Multiple Objective Optimization of Fuzzy Rules for Obstacles Avoiding by an Evolutionary Algorithm with Adaptative Operators

L.Gacôgne * **

* LIP6 - Université Paris VI 4 place Jussieu 75252 Paris 5°
tel : 01 44 27 88 07 mail : Louis.Gacogne@lip6.fr

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

Keywords fuzzy control - multiple objective optimization - evolutionary algorithms - Pareto' set

Abstract

We focus our attention on the classical problem of a simulation of obstacles avoiding. We tempt for this to make raising lists of fuzzy rules by an original evolutionary algorithm. We look for an emergence of such rules system for a robot which is able to change its direction and its speed, without precise constraints. It is really not a well-defined problem, because when there is a target (some zona to reach) or if there is a constant speed, genetic algorithms have already given good solutions. Here, we would wish to see a birth of a good road-holding comportment, that is to say to slow down in the curves and to accelerate in the straight lines. In the same time, we want to get the best response to the obstacles avoiding problem, namely solutions which are satisfying in every occasion. We are face to the well-known dilemma of learning : a good solution in a precise playground will be a set of particular rules, solutions given by randomly circuits will not be comparable themselves and it is difficult to evaluate a solution on a big number of different playgrounds. On the other hand, we wish short and readable solutions, so, as every concrete problem we have several objectives to optimize.

I THE FUZZY CONTROLER AND ITS ENCODING

In the field of problems raised by the tuning of a fuzzy controller, the first is the choice of inputs. So, if a robot is able to evaluate the distances from it to the obstacles, the rules may be very different according to two lateral distances or three, or if we take into account those distances and their variations or not between two measures of them. The difference in robot comportment may be also important when distances are measured among ±45° or ±60° or ... The second problem is the determination of predicates as "small", "middle" ... which are necessary to qualify the physical grandeurs, it may be easy only to fix an interval. For instance, if a distance is evaluated until a "maximal reconnaissance" d_m, then $[0, d_m]$ has just to be contracted in [0, 1] on which the traditional predicates ZE, PS, PB will be used. In the experiments which are described here, we focused only on the fitness functions to be optimized, therefore we fixed all other parameters as following :

Inputs of the fuzzy controller : 3 distances dl, dm, dr respectively taking the measures (in pixels) of the left, front and right distances according to ±45° from the direction of the robot.

Used predicates : N, Z, P with triangular shape on [-1, 1] cutting themselves at the height 1/2, and we use also the predicate ANY, whose the membership function is always 1, indicating, for a rule, that some input is not employed. d_m is set up with 35 pixels. Those predicates may be also in variable number and with various shapes [Gacôgne 97].

Outputs of the controller : the two outputs are the angle variation "da" given as a fraction of the maximal angle to turn, which is set $a_m = 60°$, and the variation "dp" of the step, given as a fraction of the maximal "acceleration" set up as 4 pixels.

Controller : Sugeno's type (each of the two outputs is the average of the corresponding conclusions of the rules, balanced by the rules satisfaction degrees) the "satisfaction degree" of a rule is the t-norm (conjunction) min of the premises [Sugeno 85].

Rules : the rules have not an established number, that is to say, the genetic learning may find a more or less important list of rules which may overlape themselves, for example, a found solution is: dl = P dr = Z → (da = 95/100) (dp = -62/100)
dr = Z → (da = 52/100) (dp = 66/100)
Nevertheless, a condition for symmetric situations, is imposed, that means four rules for the example above.
A solution, a chromosome in the evolutionary algorithms language, is here a list of rules, and each rule is a quintuplet, for instance (P, ANY, Z, 0.95, -0.62) for the first one. The domain of the chromosomes, the space of research is $E = (\{N, Z, P\}^3 \times [-1, 1]^2)^*$.

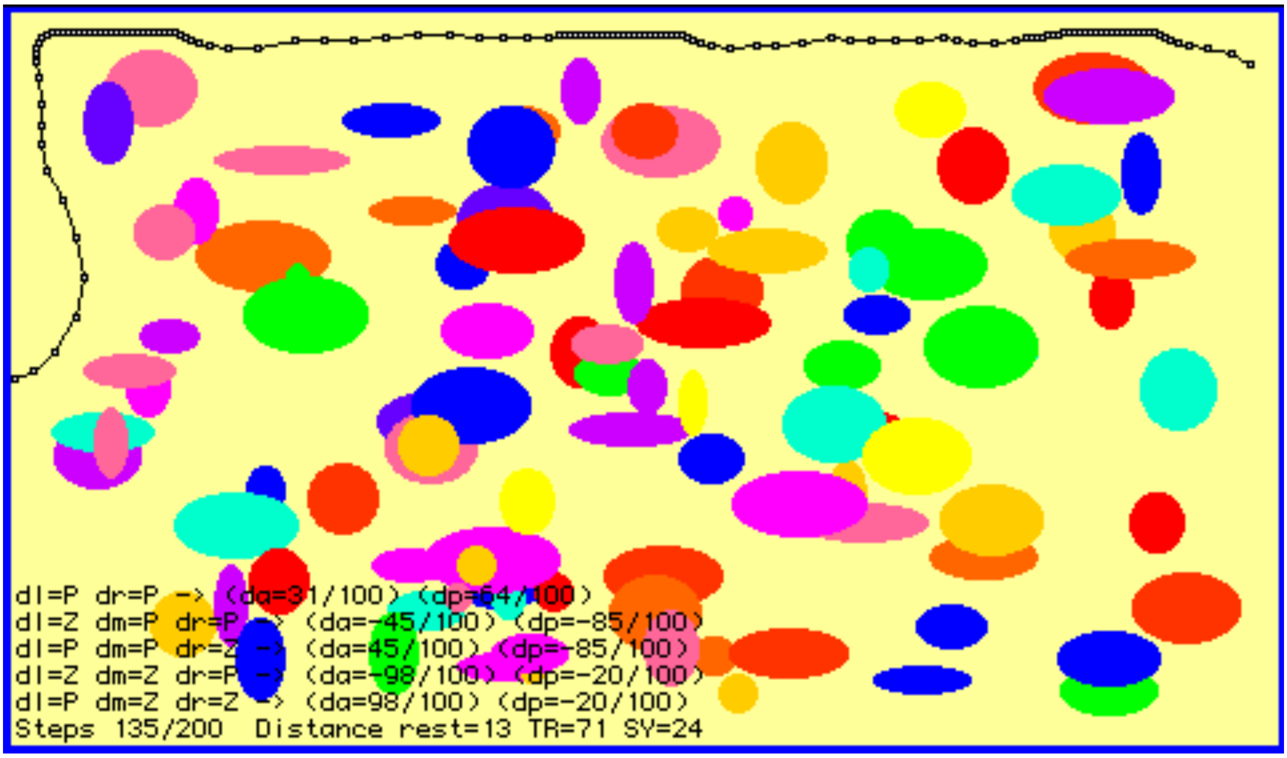

Figure 1 Example of a fuzzy controller with a symmetric rule (P ANY P → da = 0,31 dp = 0,64) to accelerate, a rule (Z P P → da = -0,45 dp = -0,85) and its symmetric, strongly contraindicated by (Z Z P → da = 0,98 dp = -0,20) and its symmetric. Contrasted with the following experiments, here, starting from the left side, the aim is to reach the right side. The main fitness is only defined by the resting distance.

II THE USED EVOLUTIONARY ALGORITHM

We introduce a genetic strategy which is strongly different than the classical approach, our aim is to leave apply the genetic operators along the run of evolution according to their capacity to make evolution inside the population. On the other hand, we always search to explore the domain, the training showing often a too big homogeneity of the population and a too fast convergence [Schwefel 95]. Several ideas may be tested for that, the "niches" or sub-populations where evolution inside is dragged by different fitness functions, the "eras" or evolution times followed by a special treatment is done... The idea to accommodate such strategies to a multiple objective optimization to reach a population with "good" solutions instead of a unique solution, is appeared with [Schaffer 85] who considers a partition of the population according to each one of the criterias, next with [Kursawe 91] and [Horn 93].

To keep, a most possible long time, several local minimas for the functions to optimize, they do a partition of the population ("niches"), and periodically do migrations between those niches, idea already given by [Franz 72].
Before implementing those ideas, let us expose our algorithm ESAO (Evolution Strategy with Adaptative operators) [Gacôgne 94].

1) A chromosome is a rule list, the rules are described as above as quintuplets. The initial population P_0 is composed with random individuals in a number N_{min} (we set it as 3).

2) A random operators population OP_0 (mutations, transpositions for two genes, Gaussian noises for numeric genes, creation or suppression of a rule, cross-over with one site, with two sites, ...) is composed among a lot of operators chosen in relation with the problem. Each of them is associated with a score (zero at the beginning) and then cumulated on all the generations with their algebraic capacity to decrease the fitness of the individuals.

3) Each generation, operators of OP_t are applied in the order of their score, to the sorted population P_t. When a such transition t is applied to an individual x, we immediately sort the father and the son or the four in the case of a cross-over, to keep only the best. The score of t is incremented by f(t(x)) - f(x) if f is the fitness. Thus we avoid an accumulation of neighborhood elements in local minimas.

4) If, at least, one progress is observed, at the generation t, we sort the operators according their score, and we create a new one operator, with the same kind of the best operator. We have thus an adaptation of the operators population OP_{t+1}.

5) In the opposite case where any amelioration is observed, all the operators are reinitialized and randomly mixed to avoid a too fast convergence of this population towards a unique operator. It could be the case in which for example of a cross-over giving fast ameliorations in a first phase, could not be able to leave place for other operators like mutations which could be better to refine the solutions in a second time. The population P_{t+1}, (a part or the whole except the best) may be itself reinitialized.

6) If the number of evaluations of the fitness goes over the threshold (for example 500000), the algorithm is stopped, in the other case one goes in the third point.

Adaptation to the research of the Pareto' set in the case of multiple objective optimization

For several criterions f_1, f_2, f_3, ... f_p, to minimize, we say that two of them i and j of the domain E of the functions f_1, ... f_p, are comparable, if for every criterions, they are in the same order. We say that they are incomparable in the opposite case. For example, for two criterions, the results (0, 1) and (1, 2) are comparables, (0, 1) and (1, 0) are not. An individual is Pareto minimal (or prime) if it is minimal for this partial ordering. The Pareto' set is defined as the set of all prime elements in E (and so incomparable between themselves).

Instead of following the [Fonseca, Fleming 95] technic whose the main principle is to keep a total ordering on the whole population, our main idea is a move of the population towards the Pareto' set. It consists to obtain a population of solutions, the bests according to all the criterions, but incomparable by pairs, leaving the choice to the user beyond a small number of those solutions. A concrete view of this problem was studied by [Korhonen, Laakso 85]. For two dimensional encoding solutions, a graphic view of an approximation of the Pareto' set is submitted to the user [Korhonen, Wallenius 88, 90]. This one may change the relative weights for some aims to modify the direction of research and its speed.
We find, about a multiple objective problem in chemistry, in [Viennet Fonteix, Marc 95] an application of genetic algorithms, but they employ it on each criterion successively.

1) We don't follow any of those two ways, our option is to simulate (in parallel) the evolution of several sub-populations. Each of them being sorted according to the lexicographic ordering, corresponding only with the priorities we want to give for the objectives functions. More precisely, if we are interested with three criterias for example : f_1, f_2, f_3, and if the optimization of f_3 is obviously

less important than those of the two firsts, but if we have some hesitation between them, about their importance, then we will sort a first population P_1 according to (f_1, f_2, f_3) and a second population P_2 according to (f_2, f_1, f_3).
For all multiple objective problem, it is indeed, patent that it is difficult to conceive a linear or not combination of the criterias with or without bonus/malus which satisfy the experts. On the other hand, it seems easier just to sort the criterias in order according to their importance (price, fitness, safety...).

2) The second modification, we bring to the above algorithm, is that each individual created by a genetic operator, is compared with all its sub-population, and then, either is deleted, or is kept, may be doing some eliminations. The population thus, always remains composed with incomparable (by pairs) elements, and then the population converges towards a part of the Pareto' set.
A consequence of this choice is the variable size of each component of the population, it may increase, so we are obliged to set a maximal size N_{max} in order of 25.

3) The two populations P_1 , P_2 according to this strategy are evolving towards the bests with the respective meaning of f_1 and of f_2 but they are not crossed at the time of eras, but permanently. It is not an endogamy but two different sortings for those populations.

Example : We give here two examples of the Pareto' set for two multiple objective problems. Beginning with a random weak sized population in $[-1, 1]^2$, we always keep the best individuals incomparable in the Pareto' meaning, and maintain exploration of the space. We use for this, a variable sized population between limits, N_{min} and N_{max} (several hundreds) and also a population of genetic operators defined in relation with the considered problem and evaluated according to the cumulation of their fitness to ameliorate the individuals.
Of course, our problem is much more difficult, because, even with three criterias to measure the fitness about the obstacle avoiding, the road-holding and the encoding, the research space is not R^2 as here, but $E = (\{N, Z, P\}^3 \times [-1, 1]^2)^*$.

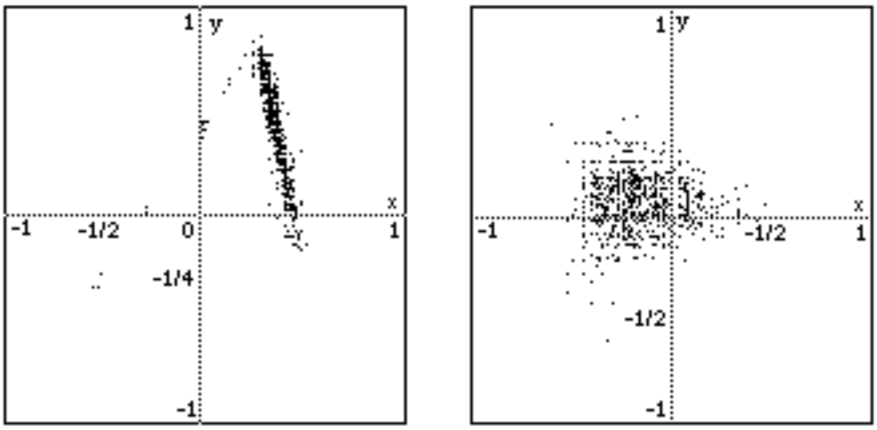

Figure 2 Left, a 200 points population got in 50 generations for two criterias
$f_1(x, y) = 200(1 + 3x - 2y)^2 + 4(1 + 4x - 4y)^2$ and $f_2(x, y) = 200(1 - 2x)^2 + 7(1 + 4y)^2$ whose minima 0 are respectively for (-1/2, -1/4) and (1/2, -1/4).
Right the 50^{th} generation of a population with the size 300 for the 3 criterias without any ordering
$f_1(x, y) = 50(x - 0.5)^2 + 80y^2$, $f_2(x, y) = 100(x + 0.5)^2 + 50(y + 0.5)^2$
and $f_3(x, y) = 50(x + 0.5)^2 + 100(y - 0.5)^2$,
Their minima 0 are respectively reached for (0.5, 0), (-0.5, -0.5) and (-0.5, 0.5).

III APPLICATION TO THE OBSTACLES AVOIDING

The searched performance

We aim to provide an adaptation of the evolutionary strategy for multiple objective problems, with different original features, and also to adapt it for the rules learning problem for a fuzzy controller to guide a robot.

Many other similar works use a genetic algorithm to optimize a numeric function. For that, it is generally a linear combination of the criterias which is empirically chosen. The wrong side of this choice resides in the fact we make an aggregation of heterogeneous physical grandeurs, and we are obliged to set the coefficients of this combination in an empirical way. An other current point of view consists to define the fitness by cases where malus and bonus are allocated.
For example, it is possible to encode the famous traveler problem by a sequence of towns, giving a handicap for each repetition of town and also for each missing town. We must confess that if those "taxes" are well done (respectively to the distances between towns), the algorithm gives a quite good solution and also avoid to correct the chromosomes born by the genetic operators. It is the reason why evolutionary algorithms are seductive for their capacity to raise solutions starting from very few parameters, we did not want to choose an arbitrary numeric fitness as a combination, but a vectorial function. In all the following text, the fitness function is a tri-dimensional vector whose components are the three criterias in the chosen ordering of importance [Korhonen, Wallenius 90].

The obstacles avoiding: the main indicator or this one which seems the most in priority should be the number of run steps to be maximized or the browsed distance. In the second alternative, we can favorize the fastest individuals.
We take this second one, looking to minimize $DR = 1 - (\sum pas) / (k_m.p_m)$ where k_m is the maximal number of steps and p_m the maximal step. We bring DR into an integer among [0, 100]. The value 0 would be so the fitness of a robot capable to do the maximal number k_m of steps with the maximal speed.

Road holding : we aim to find a coefficient (to minimize) among [0, 100] to translate an idea of road holding.
Let us say again that da is the change of direction between two inputs, a_m is the maximal angle to turn, ds the minimal step, dp the variation of step and "acc" maximal acceleration. If da and dp are the real parameters in all that follows, we note a global comportment, but, on the other side, if we judge the outputs of the controller, we note more a rules system, of course the real parameters must be taken into account. So, we consider that the more we have a road holding, the more the following equivalences are satisfied :

$$(da/a_m \text{ is Z} \Leftrightarrow dp/acc \text{ is P}) \text{ and } (|da|/a_m \text{ is P} \Leftrightarrow dp/acc \text{ is N})$$

We could define then TR as the averaging of $\big|\, |da| / a_m + dp / acc - 1/2 \big|$,
or $|\mu_Z(da/a_m) - \mu_P(dp/acc)| + |\mu_P(|da|/a_m) - \mu_N(dp/acc)|$,
or $1_{\{0 \le dp\}}|\mu_Z(da/a_m) - \mu_P(dp/acc)| + 1_{\{dp<0\}}|\mu_P(|da|/a_m) - \mu_N(dp/acc)|$.

In every case, it is not the speed but the acceleration which is taken into account, that means for example, with the minimal speed, a bend is not good, it is only if it is still possible to brake. We wish to be sure that :

$$(da/a_m \text{ is Z} \Leftrightarrow (step - ds)/(p_m - ds) \text{ is P}) \text{ and } (|da|/a_m \text{ is P} \Leftrightarrow (step - ds)/(p_m - ds) \text{ is Z})$$

Finally, we define :

$$TR = (50/k)\sum |\mu_Z(da/a_m) - \mu_P((step - ds)/(p_m - ds))| + |\mu_P(|da|/a_m) - \mu_Z((step - ds)/(p_m - ds))|$$

This indicator between 0 and 100 will be the second criterion to minimize.

Importance of the encoding: because some rules may be more general than others (taking into account one or two distances on the three), rather than evaluate the number of rules, we fix the third objective to minimize as the number of meaningful symbols (ANY not being meaningful).

Experiments

The dilemma we talked in the introduction was successively determined averaging all criterias to minimize on 5, 10, 20 finally 30 runs. Those runs are accomplished with randomly located obstacles (a number 20, 40 and next 80). The number of steps was 100, 200, 300 and at last 500. It appears indeed, that few experiments for the same solution do not give a reliable evaluation, too few obstacles advantages individuals which, for instance, are able to turn in a without obstacles part of the playground. On the other hand a minimal step must be set, if not we can see individuals which are able to optimize their life in turning back or in unmoving stages.

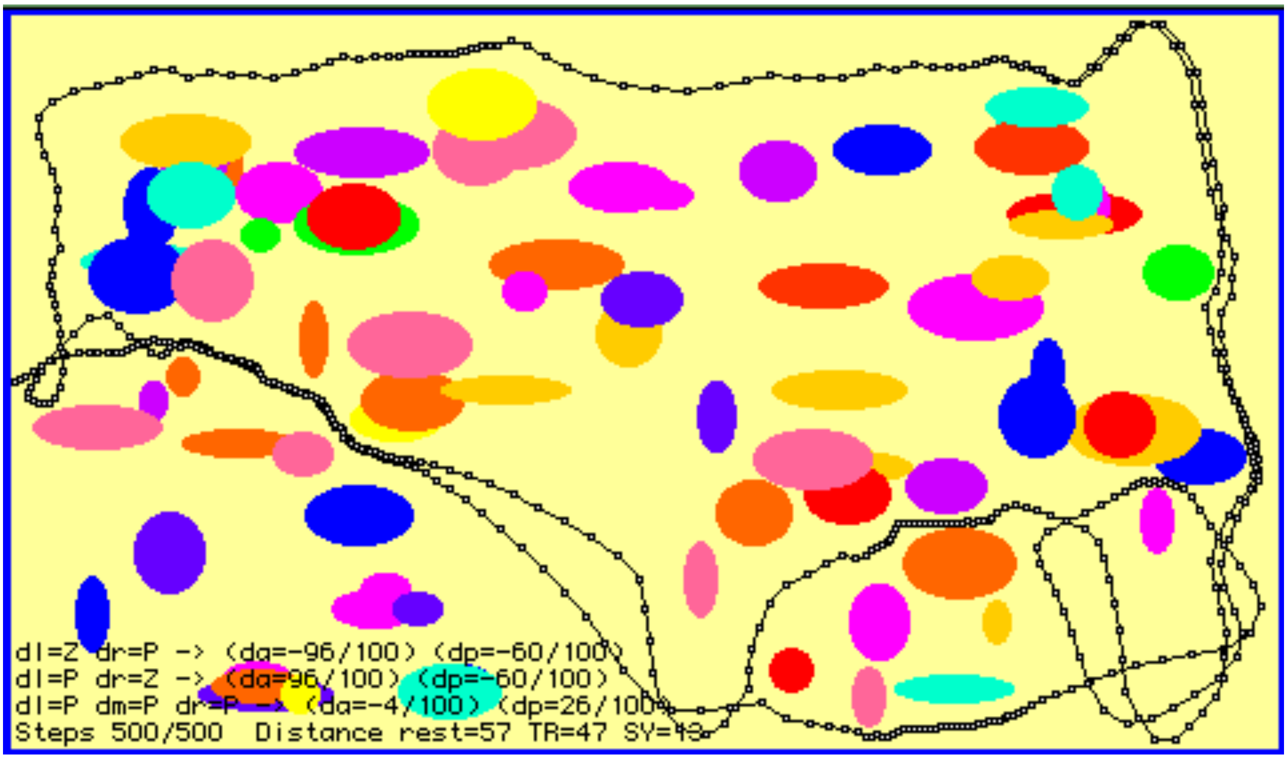

Figure 3 One of the found solutions doing the fitness (57, 47, 13) with a rule to accelerate (P P P → da = -0,04 dp = 0,26) and two symmetric rules (_ Z P → da = -0,96 dp = -0,6) (P Z _ → da = 0,96 dp = -0,6) to turn and go slower. (Departure in the middle of the left side, the last step being in the right high corner.)

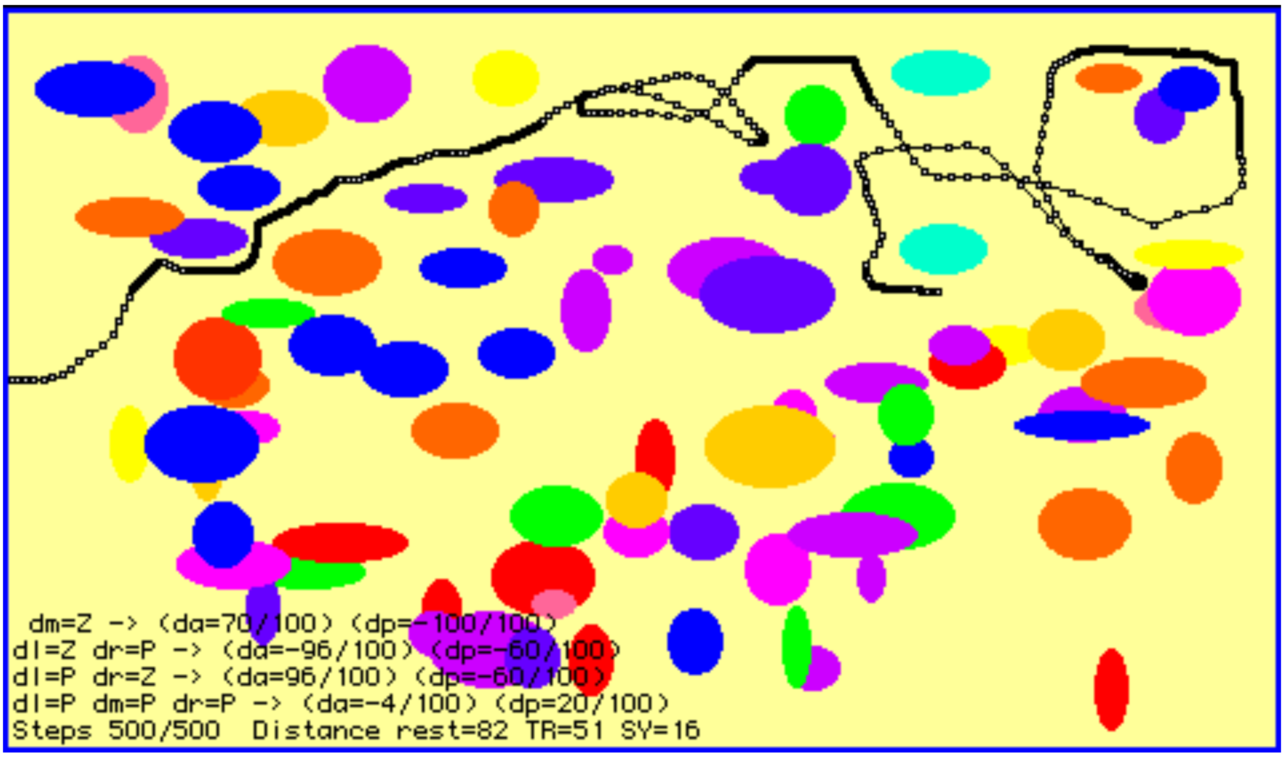

Figure 4 An other very close solution realizing a fitness (82, 51, 16) with a smaller minimal step, showing a rule to brake and turn in case of frontal obstacle (_ Z _ → da = 0,7 dp = -1), two symmetric rules again to turn and go slower (_ Z P → da = -0,96 dp = -0,6) (P Z _ → da = 0,96 dp = -0,6) and finally a moderate rule to accelerate (P, P, P → da = -0,04 dp = 0,2).

Conclusion

After many experiments, the problems we met, are not really concerning the evolutionary algorithm used and its variants, because this one is very flexible and robust, and moreover it does not require empirical initialization of parameters. More precisely N_{max} may be in the order of 20 or 100 without appreciable modifications. The difficulty is really in an other field : setting up all the parameters linked to the simulation : min and max steps, view angle for the captors, how many of them, maximal reconnaissance distance ... In every concrete problem, the difficulty resides more in its definition itself, particularly in the setting of the fitness function(s) which must encode what we want and we have to optimize, than in evolutionary heuristics used.

References

Fonseca C.M. Fleming P.J. *Multiobjective genetic algorithms made easy : selection, sharing and mating restriction,* G.A. in engineering systems : innovations and applications, IEEE 1995

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

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

Gacôgne L. *Optimisation multicritère de contrôleurs flous par une stratégie d'évolution approchant la zone de Pareto,* Rencontres francophones sur la logique floue et ses applications, Nancy, 1996

Gacôgne L. *Research of Pareto set by genetic algorithm, application to multicriteria optimization of fuzzy controler*, EUFIT (European Congress on Intelligent Techniques and Soft Computing) p.837-845, Aachen, 1997

Goldberg D.E. Richardson J. *Genetic algorithms with sharing for multimodal function optimization*, GA and their applications p.41-49 Hillsdale New Jersey, Lawrence Erlbaum ass., 1987

Holland J.H *Adaptation in natural and artificial system.* Ann Arbor University of Michigan Press 1975

Horn *Finite Markov chain analysis of genetic algorithms with niching,* Proc. 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 multiple objective linear programming decision support system,* Decision Support System, North Holland vol 6 p.243-251, 1990

Kursawe F. *A variant of evolution strategies for vector optimization,* Proc. PPSN I p.193-197 Sringer Verlag 1991

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. *Evolution and optimum seeking.* Sixth generation computer technology series. Wiley, 1995

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

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