3D airspace design by evolutionary computation

Daniel Delahaye, Stéphane Puechmorel

3D AIRSPACE DESIGN BY EVOLUTIONARY COMPUTATION

Daniel Delahaye and Stephane Puechmorel ENAC 7, Ave Ed Belin 31055 Toulouse France

Abstract

This paper presents a new method for 3D cutting of geometrical space with application to airspace sectoring. This problem comes from the air traffic management but the proposed method may be applied to many other areas. This problem consists in finding a cutting of a 3D volume into sectors in order to balance the weights of sectors and which minimizes the flow cut on sector boundaries. A mathematical modeling of this problem has been proposed for which state space, objective functions and constraints are defined. The complexity of such problem being NP_Hard, stochastic optimization have been used to address it. An Evolutionary Algorithm has been implemented for which chromosome coding and operators have been developed. Realistic problem instances have been tested on this algorithm for which the solutions produced fulfill our objective.

Introduction

Air traffic controllers monitor the traffic and check that aircraft follow their planed trajectories. When two or more aircraft are converging to the same point, the controllers have to change the heading, the speed or the altitude of some aircraft in order to ensure a minimum distance between all aircraft. Everyday about 8000 aircraft are flying in the French airspace and such amount of traffic generates a huge control workload which cannot be managed by a single controller (as any human being, a controller has working limits and may usually be able to manage a maximum of 15 aircraft with 3 conflict points); the airspace is then partitioned into sectors each of them being assigned to a controller team. When airspace is partitioned into sectors, several rules have to be applied in order to insure the safety and an effective traffic flow. The workload in sectors has to be balanced in order to ensure that controllers have roughly the same amount of work to do in all sectors. When an aircraft crosses a sector boundary, controllers in charge of those neighboring sectors have to exchange information with pilot and between each other in order to insure a safe transfer of the flight between sectors. This transfer is called a coordination of an aircraft between sectors. The way airspace is partitioned may induce different amount of coordination and a good sectoring has to minimize this number of coordination because it cost efforts to the controllers and to the pilots too. Those objective have to be optimized by taking into account several operational constraints. The first constraint ensures that sectors are convex in the airway link directions (An airway is materialized by a list of linear links). By this mean, a pilot will not encounter the same controller twice during his flight. This constraint will then prevent superfluous coordinations; this means that an aircraft crossing a sector will encounter 2 and only 2 sector boundaries (*route convexity constraint*). The second constraint will ensure that conflict point will be located at a minimum distance from the sector boundaries. Aircraft have to follow airways and conflict between aircraft may appear only at airways crossing. A sector boundary has to be at least at a given distance from each airways crossing (*safety constraint*). When a controller has to solve a conflict, he needs a minimum amount of time to develop a solution. Each controller managing individually his sector, if a sector boundary is too close to a crossing point, he is not able to solve any conflicts because he has not enough time between the coordination step (with the previous sector where the aircraft comes from) and the time the aircraft reaches the crossing point. The minimum delay time is fixed at 7 minutes and can be converted into a distance once the aircraft speed is known. The third constraint will ensure that aircraft stay enough time in sectors for controllers to be able to manage them(*min stay time constraint*). As a matter of fact, if an aircraft stays a few seconds in a sector, the controller will not be able to manage such aircraft; then it brings only superfluous coordination and the controller is not able to produce any services for this kind of aircraft. We express this constraint by a minimum distance between two boundaries cutting the same airway. Finally, a fourth constraint will limit the possible shape of synthesized sectors (*geometrical constraint*). Controllers work on a two dimensional

radar screen with aircraft tag giving the altitude of traffic. This HMI limitation may induce uncertainty on the sector which contains the plane. As a matter of fact, if the sector border are not perpendicular to the ground in the third dimensions, controller may have problem to know if an aircraft is still in a neighbor sector or already in its own sector. This problem is shown in the Figure 1.

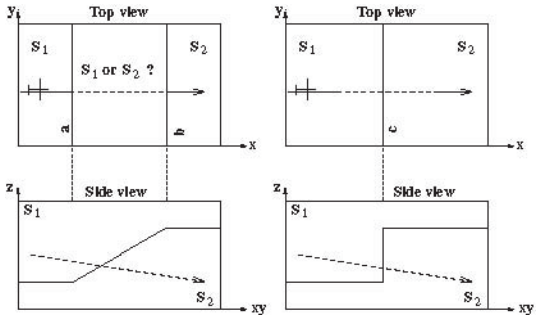

Fig. 1. Vertical Border Constraint

In Figure 1, the second situation (right side) is easier to manage from the controller point of view than in the first one (left part) because of the shape of the border in the side view.

Figure 1 shows an aircraft crossing two sectors *S*1 and *S*2. The upper left, shows the radar top view of sectors *S*1 and *S*2. The dash line represents the trajectory of an aircraft going from *S*1 into *S*2. The bottom left shows the side view of *S*1 and *S*2 with the aircraft descending from *S*1 into *S*2. As it is shown on this figure, the sector border is not perpendicular to the ground (xy). The controllers having only the top-view of sector, they have difficulties to identify the point of coordination when the aircraft is exiting *S*1 and entering *S*2. On the left of line "a" the aircraft is in sector *S*1, on the right of line "b" the aircraft is in sector *S*2 but between "a" and "b" it is difficult to know when the aircraft will change of sector mainly if it is descending or climbing. The way to solve this problem is to ensure that sector boundaries are perpendicular to the ground in the third dimension as it is shown in the right part of Figure 1. The bottom right shows the new structure of the vertical border which is now perpendicular to the ground. The upper right shows the sectors top view with the "c" line materializing the sector transition. When the aircraft crosses this line, the controllers are sure that it is changing of sector even in case of some uncertainty in the altitude.

Furthermore, the shape of sectors in the horizontal dimension has to be polygonal ensuring linear borders which are necessary for controllers to be effective. Then, sectors are cylinders with polygonal section.

Based on the problem objective and constraints definition one can summarized the framework of this study: one considers a traffic distribution in the airspace which generates a control workload density. The problem consists of partitioning this airspace into *K* sectors, in order to minimize both objectives: (1) workload balancing between sectors and (2) the number of coordinations (flow cut by the sector boundaries). This optimization process has to meet four constraints : (1) route link convexity constraint; (2) safety constraint; (3) minimum stay time constraint; (4) vertical border constraint.

This problem can be viewed as a 3D geometrical cutting problem with typical constraints and the method presented in this paper may be applied to others cutting problem in 3D space. The next part gives a summary of the previous related works. The third part gives a description of the complexity metric used to model the workload. The fourth part proposes a mathematical modeling and identifies the associated complexity of this 3D cutting problem. A fifth part gives a brief overview of the Evolutionary Algorithms which has been used to address such problem. The sixth part gives some details about the chromosome coding and the associated operators dedicated to this problem. Finally, the last part presents results on realistic instances of this problem.

Previous Related Works

The problem of cutting optimization can be summarized the following way. Having a limited domain in R^n one has to cut this domain into *K* sectors in order to optimize an objective function. This optimization process has to take into account several problem dependent constraints.

This problem may find many application in the industry mainly for the stock cutting optimization for which a factory produces material which has to be cut into different pieces ordered by the clients. The objective consists in minimizing the waste after the cutting. This problem has been studied in one, two or three dimensions and may be extended to

higher dimensionality [1-4]. The associated complexity increases with the dimension of the space which has to be cut. For stock cutting problems, the shape and the size of the synthesis sectors are fixed and controlled by the demand. From the mathematical point of view, this kind of cutting problem can be view as an packing problem [5-7] which is a typical NP_Hard assignment problem. Those techniques cannot be applied to our problem because the size and the shape of our sectors are not known in advance. Our objective is also different as we are not looking at minimizing the waste but we try to find a full partition of the airspace (no part of the airspace is left out of sectors) in order to balance the weight of the synthesis pieces with a minimum cutting section (coordination minimization). Depending of the dimension of the object that has to be partitioned into sectors, the associated complexity may strongly change.

For the 1D space, the object that has to be cut is materialized by a line with a linear weight density ($\rho(x)$) (no punctual weights is considered). The total weight of the object is $M = \int_0^L \rho(x)dx$ (where L is the total length of the object).For this simple case, a trivial solution is given by a dichotomy search which first fix X_1 then fix X_2 and so on till X_K (where K is the total number of sectors). When punctual weights are introduced in the problem (see Figure 2), one are not sure to reach an exact solution.

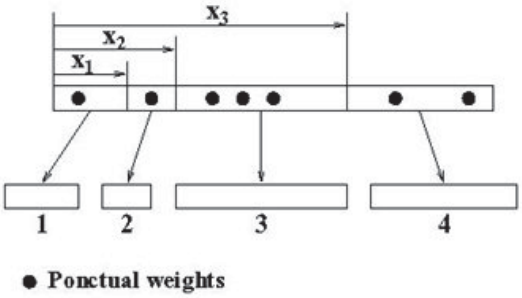

Figure 2. 1D Case where the Strip is Partitioned into 4 Balanced Sub-Strips

The weight of an individual segment k limited by and is given by:

$$M_k = \int_{x_{inf}^k}^{x_{sup}^k} \rho(x)dx + \sum_{n|(x_n)\in[x_{inf}^k, x_{sup}^k]} w_n \quad (1)$$

where is the position of the punctual weight and the associated weight (w_n). We build the following criterion:

$$J = \sum_{k=1}^{K} \left(m_k - \right) \quad (2)$$

and use a classical gradient method to find the minimum of J. When the coordination minimization is consider the criterion J is modified the following way:

$$J = \sum_{k=1}^{K} \left(m_k - \frac{M}{K}\right)^2 + \sum_{k=1}^{K} \left(\rho(x_{inf}^k) + \rho(x_{sup}^k)\right) \quad (3)$$

where the second part represents the cutting section minimization. The coordination being a workload for the controller, it means that when the boundary is built, the associated sectors have to be updated with this new weights. This is taken into account in the definition:

$$m_k = \int_{x_{inf}^k}^{x_{sup}^k} \rho(x)dx + \sum_{n|(x_n)\in[x_{inf}^k, x_{sup}^k]} w_n + \beta(\rho(x_{inf}^k) + \rho(x_{sup}^k)) \quad (4)$$

where β is a weight factor. The way to solve such problem is the same as in the case where the is no punctual weight. The two dimensions case is much more difficult to solve because of the size of the state space. When the cutting direction is fixed a priori the problem is the same as the 1D case for which the linear weight density is the projection of the surface density $\rho(x,y)$ on the cutting direction. Depending of this direction, one can find several solutions to the problem. When two cutting directions are possible (the sectors are then rectangles), there is no way to translate this problem into one dimension and the associated complexity increases drastically. Having polygonal shape of sectors, the number of cutting direction may be much higher than two and the associated complexity increases accordingly. This problem has been studied in [8,9] for which the sectors are synthesized by the mean of Voronoi diagram and optimized by genetic algorithms. In this approach, points called class centers are randomly spread in the 2D plan that has to be partitioned and the

associated Voronoi diagram is built producing a random sectoring of the plane (see Figure 3).

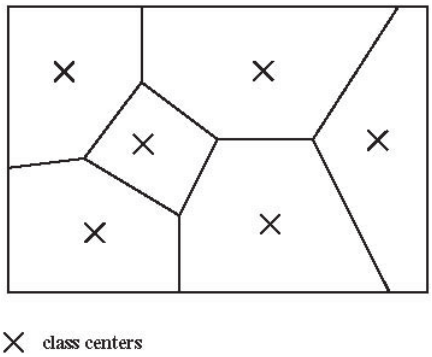

Figure 3. Random Sectoring by Voronoi Diagram

The Voronoi diagrams produce convex sectors ensuring that any line crossing the sector will encounter two and only two borders. Having this property, such sectoring will always met the route convexity constraint. The other constraint are taken into account by relaxation of the objective function. A discrete model of this approach has been proposed in [10]. The approach consists in using a mosaic of elementary cells (this maybe the result of a Voronoi diagram synthesis) and in building the associated Delaunay triangulation (see Figure 4) in order to create a connected planar graph. This principle makes the constraint management easier. As a matter of fact, when a mosaic cell is too small or has a conflict point too close to one of its boundaries, it is merge with one of its neighbors in order to create a bigger cell which met the constraint. After such a cell checking, all the cells met the constraint and the method consists in merging cells by the mean of connected component optimization in the Delaunay graph [10]. This optimization is based on an evolutionary process. This discrete method enable the management of bigger airspace and is faster to reach the optimum. The two previous models may be straightly extended in the third dimension. For the first one, it is only necessary to used 3D position for the class centers. This modification will induce the synthesis on 3D polyhedral sectors. For the second one, one has only to consider Voronoi cells in three dimensions and build the associated Delaunay triangulation inducing a graph with vertex in a 3D space. In both cases, such 3D straight extension will result in sectors which will not met the geometrical constraint. These previous works have been mainly done in the 2D case and may not be extended straightly in the third dimension. The next section introduced the problem modeling which has been used in this new 3D approach.

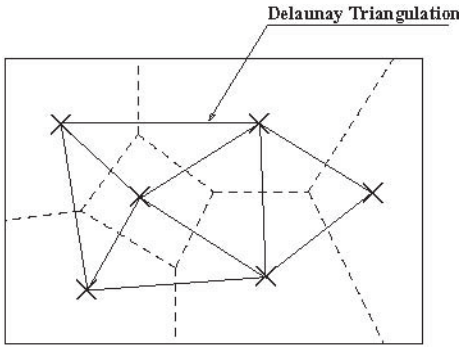

Figure 4. Delaunay Triangulation (The Dash Line Represents the Associated Vonny Diagram)

Complexity Metric

Traffic complexity metrics quantify congestion in airspace more accurately than a simple number of aircraft which is independent of the traffic configuration. This metric captures the level of disorder (or organization) of any set of trajectories in a 4D space (3D for the space and 1D for the time) and represents a good indicator of the underlying control workload. The key idea of this approach is to model the set of aircraft trajectories by a dynamical system which is summarized by the following equation:

$$\dot{\vec{X}}(t) = V(\vec{X}(t), \omega) \qquad (5)$$

where is the state vector of the system ($=[x(t),y(t),z(t)]^T$) and $V : C^2$ vector field depending on parameters ω, describe systems which integral curves may fit the observed trajectories. This equation associates a vector speed to a position in the space coordinate and then synthesis a particular vector field.

Based on the observations of the aircraft (positions, speed vectors and times), the dynamical system has to be adjusted with the minimum error. This fitting is done with a Least Square Minimization (LMS) method. For each considered aircraft *i*, it is supposed that position $_i=[xi,yi,zi]^T$ and speed vector $Vi=[vx_i,vy_i,vz_i]^T$ are given (radar tracker data). An error criterion between the dynamical system model and the observation is computed:

$$E = \sum_{i=1}^{i=N} \left\| \vec{V}_i - V(\vec{X}_i(t), \omega) \right\| \quad (6)$$

where N is number of aircraft. There are many classical ways of obtaining a class of parametrized vector fields which fulfill the fitting requirement. Among them, vector splines allow a control on the smoothness of vector fields, which is important in our case since civil aircraft maneuvers are based on low acceleration guidance laws. Computing traffic complexity for a given traffic situation requires interpolating a vector field given only samples (positions and speeds of aircraft at a given time).Vector spline interpolation seeks the minimum of a functional of the form:

$$\frac{1}{2}\int_D \|L\vec{V}(\vec{X})\|^2 d\vec{X} + \frac{1}{2}\sum_{i=1}^{m} \|\vec{V}(\vec{X}_i) - \vec{V}_i\|^2 \quad (7)$$

where is a vector field defined on a domain $D \subset R^n$, L is an elliptic differential operator controlling smoothness of the solution and (i, i)i=1...m are the interpolation data. By introducing the adjoint operator L^T, optimal vector field can be shown to be a linear combination of shifted version of the elementary solution kernel of the differential operator $L^T L$.A special case is the so-called "div-curl" splines with the criterion:

$$\int_{R^2} \alpha \|\nabla div\vec{V}(\vec{X})\|^2 + \beta \|\nabla curl\vec{V}(\vec{X})\| d\vec{X} \quad (8)$$

with α,β positive reals controlling the smoothness of the approximation by focusing on constant divergence or constant curl. The metric chosen for complexity computation relies on a measure of sensitivity to initial conditions of the underlying dynamical system called Lyapunov exponents. In order to figure out what Lyapunov exponents are, let consider a point and look at its evolution when transported by the dynamical system. Let be fixed (initial point) and let φ be a point trajectory of the dynamical system given by:

$$\phi(t, \vec{X}) = \vec{X} + \int_0^t \vec{V}\phi(s, \vec{X}) ds \quad (9)$$

Assume now that trajectory is disturbed by a small perturbation we have :

$$\phi(t, \vec{X} + \vec{\varepsilon}) = \phi(t, \vec{X}) + \mathbf{D}_{\vec{X}}\phi(t, \vec{X}) \cdot \vec{\varepsilon} + o(\|\vec{\varepsilon}\|) \quad (10)$$

where $\mathbf{D}_{\vec{x}}\phi($ is the differential of the vector field at that satisfies:

$$\frac{\mathbf{D}_{\vec{X}}\phi(t, \vec{X})}{dt} = \mathbf{D}_{\vec{X}}\vec{V}(\phi(t, \vec{X})) \cdot \mathbf{D}_{\vec{X}}\phi(t, \vec{X}) \quad (11)$$

The Lyapunov exponents are closely related to the singular values of the matrix $\mathbf{D}_{\vec{x}}\phi($ and can be thought as local shear values for the dynamical system. When Lyapunov exponents are high, the trajectory of a point under the action of the dynamical system is very sensitive to initial conditions (or parameters on which the vector field may depend), so that situation in the future is unpredictable. On the other hand, small values of the Lyapunov exponents mean that the future is highly predictable (expected to be comfortable for a controller). *So, the Lyapunov exponent map determines the area where the underlying dynamical system is organized. It identifies the places where the relative distances between aircraft do not change with time (low real value) and the ones where such distance change a lot (high real value).*

Let us now describe the practical procedure for computing 3D complexity maps. First of all, the optimal dynamic div-curl approximation for the observed trajectories is computed, based on the defining equations. That step requires a linear system solving. The second step computes the second derivatives matrix at each point of the grid for φ trajectory starting at . This is done by solving the differential equation 11 with a Runge-Kutta integrator. The complexity value at point is then obtained by averaging Lyapunov exponents over the time :

$$\kappa(\vec{X}) = \frac{1}{n}\sum_{i=1}^{i=n} \|\mathbf{D}_{\vec{X}}\vec{V}(\phi(t, \vec{X}))\|_2 \quad (12)$$

The result of this process is given by a 3D Lyapunov exponents map which has to be cut into sectors. More information about this metric may be found in [11]. The next section introduces the model which has been used to address this problem.

Problem Modeling

The current approach is based on an initial cutting of the airspace into elementary cells. This initial sectoring mayby done by the means of a 2D Voronoi diagram for which the class center are the plane projection of the 3D crossing point between air flows. Based on this initial Voronoi diagram a merging process remove the cells with too close class centers and the too small ones. After this merging step, one has a 2D mosaic on the projection of the 3D airspace (see Figure 5). The right sideof this figure shows the voronoi diagram of the 2D prejection of the 3D cells.

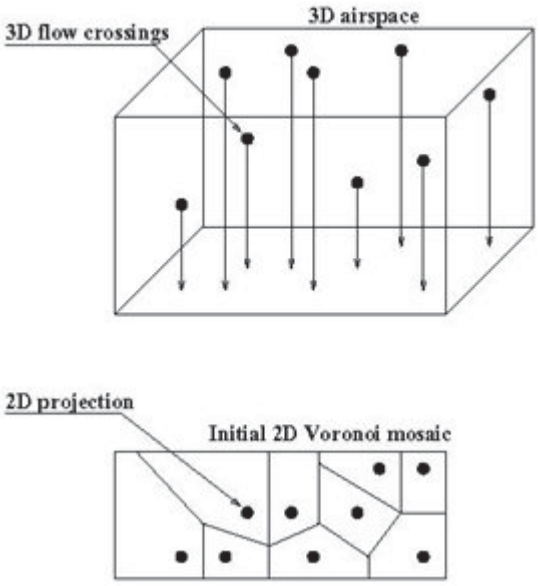

Figure 5. Initial Voronoi Mosaic

A set of altitude layers is then considered and each mosaic cell is then extended in the third dimension by duplication of the 2D cell in each layer (see Figure 6).

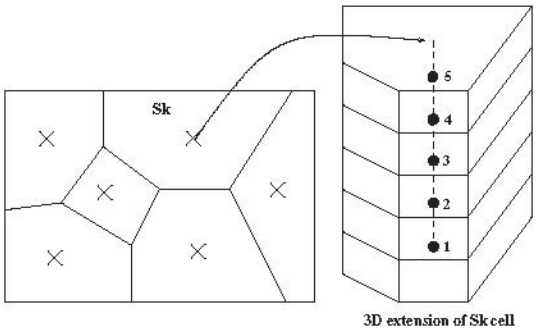

Figure 6. 3D Extension

For each cell and for each layer, the associated workload is summarized by a weight located at the position of the cell class center with height given by the average altitude of the given layer. Those weights are symbolized by black circles on the Figure 6. A Delaunay triangulation of this initial mosaic is built in order to create a graph for which the link value represents the flow going from one cell into one of its neighboring cells. This initial Delaunay triangulation is extended to each altitude layer. This information will be used in the flow cut minimization process. Based on this initial mosaic, the following sector building process has been used in our application. First a set of *sector centers* is defined: *S*. Those sector centers are points with coordinates in the 2D mosaic plane. The number of sector centers is *K* which is the number of sectors that has to be synthesized ($|S| = K$). Then, *K* layer intervals are defined in a way that ensure that each altitude layer is assigned to an interval or more. Such association is called a covering. For instance, if 5 layers are considered (1,2,3,4, 5) a possible covering of this5 layers with 3 intervals is given by :I_1 =[1,4]; I_2 =[3,5]; I_3 =[2,4]. As it can be noticed in this example all layers are assigned to one or more intervals. A cell association process to the sector centers is then applied. This process consists in taking each cell and finding the closest sector center for which the layer interval contains the cell layer. Based on the covering property of intervals, we are sure that, after this association process, all the cells are associated to one and only one sector center. In order to build a mathematical model, the following notation have to be introduced. We call *C* the set of elementary cells from the initial mosaic. A cell c_i is characterized by its position $p_{ci} = (x_{ci}, y_{ci}$ where is the layer number of the cell and its weight . The cardinality of this set is given by the number of 2D Voronoi cells , times the number of layers , so $|C|$. We introduce *K* subsets each of them representing a potential sector (*K* is the number of sectors). A sectoring being a partition of the initial cell set, we have the following properties:

$$\bigcup_{i=1}^{i=K} s_i = C \quad s_i \cap s_j = \emptyset \; \forall i \neq j \qquad (13)$$

The Delaunay triangulation is represented by a graph A for which the link symbolize the neighborhood relation between cells; this means thata link is built when two cells are neighbor. Each link is weighted by the amount of flow transiting from cell i into cell j. Based on those notations, one can formalize the objectives of this problem. The first objective consists in balancing the weights of sectors. The weight w_k of sector k is given by:

$$m_k = \sum_{c_i \in s_k} w_{c_i} \tag{14}$$

The total amount of workload is given by:

$$M = \sum_{k=1}^{k=K} m_k \tag{15}$$

Sector weight are balanced when m_k. So the first criterion is modeled by:

$$f_1 = \sum_{k=1}^{K} \frac{\left| m_k - \frac{M}{K} \right|}{\frac{M}{K}} \tag{16}$$

This criterion may vary from 0, when sectors are full balanced, to $(K-1)$ when all the cells are into one sector and all the others sectors are empty. The denominator induces to be independent of the total weight. The second objective aims at minimizing the flow cut. Based on the Delaunay graph A this second objective may be modeled the following way:

$$f_2 = \frac{1}{F} \sum_{\substack{i,j | c_i \in s_k \\ c_j \notin s_k}} f_{ij} \tag{17}$$

where ive are gathered e f by the mean of a weighted sum: $f = \alpha . f1 + (1-\alpha) . f2$. There is not relaxation of constraints in this objective because they are already taken into account by the construction method of sectors. As a matter of fact, the initial Voronoi mosaic ensure that elementary cells met the min stay constraint and the safety constraint. The aggregation of cells to their nearest sector centers ensures that the route convexity constraint is met by sectors. Finally, the use of a covering of layers by intervals ensure that the synthesized sector will have cylinder shapes with polygonal section. Having now a mathematical model of this problem, let have a look to the associated complexity. The problem to be solved can be divided into two separate parts corresponding to the two different goals:

1. Equilibrium of the different sector weights;
2. Minimization of the coordination workload.

The second criterion is typically a discrete graph partitioning problem with topological constraints and is then NP-hard [12]. The first criterion is a classical classification problem with a connectivity constraint. One must find an optimal grouping among possibilities; where is the second Stirling number:

$$S_K^{n_c} = \frac{1}{n_c!} \sum_{i=0}^{i=n_c-1} (-1)^i \left(\frac{n_c!}{i!(n_c-i)!} \right) (n_c - i)^K$$

where is the number of cells and K the number of sectors. This problem is NP-hard and stochastic optimization is a good candidate to address it. Moreover, this kind of problem may have several optimal solutions (or near optimal) due to the different possible symmetries in the topological space and one must be able to find all of them because they have to be refined by experts (it is impossible to know at this step which one is really the best). This last point makes us reject classical simulated re-annealing [13] optimization which updates only one state variable. On the other hand, Evolutionary Algorithms (EAs) maintains and improves a population of numerous state variables according to their fitness and will be able to find several optimal (or near-optimal) solutions. EAs appear to be relevant to solve this sectoring problem.

Evolutionary Algorithms

Evolutionary algorithms use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover) to find approximate solutions to optimization problems. An individual, or solution to the problem to be solved, is represented by a list of parameters, called

chromosome or genome. Initially several such individuals are randomly generated to form the first initial population (POP(k) in Figure 7). As can be seen, The first step consists in the selection of the best individuals from population *POP*(*k*). Afterward, recombination operators are applied in order to produce the *POP*(*k*+ 1) population.

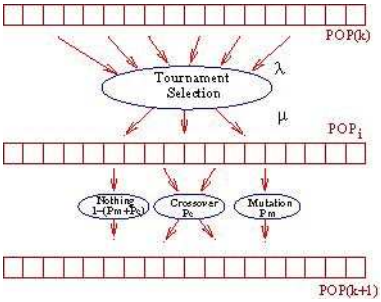

Figure 7. Genetic Algorithm with Tournament Selection

Then each individual is evaluated, and a value of fitness is returned by a fitness function. This initial population undergo a selection process which identify the most adapted individual. The one which has been used in our experiments is a deterministic (λ,μ)-tournament selection. This selection begins by randomly selecting λ individuals from the current population (POP(k) and keep the μ bests(($\lambda > \mu$)). This two steps are repeated until a new intermediate population (POP*i*) is completed. Following selection, one of the three following operators is applied : *nothing, crossover,* and *mutation*. The associated probability of application are respectively ($1- pc - pm$)), *pc* and *pm*. Crossover results in two new child chromosomes, which are added to the next generation population. The chromosomes of the parents are mixed during crossover. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. The purpose of mutation in EAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. These processes ultimately result in the next generation population of chromosomes (POP(k+1) in Figure 7) that is different from the initial generation. This generational process is repeated until a termination condition has been reached. The next section presents the application of EA to our problem.

Application to Airspace

Based on the previous problem modeling, a data coding has to be developed in order to make run EA on this problem. The state space may be summarized by a set of points in a 2D space with an altitude layer interval. The initial chromosome of our EA has the following structure:

x_1	x_2	…	x_i	…	x_K
y_1	y_2	…	y_i	…	y_K

$L(M_1)$	$L(M2)$	…	$L(Mi)$	…	$L(MK-1)$
Ex_1	$Ext_{2_{inf}}$	…	$Ext_{i_{inf}}$	…	$Ext_{K-1_{inf}}$
$Ext_{1_{sup}}$	$Ext_{2_{sup}}$	…	$Ext_{i_{sup}}$	…	$Ext_{K-1_{sup}}$

Each column of this table represents a gene of the chromosome. A gene numbered *i* consists in position coordinates (x_i,y_i), a layer marker level $L_{(Mi)}$ and two marker extension limits Ex_{inf} and Ex_{sup}. The sector center enable the building of the sectoring for a given layer by the mean of an aggregation process of the Voronoi cell centers to their nearest sector centers (see Figure 8).

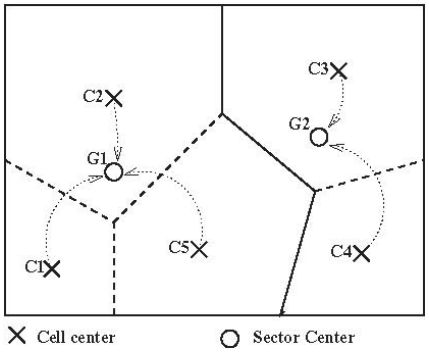

Figure 8. Aggregation Process Example

The markers are used to easily build a covering of the altitude levels by several intervals. Consider that *K* intervals have to be built then *K*− 1 marker have to be generated on the layer domain (on Figure 9, four intervals are built with three markers).

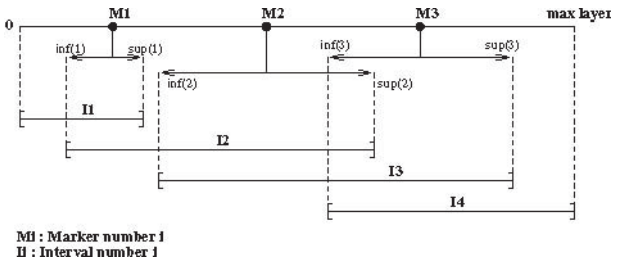

Figure 9. Layer Interval Building Process by the Mean of Layer Markers

The makers $M_1,M_2,...,M_K$ are ordered according to their layer number :$L(M_i)$ ($L(M_1) \geq L(M_2) \geq ... \geq L(M_K)$). The first interval is built by using minimum layer number (0) and $L(M_1)$+ built by using minimum layer number (0) and $L(M1)+ Ext1_{sup}$. The second interval begins at $L(M_1) -$ and ends at $L(M_2)+$. More generally the interval numbered has the following structure:

$$I_i = [L(M_{i-1}) - Ext_{i-1_{inf}}, (L(M_i) + Ext_{i_{sup}}]$$

(18)

The final interval labeled K, is given by:

$$[L(M_{K-1}) - Ext_{K-1_{inf}}, max\_layer\_number]$$

(19)

The chromosome is initialized by randomly drawing 2D points in the underlying surface (ground) and makers in the altitude layer domain (those markers are then ordered in the increasing layer number). This process is then repeated for all individuals in the population. An example of chromosome coding is given on Figure 10. This example presents a mosaic with 5 initial Voronoi cells (dash lines) for which 2 sectors are built. The number of layers considered is 10 and the first sector addresses the layer 0 to 7 and the second one addresses the layers 5 to 10. This has been produced by the marker M_1 located at the layer 4 with = 1 and = 2. The Figure 10 shows one of the3 common layers of the two sectors (5,6,7). The sector centers are represented by symbol G_1 and G_2. For the layers 0 to 4 only the first sector addresses those layers meaning that all the associated cells belong to this sector. On the same way cells of the layers 8 to 10 are associated to the second sector.

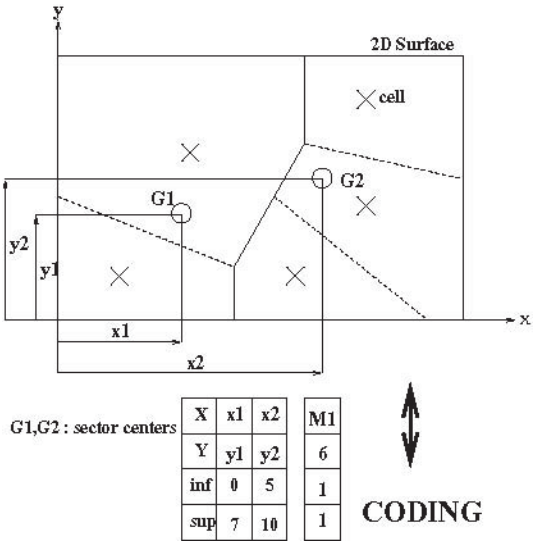

Figure 10. Example of Chromosome Coding

This coding process has been completed by operators. Those operators are used to increase the diversity of the population. The crossover operator begin by randomly drawing two parents in the population of individuals after selection. The crossover operates in two steps. The first step consists in changing the position on one sector center in each parents (see Figure 11). In this figure one sector center is selected in each chromosome (one circle and one triangle). Afterward, those sector centers are connected by a straight line and are randomly moved on this line to create two new sectors (triangles).

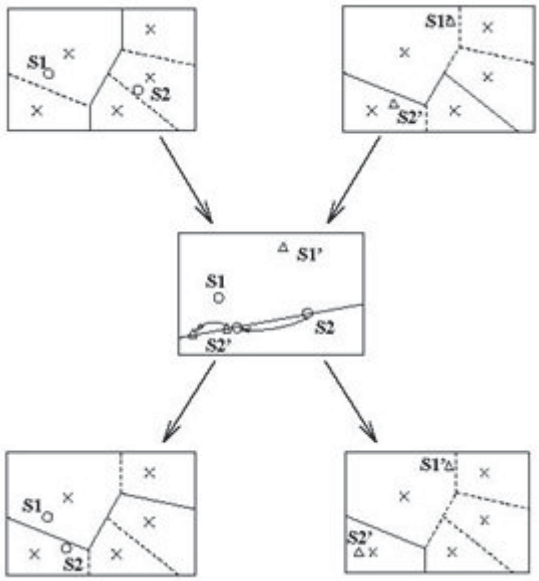

Figure 11. Crossover Operator

First, two genes are randomly selected in each parent. The associated sector centers are then connected by a straight line. Sector centers are randomly moved on this line in order to create two children sector centers. The second step of the crossover operator exchanges markers between parents (see Figure 12). The markers of the children come from both parents. The parents are labeled *P*1 and *P*2, the children *C*1 and *C*2. First the marker of both parent are gathered together and ordered according their layer level. Then, the first child *C*1 take one maker over two in the ordered list, and the child *C*2 takes the remaining ones

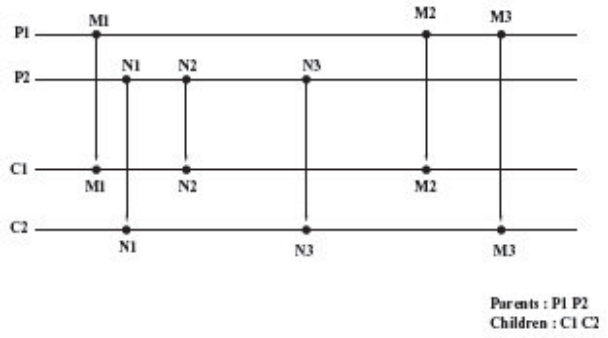

Figure 12. Marker Exchange between Parents

The second operator used for our EA is the mutation operator as shown in Figure 13. To mutate a chromosome, one sector center is randomly selected and is moved by adding noises to its geographical coordinates

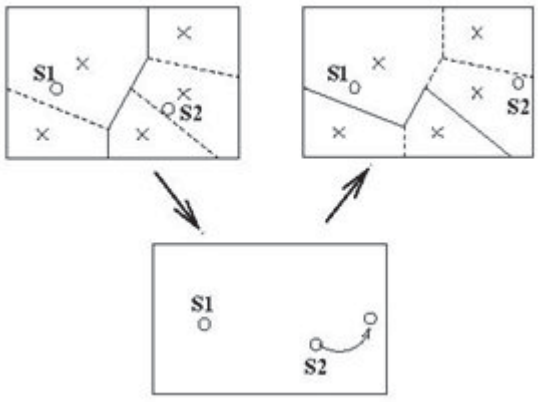

Figure 13. Mutation Operator

This operator may be decomposed into two classes:

1) strong mutation. This first operator takes an individual and randomly initializes the position of the sector centers or the makers. The choice between those two options is random.
2) medium mutation. First, the more unbalanced genes are identified. This identification, is used to put some bias in the drawing of the gene which will undergo the mutation. The extension of the selected marker are then changed by taking into account the sign on the unbalanced weight of their genes. If the sector weight is under the balanced weight, the vertical extension of the associated interval will be increased and it will be decreased if it is over the balanced weight.

Those mutations are randomly selected. The fitness used by our EA has been built by mixing both objective the following way:

$$fitness(I) = \alpha \frac{1}{0.01 + f_1(I)} + \beta \frac{1}{0.01 + f_2(I)}$$

where I is an individual and $\alpha = 0.8$ and $\beta = 0.2$. This fitness has to be maximized and reach its maximum when $f1 = f2 = 0 \Rightarrow fitness = 100$. Having presented the structure of our Evolutionary Algorithm, the next section shows some results on large instance of the problem.

Results

The parameters of the algorithm have been adjusted in order to reach a good performances for realistic instance of the problem (instances with several hundred cells in the initial mosaic).

Table 1. Parameters of the Algorithm

Number of generations	500
Population size	500
Probability of Crossover	0.4
Probability of Mutation	0.2
Elitism	Yes
(λ, μ)	(5,2)

In order to validate the method, one has to produce problems for which exact solutions are known. To reach this goal, one has to create

symmetries in the problem which are easy to investigate from the human being perception point of view but which have the same complexity from the computer point of view.

The first test considers 500 cells which have been randomly thrown on a 10×10 square (see Figure 14).

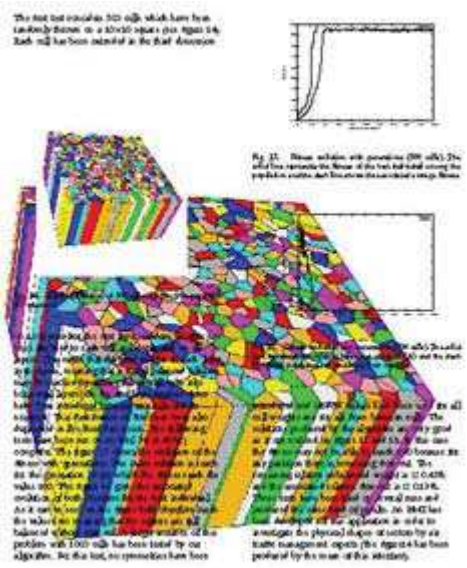

Figure 14. Initial Distribution of 500 Cells in a 10×10 Square with 3D Extension

Each cell has been extended in the third dimension on 10 layers. For the first layer, random weights has been attached to each cell and duplicated on all layers. This mean that the total weight on each layer is the same, meaning that a trivial balanced solution may be reached by cutting the overall cube into horizontal layers. On the first layer random flows have been introduced between any cells and its neighbor. This distribution of flows has been also duplicated in the third dimension. The following tests have been run on an Intel P4 (1.4Ghz) computer. The Figure 15 shows the evolution of the fitness with generations. The solid line represents the fitness of the best individual among the population and the dashed line shows the associated average fitness.

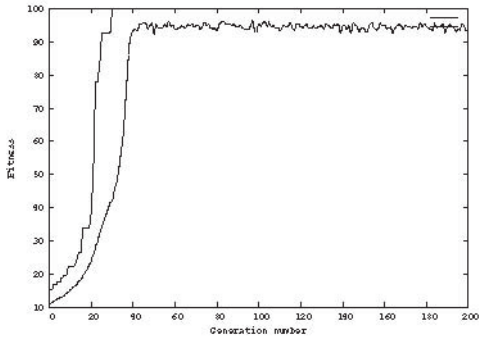

Figure 15. Fitness Evolution with Generations (500 Cells)

The exact solution is reach for the generation 30 for which the fitness reach the value 100. The Figure 16 gives the associated evolution of both objective for the best individual. As it can be seen on the figure both objective reach the value zero meaning that the sectors are full balanced without flow cut. A larger instance of this problem with 1000 cells has been tested by our algorithm. For this test, no symmetries have been introduced and random values have been used for all cell weights and for all flows between cells. The solutions produced by the algorithm are very good as it can noticed on Figures 17 and 18.

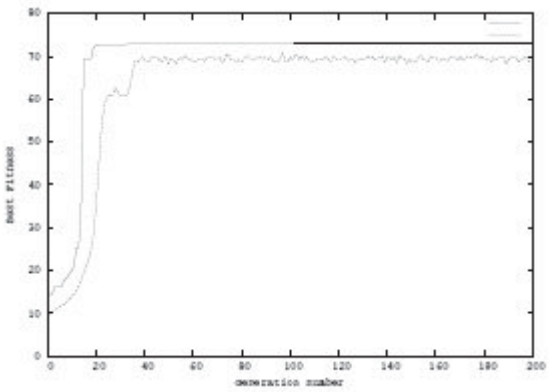

Figure 17. Fitness Evolution with Generations (1000cells)

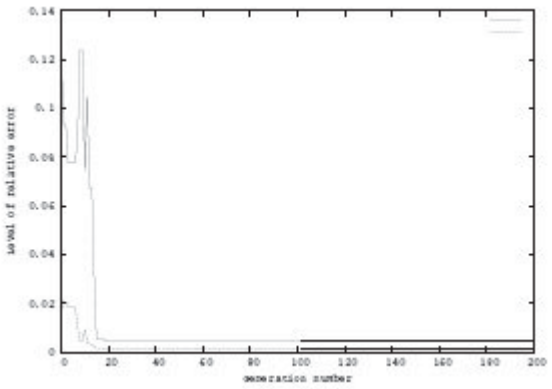

Figure 18. Criteria Evolution with Generations (1000cells)

In this case the fitness may not be able to reach 100 because for any partition there is remaining flow cut. The remaining relative unbalanced weight is 0.45% and the associated relative flow cut is 0.13%. Those tests have been tried on several runs and produced the same kind of results. An HMI has been developed for this application in order to investigate the physical shapes of sectors by air traffic management experts (the Figure14 has been produced by the mean of this interface).

Conclusion

This paper has presented a new airspace cutting method which synthesizes balanced sectors with minimum flow cut. This problem is known to be NP Hard and stochastic optimization approach has been developed in order to address real instance of airspace cutting. A mathematical model has been developed for which two objective functions has been defined in order to quantify the sector balance and the flow cut objective. The 3D cylinder sector shapes ensure that the constraint are satisfied. A coding process and recombination operators have been dedicated to this problem in order to make run a classical evolutionary algorithm scheme. This algorithm has been tested on realistic instance of the problem for which symmetries has been introduced in order to identify exact solutions. The results show that our EA find the exact solution. A second test with 1000 cells has been tried for which no symmetry has been included. The algorithm find very good solution with the same convergence rate as in the symmetrical case showing the robustness of such method.

References

[1] K.K. Lai andW.M. Chan. Anevolutionary algorithm for the rectangular cutting stock problem. *International Journal of Industrial Engineering*, 4:130–139, 1997.

[2] Yaodong Cui. Generating optimal multi-segment cutting patterns for circular blanks in the manufacturing of electric motors. *EuropeanJournal of Operational Research*, 169(1):30–40, 2006.

[3] Yaodong Cui. A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Computera and Operartion Research*, 32(7):1709–1721, 2005.

[4] T. Kampke. Simulated annealing: Use of a new tool in bin packing. *Annals of Operations Research*, 16:327–332, 1988.

[5] H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44:45–159, 1990.

[6] D. SuterF. Chen.Fastevaluationofvector splinesin three dimensions. Computing.

[7] E.Falkenauer.Ahybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2(1):5–30, 1996.

[8] D. Delahaye, JM. Alliot JM., M. Schoenauer, and J.L. Farges. Genetic algorithms partitioning airspace. In *Proceeding of theTenth IEEE Conference on Artificial Intelligence Application*. IEEE, 1994.

[9] D. Delahaye. *Spatial Evolutionary Modelling (chapter 7)*. Oxford University Press. Oxford, 2001.

[10] D. Delahaye, M. Schoenauer, and J.M. Alliot. Airspace sectoring by stochastic optimization. In *Proceeding of the IEEE International Conference on Evolutionary computation ICEC98*. IEEE, 1998.

[11] D. Delahaye and S.Puechmorel. Air traffic complexity based on non linear dynamical systems. In *Proceeding of the 5thATM Seminar*.FAA/EUROCONTROL, 2003.

[12] C.K. Cheng. The Optimal Partitioning of Networks. *Networks*, 22:297–315, 1992.

[13] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley and sons, 1989. ISBN: 0-471-92146-7.

Email Addresses

Daniel Delahaye: delahaye@recherche.enac.fr
Stephane Puechmorel: puechmor@recherche.enac.fr

27th Digital Avionics Systems Conference
October 26-30, 2008