Towards an Operational Sectorisation based on Deterministic and Stochastic Partitioning Algorithms

Judicaël Bedouet, Thomas Dubot, and Luis Basora
Department of System Design and Performance Evaluation
ONERA, The French Aerospace Lab - 31055 Toulouse, France
Email: (Judicael.Bedouet|Thomas.Dubot|Luis.Basora)@onera.fr

***Abstract*—This paper describes a method combining operational deterministic and stochastic approaches to generate optimised and smooth sector configuration plans. First, sector configurations commonly used within an Area Control Centre (ACC) are sorted according to a set of objectives for each time period. From these Pareto-optimal solutions, we determine through a stochastic method new sector configurations, mainly unstructured, to improve criteria such as the workload distribution. Then these optimized configurations are remodelled with another stochastic function to compute a set of final configurations acceptable to air traffic controllers. Secondly, we integrate these good solutions throughout the day to build the smoothest sector configuration plan possible, using the minimization of a dedicated distance function between successive configurations. Results of the SESAR 07.05.04 VP-755 experiments demonstrate that such methods could improve several criteria at the level of an ACC, such as the French Reims ACC, and pave the way for the development of an automated decision support tool integrating such algorithms.**

I. Introduction

The challenge of cleverly combining airspace sectors for an efficient use of air traffic control resources [1], or sectorisation process, is a classic optimisation problem in Air Traffic Management. Various optimisation methodologies have been explored [2], such as Constraint Programming [3], Mixed Integer Programming [4], Global Optimisation and Evolutionary Algorithms [5, Chapter 5], [6, Chapter 3].

Nevertheless, most control centres still rely today on a human expertise to compare the traffic demand with the sector capacities and choose the adequate airspace configuration for each time period of the day. In this way, only a small subset of predefined configurations is used, instead of exploring all the possible combinations of sectors. In addition, the usual hotspot resolution method consists in splitting the overloaded sector into two sectors, which is not optimal (increase of the number of control positions) and sometimes impossible (e.g. if the hotspot occurs at the level of an elementary sector) [7].

The European SESAR program [8] experiments with modular and flexible dynamic airspace configurations to better adapt to demand pattern changes and traffic flows volatility induced by an extensive implementation of free route operations [9], [10]. In this context, large airspace blocks are decomposed into airspace building blocks, smaller than current elementary sectors, and delineating typical demand forecast patterns, e.g. traffic flows. These building blocks, which are not necessarily controllable, are then grouped into control sectors named Controlled Airspace Blocks (CAB). In this way, control sectors are more adapted to traffic specificities, which enables to solve hotspots by reorganising the frontiers of control sectors, without modifying their number, instead of splitting one of the existing control sectors. We presented in [11] the methodologies and tools developed within the SESAR VP-755 exercise of the SESAR 07.05.04 project, which consisted in a performance assessment of sectorisation algorithms based on this new paradigm applied to the French Reims ACC.

We describe in this paper the sectorisation algorithms introduced in this communication paper, with the ambition of providing:

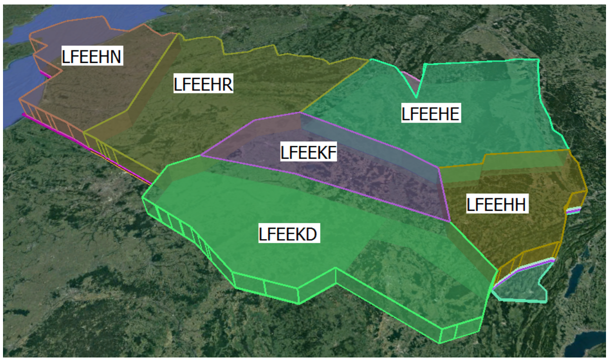
- for each time period a set of optimized sector configurations acceptable to controllers, i.e. composed of sectors commonly used in the operational context, or new sectors with admissible shape;
- a smooth sector configuration plan ensuring the shape stability of sector configurations (no abrupt sector changes between two consecutive periods of time) throughout the day;
- a decision support tool integrating these algorithms to automatically guide the operational experts in the process of building a sector configuration plan, whatever the time horizon considered.

This paper focuses on the way to combine deterministic and stochastic algorithms to compute optimized and operational sector configurations, the acceptability being the key challenge. This approach is based on our previous work, presented in [7] to build smooth configuration plans, which is a relatively unexplored field of research [12], [13]. Nevertheless all algorithms have been reshaped to take into account operational constraints. We describe in section II the mathematical model of this graph partitioning problem, and the different objectives considered, such as the workload distribution. In section III, we detail how to determine initial good solutions using a deterministic algorithm, which explores known sector configurations. Section IV explains how to disorganise and reform these solutions, using stochastic methods. Finally we detail in section V the distance function introduced to integrate these solutions in a smooth sector configuration plan. Section VI summarizes the results obtained with the Reims ACC use case, and section VII concludes on the possibility to integrate such algorithms in an operational automated decision support tool.

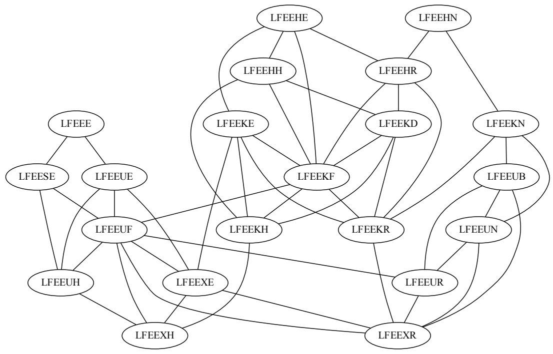
II. Modelling the problem

A. Graph model

A sectorisation problem is equivalent to a graph partitioning problem [14]. As shown in Figure 1 for the Reims ACC, each building block is mapped to a node of the graph $G = (V, E)$. The graph G denotes the representation of the airspace, where V (the set of vertices) is the set of building blocks and E (the set of edges), is such that (u, v) belongs to E only if there can be a direct trajectory from u to v.



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



(b)

Figure 1. 21 Reims building blocks (a) vs. their graph representation (b)

The set of vertices is always the same along the day[1]. We build the edges of the graph by analysing the geometrical connectivity between the vertices [15].

For a given time period δt, the weights of the nodes and the costs of the edges are determined by analysing the entries/exits of the flight plans in the different airspace building blocks. These values are computed as follows:

- $D_v(\delta t)$ density workload that occurs at vertex v during a given time period δt. Density is notably proportional to the time spent by aircraft in the block. It would be possible to refine this value with, for example, the number of potential conflicts, like in [16] or [17]. Nevertheless, the number of aircraft remains the main variable operationally used to decide to open or close a sector.
- $C_e(\delta t)$ coordination workload assigned to the edge e during a given period δt. It depends on the number of aircraft flying from a block to another.

For a given time period δt, we call $P_k(\delta t)$ a partition of this graph G in k subsets such as $P_k(\delta t) = S_1, ..., S_k$. Each subset S_i of the partition must be non-empty and disjoint from the other subsets. Moreover, the union of all the subsets must entirely cover graph G and satisfies the connectivity constraint: the different elements of a subset, the aforementioned vertices, must be connected. In the operational jargon, a partition is called a sector configuration and a subset a CAB.

B. Partition functions

For a given time period δt, we introduce three functions that will be useful to characterise the properties of a partition: balance, cut and compactness.

We define the balance of a partition $P_k(\delta t)$ as the sum of the differences between the density workload of a subset S_i, $D_{S_i}(\delta t)$, and the average density of the subsets:

$$\text{balance}(P_k(\delta t)) = \sum_i \left| D_{S_i}(\delta t) - \frac{\sum_i D_{S_i}(\delta t)}{k} \right|$$

$$\text{where } D_{S_i}(\delta t) = \sum_{v \in S_i} D_v(\delta t).$$

This definition is inspired by [14] and is used to equally share the workload between the different sectors. However, this definition is based on the infinity norm whereas ours is based on the Manhattan norm. A 2-norm could also be considered.

We define the cut of a partition $P_k(\delta t)$ such as the sum of the edges cut by the partition during a given period δt:

$$\text{cut}(P_k(\delta t)) = \sum_{i<j} \text{cut}(\delta t, S_i, S_j)$$

$$\text{where cut}(\delta t, S_i, S_j) = \sum_{v_1 \in S_i, v_2 \in S_j} C_{(v_1,v_2)}(\delta t).$$

It estimates the cost needed to coordinate traffic between the different sectors and the initial idea is presented in [18].

To better handle the shape of sectors, another function, called compactness, has recently been introduced by Jägare, Flener and Pearson in [19]. Their definition of compactness is inspired by the sphericity of the shape of a quartz particle in crystallography [3]. We give another definition based on the ratio of two prisms[2]' volumes, avoiding balconies:

$$\text{compactness}(P_k(\delta t)) = \prod_i \text{compactness}(S_i)$$

$$\text{where compactness}(S_i) = \frac{\sum_{j\ prisms\ of\ S_i} \text{volume}(j)}{\text{volume}(\text{cover}(i))}$$

$$\text{cover(i) is the smallest prism which includes all the prisms of i.}$$

[1]In the context of operations, it may change but every six months or more.

[2]A prism is a polyhedron with two parallel n-sided polygonal bases and n other parallelogram-shaped faces joining the corresponding sides of the two bases. A sector can be seen as a set of contiguous prisms as represented in Figure 2.

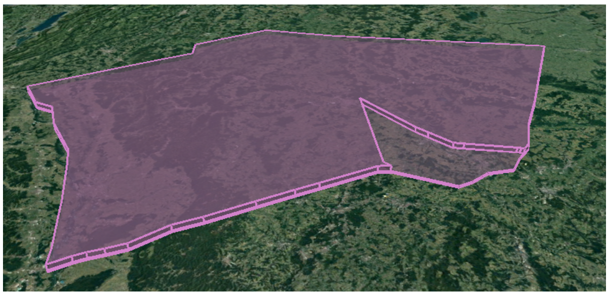


Figure 2. Sector LFEEUE composed of two contiguous prisms

When a sector can be described by one and only one prism, it is said to be compact. Otherwise, balconies appear, like LFEEUE in Figure 2, which makes more complicated the controllers' task. However, as the altitude of the floor or the ceiling varies inside the ACC (see Figure 3), balconies are unavoidable in some cases. This is the case of LFEEUE, which is operationnally acceptable. The volume of the covering airspace is then corrected to ignore volumes outside the ACC so that compactness of those sectors can be valued to 1.

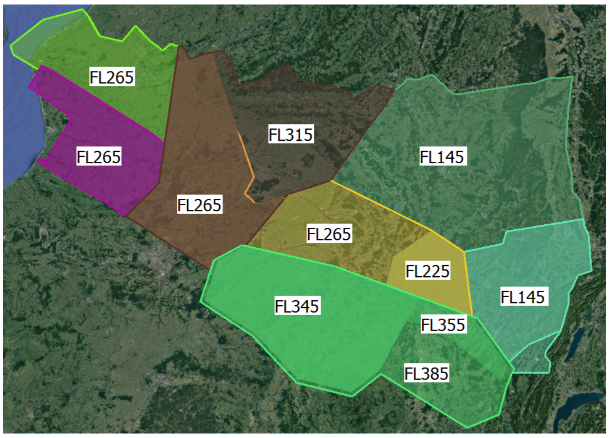


Figure 3. Flight Levels (FL) of the floor of the Reims ACC airspace

It should be noted that the perfect partition is found when its balance is 0, its cut is 0 and its compactness is 1. Unlike balance and cut, compactness is computationally expensive, but since we often manipulate the same sectors, values of $\mathrm{compactness}(S_i)$ are put into a cache to speed up computations.

C. Objectives and constraints

To build the optimal partition of airspace into k sectors, we therefore consider different objectives:

- minimising the workload distribution, which consists in minimising the function $\mathrm{balance}(P_k(\delta t))$ for each period;
- minimising the total number of transfers, which consists in minimising the function $\mathrm{cut}(P_k(\delta t))$ for each period;
- ensuring that resulting sectors have acceptable geometric shapes with a minor number of balconies, which consists in maximising the function $\mathrm{compactness}(P_k(\delta t))$.

In some contexts, we may also minimise the trajectory-convexity [2], i.e. minimising the number of re-entries (a re-entry corresponds to a flight that enters at least twice in the same sector) or ensuring the minimum dwell time [2], i.e. minimising the total number of short transits (a short transit corresponds to a flight that spends less than x minutes in a sector, e.g. 4 minutes). In the context of the VP-755 exercise, we assume that these two objectives are not essential, since they are intrinsic to the traffic and the building blocks. They are more useful to design the airspace building blocks since the same sectorisation methodology can be applied to very small cells of airspace [12].

Finally, resulting sectors must respect two hard constraints:

- minimum size: a sector must contain at least one block;
- connectivity: a sector can not be fragmented i.e. its different elements must be connected.

III. Determining an initial good solution

A. Exploring the complexity

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

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

where n is the number of building blocks,

k is the number of parts,

$\binom{k}{j}$ is the binomial coefficient $\dfrac{k!}{j!(k-j)!}$

For instance, for 8 positions to be opened in the Reims ACC, we have 1.3×10^{14} possible solutions with the conventional 21 blocks and 2.0×10^{33} with 42 blocks. It is impossible to assess such a number of sector configurations in a reasonable time. But the Reims controllers do not exploit much more than 79 control sectors including the 21 elementary sectors. If we only consider those sectors, we have a limited set of possible sector configurations: 180 315, going from 1 open position to 21[3]. To reach this result, we enumerate all the valid combinations with an exhaustive search tree. All branches not leading to a valid sector configuration are cut as soon as the connectivity constraint or graph coverage are broken.

We are then able to enumerate all the conventional sector configurations based on all the control sectors used by Reims controllers. To validate the method, we also did some tests in other French ACCs. The complexity of the Reims ACC is a bit smaller than other ACCs in terms of elementary sectors and control sectors. If we consider the Brest ACC, which is representative of the most complex European ACCs, we are able to enumerate all conventional sector configurations but there are much more than in Reims. We get 1.9×10^8 sector configurations from 1 to 18 positions. It is hence reasonable to assess all these configurations in a strategic timeframe (preparation of the sector configuration plan for the next day)

[3]In the reality of operations, only a maximum of 17 positions are opened simultaneously.

but not in a tactical one. To apply the method to bigger ACCs like Brest, it would be necessary to find heuristics to increase the number of cuts in the search tree in order to only keep interesting sector configurations: e.g. eliminate configurations that would clearly conduct to an imbalance, independently of the traffic, like a configuration combining large and very small sectors.

B. Assessing the complexity

For each time period of the day, we have access to a database of sector configurations based on conventional sectors. Each sector configuration can be assessed for each of the following objectives: workload distribution, flow cut and if needed trajectory-convexity and minimum dwell-time. It is not necessary to assess the compactness since they are based on conventional sectors that are already accepted by controllers. Then, for each period, we sort each sector configuration according to the value of the n multi-objectives and look for

$$\min_{P_k \in SC_k} (\text{balance}(P_k), \text{cut}(P_k), ..., f_n(P_k))$$

where SC_k is the set of configurations with k sectors

f_n is the n-th objective function.

It is rare to find a solution that minimises all the objectives and according to the theory of multi-objective optimisation, a solution P_k^i is said to dominate another one P_k^j if

$$\forall i \in 1, 2, ..., n, f_i(P_k^i) \leq f_i(P_k^j)$$
$$\exists j \in 1, 2, ..., n, f_i(P_k^i) < f_i(P_k^j)$$

All solutions that are not dominated are said to be Pareto optimal. The set of Pareto optimal solutions forms the Pareto front. Solutions that are only dominated by solutions of the Pareto front form the 2^{nd} front, and so on. Following the assessment of the different sector configurations, it is possible to associate each configuration to a front, as illustrated in Figure 4. We limit the total number of solutions in the different fronts to only keep good solutions.

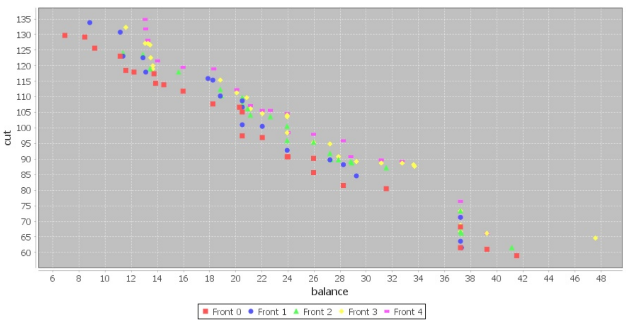


Figure 4. Example of Pareto fronts with two objectives

For each time period of the day, we have a set of fronts in which we can pick good solutions. To form the sector configuration plan, i.e. the set of sector configurations to be executed, the different good solutions are combined throughout the day, as explained in section V. Next section explains how these solutions can be refined to improve them.

IV. Refining initial solutions

Since the partitioning problem is NP-complete [20], one efficient way to find, not the optimal solution, but a very good solution, consists in exploring metaheuristics. A complete and detailed review of the different techniques can be found in [14]. These techniques are very efficient if they can start with a good initial solution. In our case, such a solution is provided by exploring the catalogue of existing sector configurations.

Our stochastic algorithms rely on the Simulated Annealing (SA) metaheuristic [21]. The main idea is to accept worse moves to not be stuck in a local optimum and to slowly decrease the frequency of acceptance as the search progresses. The progress is given by a temperature variable T, which is decreased at each iteration. Our objective is to explore new airspace configurations by exchanging building blocks, in order to check if criteria mentioned previously, e.g. the workload distribution, could be improved without degrading the compactness and consequently their acceptance by controllers. Our method is based on two successive simulated annealing algorithms: the first one will disorganise the initial solution to largely improve the balance and the cut. However, the resulting solution is often composed of ill-shaped sectors. The objective of the second algorithm is to make them compact.

During the first phase, the objective is to minimise the balance and the cut while keeping a good compactness. Then, we define the following objective function:

$$\min \left(\alpha \text{cut}(P_k) + \beta \text{balance}(P_k) + \frac{1}{\text{compactness}^2(P_k)} \right).$$

In our experiments, we chose for instance $\alpha = 0.1$ and $\beta = 0.9$ to homogenise the two first terms. The goal of the third term with the compactness is to control the shape of the sectors but not necessarily to guarantee compact sectors. The outcome solution will be then refined during the second phase to be acceptable to controllers. This is done by optimising the compactness under the constraints of not worsening the balance and cut of the initial solution. The objective function to minimise is then

$$\min \ (1/\text{compactness}(P_k))$$
$$\text{such that } \text{balance}(P_k) \leq \text{balance}(P_k^{initial})$$
$$\text{cut}(P_k) \leq \text{cut}(P_k^{initial}).$$

V. Elaborate a stable solution over time

We have seen in section IV how to determine for each time period a large number of good solutions, through the use of Pareto fronts. The next step is to combine them to form a smooth sector configuration plan throughout the day. For two consecutive time periods, we must find two sector configurations that minimise the distance between both. It is a classic of the shortest path problems in graph that can be solved by applying the Bellman's Principle of Optimality [22], for which we just have to define a distance function between two partitions. Gusfield introduces the notion of partition-distance in [23]. Given two partitions P and P' of the same graph G, he defines the distance $D(P, P')$ between these two partitions as the smallest sum of weights of any nodes of G whose removal causes the two induced partitions to be identical. This distance function is categorised by Yousefi et al. as a shared-cell metric [24]. Based on the blocks shared or not by the different sectors, it measures the degree of difference between two sectorisations. Yousefi et al. also introduces another metric based on the Hausdorff distance [25]. It is purely based on the geometric differences between two partitions. Some additional details about associated reconfiguration complexity metrics can be found in [26]. It should be noted that these metrics can measure a distance between two configurations even if the number of sector changes.

We define a distance based on the shared-cell metric:

$$\text{distance}(P_k(\delta t), P'_{k'}(\delta t')) = \sum_{j \notin R_{tt'}} \frac{D_j(\delta t) + D_j(\delta t')}{2}$$

where $R_{tt'} = \sum_{i=0}^{\min(k,k')} \bigcup_{v \in S_i(\delta t), v \in S_i(\delta t')} v$ is the set of cells that do not change after reconfiguration.

This distance is defined as the total density workload of the building blocks that are moved from a subset S_i to another while switching from the first partition to the second one. To reflect the reality of operations, we alter the results to favour the collapsing/de-collapsing operations. A collapsing operation corresponds to two sectors that are merged together to form a unique sector. A de-collapsing option corresponds to the splitting of a sector. In such cases, we consider that these operations are the least disturbing for the controllers and we sum 0 instead of $D_j(\delta t) + D_j(\delta t')$.

Once we have defined the distance between two sector configurations, we apply a shortest path algorithm to determine the sector configuration plan that minimises the total distance, which is the sum of the distances between the different sector configurations composing the plan. It should be noted that the transition between two sector configurations only depends on the previous configuration. However, once one is chosen, some interesting future transitions are eliminated and it is not possible to come back to one of them without deteriorating the smoothness. Such a process is already known from controllers who may decide to break the smoothness to switch to another succession of transitions. With a shortest path algorithm, we can better explore the catalogue of sector configurations and have a smooth plan for the entire day.

VI. Results

A. Protocol

A reference scenario was provided by DSNA, the French Air Navigation Service Provider. This scenario contains the sector configuration plan of a very busy day (2015, June 26^{th}), i.e. the different sectors that were opened for each time period this day. Traffic data were provided by EUROCONTROL in the DDR2 format [27]. One of the purposes of the VP-755 exercise was to explore the granularity of building blocks on the sectorisation process. For this reason, a set of new building blocks was manually designed with the expertise of an operational expert from Reims. Whereas Reims controllers use 21 blocks nowadays, we explored in this study the use of 42 manually designed blocks. These blocks do not necessarily correspond to the cutting of a 21-block into two blocks but to a further refinement according to major flows (see Figure 5).

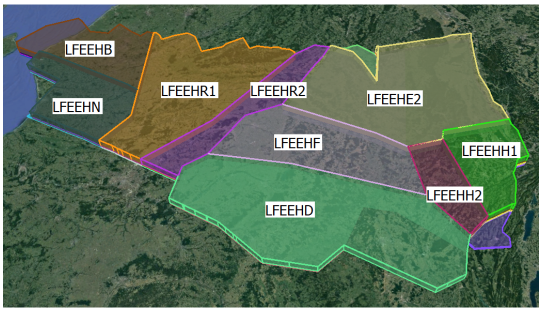


Figure 5. 42 Reims blocks (some are covered by top blocks)

The optimisation process follows four steps. During the first step, all the possible sector configurations with the 79 conventional sectors are enumerated (see section III-A). In a second step, according to the sector configuration plan from DSNA, sector configurations for each time period are classified into Pareto fronts (see section III-B). The third step consists in refining the solutions of each first front with a good balance[4] (see section IV). The refined configurations are injected in the catalogue of conventional configurations. In the final step, conventional and refined configurations are classified into Pareto fronts again and used by the smoothing algorithm to determine a configuration plan stable over time (see section V). We chose to limit the maximum number of Pareto solutions for each time period to 200, which is sufficient to have a plan mainly based on collapsing/decollapsing operations. Our tests show that we need to increase this number to 500 if we want a plan entirely based on collapsing/decollapsing operations. However, this will increase the computation time and may inject less optimal configurations in the plan.

In the next paragraphs, we will discuss the results of the refining algorithm and the final plan found by the smoothing algorithm.

B. Refining algorithm results

In this section, we have a look at a solution found for the period 07:08 - 09:05, where 6 positions were opened. 15 configurations were present in the first front and we will focus on the refinement of a randomly chosen solution denoted by P_6^{14}. This solution illustrates most of the characteristics of the found solutions. The figure 6 shows the initial and refined configurations and exploded views of both.

The refined configuration is based on three conventional sectors (LFEE4N, LFEEKD4R and LFEEKHEH), rediscovered by the refining algorithm, and three new sectors. The algorithm

[4]We can find in the first front ill-balanced but very well-cut solutions. We decided to give priority to well-balanced configurations.

decides to collapse LFEEUKBN and LFEEHBN, which were clearly underloaded, into the single sector LFEE4N. One of the sectors is a bit more overloaded (LFEEKD4R) but this sector is interesting because it contains major flows from East and South to the Paris TMA (Terminal Manoeuvring Area): the number of transfers is relatively small compared to other sectors. However, even if this sector is conventional, it is not often exploited by controllers because its large size makes it difficult for them to precisely visualise it on the Operational Display Systems.

Table I shows the values of the objective functions of the initial, unstructured and well-shaped solutions. Since the goal of the refining process is to minimise balance and cut and maximise compactness, we see that the balance is very good at the end of the disorganising phase but the compactness is not so good and needs to be reworked to be acceptable to controllers. At the end of the refining phase, the solution is not so well-balanced but well-shaped without degrading the metrics of the initial configuration. Table III shows that all sectors are compact except "HF KF UF2 XF2", which is described with only two prisms. A sector with two prisms often corresponds to a sector with one balcony, which is acceptable to controllers if it is justified. It should be noted that the initial solution already contains a sector with one balcony (see table II).

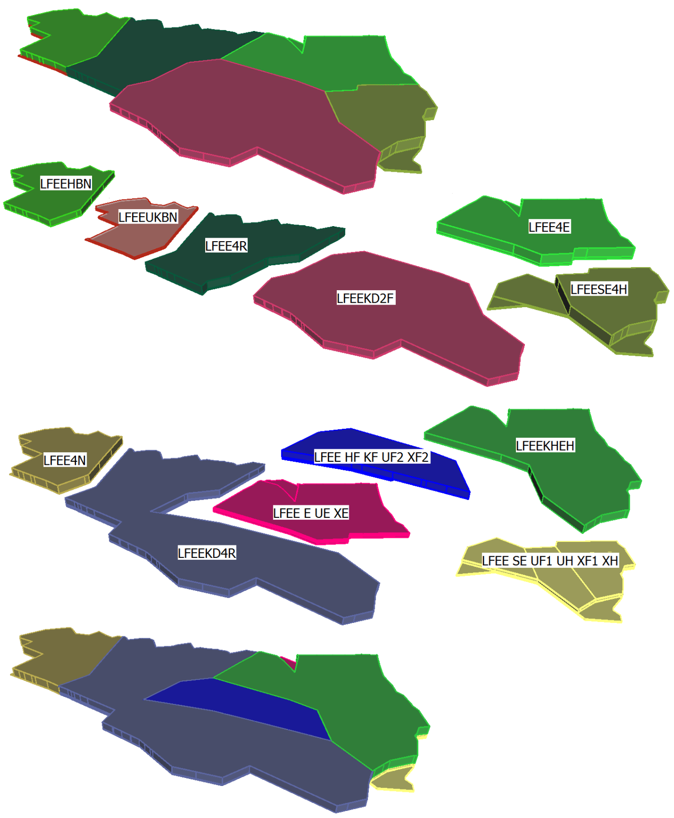


Figure 6. From the initial solution P_6^{14}, at the top, to its refinement, at the bottom

TABLE I
VALUES OF THE OBJECTIVE FUNCTIONS OF THE SOLUTION P_6^{14}

Solution step by step	Balance	Cut	Compactness
Initial solution	16.02	121.03	0.759
Unstructured solution	2.432	126.67	0.653
Well-shaped solution	8.434	111.28	0.939

TABLE II
SECTOR METRICS OF THE INITIAL SOLUTION P_6^{14}

Sector S	D_S	C_S	Compactness	Prisms
LFEESE4H	9.633	57.948	0.759	3
LFEEUKBN	4.313	18.974	1.0	2
LFEEKD2F	9.845	52.820	1.0	5
LFEEHBN	2.483	16.410	1.0	1
LFEE4R	8.417	42.051	1.0	2
LFEEE4E	9.739	53.846	1.0	2

TABLE III
SECTOR METRICS OF THE REFINED SOLUTION P_{6r}^{14}

Sector S	D_S	C_S	Compactness	Prisms
LFEEKHEH	7.394	29.743	1.0	1
HF KF UF2 XF2	5.497	52.820	0.939	2
LFEEKD4R	11.622	41.538	1.0	5
E UE XE	5.923	34.871	1.0	2
SE UF1 UH XF1 XH	7.198	42.564	1.0	2
LFEE4N	6.797	21.025	1.0	2

C. Smoothing algorithm results

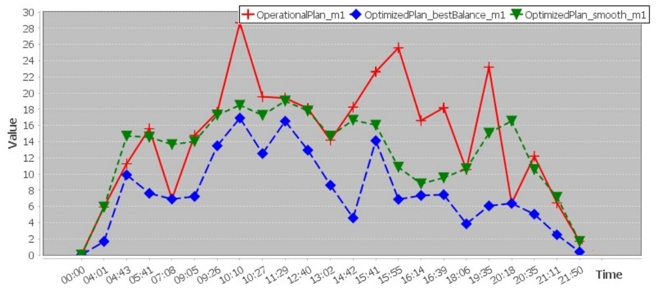
In this section, we focus on the properties of the resulting sector configuration plan. We compared this plan to the reference operational plan and a best-balanced plan. The latter is formed by juxtaposing the best-balanced configuration for each time period, without considering cut or stability objectives. The results of those different plans are presented in the following graphics.

Figure 7a shows that the balance of the resulting plan is often better than the reference plan and does not reach excessive values. The average balance gain is equal to 12.9%. The curve *OptimizedPlan_bestBalance_m1* shows that there are better balanced configurations but using those configurations will probably lead to an instability over time.

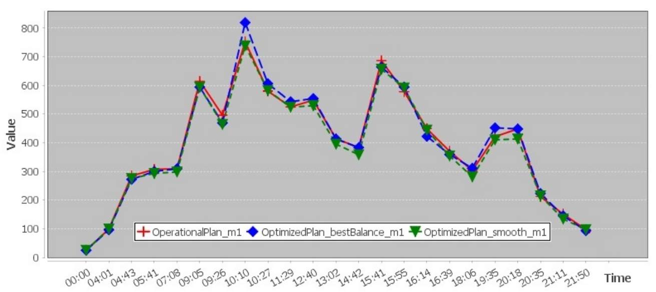
Figure 7b shows that the cut of the resulting plan is often a bit better than the reference plan. The average cut gain is equal to 3.4%. Compared to balance, it is not so easy to reach high gains because any other sectorisation would also cut flows somewhere.

As shown in figure 7c, the compactness oscillates between 0.95 and 1, a good performance, which suggests a large acceptability by controllers.

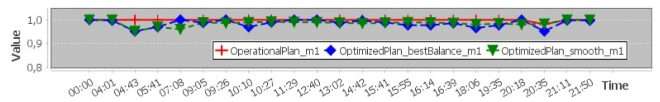
Figure 8a shows the stability of the configuration plan. The curve *OptimizedPlan_bestBalance_m1* shows that a plan with the best-balanced configurations is very unstable; there are too many differences between each sector configuration. The stability of the smooth plan is similar to the stability of the



(a) Balance

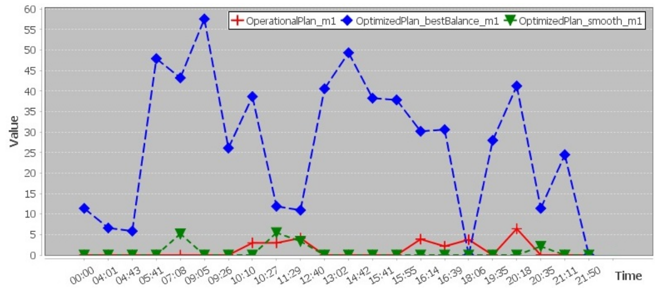


(b) Cut

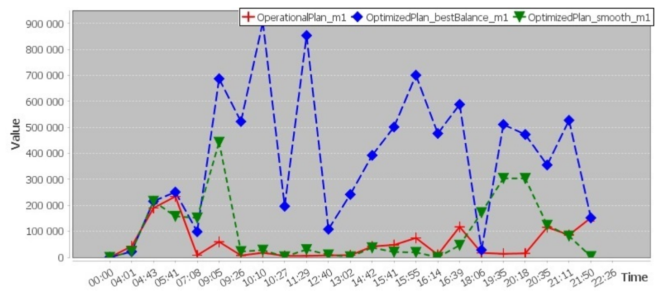


(c) Compactness

Figure 7. Balance (a), cut (b) and compactness (c) along the day



(a) Cell-based distance



(b) Hausdorff distance

Figure 8. Cell-based (a) and Hausdorff distances (b) along the day

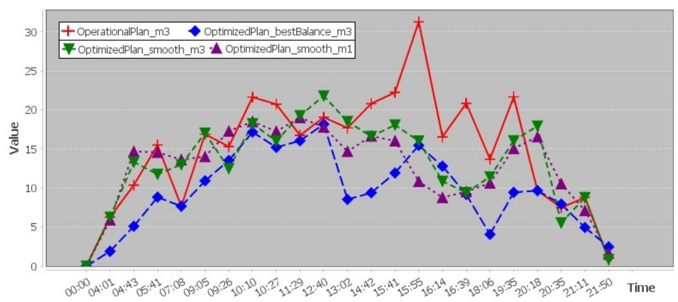


Figure 9. Balance along the day - comparison between the flight plan assessment (M1) and the trajectory assessment (M3)

reference plan. It would be possible to have a stability always equal to 0 by injecting more points in the Pareto fronts so that the smoothing algorithm has more possibilities to smooth the plan but it would degrade the balance and the cut.

To assess stability over time, we also implemented the Hausdorff metric [24] (see Figure 8b). It also shows that a plan with the best-balanced configurations is very unstable. We can see that the smooth plan is as stable as the reference plan except that there are two peaks. It would be possible to remove those peaks by considering this metric to measure the distance between two configurations in the smoothing algorithm instead of the cell-based metric. In all cases, it confirms the results given by the cell-based metric. We are able to build a plan as stable as the operational plan, as compact as the operational plan, and improving the workload distribution and the number of transfers.

Finally, we study the robustness of this plan (see Figure 9). Instead of using flight plans, we assess the balance of this plan with corresponding real trajectories. As expected, compared to the assessment with the flight plans used to elaborate the optimisation, the plan is slightly less balanced. Nevertheless, the final average balance gain is equal to 12% whereas the initial assessment gives 12.9%, which demonstrates a good robustness.

VII. Conclusion and perspectives

We described in this paper a new approach using deterministic and stochastic methods to enumerate, assess and refine sector configurations based on conventional sectors and combine them into a smooth sector configuration plan. The application of this method to the Reims scenario of the SESAR VP-755 exercise [11] demonstrates the possibility to improve several criteria, such as the workload distribution (average gain of 12%) along the day, without degrading the compactness of the configurations, and hence build a smooth configuration with reasonable transition costs.

Such algorithms could be integrated in a decision support tool provided to operational experts in charge of building and updating the sector configuration plans, notably to help them to deal with the dynamicity required by free route operations and traffic uncertainty. Enumerating all sector configurations is time-consuming but would only need to be done when a new sector design is implemented. Then conventional and optimised solutions would be rapidly provided to operational experts. Fine-tuned solutions generated by stochastic algorithms using the Simulated Annealing metaheuristic would be proposed to operational experts for addition to the catalogue, if deemed acceptable to air traffic controllers. Such an acceptance, and the associated training required, should be assessed within the framework of SESAR 2020 Advanced Airspace Management exercises. Besides, the last algorithm would help operational experts to build a smooth sector configuration plan in line with the operational need of stability for controllers and would bring more reactivity in case of crisis management.

As mentioned during the VP-755 exercise, several algorithms could hence co-exist within such as tool, depending on the considered time horizon (uncertainty of the traffic demand and necessity to rapidly compute solutions). The methods presented in this paper could support the development of some of these algorithms, but the final tool would certainly require to complement them with other methods. Many evolutions are already foreseen, such as:

- the application of this method to a larger number of blocks, for instance automatically generated by EUROCONTROL SAGA algorithm [28];
- the use of multi-objective optimisation methods applied to graph partitioning [29] to easily build Pareto fronts with multiple objectives;
- the use of an overload constraint based on occupancy "peak" and "sustain" values, that need to be defined for each new sector, in the same way as the Monitor Alert Parameter [30];
- the refinement of the flow cut objective to integrate transfers from neighbouring ACC;
- the use of trackflows to guide the selection of proposed configurations thanks to the respective weights of trajectories options for the flights, combined with complexity metrics [31].

Acknowledgement

Authors would like to thank Nathalie Bartoli, Julien Bourget and Antoine Joulia for their constructive suggestions during the reviewing of this paper. They would also like to express their gratitude to their ONERA colleagues and DSNA partners for the execution of the VP-755 exercise: Sébastien Aubry, Thomas Chaboud, Stéphane Degrémont, Rémi Lafage, Isabelle Luxembourg, Sylvain Mondou and Valérie Neyns. This study was co-financed by the European Union, through the SESAR Joint Undertaking. Authors would like to thank Kris Delcourte, project manager of the SESAR 07.05.04 project and all the European partners involved in this project.

References

[1] M. Bloem and P. Kopardekar, "Combining airspace sectors for the efficient use of air traffic control resources," in *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI*, 2008.

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

[3] P. Flener and J. Pearson, "Propagators and violation functions for geometric and workload constraints arising in airspace sectorisation," *arXiv preprint arXiv:1401.7463*, 2014.

[4] M. Drew, "Analysis of an optimal sector design method," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, 2008, pp. 3–B.

[5] D. Delahaye and S. Puechmorel, *Modeling and optimization of air traffic*. John Wiley & Sons, 2013.

[6] N. Durand, D. Gianazza, J.-M. Alliot, and J.-B. Gotteland, *Metaheuristics for Air Traffic Management*. John Wiley & Sons, 2016, vol. 2.

[7] T. Dubot, S. Aubry, and J. Bedouet, "Building a smooth and dynamic opening scheme from graph partitioning-exploring dynamic airspace configurations," in *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015, p. 3403.

[8] (2016) SESAR JU. [Online]. Available: http://www.sesarju.eu/

[9] "P07.05.04 Dynamic Airspace Configuration Step 2-V2 OSED, SESAR D54, edition 00.02.00," Tech. Rep., 2016.

[10] "P07.02 Step 1 Release 5 Detailed Operational Description (DOD), SESAR D29, edition 00.04.01," Tech. Rep., 2016.

[11] T. Dubot, J. Bedouet, and S. Degrémont, "Modelling, generating and evaluating sector configuration plans methodology report of the SESAR VP-755 exercise," in *30th Congress of the International Council of the Aeronautical Sciences (ICAS 2016)*, 2016.

[12] M. Sergeeva, D. Delahaye, L. Zerrouki, and N. Schede, "Dynamic airspace configurations generated by evolutionary algorithms," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 2015, pp. 1F2–1.

[13] K. Leiden, S. Peters, and S. Quesada, "Flight level-based dynamic airspace configuration," in *Proc. of the 9th AIAA Aviation Technology, Integration and Operations (ATIO) Forum. American Institute of Aeronautics and Astronautics*, 2009.

[14] C.-E. Bichot, "Élaboration d'une nouvelle métaheuristique pour le partitionnement de graphe: la méthode de fusion-fission. Application au découpage de l'espace aérien," Ph.D. dissertation, INP, 2007.

[15] S. M. Sale, "Development of a computer based airspace sector occupancy model," Ph.D. dissertation, Virginia Tech, 1998.

[16] D. Delahaye, J.-M. Alliot, M. Schoenauer, and J.-L. Farges, "Genetic algorithms for partitioning air space," in *AI 1994, 10th Conference on Artificiel Intelligence for Applications*. IEEE, 1994, pp. pp–291.

[17] A. Yousefi and G. L. Donohue, "Temporal and spatial distribution of airspace complexity for air traffic controller workload-based sectorization," in *AIAA 4th aviation technology, integration and operations (ATIO) forum, Chicago, Illinois*, 2004.

[18] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

[19] P. Jägare, P. Flener, and J. Pearson, "Airspace sectorisation using constraint-based local search," in *ATM*, 2013.

[20] P. Brucker, "On the complexity of clustering problems," in *Optimization and operations research*. Springer, 1978, pp. 45–54.

[21] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[22] R. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*. Citeseer, 1965, vol. 81.

[23] D. Gusfield, "Partition-distance: A problem and class of perfect graphs arising in clustering," *Information Processing Letters*, vol. 82, no. 3, pp. 159–164, 2002.

[24] A. Yousefi, R. Hoffman, M. Lowther, B. Khorrami, and H. Hackney, "Trigger metrics for dynamic airspace configuration," in *AIAA Aviation Technology, Integration, and Operations Conference, Hilton Head, SC*, 2009.

[25] N. Gregoire and M. Bouillot, "Hausdorff distance between convex polygons," *Web Project for the Course CS 507 Computational Geometry*, 1998.

[26] S. Zelinski and C. F. Lai, "Comparing methods for dynamic airspace configuration," in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*. IEEE, 2011, pp. 3A1–1.

[27] (2016) DDR repository. Eurocontrol. [Online]. Available: http://www.eurocontrol.int/ddr

[28] (2016) NEST. Eurocontrol. [Online]. Available: http://www.eurocontrol.int/services/nest-modelling-tool

[29] A. Rummler and A. Apetrei, "Graph partitioning revised - a multiobjective perspective," 2002.

[30] M. C. Drew, "A method of optimally combining sectors," in *AIAA Aviation Technology, Integration, and Operations Conference, Hilton Head, SC*, 2009.

[31] "P07.05.04 Step 2 V2 Flexible Airspace Management Validation Report (VALR), SESAR D66, edition 00.01.00," Tech. Rep., 2016.