Genetic Algorithms for automatic regroupement of Air Traffic Control sectors

Daniel Delahaye Jean-Marc Alliot Marc Schoenauer Jean-Loup Farges

CENA[*] IRIT[†] CMAPX[‡] CERT[§]

25 September 1994

Abstract

In this paper, we show how genetic algorithms can be used to compute automatically a balanced regroupement of Air Traffic Control sectors to optimally reduce the number of controller teams during daily low flow periods.

AI topic: Genetic Algorithm, Network partitioning, Sector Classification.
Domain Area: Air Traffic Control
Status: Operational mock-up.
Effort: 1 man/year
Impact: Automatic (and dynamic) regroupement of Air Traffic Control sectors according to the flow on the network by balancing workloads groups.

1 Introduction

The CENA is the organisation in charge of studies and research for improving the French ATC systems. Studies on the use of genetic algorithms for conflict resolution [2], [9] , airspace sectoring [7] and air traffic assignment [8] have given encouraging results, and a new study has been funded to solve the Air Traffic Control sectors regroupment problem which is the following of [7]. When joining two airports, an aircraft must follow routes and beacons ; these beacons are necessary for pilots to know their position during navigation and because of the small number of beacons on the ground they often represent crossing points of different airways.

Crossing points may generate conflicts between aircraft when their trajectories converge on it at the same time and induce a risk of collision.

At the dawn of civil aviation, pilots resolved conflicts themselves because they always flew in good weather conditions (good visibility) with low speed aircraft. On the other hand, modern jet aircraft do not enable pilots to resolve conflicts because of their high speed and their ability to fly with bad visibility. Therefore, pilots must be helped by an air traffic controller on the ground who has a global view of the current traffic distribution in the airspace and can give orders to the pilots to avoid collisions.

As there are many aircraft simultaneously present in the sky, a single controller is not able to manage all of them. In France, airspace is partitioned into different sectors, each of them being assigned to a controller.

During the day the flows on the airways change according to the demand between each Origin Destination pair. So one can identify a maximum of the control workload during the day, in the same way, a minimum during the night. Maximum daily worload is always used to compute a new sectoring to be sure that it is well adapted for this period with no overloaded sector even if it is not so well adapted other periods.

During the night, we know that traffic demand is reduced inducing less workload in the airspace and sector are gathered together into groups each of them being assigned to a controller who manage several sectors.

In [7] we tried an automatic approach to compute a balanced sectoring of the air space which induced a convex sectoring. In this paper we present an automatic regroupement algorithm (based on Genetic Algorithms) which uses convex sectoring to synthesize an optimal regroupment. In the first part we describe more precisely our problem and make some relevant simplification to develop a mathematical model. In the second part we present a complete example of resolution with genetic algorithms.

2 A simplified model

2.1 Introduction

Before specifying a mathematical description of our problem, it is necessary to set out our framework to introduce some simplifications for our model. Since training period of an air traffic controller on his sector is long (from 3 to 4 months), we must not investigate a real time sectoring optimization according to the

[*] Centre d'Etudes de la Navigation Aérienne

[†] Institut de Recherche en Informatique de Toulouse

[‡] Centre de Mathématiques appliquées de l'X

[§] Centre d'Etudes et de Recherche de Toulouse

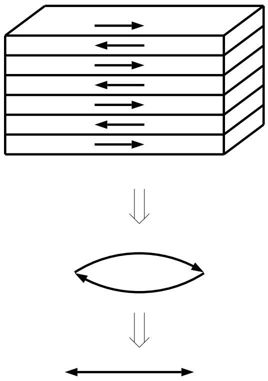

Figure 1: Airways modeling

variations of the traffic load. Instead we have to consider a registered maximum load traffic period on the working network. This worload is supposed to be partitioned into balanced sectors. Our problem is then to find a balanced regroupement of sectors in a way that minimizes coordinations during underloaded period.

When examining the physical air traffic network, we notice that airways are superposition of several routes which have the same projection on the ground but different altitudes according to their azimuth (semi circular rule[1]). So an airway can be modeled by a bidirectional link which gathers several individual aircraft routes (see figure 1).

Then, our 3 dimensional transportation network will be modeled by a classical 2 dimensions network on a horizontal plan. This network is supposed to be sectorised into K convex sectors for each of them we can compute the associated control workload.

The controller workload has several sources that can be divided into two categories:

1. there are quantitative factors which include the number of flights, the number of conflicts etc..., which can be precisely modeled in a mathematical way and handled by an optimization algorithm;

2. there are psychological factors such as stress, concentration etc... which have no evident mathematical formulation but are in direct relationship with the previous ones according to the controllers themselves.

So, we will only take into account quantitative elements in our application on first approximation.

Having a model, we can now define more precisely our goals in the following way:

> one considers an air traffic transportation network in a 2 dimensional space with flows on it inducing a workload distributed over the space and supposed to be partitioned into K convex sectors. Those sectors must be regrouped into N equilibrated groups in a way that minimizes coordinations[2].

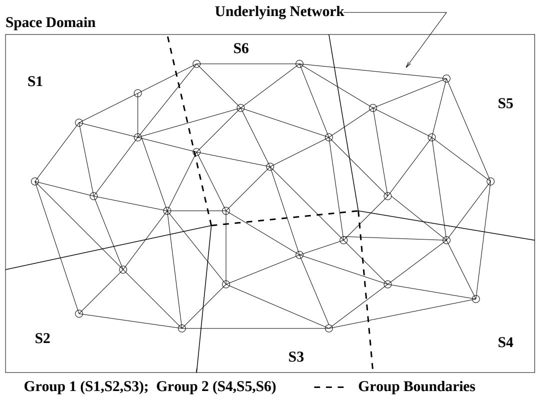

Figure 2: Example of sectoring

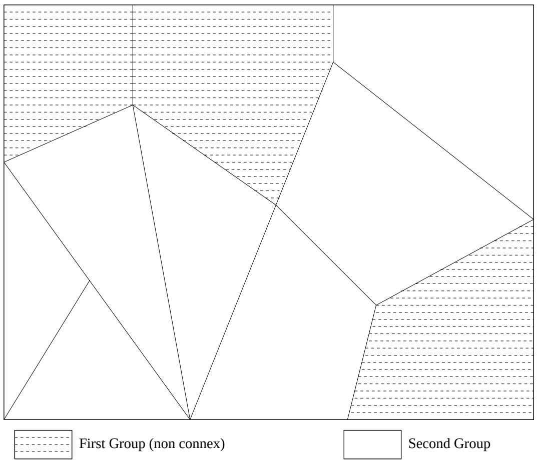

Figure 3: Connexity constraint

Figure 2 shows an example of network sectoring with 6 sectors and 2 groups.

This regroupement must respect the following constraint which is called group connexity: all the sectors belonging at the same sector must be connex.

Figure 3 gives an example of regroupement where the previous constraint is not satisfied. As we can see, there is a group with two separate subsets which would be managed by the same contoller, furthermore it induces superfluous coordinations when an aircraft crosses the two subset.

[1] Aircraft with heading between 0 and 180 have to fly with odd altitude (in hundred of feet) and even altitude for headings between 180 and 360

[2] When an aircraft crosses a sector frontier, controllers in charge of those sectors have to exchange information about the flight inducing a workload called coordination

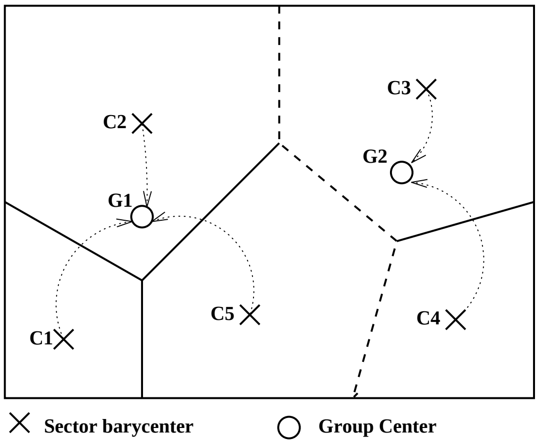

Figure 4: Example of a 5-partitioning

2.2 Mathematical formulation

2.2.1 The transportation network

We define our transportation network as a doublet (N, L) in which N is the set of nodes (with their positions in a topological space) and L is the set of links each of them transporting a quantity f_{ij} of flow from node i to node j [3].

2.2.2 Construction of groups

According to the previous section, the groups we have to build must gather together several sectors in a way that garantees the connexity between the sectors. To reach this goal we use a Forgy aggregation method [14] coming from dynamic clustering in exploratory statistics which aims at extracting clusters from a set of points randomly distributed in a topological space (see [5, 14]). This method randomly throws N points (the group centers) in the space domain containing the transportation network and aggregates all the sector barycenters to their nearest group center. This method ends up in a N partitioning of our domain into connex groups. Figure 4 gives an example of a 5-partitioning of a rectangle with two groups.

The way working gives an automatic way for making groups which automatically respect the connexity constraint. As a matter of fact, when the sector barycenters are aggregated to their nearest group center, the polygonal shape induced by this aggregation is convex, then the associated sectors are connex.

2.2.3 Workload induced in a control sector

As said before, we just take quantitative criterions into account to compute controller workload (see [4]). According to the controllers themselves, workload can be divided into three parts which correspond respectively to the conflict workload, the coordination workload and the trajectories monitoring workload of the different aircraft which are present in a sector :

- the conflict workload gathers the different actions of the controller to solve conflicts.

- the coordination workload corresponds to the information exchanges between a controller and the controller in charge of the bordering sector or between a controller and the pilots when an aircraft crosses a sector frontier.

- the monitoring aims at checking the different trajectories of the aircraft present in a sector and induces a workload.

3 Complexity of our problem

The problem we have to solve can be divided into two separate parts corresponding to our two different goals:

1. equilibrium of the different groups workload according to the number of aircraft and conflicts in each group;

2. minimization of the coordination workload.

The second criterion is typically a discrete graph partitioning problem with topological constraints and then is NP_HARD [6]. The first criterion is a classical classification problem with a connexity constraint.

One must find an optimal grouping among C_k^n possibilities where k is the number of sectors and N the number of groups. So our problem is NP HARD and classical combinatorial optimization is not relevant and stochastic optimization seems to be more suitable.

Moreover this kind of problem may have several optimal solutions (or near optimal) due to the different possible symmetries in the topological space etc..., and we must be able to find all of them because they have to be refined by experts and we do not know at this step which one is really the best. This last point makes us reject classical simulated reannealing optimization which updates only one state variable, even if it might give better results in some cases [11].

On the other hand, Genetic Algorithms (GAs) maintain and improve a population of numerous state variables according to their fitness and will be able to find several optimal (or near optimal) solutions. Then, GAs seem to be relevant to solve our sectoring problem.

4 Genetic algorithms

4.1 Principles

We are using classical Genetic Algorithms and Evolutionary Computation principles such as described in the litterature [10, 13]; Figure 5 describe the main steps of GAs.

First a population of points in the state space is randomly generated. Then, we compute for each population element the value of the function to optimize,

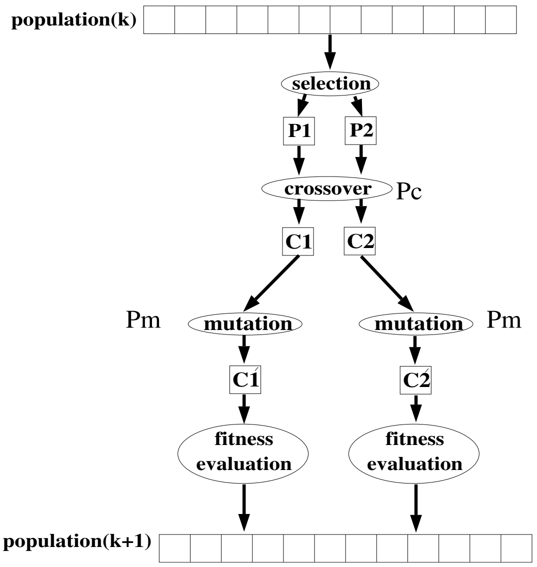

Figure 5: GA principle

which is the *fitness*. In a second step we select[3] the best individuals in the population according to their fitness. Afterward, we randomly apply classical operators of crossover and mutation to diversify the population (they are applied with respective probabilities P_c and P_m). At this step a new population has been created and we apply the process again in an iterative way.

This GA can be improved by including a Simulated Annealing process after applying the operators [12]. For example, after applying the crossover operator, we have four individuals (two parents $P1$,$P2$ and two children $C1$,$C2$) with their respective fitness. Afterward, those four individuals compete in a tournament. The two winners are then inserted in the next generation. The selection process of the winners is the following:if $C1$ is better than $P1$ then $C1$ is selected. Else $C1$ will be selected according to a probability which decreases with the generation number. At the beginning of the simulation, $C1$ has a probability of 0.5 to be selected even if its fitness is worse than the fitness of $P1$ and this probability decreases to 0.01 at the end of the process. A description of this algorithm[4] is given on figure 6.

Tournament selection brings some convergence theorems from the Simulated Annealing theory. On the other hand, as for Simulated Annealing, the (stochastic) convergence is ensured only when the fitness probability distribution law is stationary in each state point [1].

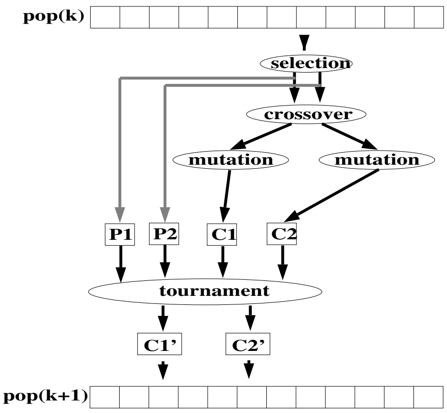

Figure 6: GA and SA mixed up

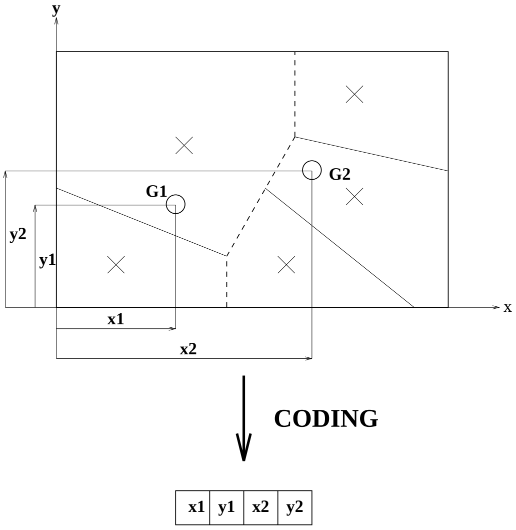

Figure 7: Structure of the chromosome

4.2 Coding our problem

To code our problem, we did not use binary chromosomes. The problem is not well suited for binary coding, and, as it has been advocated already by different experts, a specific coding with specific operators is usually more efficient.

An example of the coding of a chromosome is given in figure 7.

To create a new classification we just need to know the coordinates of the associated group centers in the space domain. So the chromosome can be coded as a vector regrouping the coordinates of group centers. For each chromosome there is only one associated classification (the reverse is false).

To make the GAs run one must be able to randomly initialise a numerous population of individuals. To reach this aim, random coordinates are generated in

[3] Selection aims at reproducing better individual according to their fitness. We tried two kinds of selection process, Roulette Wheel Selection" and "Stochastic Remainder Without Replacement Selection", the last one always gave better results.

[4] We are using our own GA simulator, which includes some goodies usually not available on public domain GA, such as Simulated Annealing, very simple parallelism, etc.

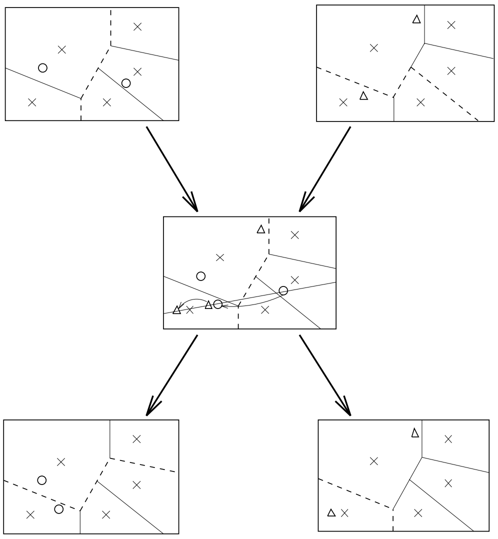

Figure 8: Crossover operator

the space for each individual.

We had then to create operators for crossover and mutation. The efficiency of the algorithm depends of the ability of those operators to create new individuals that respect the constraint of our problem.

The crossover is implemented as a floating crossover: After selecting two parents in the current population, we randomly chose an allele position (so we select two groups at the same allele position, one in each grouping). Afterwards, we join by a straight line the associated group centers.Then, we move the groups centers on this line according to a uniform random variable. An example of this kind of crossover is given in figure 8.

To mutate a chromosome we randomly select an allele position and we move its associated class center by adding noise to it[5] (An example, of mutation is given on figure 9).

5 Results

To validate our algorithm, we used a toy network for which we knew a trivial grouping solution (this network is drawn on figure 10). This network is sectorised into 81 sectors (see figure 11) for which it is very easy to find 9 balanced groups of 9 sectors.

The parameters for the simulation were:
Population size: 200
Number of generations: 100
Probability of crossover: 0.6
Probability of mutation: 0.1

The evolution of best-ever chromosome fitness and average chromosome fitness is displayed on figure 12:

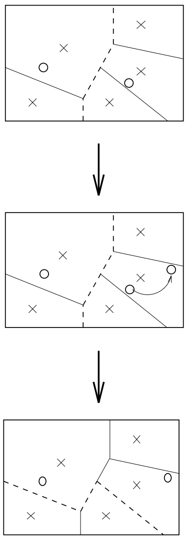

Figure 9: Mutation operator

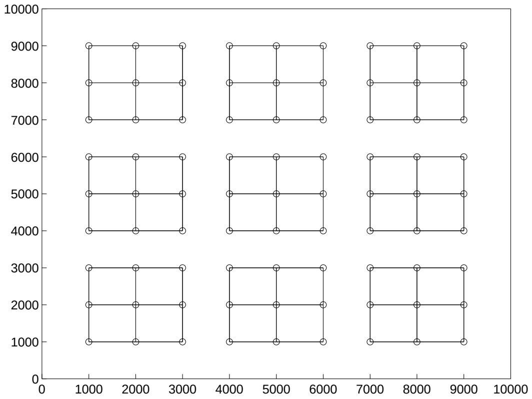

Figure 10: Square network

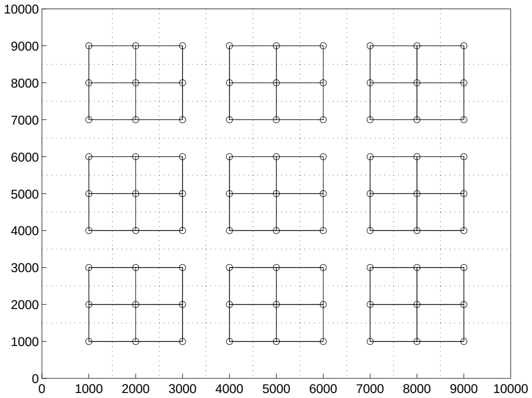

Figure 11: Initial Sectoring

[5] it seems that best results are given with an affine distribution and not with a Gaussian

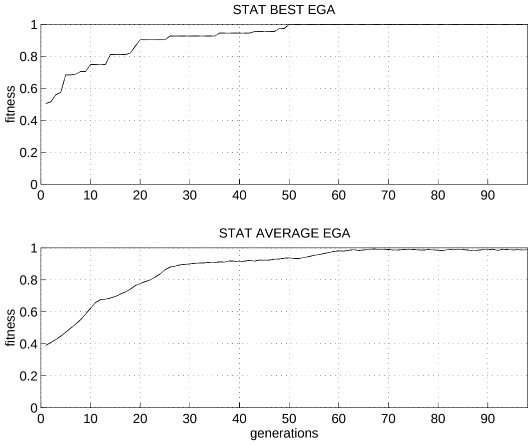

Figure 12: Statistic results

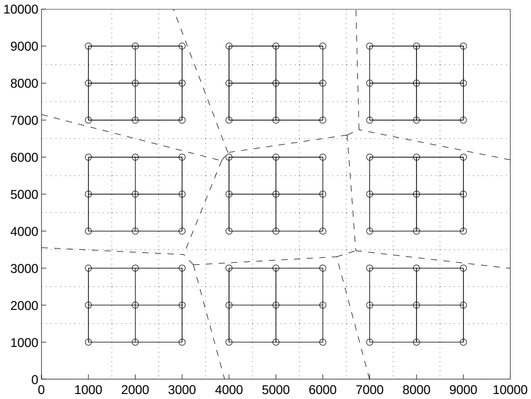

Figure 13: Geometrical results

an optimal solution is found at generation 50.

The solution is displayed in figure 13. It is clearly a correct solution. It must be noted that, even if this solution is trivial to find for a human being because of the symmetries of the problem, it remains as difficult as any other problem for our algorithm. The classification induced by this geometrical result is given on figure 14

6 Conclusion

This study showed us how Genetic Algorithm were suitable to solve the sector regroupement problem with very special constraints. We added a tournament operator and used dynamic parameters to improve the space exploration as well as the selection process. This change brought good improvements to the algorithm convergence rate. As every Genetic Algorithm, the key of success lies in the modeling and the operators. Both must be as close as possible to the application problem. In our case, the representation seems to be very close to the physical application but operators

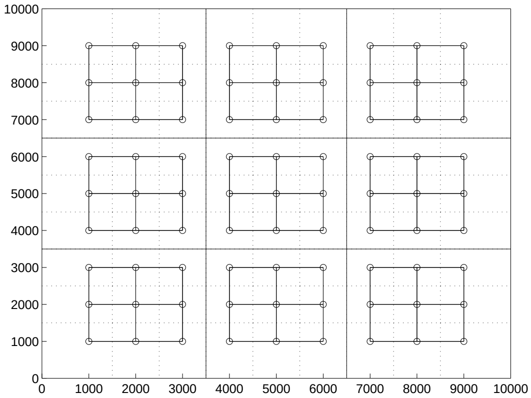

Figure 14: Induced regroupement

can still be improved, though the ones we have used gave very good results.

One possibility to improve this algorithm would be to reinforce the Simulated Annealing concept used in the different operators as Goldberg does in his PRSA [12] algorithm (with binary chromosome). This brings some convergence theorems coming from the Simulated Annealing theory. As for Simulated Annealing, the (stochastic) convergence is only ensured when the fitness probability distribution law is stationary at each state point [1]. We would then have the same drawbacks when this hypothesis is not satisfied.

Finally it would be very interesting to try different acceptance probability laws and different moving probability laws to change the way of exploring the space in our last algorithm.

References

[1] Emile Aarts and Jan Korst. *Simulated annealing and Boltzmann machines.* Wiley and sons, 1989. ISBN: 0-471-92146-7.

[2] Jean-Marc Alliot, Hervé Gruber, and Marc Schoenauer. Using genetic algorithms for solving ATC conflicts. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence Application.* IEEE, 1993.

[3] D. Klingman and J. M. Mulvey, editors. *Networks models and associated applications.* North-Holland, 1981. ISBN: 0-444-86203-X.

[4] P. L. Tuan, H. S. Procter, and G. J. Couluris. Advanced productivity analysis methods for air traffic control operations. FAA Report RD-76-164, Stanford Research Institute, Menlo Park CA 94025, December 1976.

[5] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, and H. Ralambrondrainy. *Classification automatique des données: environnement statistique et informatique.* Dunod, 1989.

[6] Chung-Kuan Cheng. The optimal partitioning of networks. *Networks*, 22:297–315, 1992.

[7] Daniel Delahaye, Jean-Marc Alliot, Marc Schoenauer, and Jean-Loup Farges. Genetic algorithms for partitioning airspace. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence Application*. IEEE, 1994.

[8] Daniel Delahaye, Jean-Marc Alliot, Marc Schoenauer, and Jean-Loup Farges. Genetic algorithms for air traffic assignment. In *Proceedings of the Conference on Artificial Intelligence Application*. CAIA, 1994.

[9] Nicolas Durand, Nicolas Alech, Jean-Marc Alliot, and Marc Schoenauer. Genetic algorithms for optimal air traffic conflict resolution. In *Submitted to the Second Singapore Conference on Intelligent Systems*. SPICIS, 1994.

[10] David Goldberg. *Genetic Algorithms*. Addison Wesley, 1989. ISBN: 0-201-15767-5.

[11] Lester Ingber and Bruce Rosen. Genetic algorithm and very fast simulated reannealing: a comparison. *Mathematical and Computer Modeling*, 16(1):87–100, 1992.

[12] Samir W. Mahfoud and David E. Goldberg. Parallel recombinative simulated annealing: a genetic algorithm. IlliGAL Report 92002, University of Illinois at Urbana-Champaign, 104 South Mathews Avenue Urbana IL 61801, April 1992.

[13] Zbigniew Michalewicz. *Genetic algorithms+data structures=evolution programs*. Springer-Verlag, 1992. ISBN: 0-387-55387-.

[14] Gilbert Saporta. *Probabilités, analyse des données et statistique*. Technip, 1990.