3D Sectors Design by Genetic Algorithm Towards Automated Sectorisation

Marina Sergeeva, Daniel Delahaye, Catherine Mancel
MAIAA Laboratory, French University of Civil Aviation (ENAC)
7, avenue Edouard-Belin, 31055 Toulouse CEDEX 4, France

Leila Zerrouki, Nick Schede
EUROCONTROL Brtigny-sur-Orge, France

***Abstract*—The aim of this work is to develop a research prototype to support the validation of new airspace sector design methodology. To do this, an algorithm has been developed that manages main features of the sector design process. The proposed method is based on a mathematical modeling and heuristic optimization techniques. In order to run this algorithm efficiently a pre-processing step has been proposed, which creates an initial division of the airspace into Voronoi cells using k-means clustering algorithm. Then, due to the induced combinatorial complexity, a stochastic optimization algorithm based on artificial evolution has been applied to solve the sectorisation problem. An evaluation of the algorithm is presented as well, with a comparison to existing sectorisation with the support of the operational expertise.**

***Index Terms*—genetic algorithm; k-means clustering; sector design**

I. Introduction

This paper presents an ongoing research work performed in the scope of the SESAR program to assess the concept of the sector design, supported by automated tools. The objective is to optimize the sector design of the European airspace with the support of automation in order to increase an adaptability of sector configurations.

In comparison with a currently used fixed route network, the SESAR program introduces the *User Preferred Routing* (UPR) concept, to enable the airspace users to freely plan the 4D trajectory that best fits their business needs. On contrary to a fixed-route network, a *free route* environment will produce a larger number of different trajectories for which a new and more adaptable sectorisation should be proposed. The dynamic nature of the automatic sector design process will support most efficiently the implementation of the UPR concept.

First, we start with a brief description of the current *Air Traffic Management* (ATM) system in Europe. The airspace is partitioned into sectors, each of them being controlled by a group of *Air Traffic Controllers*. Air traffic controllers monitor the traffic and check that aircraft follow their planned trajectories. An elementary sector is defined as a volume of the airspace, within which the air traffic controller can perform his controlling function. The number of sectors in the airspace is usually determined by the capacity of one controller to manage several aircraft simultaneously. *Area Control Centers* (ACC) consists of several elementary sectors. When one elementary sector is regularly close to saturation, or significant and permanent change of traffic patterns occurs, a new sectorisation of the ACC should be proposed.

Currently airspace is redesigned by qualified operational airspace experts, who first, identify a problem in the existing sectors and then propose a new sector design, which they build manually, or with the support of visualization and *what-if* tools for the assessment of the sectors design. However, due to the complexity of the task, it requires a significant amount of time for the experts to find an acceptable and satisfying solution. The aim of the present work is to propose a mathematical modeling and heuristic optimization methods as part of a global methodology for the sector design of the European airspace, developed in the scope of SESAR project P07.05.04, co-financed by the EU and EUROCONTROL ([1]). The aim of this project is to develop a research prototype (decision-support tool) to support new sectorization methodologies based on 4D trajectories to deal with the implementation of the Free Routing concept in the short-term future. Resulting algorithms are integrated within EUROCONTROL research prototype to support SESAR validation exercises.

This paper is organized as follows: in section 2, a short literature review will be presented; in section 3, a mathematical model of the problem of the airspace sector design is proposed. In this section, we also describe in detail the creation of initial blocks that are used for the design of elementary sectors. In section 4, a GA approach for the sectorisation problem is described. Finally, in section 5, results and operational analysis will be presented.

II. Previous Related works

A number of different methods are proposed in literature which solve airspace sectorisation problem [2], [3]. The choice of the appropriate algorithm derives from the way the sector design process is defined. Existing approaches can be divided on approaches that are aiming to change existing sectors borders and on approaches that start from "scratch". Most of the existing works are concentrated on a 2D sector design, while only few studies include the third dimension [4], [5]. Different optimization methods have been used in research of the sector design problem. The most common methods are Constraint Programming [6], [7], Mixed Integer Programming [8]–[10], Clustering Algorithms [11], Evolutionary Algorithms (EAs) [4], [12]–[14], Computational Geometry [15], Global optimization and several other techniques [16], [17]. Works can also vary in the way how the ATC workload is quantified during the sectors evaluation process. Several complexity metrics have been proposed in works [18], [19] to evaluate the workload of the airspace sectors.

Here, is a short overview of the most promising methods found in recent literature. Research works [4] and [13] use Genetic algorithm (GA) for sectors design. In work [4], on a first step, airspace is divided into elementary cells using 2D Voronoi diagram, where each center of cell is a conflict point between two aircraft trajectories. At the beginning of the second step, centers of sectors are chosen randomly. Then, sectors are composed from Voronoi cells during an association process. Resulting sectors are supposed to be balanced, regarding the workload, due to the optimization process. A fitness function used in this work includes the total workload imbalance and the total flow cut (the transfer traffic between neighboring sectors). Almost the same approach is proposed in [13]. In this work, a methodology based on the Voronoi Diagram and Genetic Algorithm (GA) is investigated. The Voronoi Diagram is applied to divide the airspace into a group of convex polygons with no overlap. Genetic Algorithm is used to perform the multi-objective optimization. In this work, the following objectives were included: minimizing monitoring workload variance, minimizing coordinating workload, and maximizing dwelling time in a sector.

In recent research, a big attention has been paid to methods based on Constraint Programming [3], [5]–[7]. Authors of [6] have studied constraints that arise in airspace sectorization. In this work, authors give for each constraint an analysis of what algorithms and properties are required under systematic search and stochastic local search. Under stochastic local search, efficient algorithms are proposed for maintaining constraint and variable violations, as well as efficient algorithms for probing the effect of local search moves. The main limitation on using the Constraint Programming is that it works satisfactorily only with small instants of the problem [7]. It is not easy to adapt it to a 3D extension of sectors design as well [5].

Most of the previous approaches do not take into account the third dimension as well as some important airspace design aspects, such as shape of the sector. As a matter of fact, one must be able to generate sectors that have to be balanced in terms of the workload and operationally acceptable by the airspace experts in terms of shape.

III. Problem modeling

First, a sector design problem should be explicitly defind. The problem description presented here is developed according to EUROCONTROL requirements and based on operational expertise of sector design [1].

A. Problem description

Given a forecast air traffic demand, the airspace sectorization problem consists in searching a partition of a given airspace domain D into a set of N operationally workable sectors [s1, ... ,sn], so as to minimize some cost function. A sector design is a process which delineates shape of the sectors in order to optimize performance objectives and fulfill operational constraints. Moreover, sectors have to satisfy some geometrical and operational constraints in order to be accepted by ATC experts.

The quality of the airspace sectorization can be evaluated according to several kinds of criteria. In this work, the cost function includes the following criteria:

- The imbalance between the workload of the resulting sectors.
- The coordination workload.
- The number of flight re-entry events.
- The number of conflict points close to the sector borders.
- The number of short transits flight through sectors.

In our model, sector shapes have also to satisfy the following constraints:

- An airspace sector must be a continuous portion of the airspace and should not be composed of disconnected blocks.
- A sector shape should follow main traffic flows and preferably be a convex polygon.
- A sector should span over several flight levels, but should not have too much different lateral shapes at each altitude layer.
- Shapes of sectors such as ”stairs” or ”balconies” should be restricted or avoided.

Next step consists in modeling the problem using a mathematical abstraction which should be as faithful as possible. The main challenge, which researchers of sectorisation domain have to face, is the development of a relevant mathematical model of the airspace.

B. Airspace modeling

Let’s consider a given airspace, represented as a 3D polygon. The airspace area is split into several altitude layers, each layer includes at least 5 Flight Levels (FL). The shape of this polygon can vary depending on the altitude level. A set of aircraft trajectories that are crossing this area is known as well. Aircraft trajectories are computed using flight plans. For the purpose of the airspace sectorisation, traffic data are usually taken just for several peak hours of the day(s), when the workload is the highest.

Our model is based on a discretization of the 3D airspace into hexagonal or square cells, with a size less than 5nm (in vertical direction $<$ 5 FL). The aim of the discretization of the airspace using grid-cells is to simplify the problem and to reduce the computational cost. We need to determine the distribution of the traffic complexity i.e., the distribution of the ATC workload in 3D space using a known traffic data. Instead of working with a list of all trajectories, for each grid cell the aggregated workload is computed. For the same reason of simplification, each trajectory instead of being represented as a list of samples is represented as a list of cells with a time line.

The metric of the ATC workload that is used in this work is based on two metrics: flights crossing time and a conflict count. The crossing time cumulated for all aircraft inside the airspace volume is multiplied by the time required by the controller to monitor one aircraft per minute flown in the area (e.g. 3 seconds

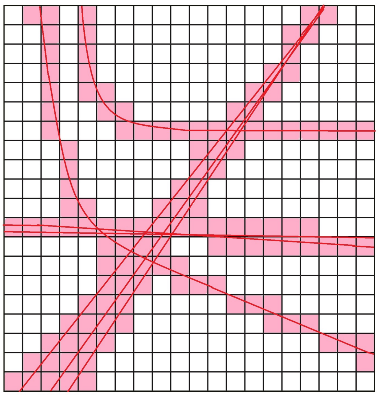

Fig. 1. 2D Projection of air traffic on the grid

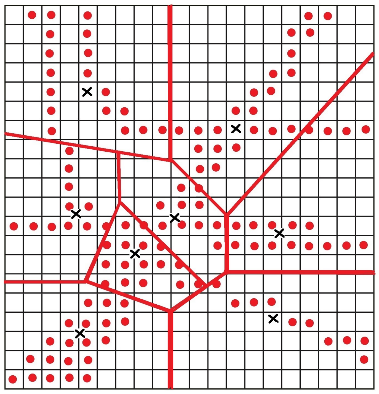

Fig. 2. Voronoi cells construction. Projections of the centers of loaded cells are shown with points and the associated cluster centers are shown with crosses

per 1 minute flown). The conflict count metric is computed as a sum of a total number of conflicts inside the area multiplied by the time required by the controller to solve one conflict. A bigger resolution time is used for the conflicts located close to the border of a sector (entry-conflicts).

Having the workload associated with each grid cell, it is possible now to aggregate cells into elementary sectors. However, computational cost required for the cells aggregation process will be high and shapes of resulting sectors may not be satisfying (non-convex). Instead, we propose to reduce the number of cells by aggregating them into bigger cells, using Voronoi diagram [20], which have a shape of convex polygons.

C. Preparing airspace blocks for sector building process

The traffic is not equally distributed in the airspace, therefore, areas, where the traffic load is low, should be divided on cells with a bigger size. On the other hand, in areas with high traffic, cells should be smaller, in order to be more flexibly combined during the sector design process. To reach this goal, we propose to create a mosaic of cells using k-means clustering algorithm and Voronoi Diagram. The size of a cell, created by the algorithm, will depend on the level of the traffic complexity associated with such a cell.

Having the set of grid cells with their associated workload, the objective of the k-means clustering algorithm, is to group those cells into bigger cells (Voronoi cells), with the size that varies depending on the workload in this area. To reach this goal, grid cells are first projected on the 2D plane corresponding to the ground (see Fig. 1). The workload of each projected cell is then computed as a sum of all cells that have the same ground index, i.e. same horizontal coordinates. In the clustering process we use only loaded cells (with workload > 0). We compute centers of a Voronoi cells using the k-means clustering algorithm, as geometrical barycenters of loaded grid cells (see Fig. 2). Based on the positions of computed geometrical barycenters, each grid cell is aggregated to its nearest center, designing a Voronoi diagram represented in Fig. 2. As a matter of fact, this clustering process indirectly ensures that the main flows and the crossing point between trajectories will be located in the middle of the new cells.

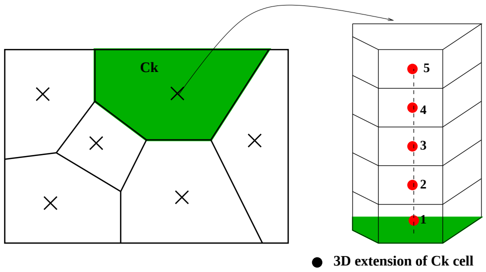

Fig. 3. Extension of Voronoi cells in the third dimension

The Voronoi diagram is built in 2D and then extended in the third dimension as shown in Fig. 3, leading to a set of airspace building blocks that covers all the given airspace. Building blocks do not change their shape depending on the layer, however the workload of blocks, is different on each altitude layer.

D. State space modeling

The state space represents a set of parameters of the system upon which we may act in order to optimize one (or more) objective(s). Examination of the properties of the state space helps us to choose a suitable optimization method.

First, we characterize data that will be used as an input of the sector design algorithm. During the optimization process we should be able to evaluate the workload of each sector built from a set of blocks. We should also be able to check if blocks belonging to the same sector are connected. In order to do that, we build a 3D graph (see Fig. 4), in which each node represents the center of the building block and each link represents the relation *is neighbor with* between two blocks. Nodes and links of the graph are loaded with weights. Nodes are loaded with cumulated complexity of the associated grid cells and links with an air traffic flow i.e. the number of aircraft passing from one block to another one. Each link also carries the number of conflicts that are close to the border between two neighboring blocks. Weights of nodes and links are computed separately for each altitude layer. The constructed 3D graph is only used during the evaluation of the results, and not in a process of

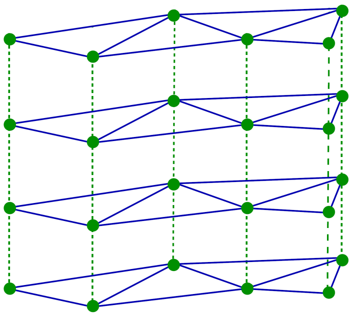

Fig. 4. A 3D graph, for which each node represents a building block (center of the Voronoi cell) and each link represents the connection between two nodes on the same level

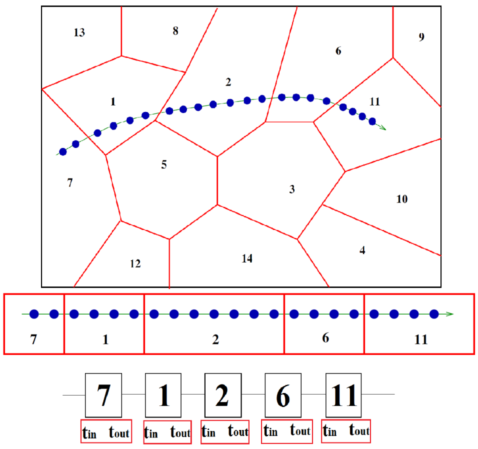

Fig. 5. List of blocks associated with each trajectory

building elementary sectors.

Once the 3D graph is built, the traffic data should be modeled in order to be able to evaluate re-entries and short transits criteria during the sector design process. To compute the number of re-entries events, we need to check if any trajectory enters twice the same sector. In the same manner we compute the short transits: for each trajectory we register the time that the aircraft has spent in the sector. In order to compute this in an optimized way, we propose to summarize each aircraft trajectory by the list of blocks, crossed by this aircraft with registered entering and exit time (see Fig. 5). Based on this list of blocks, it is easy to check if an aircraft enters twice the same sector and if it stays too shortly in it.

Now, we briefly describe how elementary sectors are built during the optimization process. First, we consider positions of the N centers of blocks in the airspace as shown in Fig. 6. Those positions are included in the state space. Sectors are constructed from blocks using sector centers, which are computed during the optimization process. One sector can occupy different number of layers and can be built from different number of blocks at each layer. The sectorization process starts from associating each block to its nearest sector center at each layer (see Fig. 6).

Block centers and sector centers enable the two dimensional

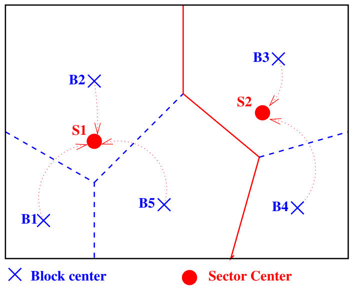

Fig. 6. Sectors building process. Each block center is aggregated to its nearest sectors center

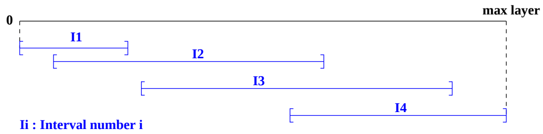

Fig. 7. Altitude interval set covers the whole altitude range

design of sectors. For the third dimension design, a set of altitude intervals are used. This set ensures altitude layer covering. Each sector center S_k addresses a limited altitude interval I_k. The intervals set covers all range of altitude layers(see Fig. 7).

E. Connectivity Constraint

To ensure that sectors are composed of connected blocks, we run a test to check if sectors are connected or disconnected using the modeled 3D graph just after sectors delineation. We have developed a graph coloring algorithm which is applied on our 3D graph in order to highlight disconnections between parts of resulting sectors. After the aggregation process, each node is associated with one of the sectors, so that for each sector it is possible to know the exact number of nodes included in it. Then for each sector, we first take one of its nodes in one layer and associate it with a color accordingly. Then, using known neighborhood of this node we propagate its color to each neighbor node, if it belongs to the same sector (color is propagated in other layers as well). This step is repeated for each new colored node. At the same time we compute the number of nodes that receive the same color. This number is then compared with the actual number of nodes inside the sector, and if those two numbers are different we indicate a disconnection.

F. Objective function

The objective function as well as the constraints included in our model are designed according to EUROCONTROL requirements and developed jointly with the operational experts. Six criteria have been included in our objective function for evaluation of a solution. The first criterion measures the level of the workload imbalance among sectors. The workload imbalance of the sector can be modeled by the following formula :

$$\delta(k) = \frac{|w(k) - \frac{W}{K}|}{\frac{W}{K}} \quad (1)$$

where $W = \sum_{k=1}^{K} w(k)$ is the capacity, K is the number of sectors and $w(k)$ is the workload of the sector k.

The capacity is equal to an average value of the workload and is computed as a total workload of the airspace divided by the desired number of sectors. Then, the total workload imbalance of all sectors is given by:

$$y_1 = \bar{\Delta} = \sqrt{\frac{\sum_{k=1}^{K}(\delta(k))^2}{K}} \quad (2)$$

The second criterion measures the coordination workload between neighboring components. When two neighboring blocks belong to different sectors on a considered layer, the traffic flow between them is cut by the sector border, as a result, the coordination workload is increasing. The total flow cut between cells is computed as follows :

$$F_T = \sum_{z=0}^{z=N_Z} \sum_{l=0}^{l=|\mathcal{L}|} f_l^z \quad (3)$$

where N_z is the total number of altitude layers, $|\mathcal{L}|$ is the total number of 2D links in the graph and f_l^z is the flow crossing the link l in the altitude layer z.

Then, the total flow cut is given by:

$$F_c = \sum_{\substack{l,z|O(l^z) \in s_k \\ D(l^z) \notin s_k}} f_l^z \quad (4)$$

where $O(l^z)$ represents the origin and $D(l^z)$ represents the destination of the link l in the altitude layer z and s_k represents the sector k.

Finally, the four following criteria are computed: the number of re-entries events (y_3), the number of short transits inside each sector (y_4), the number of conflicts located too close to the borders of sectors (y_5) and the the number of balconies in each sector (y_6). It is relatively easy to compute efficiently the number of re-entries y_3 and the number of aircraft with short transit time inside one sector y_4 using trajectories that are represented by its associated list of blocks with entering and exit times. The same concerns the computation of the number of conflicts close to the sectors borders y_5. It is computed almost the same way as we have computed the flow cut, but using the second value of links weight.

All those criteria are normalized in order to have values $\in \{0, 1\}$ and aggregated into one objective function :

$$y = \alpha_1.y_1 + \alpha_2.y_2 + \alpha_3.y_3 + \alpha_4.y_4 + \alpha_5.y_5 + \alpha_6.y_6 \quad (5)$$

IV. Application of EAs to the sector design problem

Based on the airspace model described above, the sectorization problem is simplified to a task of finding centers of the sectors. The complexity of this problem is linked to the number of blocks N_b, the number of elementary sectors N_s and also to the number of altitude layers N_l. The number of combinations for grouping such N_b blocks multiplied by N_l into N_s blocks is given by the second Stirling number [21]. In our work, we consider that the number of initial blocks is more then 100 and the number of layers is bigger then 2, so the number of possible combinations can become extremely high. This makes impossible the application of the mathematical programming, instead it is more realistic to choose algorithms based on ordered and targeted random search. Typically, this type of problems falls under the category of NP-hard problems and stochastic optimization is a good candidate to address it. Moreover, this problem can have several different optimal solutions, due to the different possible symmetries in the topological space. Thus, one must be able to find most of them, as they have to be evaluated and refined by experts. EAs maintain and improve a population of numerous state variables according to their fitness and are able to find several solutions. On the other hand, EAs would be a best choice if we would like to extended a developed model to multi-objective. EAs appear to be faster then other methods, such as, for example, a simulated annealing, applied to our problem. Thus, EAs is relevant to solve the sector design problem.

A. Coding the chromosome

EAs [22], [23] use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover) to find an approximate solution of the optimization problem. An individual (a solution of the problem), is represented by a list of parameters, called chromosome. The chromosome used in our algorithm is defined as a set of sector centers and has the following structure :

x_1	x_2	...	x_i	...	x_{N_s}
y_1	y_2	...	y_i	...	y_{N_s}

$Ext1_{min}$	$Ext2_{min}$	...	$Ext_{i_{min}}$	...	$Ext_{N_s-1_{min}}$
$Ext1_{max}$	$Ext2_{max}$	...	$Ext_{i_{max}}$	...	$Ext_{N_s-1_{max}}$

Here, the first table represents coordinates of sector centers and the second table contains the associated vertical extension of each sector center.

The initial population of solutions is generated randomly. First, coordinates of sectors are generated randomly and then, altitude layer intervals are generated for each sector center. In order to generate covering of the full altitude layer range we use a set of randomly generated $N_s - 1$ markers (represented by min and max altitude layer), sorted in ascending order as shown in Fig. 8. A marker i is specified by its minimum altitude layer M_i^{min} and its maximum altitude layer M_i^{max}. In Fig. 8, a construction process of altitude intervals is described. In this figure, five sectors are generated. For designing the associated altitude intervals, four markers are used and ranked by maximum altitude layer (M_i^{max}). The first altitude interval I_1 starts from the lowest layer 1 and ends at $M_1^{max} = 5$, the second interval starts at $M_1^{min} = 2$ and ends at $M_2^{max} = 6$ and etc. Using those generated altitude intervals that fully cover range of layers, we can partition the airspace into sectors as requested by the operational constraint presented in the previous section.

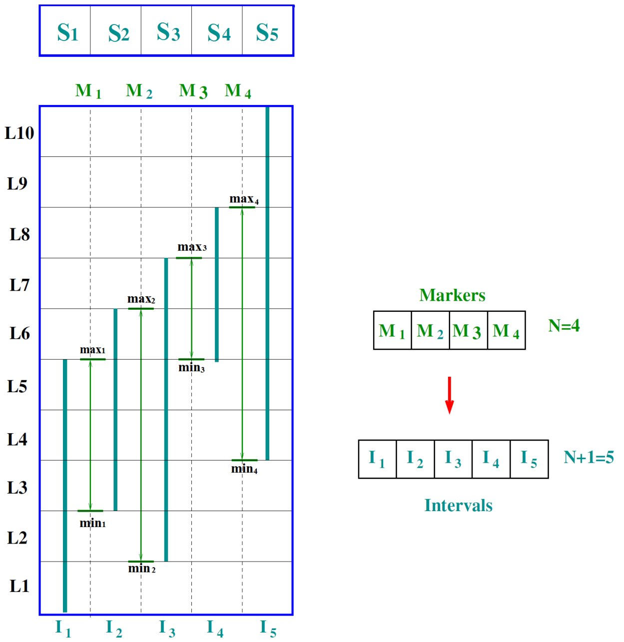

Fig. 8. In this figure, 5 sectors are generated. For layer 1, all nodes are associated to sector 1, for layer 2, all nodes are shared between sector 1 and sector 3, etc...

B. Sector building process

After generating initial chromosome, we build sectors using coordinates of centers of blocks. Each block (node) is aggregated to its nearest sector center in the considered altitude level. We can aggregate the node only to the center that locates on the same level with it.

After creating the first population, each solution is evaluated, and a value of fitness is returned by a fitness function. This initial population undergo a selection process that identifies the best solutions, which then constitute an intermediate population. Then, three following recombination operators are applied to individuals of the intermediate population : *nothing, crossover,* or *mutation*. The associated probability of application are respectively $(1 - p_c - p_m))$, p_c and p_m. These processes ultimately result in the next population of chromosomes $POP(k + 1)$ that is different from the initial generation. This generational process is repeated until a termination condition is reached.

Several different crossover operators have been developed, which can be divided on strong and weak operators. Crossover results into two new child chromosomes, which are added to the next population. Chromosomes of two parents are mixed between each other during crossover. However, mutation operators have shown to be more efficient in a solution search process. The purpose of the mutation in EAs is to allow the algorithm to avoid local minimum by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. First strong mutation operators are applied in order to obtain a wide range of results, which after some generations will narrow to the stack of near optimum solutions. Then, weak mutation operators are applied in order to obtain final results.

V. Simulations and Results

After the modeling and algorithm descriptions above, this section presents results of the sectors design algorithm applied to Maastricht UAC (Amsterdam UTA and Northern part of Hannover UIR).

A. Scenarios description

Two scenarios have been used to test the algorithm. In each scenario we are using different values of the main parameters. Results are obtained using free route simulated trajectories, which provide a sample of full free route trajectories for the 11th of July 2014 crossing Maastricht/Amsterdam Airspace (EDYYDUTA).

The reference scenario, to which the algorithm will be compared to, is the sector configuration proposed by ICO tool (Improved Configuration Optimizer). This tool computes the optimum sequence of sector configurations for a given traffic day on the basis of existing elementary sectors. For the considered time period, 19:00-20:00, the sectors proposed by ICO is the configuration 5.5.1, composed of 5 sectors. The ICO tool is implemented within NEST simulator, the Network Strategic Tool developed by EUROCONTROL.

Two solutions scenario were obtained using the algorithm, run with the parameters presented in table I.

TABLE I
Parameters for two scenario

Parameter	Scenario 1	Scenario 2
Nb of sectors	5	5
Workload balancing	0.45	0.35
Flow cut	0.15	0.15
Entry-conflict	0.4	0.4
Short-crossing	0.35	0.35
Re-entering	0.35	0.35
Balconies minimization	0.6	0.6
Layers cuts(FL)	245-345-355-365	245-345-355-365

This table shows that the algorithm shall design for both scenario 5 sectors based on the weights of 6 criteria that have been presented in the objective function description section and displayed in the table from line 2 to 7. The last line indicates that the algorithm uses 4 layers to design the sectors vertically.

B. Results Analysis

Sectors design algorithm has been implemented in C++ within the ASTAAC tool [24]. The results obtained for the first scenario are presented in table II and Fig. 9, 10.

The results allow assessing the quality of the designed sectors based on the criteria of workload balancing, avoiding entry-conflicts, number of the re-entries and short-crossing flights in order to design the most acceptable sectorisation from the operational viewpoints and also the optimum sectorisation with regards to sectors overload.

The comparison using similar criteria with the reference scenario (see in table II and Fig. 11, 12) shows that the solution

TABLE II
RESULTS OF THE ALGORITHM FOR THE SCENARIO 1

Sec tor	Work load	Imbal ance	Nb Conf.	Nb Entry Conf.	Ent ries	Re-entr.	Short Cross.
1	3138	0.15	4	0	75	0	2
2	2438	0.1	1	0	66	0	1
3	2533	0.07	3	0	80	0	3
4	2821	0.03	6	1	60	0	3
5	2719	0.004	2	0	58	0	2

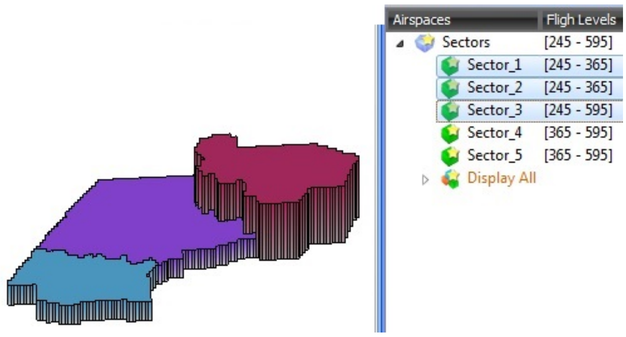

Fig. 9. Designed sectors for the scenario 1, lower layer. Sectors 1 (blue), 2 (violet) and 3 (purple)

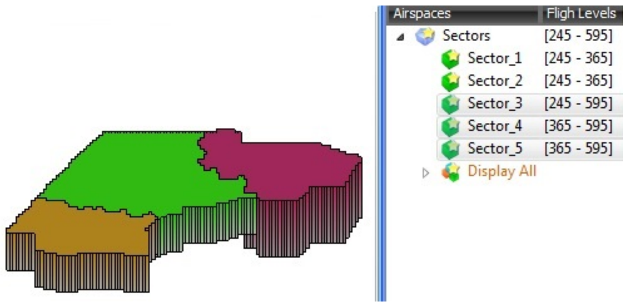

Fig. 10. Designed sectors for the scenario 1, upper layer. Sectors 4 (brown), 5 (green) and 3 (purple)

1 offers a much more balanced workload (expressed in seconds of work). The solution for the first scenario includes one entry-conflict (sector 4), which is acceptable. The number of short crossings is higher for the solution scenario, however this mainly derives from the fact that sectors are constructed from smaller cells. Smoothing of the borders will remove some of those short transits through the designed sectors.

In terms of shape and size of the sectors, first, lateral shapes of the designed sectors show some similarities with the reference sectors, main differences are with DFLs (Division Flight Level) selected by the algorithm. Secondly, a comment regarding the reference sectors: whilst the volumes exist, they would not be configured with different DFLs, as such this reference scenario does not exist. This could lead to the conclusion that the designed sectors with the selected DFLs would show even greater improvement over a really existing scenario. The difference in size of the designed sectors (sector 1 and 2) DFL 245-365 would indicate that conflicts and/or traffic are truly unevenly distributed.

While the solution for the scenario 1 seems to provide better

TABLE III
RESULTS FOR THE REFERENCE SCENARIO

Sec tor	Work load	Imbal ance	Nb Conf.	Ent ries	Re-entr.	Short Cross.
1	2403	0.11	1	68	0	4
2	1546	0.43	2	49	0	0
3	2988	0.1	2	67	0	0
4	5378	0.98	11	96	1	0
5	1213	0.55	0	47	0	2

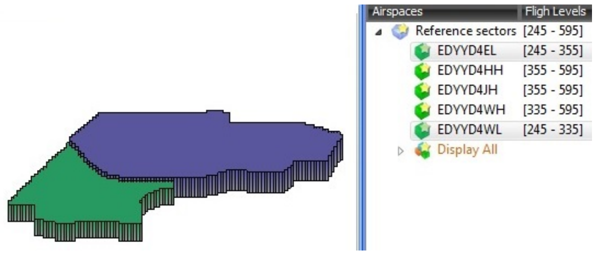

Fig. 11. Proposed sectors for the reference scenario, lower layer

Fig. 12. Proposed sectors for the reference scenario, upper layer

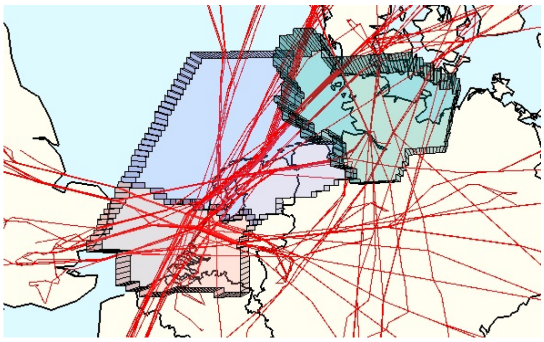

Fig. 13. Trajectories crossing designed sectors 1 (pink), 2 (light blue) and 3 (blue)

results than the reference scenario, Fig.13 shows that a large amount of trajectories are close to a boundary of the designed sector 1 (pink one). This is not ideal but this criterion has not been specified as part of the parameters of the algorithm and consequently it could not be taken into account.

Nevertheless, solution for the scenario 2 (Fig. 15), which uses the same parameters as for the scenario 1, except the weight of workload balancing criterion, offers another design of the sectors which avoids trajectories along the sector boundaries(in Fig. 14 sector number 5 in blue).

Solution for the scenario 2 (see table IV) maintains satisfy-

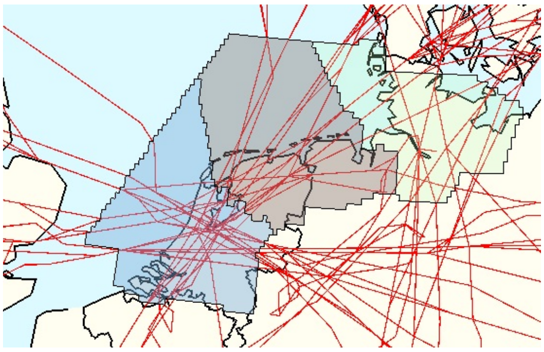

Fig. 14. Trajectories crossing designed sectors: in top view sector 5 (blue), sector 3 (brown), sector 4 (light green)

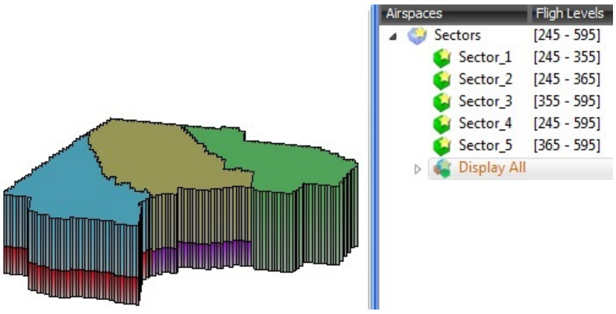

Fig. 15. Designed sectors for the scenario 2. Sectors 1 (purple), 2 (red), 3 (green), 4 (brown) and 5 (blue)

ing results regarding entry-conflicts, re-entries, short-crossing flights criteria, while sectors workload balancing is considerably deteriorated. This means that the algorithm is very sensitive to the parameters values and can be adapted to provide different sectors shape with different sectorisation performance, depending on the requirements and operational preferences.

TABLE IV
RESULTS OF THE ALGORITHM FOR THE SCENARIO 2

Sec tor	Work load	Imbal ance	Nb Conf.	Nb Entry Conf.	Ent ries	Re-entr.	Short Cross.
1	1726	0.36	0	0	53	0	0
2	3453	0.27	5	0	75	0	2
3	2863	0.06	2	0	67	0	2
4	2521	0.07	3	0	82	0	0
5	2965	0.09	6	0	61	0	2

VI. Conclusion and Next Steps

The modeling approach and algorithm solution presented in this paper have been tested and compared to existing sectorisation. The provided results demonstrate that the proposed sector design algorithm is able to provide very satisfying sectorisation with regards to sector load balancing, as well as to the number of entry-conflicts, the number of re-entering and short-crossing flights criteria. Even though operational expertise is still needed to validate the workability and acceptability of the designed sectors, the algorithm is able to offer a multitude of automated design options using different parameters values that the users can calibrate according to their own business needs and operational working preferences.

Acknowledgment

We would like to thank Nicolas BOULIN from the NEST team for his permanent support with the simulation tools and also Russell GARNHAM who provided valuable inputs, as operational expert, for the specification of the algorithm.

References

[1] S. W. project, "Dynamic airspace configuration," vol. OSED step2/V2.

[2] P. Flener and J. Pearson, "Automatic airspace sectorisation: A survey," *CoRR*, vol. abs/1311.0653, 2013. [Online]. Available: http://arxiv.org/abs/1311.0653

[3] C. Allignol, N. Barnier, P. Flener, and J. Pearson, "Review: Constraint programming for air traffic management: A survey," *Knowl. Eng. Rev.*, vol. 27, no. 3, pp. 361–392, july 2012. [Online]. Available: http://dx.doi.org/10.1017/S0269888912000215

[4] D. Delahaye and S. Puechmorel, "3d airspace design by evolutionary computation," in *Proceedings of 27th digital avionics systems conferencen*. IEEE, 2008.

[5] P. Jagare, *Airspace sectorisation using constraint programming. Masters thesis*, 2011.

[6] P. Flener and J. Pearson, "Propagators and violation functions for geometric and workload constraints arising in airspace sectorisation," *CoRR*, vol. abs/1401.7463, 2014.

[7] H. Trandac, P. Baptiste, and V. Duong, "Optimized sectorization of airspace with constraints," in *Proceedings of the ATM seminar*, 2003.

[8] S.-L. Tien and R. Hoffman, "Optimizing airspace sectors for varying demand patterns using multi-controller staffing," in *Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar*, 2009.

[9] D. M, "Analysis of an optimal sector design method," in *Proceedings of the Digital Avionics Systems Conference*, 2008.

[10] G. B. Chatterji and D. M, "Air traffic sector configuration change frequency," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.

[11] C. R. Brinton and P. S, "Airspace partitioning using flight clustering and computational geometry," 2008.

[12] O. Degtyarev, V. N. Minaenko, and M. O. Orekhov, "Solution of sectorization problems for an air traffic management area. ii. development of sectorization algorithms," 2010.

[13] M. Xue, "Airspace sector redesign based on voronoi diagrams," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 12, pp. 624–634, 2009.

[14] D. Delahaye, M. Schoenauer, and J.-M. Alliot, "Airspace sectoring by evolutionary computation," in *Proceedings of the IEEE International Congress on Evolutionary Computation*. IEEE, 1998.

[15] A. Basu, J. Mitchell, and G. Sabhnani, "Geometric algorithms for optimal airspace design and air traffic controller workload balancing," *Journal of Experimental Algorithmics (JEA)*, vol. 14, no. 3, 2009.

[16] B. Paul, W. Theeck, J. Himmelspach, A. Uhrmacher, and D. Flugsicherung, "An efficient and effective partitioning algorithm for air traffic management," in *Proceedings of the 3rd EMSS*, 2006.

[17] A. Klein, "An efficient method for airspace analysis and partitioning based on equalized traffic mass," in *Proceedings of the 6th Seminar*, 2008.

[18] A. Masalonis, M. Callaham, and C. Wanke, "Dynamic density and complexity metrics for realtime traffic flow management," in *Proceedings of 5th USA/Europe Air Traffic Management Seminar*, 2003.

[19] P. H. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhode, "Airspace complexity measurement: An air traffic control simulation analysis," in *Proceedings of the 7th USA/Europe ATM 2007 RD Seminar*, 2007.

[20] R. Klein, "Voronoi diagrams and delaunay triangulations," pp. 1–5, 2015.

[21] R. Graham, D. Knuth, and O. Patashnik, *Concrete Mathematics*. AddisonWesley, Reading MA, 1988.

[22] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New-York, 1991.

[23] Z. Michalewicz, *Genetic algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.

[24] Eurocontrol, "Nest." [Online]. Available: https://www.eurocontrol.int/services/nest-modelling-tool