RESEARCH OF PARETO SET BY GENETIC ALGORITHM, APPLICATION TO MULTICRITERIA OPTIMIZATION OF FUZZY CONTROLER

L.Gacôgne * **

* LIP6 CNRS - Université Paris VI 4 place Jussieu 75252 Paris 5 France
tel : 01 44 27 70 02 fax : 01 44 27 70 00 mail : gacogne@poleia.lip6.fr

** Institut d'Informatique d'Entreprise (CNAM) 18 allée J.Rostand 91025 Evry France

KEYWORDS Multiobjective optimization - Genetic algorithms - Fuzzy control - Default logic

ABSTRACT

In a multiobjective problem of decision, it is very often difficult to say which solution we want to keep as aggregation for the different goals we have. We present here an adaptation of an evolution strategy which is able to give a set of solutions. First, the user must define all goals as a list of numeric functions to optimize, like cost, quality... During evolution, it is possible to keep some fixed parameters and make a concentration of the research among other parameters. This is possible thanks to a familly of genetic operators defined in relation with the problem. The user may order the different functions of fitness in relation with their importance for him, but it is possible to avoid a too homogeneous population if there is not at all order in it with a randomly mixing of the whole population at each generation. In this setting, user has to choose the most convenient solution, or to direct the research in an other way by changing a part of the list of fitness functions or their order. We make a trial with fuzzy directed robots which must keep a smooth road inside rings and we give some remarks after a lot of running.

ZUSAMMENFASSUNG

Wenn wir eine Entscheidung treffen haben, wo es eine Vielfalt von Pielen gibt, ist es sehr oft schwierig zu sagen welche Lösung wir behalten wollen als eine Zusammensetzung für die verschiederen Ziele, die wir haben. Wir stellen hier eine Anpassung an einer evolutiven Strategie dar, die eine Reihe von Lösungen ermöglicht. Zuerst einmal, muβ der Verbraucher alle Ziele richtig klarstellen und einreihen und diese Ziele als numerische Functionen ordnen die zu optimisieren sind, wie beisplielweise Kosten und Qualität. Im evolutiven Prozeβ, bleiben einige Variabel unverändert und das Forschen wird auf andere Parameter konzentriert. Dies wird durch eine Familie von genetischen operatoren in Verbindung mit den Problem ermöglicht. Der Verbrauscher mag die Verschidenen Leistung Fonctionen in Verhältnis mit der Bedeutung die er darauf legt, einrein, doch kann er eine zu homogene Bevölkerung verneiden wenn er dieses Bewölkerung durch zufall vermischt. Der Verbraucher hat in diesem Fall, die beste Lösung zu wählen oder das Forschen in eine andere Richtung zu orientieren, indem er einen Teil der Liste umstellt oder aber die Reihe der Leistungfunction. Wir machen einen Versuch mit Robotten die eine weiche Richtung innererhalb eines Ringes behalten und wir verzeichnen einige Bemerkungen nach etlichen Abläufen.

INTRODUCTION

In multiobjective optimization, usually, an association to combine all constraints is setted as a linear combination of all criteria defining a single fitness function. More generally with several heterogeneous criteria, the classical approach is done by elaboration of an aggregation formula. But it is quite hard to make a choice of weights on criteria, and to choose different scores and penalties for features we want favorize or avoid. Which averaging method to choose ? What kind of aggregation formula and which weights to choose? Experiment shows that small changes in the coefficients of aggregation formula may quite different results. This is of course true for mathematical or stochastic methods like genetic algorithms.

The idea to search a Pareto' set of solutions is not to get a best solution according to an aggregation of several goals, but a set of admissible solutions for all criteria. User of an optimization' problema, will have to choose a part of this set.

The question of building a subset of a Pareto' set of solutions in a multiple criteria problem has been first studied in [Korhonen, Laakso 85]. The first idea was to implement an iterative method where directions from current point are specified at each step by the decision maker's preferences according to "flexing goals". Instead of computing a new point close to the last one, they build a subset of efficient solutions (an efficient solution will be a prime point for the Pareto partial ordering). A graphic representation of this subset is produced at each step and presented to the user [Korhonen, Wallenius 88, 90]. The user may change weights for some flexible goals, the aspiration levels of goals (some thresholds to attempt), and also the reference vector of direction of research and the speed of descent along this direction.

The idea to use genetic algorithms to find a set of non dominated solutions in the sense of Pareto instead of a unique solution with a unique fitness function appears with [Schaffer 85] considering fractions of the population in relation with each criteria. [Horn, Nafpliotis, Goldberg 94] applied a nitching algorithm to maintain solutions along the frontier of the Pareto' set.
To keep as long as possible several minima of the fitness functions, they make a partition of the population with the idea of "niching" and make periodically some "migrations" between the sub-population [Franz 72].
In [Viennet Fonteix, Marc 95] there is an application of genetic algorithms in sequence for all criteria separately, about a chemical problem.

Our purpose is to give an adaptation of strategies of evolution for multiobjective problems with the following features :

First we begin with a small random population of points in the space of research and setting a maximal size, we leave growing or a decreasing the population in that range with keeping only the best solutions, that is to say non dominated solutions. We discuss in section II about the pertinency of an ordering for the different criteria.
Secondly, we use a familly of genetic operators in relation with each specific problem, and we make a control on this familly to give a reinforcment for the best of them.
In section III, we apply the strategy of evolution to the tunning of an autonomous vehicle guided with a set of fuzzy rules defined with fuzzy predicates. All the parameters of each fuzzy controller constitute an individual robot. We improve them in an elliptic ring where they have to direct them changing direction and speed. The evaluation of a such individual robot is typically a multiobjective optimization where all criteria are quite difficult to express in view of a global qualification for the their trajectory.

I DEFINITION OF THE PARETO' SET

Let us $f_1, f_2, ..., f_p$ some functions from R^n to R. The goal to find solutions x in R^n to minimize each f_i is not possible in the most of cases, so we define a partial ordering in the space of decision R^n by :

$$x < x' \Leftrightarrow \forall i \ \ f_i(x) \leq f_i(x') \qquad \text{(We say x dominates x')}$$

We say that x and x' are comparable if (x < x') or (x' < x), and incomparable on the other case, a prime element ê is such that there is no e such that e < ê. The set of all Pareto optimal solutions associated to this ordering is the set of all prime elements of R^n according to this relation, that is to say the biggest set E such that all the pairs (x, y) in E are incomparable [Sawaragi, Nakayama, Tamino 85].

Let us give an illustration with four or two functions in R^2, all following examples are given for x, y in [-1, 1], figure 1 shows the Pareto' set for the four following functions :

$$f_1(x) = |x + y - 1|, \quad f_2(x) = |x - y - 1|, \quad f_3(x) = |x + y + 1|, \quad f_4(x) = |x - y + 1|$$

Of course it is easy to get the borders when we sort the population with the $f_{min} - f_{max}$, and the four corner sorting the population with the biggest sum of variations between the criteria, the score (f_1, f_2, f_3, f_4) being (0 2 2 0), (0 0 2 2), (2 0 0 2), (2 2 0 0) at the corners, (0 1 2 1) ... at the midle of the sides and (1 1 1 1) in the center. But any of those ordering may be generalized in view to obtain the border of the searched set.

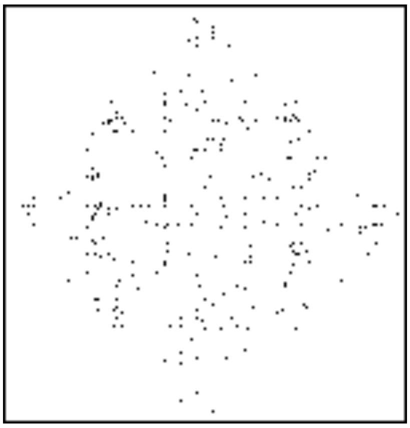

Figure 1 A population of 250 chromosomes reached in 50 generations without any ordering for the four linear functions |x + y - 1|, |x - y - 1|, |x + y + 1|, |x - y + 1| to minimize.

Figure 2 The 50th population of 500 chromosomes whithout any ordering, for the two functions $100|x^2 + y^2 - 1|$ and $100|x^2 + y^2 - 1/4|$. Border of Pareto'set is not connex.

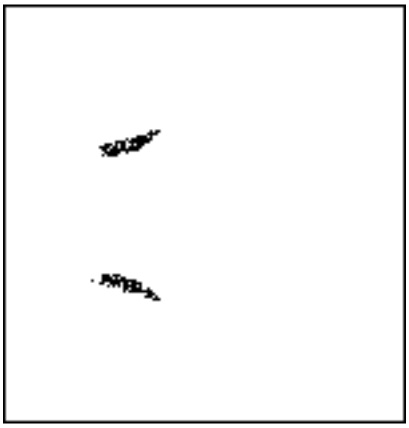

Figure 3 The 50th generation of a population of 250 points for two criters $f_1(x, y) = (2x + 1)^2$ and $f_2(x, y) = 1 + |\cos \pi y(1 - x)|$

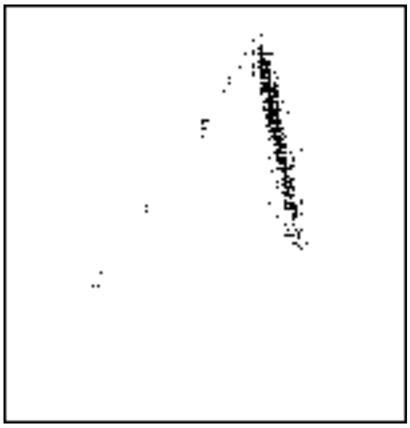

Figure 4 A population of 200 points reached in 50 generations for two criters [Viennet, Fonteix, Marc 95] $f_1(x, y) = 200(1 + 3x - 2y)^2 + 4(1 + 4x - 4y)^2$ and $f_2(x, y) = 200(1 - 2x)^2 + 7(1 + 4y)^2$

II A STRATEGY OF EVOLUTION FOR MULTICRITERIA OPTIMIZATION

II-1° SIZE OF POPULATION AND ELIMINATION

The goal of genetic algorithms is the search of a "good solution" to a problem, we now, want a "good population". We wish to give a number (as small as possible) of solutions in which it will be possible to choose.

We always start any research with a random population P of few points (P_{min} chromosomes). We fix a minimal number P_{min} and a maximal P_{max} for the population' size. At the beginning of any session, making with P_0 an elimination of the worse elements if there are comparisons and in this case, we build new chromosomes until $|P_0| = P_{max}$. A special operator called "migration" is used for that, creating a new random chromosom.

At the generation t the population P_t contains a number of chromosomes bounded by P_{min} and P_{max}, and we suppose all elements of P_t incomparable.
When an operator is applied on x, giving x', we compare x and x', and if they are comparable, we kill the worse of them. In the case where x' is the best or incomparable with x, we have to compare it with the whole population.
When a cross-over is applied, the elimination is also applied between the two children and the whole population. By this way the population may decrease, but generally will be increasing in a first step.
At the end of each generation, we can have $|P_{t+1}| > P_{max}$, so we must kill some chromosomes, then some options are possible :
We say that the "score" of a chromosome x is the p-uplet $(f_1(x), f_2(x), ..., f_p(x))$ and we can sort the population with the lexicographic order. (With this total ordering we have for instance (1 1 1 2) < (1 2 0 0), (1 2 0) < (2 1 0), (2 2 2) < (2 2 3), ...).
But this gives preponderance to the firsts criteria in the list $(f_1, f_2, ..., f_p)$. So it depends of the problema, but the user, has only to give the order of importance of his criteria.
On the other hand, an other solution we tried, is given by the union of populations obtained by the p! permutations of the p criteria.

Example If we try to optimize three functions we can do different orders of them to get an idea of the Pareto'set. For example with the three criteria f_1, f_2, f_3, we make a 50 generations' running on each of six populations respectively sorted with as in figure 6 : (f_1, f_2, f_3), (f_1, f_3, f_2), (f_2, f_1, f_3), (f_2, f_3, f_1), (f_3, f_1, f_2), (f_3, f_2, f_1). The 3 functions are the cones :

$$f_1(x, y) = 50(x - 0.5)^2 + 80y^2$$
$$f_2(x, y) = 100(x + 0.5)^2 + 50(y + 0.5)^2$$
$$f_3(x, y) = 50(x + 0.5)^2 + 100(y - 0.5)^2$$

Their minima 0 are respectively reached at (0.5, 0), (-0.5, -0.5) and (-0.5, 0.5). We can see after 50 generations a move of all population in R^2 towards those three corners when we sort it with in f_1, f_2 or f_3 first.
The centered cloud in figure 6, is obtained sorting the population with a unique average fitness $(f_1 + f_2 + f_3) / 3$.

Figure 5 The 50th generation of a population of size 300 for the 3 criteria whithout any order.

Figure 6 Superposition of 7 populations (each one is the 50th generation with 200 chromosomes) with the 6 lexicographic ordering, and the ordering according to the average of the 3 criteria.

II-2° CONTROL OF THE POPULATION OF OPERATORS

We now, present here all details about implementation of our strategy of evolution :
a) Besides of our population P_t with a variable size of solutions of a problema, we have a population OP_t of different operators.
Operators are the genetic transitions (mutation, transpositions of two genes, gaussian noise if numeric, creation or suppression, or changing priority of a randomly chosen rule as we see latter, ...) which are randomly generated at the beginning in an other population.
Moreover, as a chromosom contains coded "rules" we shall introduce the "strength" f of a rule inside a chromosome, this is the average (among 0 and 100) of the satisfaction of the rule during an improvement. Of course this parameter will be variable according to the improvment and the context of the other rules. We define then the "supmin" operator which deletes the weakest rule in a chromosome, and the "mutmax" operator which is a mutation on the strongest rule.
With not-fixed length chromosomes, we use operators which are not mathematical functions because they choose sites and genes to make transformations, each time they are applied.

b) When an operator is applied to a chromosome, we immediately compare the father and the son (the four parents and children in the case of cross-over). By this way, when decreasing in a "well" for the cost function is primed, we only keep the last descendent.

c) If the aim of the algorithm is to get a minimum of a list of fitness functions computed on the population whose elements are potentials solutions to a given problem, we also give a cost to the genetic operators and make a sort of them at each generation.
We give for this a cumulated capacity to decrease the fitness, to each genetic operator. The comput of it, is 0 in initialization and incremented by $\sum f_i(x') - f_i(x)$ each time the operator f is applied to x giving x'. This algebraic "fitness" for the operators is reinitialized to 0 when any modification occurs during a generation. In all cases, the first operator is applied to the first individual, the second to the second and so on... For crossover, a second individual is randomly taken in the biggest part of the population before or after the first parent. By this way, we don't need any probability to fix.

d) Control of the operators : at each generation, if at least one progress is made, we sort the operators and we create a new of them in the type of the best operator. So, we have an adaptative control of the whole population of operators. In the opposite case the whole population of operators is randomly reinitialized with the list described in (a) to avoid a convergence to the same operator. We note that roughly crossover is going better after action of transitions like suppression or mutations.
Observation often gives a better heteregeneous population with few applications of cross-over, and we can also see a long number of iterations before the list of fitness functions decreases, so we can also change a part of population with randomly new chromosomes.

III APPLICATION TO THE SETTING OF A FUZZY CONTROLLER

We want to apply those methods to the setting of a fuzzy controller for the guidance of an autonomous vehicle in an elliptic road. To find a set of rule for this aim is not really difficult, but difficulty is strongly increasing if we want also to find the predicates, some other parameters and moreover there is a difficulty to define what we intend to do.

There are several criteria to optimize : we want to take into account the length of the run complied, to the softness of trajectory (the vehicle makes changes for its direction and its speed too), and also to the complexity of the set of rules. We shall see that it is possible with our method, to bring our attention to all or a part of the parameters to modify according to the familly of genetic operators we choose.
Let us define first, what is a chromosome as a package of predicates and fuzzy rules.

III-1° A FUZZY DEFAULT CONTROLLER FROM $[-1, 1]^N$ TO $[-1, 1]^P$

For a fuzzy set A defined on a universe U, we note μ_A the membership function of A, and supp(A) $=\{x \in A / \mu_A(x) \neq 0\}$ the support of A and ker(A) $= \{x \in A / \mu_A(x) = 1\}$ the kernel of A.
First we define what is a fuzzy default controller from $[-1, 1]^n$ to $[-1, 1]^p$:

Rule We keep an idea to give an order of priority to separate particular and general rules [Gacôgne 93 a]. Let A be a fuzzy set with continuous membership function μ_A on $[-1, 1]^n$, we say that (k A c) is a rule of order of priority (or priority) k if $c \in [-1, 1]^p$ and k is an integer, (k A c) is a formalization of "if (x is A) then (y is c)".

Fuzzy default controller If $k \in N$ and C_k is a finite set $\{(k\ A_1\ c_1) \ldots (k\ A_i\ c_i) \ldots\}$ of rules with the same priority, the Sugeno function f_k [Sugeno 85] defined on the reunion of the supports of the A_i will be defined as usually in fuzzy control for $x \in [-1, 1]^n$ by : $f_k(x) = (\Sigma\mu_{Ai}(x)c_i) / (\Sigma\mu_{Ai}(x))$.
So a fuzzy default controller C is a finite set of rules whose orders (of priority) are all integer from 0 to n (0 for the most specific rules and n for the most general). The Sugeno's function associated with the controller will be defined for all $x \in [-1, 1]^n$ by :

$f(x) = f_k(x)$ if k is the first order of priority such that x is in the domain of f_k, and 0 otherwise.

(The last condition is equivalent to a last rule more general than all the other and giving a conclusion 0 without condition.) We assume that C is minimal that is to say it is not possible to have the same function with a proper subset of C, but of course minimal set as C is not necessary unique.

Remark : it is possible to imagine a softer system where each f_k can be extended to [-1, 1] with 0/0 = 0 as convention, and where $f(x) = \phi\ (f_0(x), f_1(x), \ldots , f_n(x))$.
ϕ can be $\phi(a_1, \ldots a_n) = (\Sigma m_i a_i) / \Sigma m_i$, where m is decreasing. To illustrate this point of view, ϕ could be $m_i = n + 1 - i$ as a prototype. See [Yager 92] for a such system.

III-2° FAMILLY OF FUZZY PREDICATES USED IN THE APPLICATION

We intend to apply our strategy of evolution for the tunning of fuzzy default controller. So, we hope not only to get sets of fuzzy rules to describe a comportment, but moreover, a set of predicates. This has been yet done with a lot of research works thanks to genetic algorithms or other topics like neural networks. But it is very difficult to consider the problema in a whole generality. More often a part of parameters are fixed in experiments, and the machine learning is devoted to the others. Therefore, we want make a compromise to learn automatically a set of predicates which can be considered as union of all famillies fluently used in applications.

We define then a familly of predicates on [-1, 1] with four numeric parameters.
The number (nb in figure 7) of the familly is deduced by $p \in \{1, 2, 3, 4\}$
(it will means 2p + 1 predicates).

The second one $r \in [0, 1]$ will be called contraction ("contr") of the predicates. The next number $h \in [0, 1]$ is a coefficient ("trap") of overlaping of them and $c \in]0, 1]$ is a fuzzy degree of them ("fuz").
The family is built on the membership functions :
Let us m_k the "top" of triangular functions for $0 \leq k \leq p$ (with $m_{-1} = m_1$) defined by :

$$m_k = r\left(\frac{k}{p}\right)^2 + (1 - r)\frac{k}{p} \quad \text{and} \quad \mu_k(x) = \min\left(1, \max\left(0, h + 1 - \frac{1}{c}\left|\frac{m_k - x}{m_k - m_{k-1}}\right|\right)\right)$$

Each membership function is built as a troncature of a "triangular function" with a maximum equal at $1 + h + r.m_k$ for m_k, (so $m_0 = 0$) and linear between it and the point $(m_{k-1}, 0)$, symmetric around m_k and bounded by [0, 1].
We call ZE the fuzzy set with μ_0 membership function, and we shall impose a symetry such that $\mu_{-k}(x) = \mu_k(-x)$. We can call the predicates PS and NS for k = 1 or -1 whose membership functions are respectively μ_1 and μ_{-1}, PM and NM whose membership functions are respectively μ_2 and μ_{-2} and PB and NB for ±3.

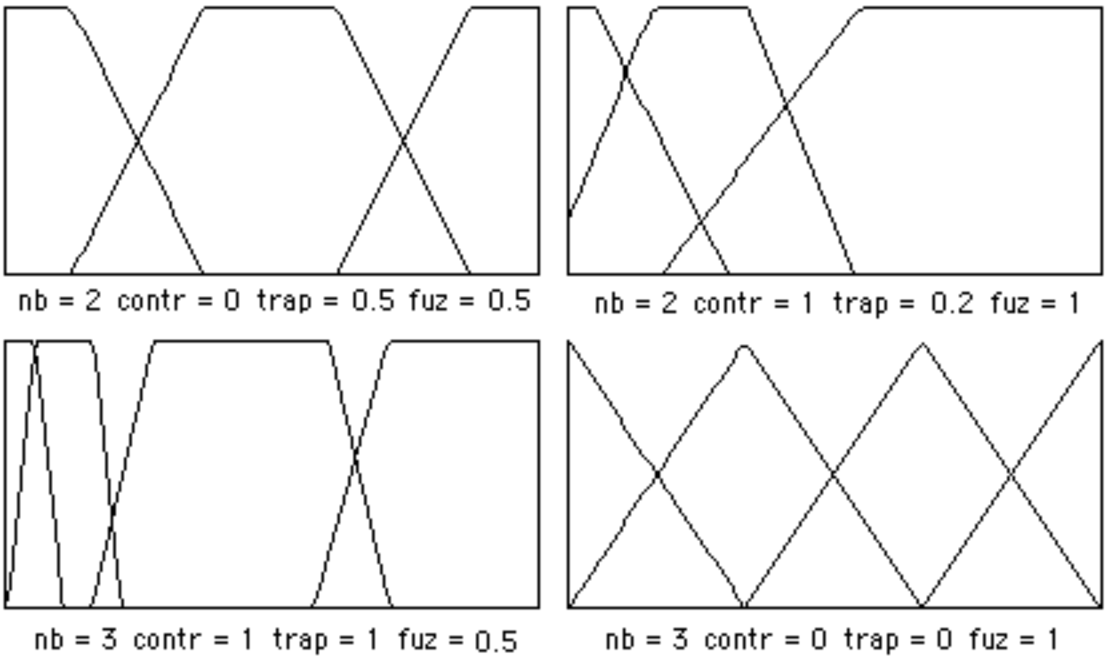

Figure 7 Example of famillies of predicates on [0, 1] with nb $\in$ {0, 1, 2, 3}, contr, trap $\in$ [0, 1] and fuz $\in$]0, 1].

III-3° APPLICATION TO THE SIMULATION OF AN AUTONOMOUS VEHICLE

Coding of a chromosome

The chromosomes don't have a fixed length, the structure of them is (nb, contr, trap, fuz, ouv, R_1, R_2, R_3, ...) where nb, contr, trap, fuz are the last described coefficients, "ouv" an angle in degrees, and R_i are the rules.

Each rule R is on the type (pr f h_1 ... h_{nh} c_1 c_2 ... c_{nc}).

The hypothesis are symbol $h_i \in$ {any, nb, ns, ze, ps, pb} and the consequents c_j are numeric integers in the field [-100, 100], pr is the priority of the rule, and f is the "strength" of it.

The meaning of the predicates NB, NS, ZE, PS, PB is respectively negative big and small, zero, positive small, medium and big, are defined with trapezoidal fuzzy sets as above thanks to the four firsts numbers of the chromosom.

More precisely, a rule will have two premisses : the two distances from the point to the borders of a road, according to an angle ± ouv with its direction, and two conclusions in [-100, 100] giving the part in percent of the change of direction (maximal is am = 45°) and the part of modification of the speed (maximal is acc = 4 pixels).

Training and distance to the ellipse

The vehicle is runing between two ellipses with same exentricity e and axis respectively a and a + gap. It has of course the constraint to remain between them and to make a maximal number of steps.

Suppose the vehicle in position (x, y), we want have the distance d under a maximal distance dm, to the ellipse (half axis = a, exentricity e) in the direction α. In respect of the actual position (x, y, α) the distance d is computed by the algorithm :

Let us $b = \sqrt{}(1 - e^2)$, co = cosα, and "out " the boolean $x_0^2 / a^2 + y_0^2 / b^2 > 1$ (the point is outside)

If co = 0 then if out then if $a < |x|$ or $0 < \alpha y$ then $d \leftarrow dm$
else let $y_c \leftarrow b \sqrt{}(1 - (x / a)^2)$, $d \leftarrow \min(|y - y_c|, |y + y_c|)$
else let $y_c \leftarrow b \sqrt{}(1 - (x / a)^2)$, $d \leftarrow |y_c - \mathrm{sgn}(\alpha).y|$
else let $t \leftarrow \tan\alpha$, $m \leftarrow tx - y$, $k \leftarrow 1 / a^2 + t^2 / b^2$, $\partial \leftarrow k - (m / ab)^2$
if $\partial < 0$ then $d \leftarrow dm$
else $\partial \leftarrow \sqrt{}\partial$, $x_c \leftarrow mt / b^2 - \partial$
if out if $co.(x - x_c) < 0$ then $d \leftarrow dm$
else $d \leftarrow \min(|x_c / k - x|, |(x_c + 2\partial) / k - x|) / |co|$
else $d \leftarrow |$ (if $0 < co$ then $x_c + 2\partial$ else x_c) $/ k - x| / |co|$

Each moment, the vehicle compute with it, the distances di and de to the inside and outside ellipses. They will be the inputs of fuzzy control.

Criteria to optimize

We tried our runs with two to four criteria among :

1 - The number of steps, it is clear that this unique function to maximise would give vehicle very slow.

2 - Second one is the ratio between the performed distance and the maximal distance for a vehicle with maximal speed.

3 - An other criter is the "road-holding" [Gacôgne 94]. It is a number in percentage : tr = $\left| \frac{|da|}{am} + \frac{dp}{2acc} - \frac{1}{2} \right| \in [0, 1]$

Where da and dp are change of direction and step realised each current run, "am" and "acc" are maxima for these changes. The nearer tr is 0, more we estimate the trajectory good, when close to 50, we estimates bad, and more tr is near 100, more the trajectory is opposite of a good driving.

4 - We also make a count of the number of meaningful symbol in the set of rules.

Experiment

A minimal size for the population is necessary to make a good exploration of the space of research, because the elimination with Pareto' order may incite to search while the minimal size is not reached. So the evolution starts only after this goal performed. With a minimal population of size 5 and a maximal size 25, it happens sometimes a decreasing under 5, and consequently a migration of new randomly individuals in the population. But generally the size is, after some generations, always maximal except for few criteria. This moving size depends of the number of fitness functions. For one we could have, of course, a population of a unique incomparable element, for three functions we have often got a moving population around 10 elements, but with four or five functions, we are obliged to fix a maximal size (P_{max} = 20 to 50 for example)

The first surprising fact, with all experiments, is the move from two captors (+ouv and -ouv from the direction of the robot) to a unique one (ouv = 0°). Then the rules are just saying if the obstacle is the left or right side of the road.

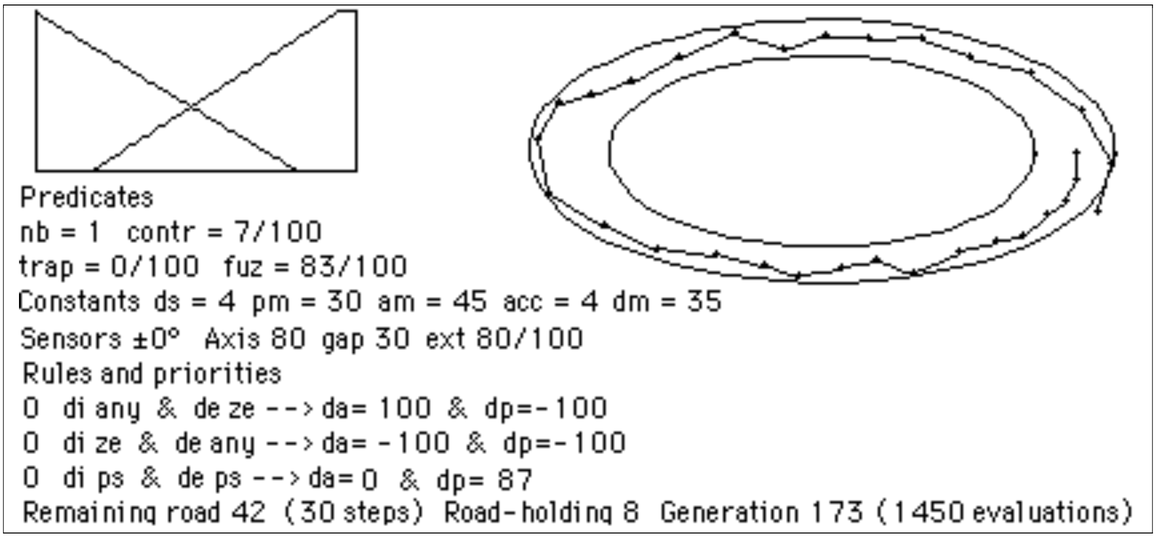

Figure 8 A simple set of rules with 2 predicates. The rules are "when near of a side, turn and brake, when far of the two sides, acceleration". All departures are from the right towards the bottom and left of the circuit.

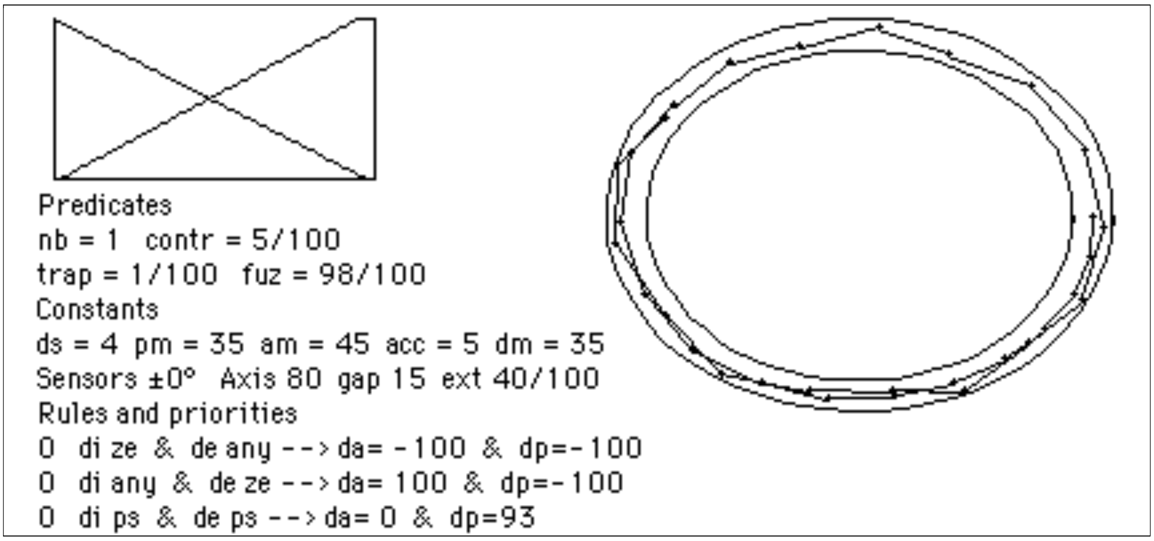

Figure 9 Evolution of the precedent solution and test on an other ring. When the last rule has a priority 1 (a general rule), we get a similar chromosome, but with a more jerky trajectory.

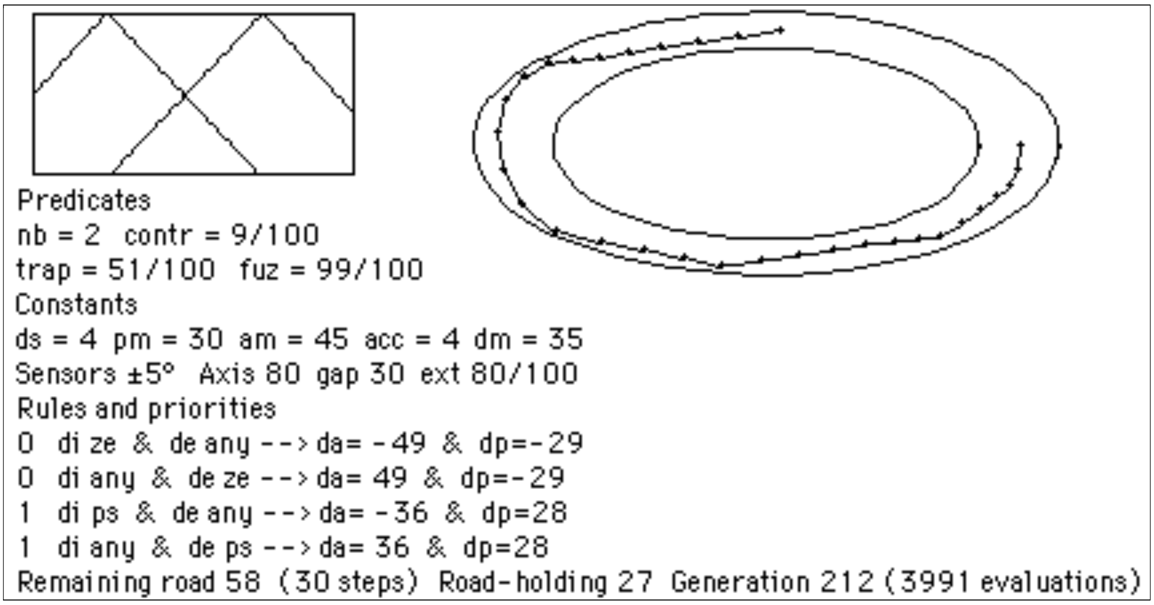

Figure 10 A solution with three predicates (remark that any number in [0, 1] is really ZE or PS or PM) and 2 general rules (priority 1) for a moderate acceleration, and 2 particular rules (priority 0) for braking. Fitness is compound by the three (number of remaining steps, remaining road, road-holding).

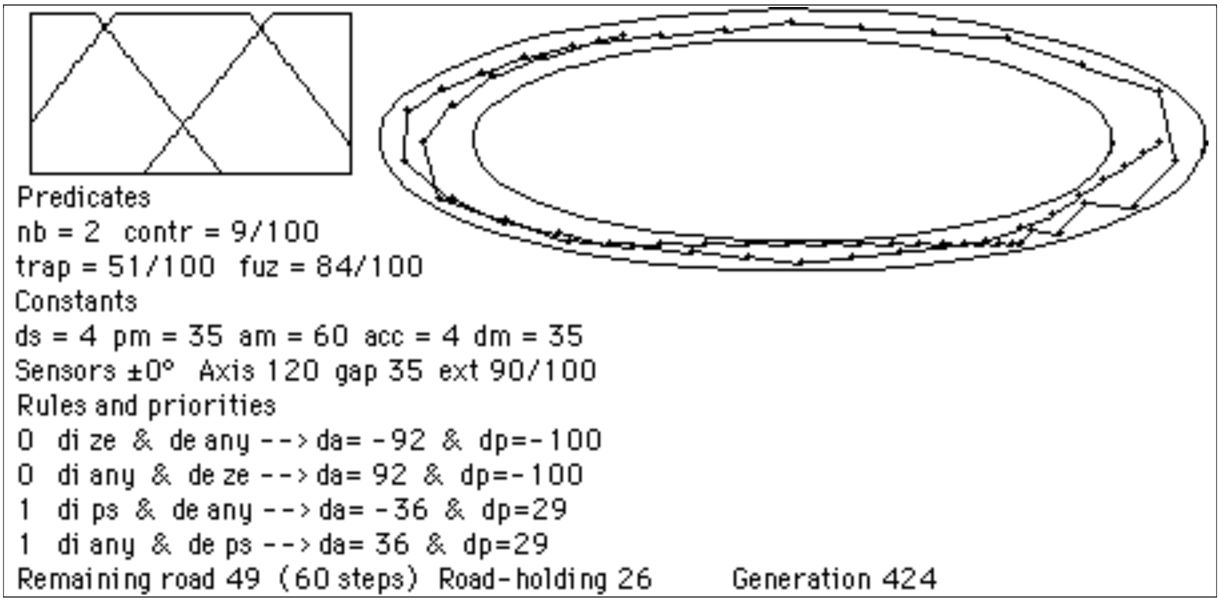

Figure 11 A "descendant" of the last solution, but with more trapezoidal predicates, and more braking when a side is near. In this running with had fixed a maximal change of direction 60° instead of 45°, because the ring is more abrupt with e = 0.9 instead e = 0.8. In spite of this more difficult ring, the comportment is quite good here, and we show here a trial with 60 steps.

Conclusion

During a lot of sessions, we always can observe emergence of solutions with a few number of predicates (nb = 1 or 2) and in the most of cases with a quite big overlaping, a few number of rules and a very small angle "ouv" (0° to 5°) for the positions of the sensors. That means the vehicle is able to run with only a frontal view.
Observation of the operators shows that noises and mutations are the best noted among the familly of operators.
We tried a lot of running with or without order for the fitness functions, with three or four functions to optimize, but we must say that it is quite difficult to formalize really what we want in a such case. It is indeed a very complex comportment we want to estimate : what is a smooth road-holding and how choose a set of circuits for tests ?

References

Bäck Th. Evolutionary algorithms in theory and practice. Oxford University Press, New York 1995

Bäck T. Kursawe F. Evolutionary algorithms for fuzzy logic IPMU p.659-664, 1994

Bäck Th. Schwefel H.P. An overview of evolutionary algorithms for parameter optimization. Evolutionary computation 1 p.1-23, 1993

De Jong K.A. Genetic algorithms are not function optimizers. Foundations of genetic algorithms 2, p.5-17. Morgan Kaufmann Pub. San Mateo, 1993

Franz Non-linearities in genetic adaptative search, Doctoral dissertation, Abstracts International 33 (11) 5240B-5241B University of Michigan, 1972

Gacôgne L. Une stratégie de hiérarchisation des règles dans les systèmes à base de connaissance, Actes des journées Volcan IA p.226-300, 1993

Gacôgne L. Apprentissage génétique global d'un contrôleur flou à deux variables basé sur la simulation d'un véhicule autonome. IPMU p.1099-1104, Paris, 1994

Gacôgne L. About the fitness of fuzzy controllers whose rules are learned by genetic algorithms. EUFIT p.1523-1531, Aachen, 1994

Goldberg D.E. Genetic algorithms in search optimization and machine learning, Add.Wesley, 1989 Holland Une échappatoire à la précarité : les possibilités de l'application des algorithmes généraux d'apprentissage aux systèmes parallèles à base de règles, Apprentissage symbolique p.523-554 Cépaduès, 1993

Horn Finite Markov chain analysis of genetic algorithms with niching, Proceedings of the 5th Int. Conf. on G.A. p.110,117, 1993

Korhonen Laakso A visual interactive method for solving multiple criteria problem, European journal of Operational Research Society, vol.32 n°7 p.577-585, 1985

Korhonen P. Wallenius J. A Pareto race, Naval Research Logistics Wiley, vol 35 p.615-623, 1988

Korhonen P. Wallenius J. A multiple objective linear programming decision support system, Decision Support System North Holland vol 6 p.243-251, 1990

Kouada I. Sur la propriété de domination et l'existence de points Pareto-efficaces en optimisation vectorielle convexe, RAIRO Op.research vol 28 n°1 p.77-84, 1994

Michalewicz Z. Genetic algorithms + data structures = evolution programs, Springer Verlag 1992

Rudolph G. Convergence analysis of canonical genetic algoritms. IEEE Transactions on neural networks, Special issue on evolutionary computation, p.63-66 IEEE Press, 1994

Sawaragi Y. Nakayama H.Tamino T. Theory of multiobjective optimization, Academic Press, 1985

Schaffer J.D. Multiple objective optimization with vector evaluated genetic algorithms, Proceedings of an International Conf. on GA and their applications p.93-100, 1985

Schwefel H.P. Numerical Optimization of computer models, J.Wiley 1981

Schwefel H.P. Systems analysis, systems design and evolutionary strategies, System analysis, Modeling and Simulation vol.7 p.853-864, 1990

Schwefel H.P. Evolution and optimum seeking. Sixth generation computer technology series. Wiley, 1995

Steuer R.E. Multiple criteria optimization, theory, computation and applications, Wiley 1986

Sugeno M. An introductory survey of fuzzy control. Information and Science n°36 p.59, 1985

Thrift P. Fuzzy logic synthesis with genetic algorithm Proceedings of the fourth International Conference on genetic algorithm pp 509-513 San Diego, 1991

Viennet R. Fonteix C. Marc L. Optimisation multicritère à l'aide d'un algorithme génétique diploïde, Evolution Artificielle, Brest, 1995

Yager R.R. Hierarchical representation of fuzzy if-then rules, Proc. IPMU Conference Mallorca, 1992