Search for optimum in dynamic environment: a efficient agent-based method

Alain Berro

CERISS / IRIT
Université des sciences Sociales – Toulouse I
Place Anatole France
31042 Toulouse Cedex
berro@univ-tlse1.fr

Yves Duthen

CERISS / IRIT
Université des sciences Sociales – Toulouse I
Place Anatole France
31042 Toulouse Cedex
duthen@irit.fr

Abstract

This paper presents an optimization method using independent software agents in dynamic environment. The first part shows the characteristics of the method. Then we analyze the results obtained by using this method on multimodal functions and on multiobjectives problems. During this analysis we compare this method to genetic algorithms based methods.

1 INTRODUCTION

This multiagents based method attempt to bring a new solution to difficult optimization problems.

a) In case of highly multimodal function, methods using genetic algorithms require a large number of individuals in order to find all the optima. Calculating times[1] are significant and quality of the solution is not so acceptable. Indeed, these methods cannot guarantee the number of optima because they ignore the small niches [2,6] that do not have a sufficient capacity of attraction.

b) In the case of multiobjectives problems, using genetic algorithms based methods try to gather the individuals within the Pareto optimal zones [4]. Calculating times are significant and recovery of the entire Pareto optimal set is not highly homogeneous.

Moreover, using the precedent methods cannot give to a decision-maker any indication a) of the quality of a found optimum nor b) of the position[2] of the solution provided within the whole Pareto optimal set. So the risk taken by the decision maker cannot be evaluated.

In case of using genetics algorithms based methods within a dynamic environment[3], the redistribution time of the population is very significant. Therefore if the environment changes more quickly than the system, the genes solving the new problem do not have time to diffuse to the majority of the individuals. The system becomes chaotic.

We have tried to develop a method that, at the same time, provides accurate solutions and reacts quickly to changes in the state of the problem.

Besides, this method uses the fundamental exploration/exploitation [3] principle of the genetic algorithms.

2 THE RESEARCH METHOD

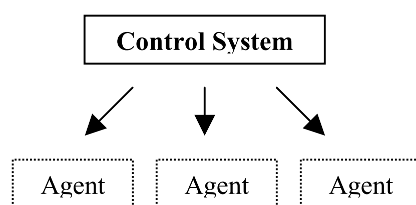

Figure 1: System schematic

The principle of the method is based on random creation of software agents that will try to colonize an optimum of the function.

The system is made of two parts:

- The control system
- The agents

The control system periodically creates agents that are positioned randomly within the space search. So each point within the research space has the same probability of being reached. This way the method has a good exploring aptitude of the research space.

The good exploitation capacity of the system is due to the characteristics of the agents. Searching and "fighting" for an optimum will generate the dispersion of the agents in the search space.

[1] Because of the use of sharing or crowding methods.
[2] Assuming that Pareto optimal set is a circle, we can assert that the center is a solution less risky than the one a point of the perimeter represents.
[3] Objective(s) function(s) or search space are modified.

2.1 CARACTERISTICS OF AN AGENT

Each agent inside the system has the following capacities:

- An agent seeks to colonize an optimum. It has a position within the search space. It does not know the function it is solving.
- An agent is autonomous i.e. it is not directed by the user or the control system. It has an evolution function that enables him to modify its position within the search space according to the goal to reach.
- An agent can reproduce. In our system reproduction means duplication. Under certain conditions an agent will create a double. Thus this last will colonize the neighbouring optima.
- After having colonized an optimum, each agent tries to protect it by calculating a zone of influence around it. Any agent trying to enter this zone will be eliminated. This zone of influence represents a region where the agent does not detect any monotony change of the function(s) of the environment.
- An agent constantly supervises its zone of influence. In case of a change causing the disappearance of the optimum, the agent immediately starts a new search.
- An agent does not communicate but it perceives the presence of the other agents.

2.2 PARAMETERS OF THE SYSTEM

This system is easily parameterized by the user. Once the functions to optimize and the dimensions of the search space both have been defined, the user must specify two parameters which will influence the searching speed and the calculating precision of the optimal points.

- *FORCE* determines the number of points that the agent will test at each stage of its evolution. The higher *FORCE* will be, the slower is the calculation, but the search the agent is doing will be more effective.
- *EPSILON* determines the accuracy the user wants. Each agent stops its search according to a difference between two successive calculating stages. When this difference becomes lower than *EPSILON* then the agent considers that the accuracy is sufficient and it stops its search for the optimum. The agent is then marked as valid.

2.3 CARACTERISTICS OF THE METHOD

2.3.1 Diffusion within the search space

When an agent has found an optimum it acquires the reproducing capacity. The newborn agents will explore the surrounding area searching for other optima. This phenomenon repetition will make it possible to explore all the search space.

2.3.2 Exploration and non continued space

To accelerate the investigation of the search space, the system periodically creates agents. Thus all the points within the search space have the same probability of being reached. Another advantage of this random creation is to allow search in no related spaces. As opposed to the hill-climbing-based methods, this method doesn't require the conditions derivability or continuity of function. Positioning an agent in a region of the search space is enough to ensure that it will be completely explored by diffusion.

2.3.3 System dynamics

The environment of an agent can change at any time. The agents affected by this change will be very quickly reallocated within the search space in order to solve the new problem. As each agent supervises its zone of influence, it will perceive the change of environment and, starting from its position, it will try to find an optimum to be colonized.

This characteristic classifies this method in the category (React on change) in the meaning of J. Branke. [1].

2.3.4 Optima comparison

In case of multiobjectives function, this method allows to compare the Pareto optimal solutions according to the size of the zone of influence around the agent that have found this solution.

On the other hand, while using genetic algorithm based method, the user must fix a number of individuals in the population. That implies certain knowledge of the problem. Some AGs based current systems control the number of individuals but to the detriment of calculating time [4].

2.3.5 Auto regulation of agents quantity

Valid agents have the capacity to eliminate the agents that penetrate in their zone of influence. Thus the system controls only the number of agents within the search space. The higher the number of valid agents is, the more the total of the zones under influences increases. So exploring space decreases as well as the total number of agents. That increases the investigating speed of the search space.

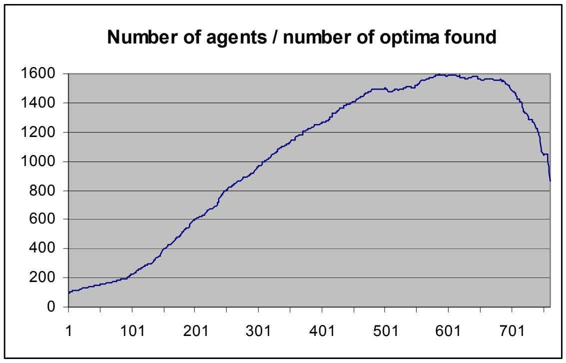

Figure 2

The curve above represents the evolution of the number of agents within the search space of the function (figure 6) :

$$\sum_{i=1}^{5} i * \cos((i+1)*x+i) * \sum_{i=1}^{5} (-i * \sin((i+1)*y+i)) * (i+1) +$$
$$\sum_{i=1}^{5} i * \cos((i+1)*y+i) * \sum_{i=1}^{5} (-i * \sin((i+1)*x+i)) * (i+1)$$

Equation 1

3 TEST FUNCTIONS

We tested this method on two different cases of optimization problems:

- A multimodal function with many optima.
- Multiobjectives function [5,6].

$$\begin{aligned}
&\min(f_1(x,y), f_2(x,y), f_3(x,y)),\ \textit{where}\\
&f_1(x,y) = x^2 + (y-1)^2,\\
&f_2(x,y) = x^2 + (y+1)^2, \qquad \textit{with } -2 \le x,y \le 2\\
&f_3(x,y) = (x-1)^2 + y^2 + 2
\end{aligned}$$

Equation 2 : Viennet 1

$$\begin{aligned}
&\min(f_1(x,y), f_2(x,y), f_3(x,y)),\ \textit{where}\\
&f_1(x,y) = x^2 + (y-1)^2,\\
&f_2(x,y) = \frac{(x+y-3)^2}{36} + \frac{(-x+y+2)^2}{8} - 17, \qquad \textit{with } -4 \le x,y \le 4\\
&f_3(x,y) = \frac{(x+2y-1)^2}{175} + \frac{(2y-x)^2}{17} - 13
\end{aligned}$$

Equation 3 : Viennet 2

3.1 MULTIMODAL FUNCTION

In this case the goal is to determine in a precise way the number *NB* of optima of the function and their position, more or less *EPSILON*, within the search space. An agent that has detected an optimum is marked as valid, so the number of the function optima equals the number of valid agents. Position of each agent gives accurate position of an optimum (figure 3b and 5b). While using genetic algorithms based methods with sharing (figure 3a and 5a) to spread out the individuals on the search space, it is uneasy to determine precisely the positions of the optima and their exact quantity (figure 5a). These AGs based methods are not so acceptable in solving problems involving functions with a large amount of optima because they often ignore the small niches (figure 5a).

An applet located on the http://eva.univ-tlse1.fr/Eva/irit/multi/main.html page shows the capacity of the method to answer the changes imposed by the user.

3.2 MULTIOBJECTIVES FUNCTION

The goal of a multiobjectives problem is to find the whole Pareto optimal set. In case of decision-making, it is necessary to be able to give to the decision-maker an evaluation of the risk.

In the figure 6a; 72% of the population is in the Pareto optimal set. But in case of dynamic problem, the shape of the Pareto set is not a priori known. So, we cannot compare a solution with another. We can only assert that both solutions are Pareto acceptable.

The figure (6b) shows the result obtained using the agents based method. More than 90% of the Pareto optimal surface is found and we can assert that the agent in the middle of the area represents a risk less decision because it has a large zone of influence.

The figure (7) shows the results obtained searching the Pareto optimal set of Viennet2's problem.

4 CONCLUSION

The first tests of this method were satisfactory in particular for the computing speed and the precision of the results (positions of the optima and their number). We now wish to improve this technique by integrating the parameters *FORCE* and *EPSILON* in the agents. Therefore the user will not have adjust the parameters of the system.

References

[1] Branke. J. "Evolutionary approaches to Dynamic Optimization Problems: A Survey" p.134-137, Evolutionary Algorithms for Dynamic Optimization Problems.

[2] Goldberg. D.E., Richardson, J.J., "Genetic algorithms with sharing for multimodale function optimization", Genetic algorithms and their applications : Proceedings of the Second International Conference on Genetic Algorithm, 1987.

[3] Goldberg. . D.E. "Genetic algorithm in search, optimisation and machine learning", Addison-Wesley, Reading, 1989.

[4] Harik. G., Lobo, Fernando., "A parameter-less genetic algorithm", ILLIGAL Report N° 99009, January 1999.

[5] Horn, J., Nafpliotis. N. "Multiobjective optimization using the niched pareto genetic algoithm", ILLIGAL Report N°93005, July 1993.

[6] Van Veldhuizen., D. A., Lamont., Gary., "Multiobjective evolutionary algorithms : analysing the state of the art", Evolutionary Computation 8(2), p125-147, 2000.

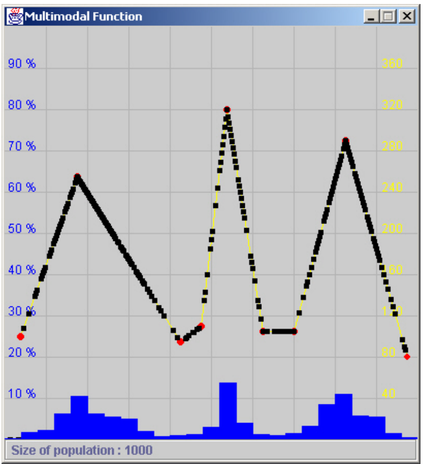

Figure 3a : AG using sharing

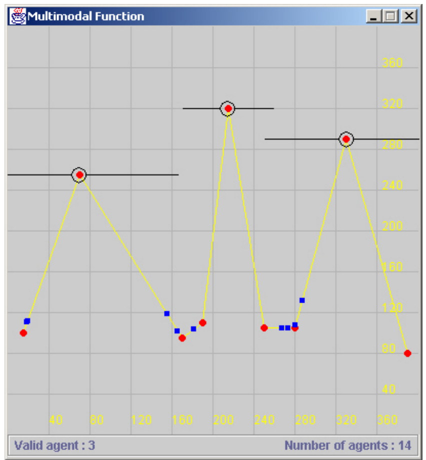

Figure 3b : Multiagents based Method (MA)

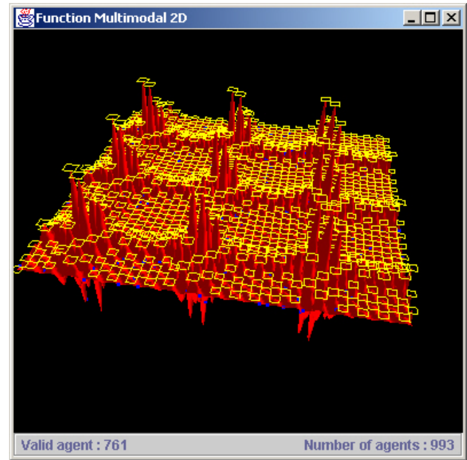

Figure 4 : MA : function with 760 local optima

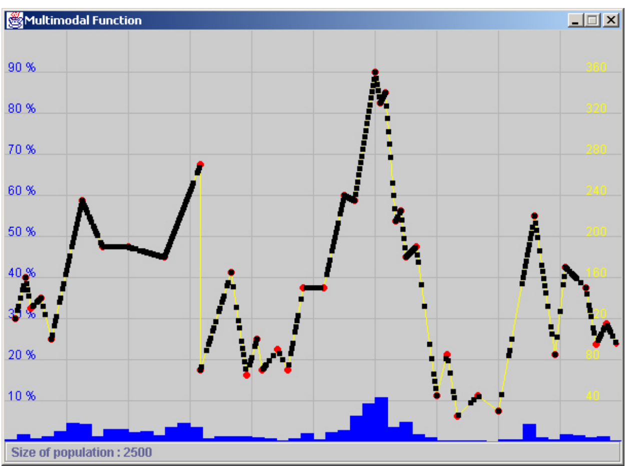

Figure 5a : AG using sharing

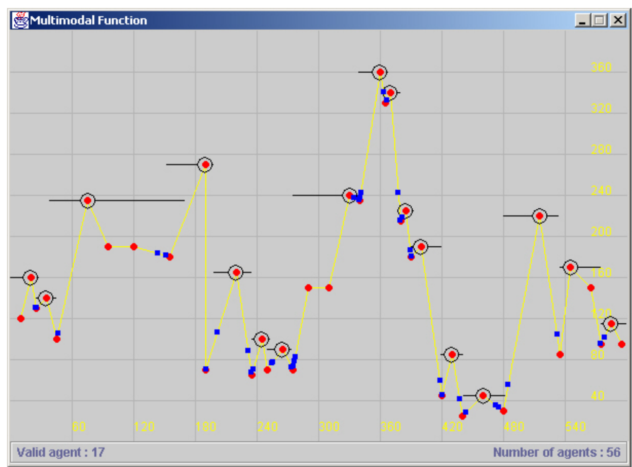

Figure 5b : MA

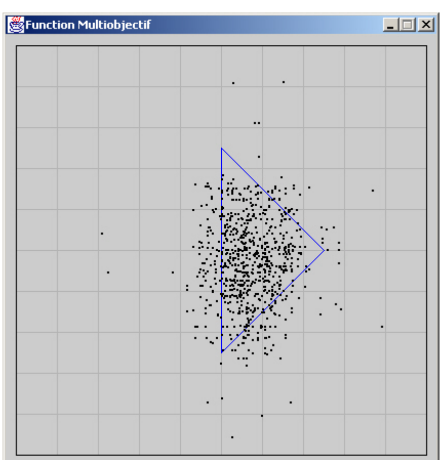

Figure 6a : AG Viennet 1

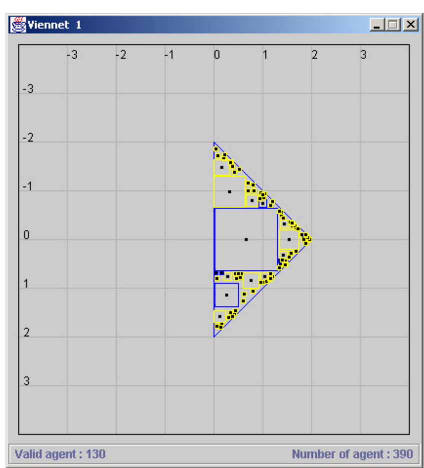

Figure 6b : MA Viennet 1

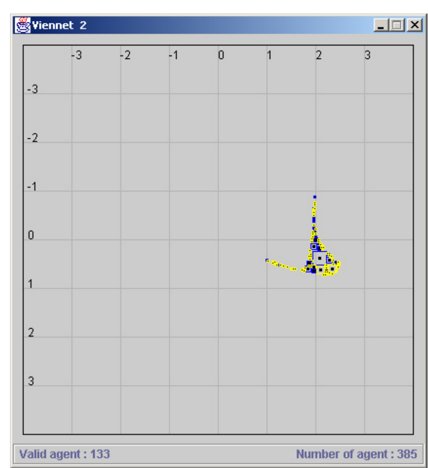

Figure 7 : MA Viennet 2