoothing the metrics and the forecast accuracy. In fact, averaging the metrics over too long a period delays the times when airspace configuration changes are predicted. This might motivate further research, using time series, instead of smoothing the input metrics with a moving average method.

Dealing with flight planning uncertainties may have an influence over the context in which the proposed algorithms could be used. A certain level of uncertainty and the use of smoothed metrics may be acceptable in a pre-tactical context, when roughly assessing how many ATC sectors are necessary to handle the planned traffic. But this same level of uncertainty may not be acceptable for a tactical tool, when overloads need to be anticipated with

greater accuracy. However, we can expect to reduce these uncertainties by using a mix of radar tracks and flight intentions.

These issues need to be further explored, through scientific studies eventually involving ATC experts in real-time experiments. In any case, it must be underlined that one cannot expect a highly realistic airspace configuration forecast without an accurate 4D trajectory forecast.

Let us finally highlight a restriction in the use of the proposed algorithms: the workload prediction model is tuned on historical data and reproduces the average past behaviour of control room managers and air traffic controllers. So it should *not* be used by control room managers actually to decide how to split or merge sectors. This would freeze the whole system and always reproduce the same behaviour. The proposed hybrid method should in fact be seen as a forecasting tool for flow managers or multi-sector planners.

### *9.3. Potential gains in operational perspective*

Considering the difference (see Figure 7) between the sector opening schedule made by the FMP operator and the actual sector operations, the potential benefits in staff costs that could be expected from a more accurate forecast are obvious.

However, several arguments mitigate this statement. One cannot expect to adjust the staff variable perfectly to the traffic demand on an everyday basis, even with a perfect airspace configuration forecast. The staff variable is adjusted to peak traffic demand, estimated well in advance, as it takes several years to train a controller: air traffic controllers are a highly trained and highly qualified work force. The total number of controllers in operation cannot be fine-tuned over the year: this would mean employing fewer controllers in the winter than during the summer for example, which is not easily feasible.

Also, air traffic management is a world of uncertainties (take-off times, unexpected delays, sector entry times, weather disturbances, etc): some researchers refer to the "stochastic" nature of air traffic planning. Consequently the overall system also needs to deal with these uncertainties. The fact that the FMP schedule requires more staffing than what is actually necessary is also a safety margin that can be useful when unexpected peaks of traffic occur. It also reflects the flexibility of the current system: merging sectors or reconfiguring the airspace in real-time when the workload is low also allows air traffic controllers to maintain their proficiency by handling a large enough amount of traffic at all times of the day. So, in fact, it is not cost efficiency

or flexibility (the system is already highly flexible in real time) that we could improve the most with the proposed algorithms, it is the predictability of the system.

It is expected that our algorithms could be used, possibly by flow management operators or multi-sector planners, to *forecast* airspace configurations and controller workload with greater accuracy than today’s empirical method. This would allow better anticipation of potentially dangerous situations, and preventive measures to be taken well in advance when some sectors are predicted to be overloaded.

When using our algorithms without constraining the maximum number of ATC sectors, they could be used to assess how many sectors would be necessary to handle a given traffic demand. Posting constraints on the number of ATC sectors could help flow managers to forecast potential overloads in the circumstances of the day (duty roster, or limited number of controller working positions).

The potential gain in operations remains to be assessed through real-time experiments, and the HMI currently under development may be used for this purpose in the future.

#### 9.4. Further work

This work was a first approach to the problem. Time-dependencies were not explicitly taken into consideration, except through the use of complexity metrics that are related to traffic flows over a period of time. A simple feed-forward neural network was used, and its output only depends on a snapshot of the metrics values at every time  $t$ .

In future work, we plan to improve the workload prediction model by considering each input metric as a time series, and by using recurrent neural networks like the Jordan network ([49]), trained with backpropagation through time ([48], [50], [51]).

The opening schedule algorithm also needs improvements, such as checking if there is a feasible path from one configuration to another. At the beginning of this conclusion, we stated two issues when forecasting airspace configurations: predict the workload, and find an optimal airspace partition. In fact, we can add a third one, which is to find an optimal *sequence* of airspace configurations throughout the day, satisfying some constraints on the transitions between airspace configurations.

We have partially addressed this third issue by allowing local recombinations of sectors from one configuration to the next. However, we still need to

explore this issue further by applying tree search methods through a whole sequence of successive candidate configurations, instead of successively applying the tree search at each time step. Constraint programming techniques could be useful for this purpose.

The work presented in this paper will be included in the contribution of the French Air Navigation Services Provider to SESAR, the major European R&D programme.

## A. Glossary of terms

**Elementary airspace module:** a volume of airspace to which a radio frequency is assigned that may be used for communications between the aircraft flying in this volume and the air traffic controller in charge of aircraft separation. Modules can be combined (i.e. grouped onto one working position) to form larger ATC sectors, depending on the controller workload. Note that the usual term to denote these airspace modules is *sector*, which might be confusing as *sector* can either denote a single airspace module, or an air traffic control sector made up of one or more modules. So in this paper these elementary geographic sectors are referred to as *modules*, in order to avoid confusion with *ATC (Air Traffic Control) sectors*.

**ATC sector:** volume of airspace made up of one or more elementary airspace modules, operated as a single unit by a team of usually two controllers ("radar" and "planning").

**Controller working position:** a piece of furniture with radar screens, radio-communications equipment, telephones, systems interfaces, etc, operated by two controllers in charge of a control sector.

**Airspace configuration:** mapping of the set of elementary airspace modules onto a smaller number of controller working positions. It can also be seen as a partition of the ATC centre's airspace into control sectors.

**Sector opening schedule:** (or opening scheme) the successive airspace configurations that are predicted over a period of time in the future.

**ATC:** Air Traffic Control.

**ATCC:** Air Traffic Control Centre. A managerial unit in charge of a volume of airspace divided into a number of sectors.

**ATM:** Air Traffic Management.

**ATFM:** Air Traffic Flow Management

**CFMU:** European Central Flow Management Unit

**FMP:** Flow Management Position

**SESAR:** Single European Sky ATM Research Programme.

## References

- [1] S. Loft and P. Anderson and A. Neal and M. Mooij Modeling and Predicting Mental Workload in En Route Air Traffic Control: Critical Review and Broader Implications. In *Human Factors: The Journal of the Human Factors and Ergonomics Society*, volume 49, N3, p. 376-399, June 2007.
- [2] D. Gianazza and K. Guittet. Evaluation of air traffic complexity metrics using neural networks and sector status. In *Proceedings of the 2nd International Conference on Research in Air Transportation*, ICRAAT 2006.
- [3] D. Gianazza and K. Guittet. Selection and evaluation of air traffic complexity metrics. In *Proceedings of the 25th Digital Avionics Systems Conference*, DASC 2006.
- [4] C. Verlhac and S. Manchon. Optimization of opening schemes. In *Proceedings of the fourth USA/Europe Air Traffic Management R&D Seminar*, 2001.
- [5] D. Delahaye and J.M. Alliot and M. Schoenauer and J.L. Farges. Genetic algorithms for partitioning airspace. In *Proceedings of the Tenth Conference on Artificial Intelligence Application*, 1994.
- [6] D. Delahaye and J.M. Alliot and M. Schoenauer and J.L. Farges. Genetic algorithms for automatic regroupement of Air Traffic Control sectors. In *Proceedings of the Conference on Evolutionary Programming*, 1995.

- [7] D. Delahaye and M. Schoenauer and J.M. Alliot. Airspace Sectoring by Evolutionary Computation *IEEE International Congress on Evolutionary Computation*, 1998.
- [8] A. Klein. An efficient method for airspace analysis and partitioning based on equalized traffic mass. In *Proceedings of the 6th USA/Europe Air Traffic Management R & D Seminar*, 2005.
- [9] H. Trandac, P. Baptiste, and V. Duong. Optimized Sectorization Of Airspace With Constraints In *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, June 2003.
- [10] H. Trandac, P. Baptiste, and V. Duong. Airspace sectorization by constraint programming. In *Proceedings of the "1ère conférence en Recherche Informatique Vietnam & Francophone"*. RIVF, 2003.
- [11] P. Flener, J. Pearson, M. Agren, C. Garcia-Avello, M. Celiktin, and S. Dissing. Air-traffic complexity resolution in multi-sector planning using constraint programming. In *7<sup>th</sup> USA/Europe Seminar on Air Traffic Management Research and Development*, 2007.
- [12] A. Basu, J. S. B. Mitchell and G. Sabhnani. Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing. In *Proceedings of the 9th Workshop on Algorithms Engineering and Experiments (ALENEX)*, January 2008.
- [13] C.E. Bichot and N. Durand. A tool to design functional airspace blocks. In *7<sup>th</sup> USA/Europe Seminar on Air Traffic Management Research and Development*, 2007.
- [14] C. E. Bichot. A New Meta-method for Graph Partitioning. In *IEEE Congress on Evolutionary Computation (CEC)*, 2008.
- [15] Cognitive complexity in air traffic control, a literature review. Technical report, Eurocontrol experimental center, 2004.
- [16] R. H. Mogford and J. A. Guttman and S. L. Morrow and P. Kopardekar. The complexity construct in air traffic control: A review and synthesis of the literature. FAA WJHTC DOT/FAA/CT-TN95/22, Federal Aviation Administration, 1995.



- [17] P. Kopardekar. Dynamic density: A review of proposed variables. FAA WJHTC internal document. overall conclusions and recommendations, Federal Aviation Administration, 2000.
- [18] P. Kopardekar and S. Magyarits. Measurement and prediction of dynamic density. In *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, 2003.
- [19] A. J. Masalonis, M. B. Callahan, and C. R. Wanke. Dynamic density and complexity metrics for realtime traffic flow management. In *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, ATM 2003.
- [20] A. Majumdar, W. Y. Ochieng, G. McAuley, J.M. Lenzi, and C. Lepadetu. The factors affecting airspace capacity in Europe: A framework methodology based on cross sectional time-series analysis using simulated controller workload data. In *Proceedings of the 6th USA/Europe Air Traffic Management R & D Seminar*, ATM 2005.
- [21] G.B. Chatterji and B. Sridhar. Measures for air traffic controller workload prediction. In *Proceedings of the First AIAA Aircraft Technology, Integration, and Operations Forum*, 2001.
- [22] B. Sridhar and K. S. Seth and S. Grabbe Airspace Complexity and its Application in Air Traffic Management In *Proceedings of the 2nd USA/Europe Air Traffic Management R & D Seminar*, ATM 1998.
- [23] I. V. Laudeman, S. G. Shelden, R. Branstrom, and C. L. Brasil. Dynamic density: An air traffic management metric. Technical report, 1999.
- [24] J.H. Crump. Review of stress in air traffic control: Its measurement and effects. *Aviation, Space and Environmental Medecine*, 1979.
- [25] P. Averty, S. Athènes, C. Collet, and A. Dittmar. Evaluating a new index of mental workload in real ATC situation using psychological measures. Note CENA NR02-763, CENA, 2002.
- [26] A. Yousefi, G. L. Donohue, and K. M. Qureshi. Investigation of en route metrics for model validation and airspace design using the total airport and airspace modeler (TAAM). In *Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar*, 2003.

- [27] ACT-540 NAS Advanced Concepts Branch. An evaluation of dynamic density metrics using RAMS. Technical report (draft) DOT/FAA/CT-TN, Federal Aviation Administration, April 2001.
- [28] 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.
- [29] P. Averty. Conflict perception by ATCS admits doubt but not inconsistency. In *Proceedings of the 6th Air Traffic Management Research & Development Seminar*, 2005.
- [30] A. Klein, P. Kopardekar, M. D. Rodgers, and H. Kaing. Airspace playbook: Dynamic airspace reallocation coordinated with the national severe weather playbook. In *Proceedings of the 7th AIAA Aviation Technology, Integration and Operations Conference*, 2007.
- [31] Global air traffic management operational concept. Technical report, International Civil Aviation Organization, 2005.
- [32] H. Swenson, R. Barhydt, and M. Landis. Next Generation Air Transportation System (NGATS) Air Traffic Management (ATM)-Airspace Project. Technical report, National Aeronautics and Space Administration, 2006.
- [33] E. S. Stein, P. S. Della Rocco, and R. L. Sollenberger. Dynamic resectorization in air traffic control: A human factors perspective. Technical report DOT/FAA/TC-TN06/19, Atlantic City International Airport, NJ: Federal Aviation Administration William J. Hughes Technical Center, 2006.
- [34] D. Gianazza, J. M. Alliot, and G. Granger. Optimal combinations of air traffic control sectors using classical and stochastic methods. In *Proceedings of the 2002 International Conference on Artificial Intelligence*, 2002.
- [35] D. Gianazza and J. M. Alliot. Optimization of air traffic control sector configurations using tree search methods and genetic algorithms. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002.

- [36] D. Gianazza. Airspace configuration using air traffic complexity metrics. In *7<sup>th</sup> USA/Europe Seminar on Air Traffic Management Research and Development*, 2007. best paper of "Dynamic Airspace Configuration" track.
- [37] D. Gianazza, C. Allignol, and N. Saporito. An efficient airspace configuration forecast. In *Proceedings of the 8th USA/Europe Air Traffic Management R & D Seminar*, ATM 2009.
- [38] D. Gianazza. Smooth traffic complexity metrics for airspace configuration schedules. In *Proceedings of the 3rd International Conference on Research in Air Transportation*, ICRAAT 2008.
- [39] N. Durand, J.M. Alliot, and J.F. Bosc. CATS, a complete air traffic simulator. In *Proceedings of the 16th Digital Avionics Systems Conference*, DASC 1997.
- [40] N. Durand, J.M. Alliot, and G. Granger. Optimal resolution of en route conflicts. In *Proceedings of the 4th USA/Europe Air Traffic Management R & D Seminar*, ATM 2001.
- [41] J. D. Welch and J. W. Andrews and B. D. Martin and B. Sridhar. *Macroscopic workload model for estimating en route sector capacity*. In *Proceedings of the 7th USA/Europe Air Traffic Management R & D Seminar*, ATM 2007.
- [42] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996. ISBN: 0-198-53864-2.
- [43] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996. ISBN: 0-521-46086-7.
- [44] M. I. Jordan and C. Bishop. *Neural Networks*. CRC Press, 1997.
- [45] R. A. Dunne and Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proceedings of the Eighth Australasian Conference on Neural Networks*, 1997.
- [46] P. Averty and K. Guittet and P. Lezard. An Ordered Logit Model of Air Traffic Controllers Conflict Risk Judgment. *Air Traffic Control Quarterly*, vol. 16 (2008), no. 2, pp. 101–125, 2008.

- [47] C. Gwiggner and V. Duong Averages, Uncertainties and Interpretation in Flow Planning In *Proceedings of the 2nd International Conference on Research in Air Transportation*, ICRAAT 2006.
- [48] D. E. Rumelhart. *Learning internal representations by error propagation*. University of California, Institute for Cognitive Science, San Diego, September 1985.
- [49] M. I. Jordan. *Serial order: a parallel distributed approach*. University of California, Institute for Cognitive Science, San Diego, 1986.
- [50] P. Werbos. . Proceedings of the IEEE, 1990, vol. 78, no 10, pp. 1550-1560
- [51] R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin and D. E. Rumelhart, editors, *Back-propagation: Theory, Architectures and Applications*, chapter 13, pages 433–486. Hillsdale, NJ: Erlbaum, 1992.