REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

F. BICKING
C. FONTEIX
J.-P. CORRIOU
I. MARC

Global optimization by artificial life : a new technique using genetic population evolution

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 28, nº 1 (1994), p. 23-36.

<http://www.numdam.org/item?id=RO_1994__28_1_23_0>

Recherche opérationnelle/Operations Research
(vol. 28, n° 1, 1994, p. 23 à 36)

GLOBAL OPTIMIZATION BY ARTIFICIAL LIFE: A NEW TECHNIQUE USING GENETIC POPULATION EVOLUTION (*)

by F. Bicking (1), C. Fonteix (1),
J.-P. Corriou (1) and I. Marc (1) (**)

Communicated by Jean Abadie

Abstract. – *In this paper, a new technique based on genetic algorithms principles is proposed for global optimization problems. This optimization technique uses concepts from population genetics such as population size, birth, death, mutation. Its main characteristic is to perform the search by working on real variables. Following the algorithm description, experiments and results on a set of functions are provided. The ease of implementation of this new method makes it particularly useful as a tool for nonlinear unconstrained and constrained optimization problems.*

Keywords: Global optimization, stochastic method, probabilistic rules, population genetics.

Résumé. – *Dans cet article, une nouvelle technique basée sur les principes des algorithmes génétiques est proposée pour traiter des problèmes d'optimisation globale. Cette technique d'optimisation utilise des concepts tirés de la génétique des populations tels que la taille de la population, la naissance, la mort, la mutation. Sa principale caractéristique est d'effectuer la recherche en opérant sur des variables réelles. Après une description de l'algorithme, des études de recherche d'optimum et leurs résultats sont présentés pour différentes fonctions. La facilité d'implantation de cette nouvelle méthode la rend particulièrement utile pour résoudre des problèmes d'optimisation nonlinéaire avec ou sans contraintes.*

Mots clés : Optimisation globale, méthode stochastique, règles probabilistes, génétique des populations.

1. INTRODUCTION

With the advent of advanced computer architectures and emerging of new theoretical results, there is a growing interest in developing algorithms for

---

(*) Received April 1992.
(**) Author to whom all correspondence should be addressed.
(1) L.S.G.C., C.N.R.S.-E.N.S.I.C.-E.N.S.A.I.A, 1, rue Grandville, B.P. 451, 54001 Nancy Cedex, France.

Recherche opérationnelle/Operations Research, 0399-0559/94/01/$ 4.00

nonlinear optimization problems. A large number of optimization problems can be written in the following form:

$$\mathrm{Min}\{f(x) | x \in X\} \quad \text{where } X \subset R^n ; \; f : R^n \to R$$

$$X = \{x \in R^n \mid l_j \leq x_j \leq u_j , j = 1, 2, ..., n; \\ g_k(x) \leq b_k, k = 1, 2, ..., m\}$$

where l_j and u_j are respectively the lower and upper bounds for x_j. It assumes that n, m, the l_j and u_j are known and $l_j < u_j$ for $j = 1, 2, ..., n$. Note that f is not required to be convex, differentiable, continuous or unimodal. This kind of problems can be solved with differents approaches: deterministic or stochastic, according to the classification adopted by most authors.

Among deterministic approaches, Floudas *et al.* [1, 2] distinguish covering methods, branch and bound methods, cutting plane methods, interval methods, trajectory methods and penalty methods. Among probabilistic methods, they distinguish random search methods [3], clustering methods [4], methods based on statistical models of objective functions [5, 6].

Rinnooy Kan and Timmer [7] prefer to this classical distinction into deterministic and stochastic approaches the following classification based on the philosophy of the respective methods: partition and search, approximation and search, global decrease, improvement of local minima, enumeration of local minima.

It will be shown in the following that genetic algorithms are clearly a stochastic approach which can be seen as a random search method. It is more difficult to incorporate genetic algorithms in Rinnooy Kan and Timmer [7] classification, however they have links with a global decrease method with respect to the best generated point.

Genetic algorithms constitute a class of optimization algorithms. This class of algorithms is distinguished from other optimization techniques by the use of concepts from population genetics to perform the search [8, 9]. A genetic algorithm creates a collection of initial points in the search space and makes it evolve by genetic operators such as selection, recombination, mutation and crossover. The four characteristics of these algorithms are the following:

– parameters are coded in fixed length strings that assume an implicit parallelism;

– the search is performed with a population of points, not with a single one;

– probabilistic rules are used to make the population evolve;

– no information about the function is required, for example continuity, derivability, convexity,...

In this paper, a new technique for global optimization based on genetic concept is presented. The proposed technique uses actual values of the parameters while genetic algorithms use coded parameters. Our technique works also with genetic operators to perform the search. The overall approach is stochastic in nature, which makes theoretical analysis quite difficult, especially with regard to issues about the global convergence. Therefore the behavior is characterized computationally through a series of experiments. We first describe the proposed Global Optimization through Artificial Life technique (GOAL). Then, the experiments on a set of test functions selected from the literature for their difficulty and the results are provided.

2. DESCRIPTION OF THE ALGORITHM

As mentioned earlier, in this approach, we use genetic concepts. Like genetic algorithms, our method considers simultaneously many points from the search space by generating populations of points and tests each point independently. At each iteration, called a generation, a new population is generated and tested. Probabilistic rules are used to guide the search. Our algorithm requires only information concerning the quality of the solution (objective function value). With probabilistic rules, a random choice is efficiently introduced in the exploitation of knowledge to locate near optimal solutions rapidly.

The philosophical principle can lead to a qualitative understanding of the algorithm. The proposed algorithm operates with a set of vectors which represents a population of individuals. The *phenotype* of an individual is given by a real vector x with $x \in X$. Each component of x is called a *gene*.

The algorithm can be described by a five-tuple (N, L, M, G, S):

– N: Population size.

The population size affects both the performance and the efficiency of the algorithm. It generally behaves poorly with very small populations, because the population provides an insufficient sample size for the search. A large population is more likely to contain representative individuals and discourages premature convergence to suboptimal solutions but requires more evaluations per generation, possibly resulting in a slow rate of convergence.

– L: Number of genes.

This parameter depends on the characteristics of the function to optimize $(L = \dim(x),\ x \in X)$.

– M: Mutation rate.

Mutation is a secondary search operator which increases the variability of the population and prevents complete loss of "genetic material" through selection and recombination. Approximately, $M \times N$ mutations occur per generation. The mutation rate value is chosen equal to 0.01, near the value met in natural processes.

– G: Generation patrimony.

The generation patrimony determines the population ratio which remains unchanged at each generation.

– S: strategy.

The population evolves generation after generation under the application of evolutionary rules called genetic operators. The used strategy influences the selection process. Different strategies might have been used; in particular, in genetic algorithms, the pure selection strategy is often encountered where each individual in the current population is reproduced in proportion to the individuals performance. In the present case, only the elitist strategy is retained: at each generation, $N \times G$ individuals are unchanged and $N \times (1–G)$ are eliminated; these latter will be replaced by new individuals.

The search is done by N active individuals, each of them being described by its vector of genes or phenotype

$$x^i = \left(x_1^i, x_2^i, \ldots, x_L^i\right) \qquad \left(x^i \in X\right) \quad \text{and} \quad f\left(x^i\right),$$

its objective function. First, the algorithm creates at random an initial population of N individuals satisfying all eventual constraints. In the *evaluation process*, the objective function is computed for each individual and the extrema values H_{Min} and H_{Max} are stored ($H_{\text{Min}} = \text{Min}\ \{f(x^i)\}$ and $H_{\text{Max}} = \text{Max}\ \{f(x^i)\}$ for $i = 1, 2, \ldots, N$).

The population is next sorted in increasing order by $f(x^i)$ if a minimization is operated (respectively decreasing order in case of maximization) and a *selection* is made. Only a population ratio is selected using the generation patrimony. The $N \times G$ first individuals will survive and participate to the creation of a new generation. So G is chosen to select individuals as severely as possible without destroying the diversity of the population too much. The elitist strategy guarantees that the best individual of a population P_t survives in P_{t+1}.

In the *reproductive stage*, two individuals among the $N \times G$ selected are chosen at random to create a new individual (a child), x^k $(N \times G < k \leq N)$ by a random combination of the parents genes. This new individual is tested with respect to all constraints with must be satisfied; if not, it is rejected and another child is created. This child can mutate with the mutation rate M. The mutation operator selects at random one gene j that undergoes a random change $(l_j \leq x_j^k \leq u_j)$. Next, the objective function value of this child is computed. Another selection is made: the child that satisfies $f(x^k) < H_{\text{Max}}$ is accepted, otherwise another birth takes place. The reproductive stage runs until $k=N$, so the population is entirely reconstructed and the search scheme is iterated.

In order to verify the convergence of the algorithm, the following fitness index was defined:

$$\left| \frac{H_{\text{Min}} - \bar{H}_{N \times G}}{H_{\text{Max}} - H_{\text{Min}}} + 1 \right| = \text{fit}(\text{P}'_{\text{t}})$$

where

$$\bar{H}_{N \times G} = \frac{1}{N \times G} \sum_{i=1}^{N \times G} f(x^i)$$

If the best $N \times G$ individuals (subpopulation P'_t) converge to the found minimum (H_{Min}), fit(P'_t) will tend to 1. It ensures the convergence to a minimum without knowing whether it is a local or a global one.

The conditions to stop the search are

$$|H_{\text{Max}} - H_{\text{Min}}| \leq \varepsilon \qquad \text{and} \qquad \text{fit}(P'_t) \approx 1.$$

The first condition assumes that the entire population has converged to a solution and the second one that this point is the best minimum found during the run. This means that ϵ is an important factor for the search. If ϵ is very small, the search goes on and a better minimum (hopefully the global one) can be located due to the mutation process, otherwise the found solution is likely to be a local minimum.

Due to the elitist strategy, the next generation will have a minimal function value (H_{Min}) lower or equal to the previous one. So, generation after generation, the population will converge to a solution which will be the best found during the search.

The algorithm is summarized by the block diagram (*fig.* 1).

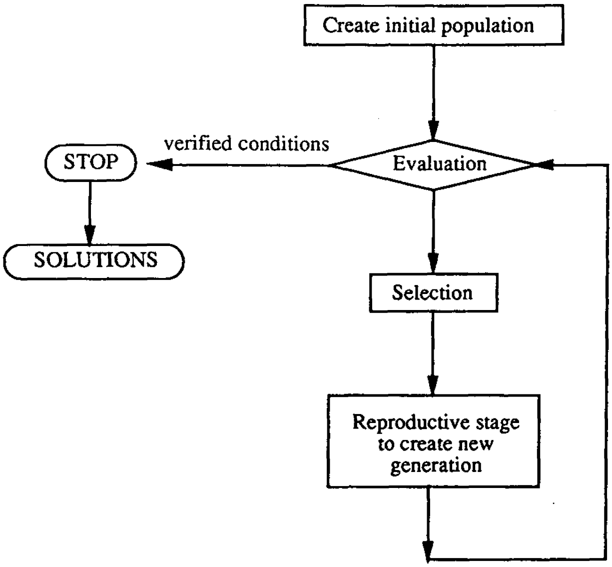

Figure 1. – Block diagram of the algorithm.

3. EXPERIMENTS AND RESULTS

Several functions taken from the literature have been tested with the proposed optimization technique. The complexity of a function optimization problem depends on the number of local minima and their distribution and the search domain defined by the constraints.

The first and simple example is presented to show the algorithm behavior in the search space. The second one is the Rosenbrock's 4-dimensional function, the third one is subjected to nonlinear constraints, the fourth one is used extensively in the Genetic Algorithms community. The two last are used in "mainstream" optimization and are multimodal.

For all runs, the mutation rate M is fixed equal to 0.01 and the used strategy is elitist, ϵ is chosen equal to 10^{-8}.

Example 1: The first experiment concerns a simple unimodal function with explicit constraints but presents a real interest to visualize the algorithm convergence.

$$\mathrm{Min}\left\{ f(x,y) = 1 + (5 - x - y)^2 + \frac{(1 - x + y)^2}{16} \right\}, \qquad 0 \le x, y \le 7$$

The global minimum is located at ($x=3$, $y=2$) with an objective function value of 1. Figure 2 shows the algorithm behavior and the search domain is also presented with the contours of $f(x,y)$.

The initial population is scattered (*fig.* 2*a*). The successive populations are presented generation after generation (*i. e.* figure 2*b* corresponds to

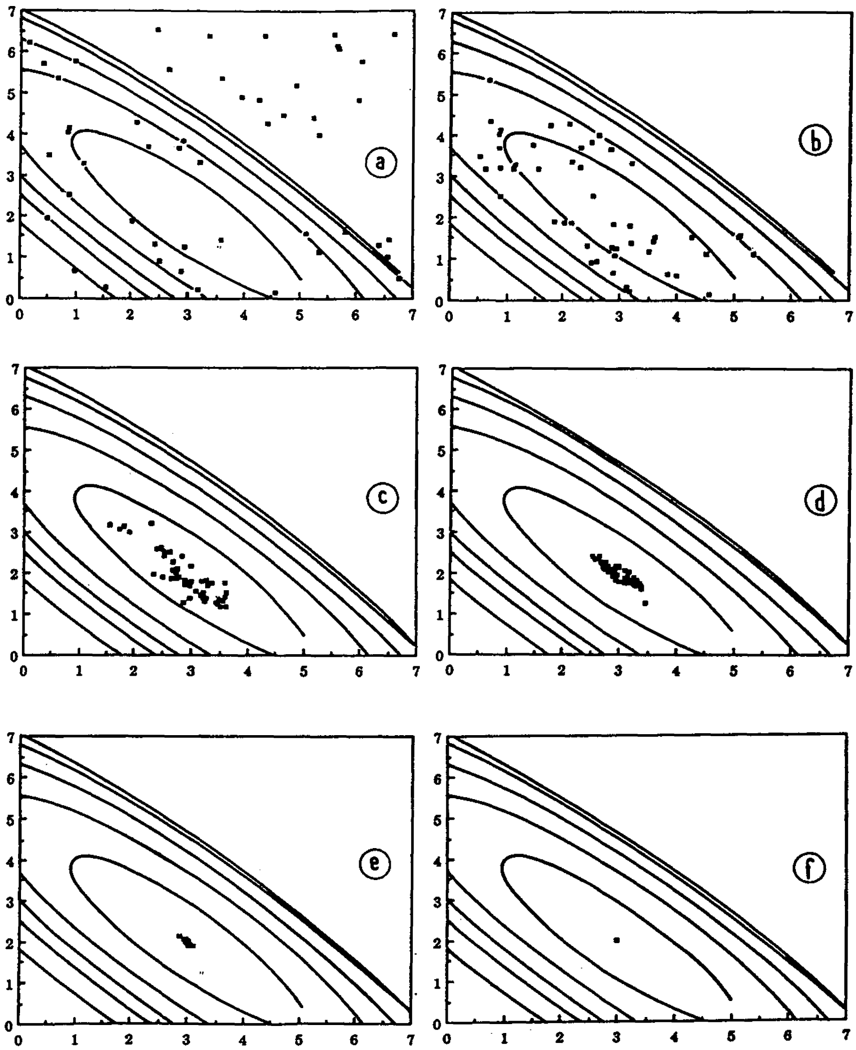

Figure 2. – Behavior of the algorithm for example 1 (successive populations for generations 1 to 6).

generation 2, figure 2*c* to generation 3 and so on). The points are progressively concentrated to a particular region of the search space where the minimum is located (figure 2*f* corresponding to generation 6). The next generations bring a further accuracy which is not distinguishable on a drawing.

The global minimum was located in 12 generations with $N=500$ and $G=0.2$, L being fixed equal to 2.

Example 2: This experiment is to minimize the Rosenbrock's 4-dimensional function:

$$\begin{aligned}\text{Min } \{f(x_1, x_2, x_3, x_4) \\ &= 100\,(x_2^2 - x_1)^2 + (1 - x_1)^2 + 90\,(x_4^2 - x_3)^2 + (1 - x_3)^2 \\ &\quad + 10.1\,[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8\,(x_2 - 1)(x_4 - 1)\}\end{aligned}$$

The global minimum $f=0$ is located at (1, 1, 1, 1). The algorithm was tried with two populations sizes $N=100$ and $N=500$, $L=4$, and a generation patrimony equal to 0.2. The variations of N and G influence only the number of generations needed to reach the solution but not the quality of the solution, the global optimum being located at each run. For $G=0.2$ and $N=100$, 318 generations were neded instead of only 25 for $N=500$.

Example 3: This experiment is to minimize the following function subject to two nonlinear constraints [2].

$$\text{Min}\{f\,(y_1, y_2) = -y_1 - y_2\}$$

with the search domain: $0 \le y_1 \le 3$ and $0 \le y_2 \le 4$

subject to constraints:

$$y_2 \le 2 + 2\,y_1^4 - 8\,y_1^3 + 8\,y_1^2$$
$$y_2 \le 4\,y_1^4 - 32\,y_1^3 + 88\,y_1^2 - 96\,y_1 + 36$$

The feasible region consists of two disconnected subregions (*fig.* 3*a*). The global solution $f=-5.50796$ occurs at $y_1=2.3295$ (point C on figure 3*b*).

The proposed algorithm located the global minimum at $y_1=2.3295$ with an objective value of -5.50797. The optimum was detected in 24 generations with $N=500$, $G=0.2$, L being fixed equal to 2 and 50 generations for $N=100$, $G=0.2$

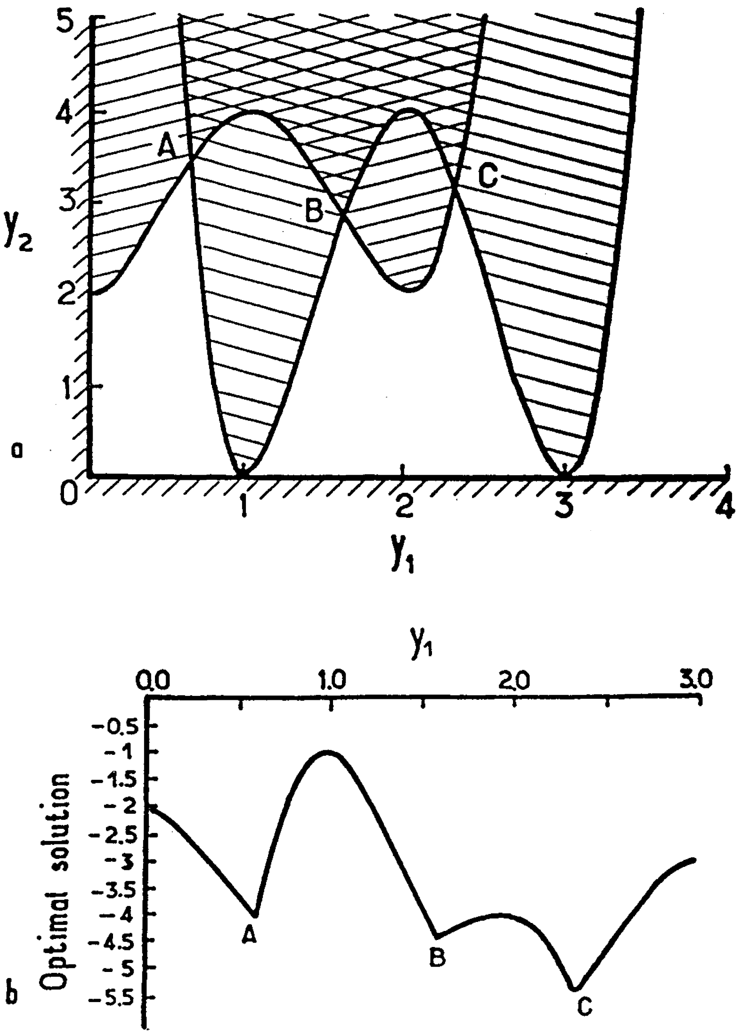

Figure 3. – (a) Feasible region for example 3; (b) Optimal solution of example 3.

Example 4: This function was suggested by De Jong [10]

$$\operatorname{Min}\{f(x, y) = 100\,(x^2 - y^2)^2 + (1 - x)^2\}; \quad -2.048 \le x, y \le 2.048$$

This function (*fig.* 4) is nonconvex and has its global minimum $f=0$ at (1,1). For all runs with different population size, the proposed algorithm has located the global optimum as shown in table.

TABLE

Influence of the population size N and the generation patrimony G on the number of generations.

N	G	Generations
500	0.2	14
500	0.5	200
50	0.2	22
50	0.5	391

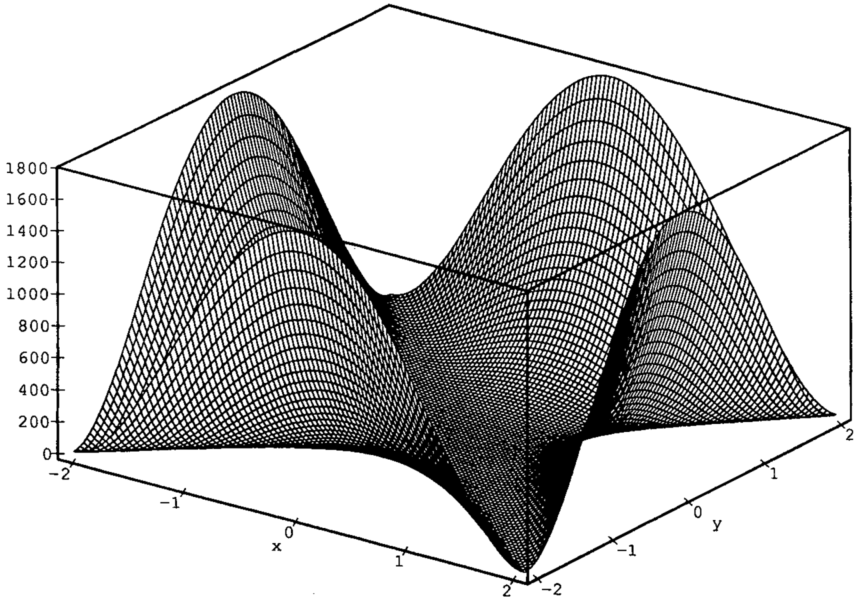

Figure 4. – Function of example 4.

Example 5: This function was suggested by Griewangk [11]

$$\text{Min}\left\{ f(\mathbf{x}) = \sum_{i=1}^{10} x_i^2/4\,000 - \prod_{i=1}^{10} \cos\left(x_i/\sqrt{i}\right) + 1 \right\}; \quad -600 \le x_i \le 600$$

The global minimum $f=0$ is located at $x_i=0$, $i=1, 2, \dots, n$. The local minima are located approximately at

$$x_k = \pm k\pi\sqrt{i}, \qquad i = 1, 2, \dots, n, \quad k = 0, 1, 2, 3, \dots$$

In ten dimensions, there are four suboptimal minima $f(x) \approx 0.0074$ at $x \approx \left(\pm\pi, \ \pm\pi\sqrt{2}, 0, \dots, 0\right)$. For all runs, the proposed algorithm has located the global optimum. In ten dimensions, with $N=500$, the number of necessary generations was 35, respectively 65, resp. 124 with G equal to 0.2, resp. 0.5, resp. 0.7.

Example 6: This function was proposed by Schwefel [12].

$$\text{Min}\left\{ f(\mathbf{x}) = \sum_{i=1}^{10} -x_i \sin\left(\sqrt{|x_i|}\right) \right\}; \quad -500 \leq x_i \leq 500$$

The global minimum is at $x_i=420.9687$, $i=1, 2, ..., n$ (*fig.* 5). The local minima are located at the points $x_k = (\pi\,(0.5+k))^2$, $k = 0, 2, 4, 6$ for the positive directions of the coordinate axis and $x_k = -(\pi\,(0.5+k))^2$, $k = 1, 3, 5$. Points with $x_i=420.9687$, $i=1,2,..., n$, $i \neq j$, $x_j=-302.5232$, give the second best minimum–far away from the global minimum. For all runs with $N=500$ and $G=0.2$, the proposed algorithm has located the global optimum in 52 generations. For $N=500$ and $G=0.5$, 196 generations were necessary to reach the solution. To emphasize the efficiency of the algorithm when faced to a multimodal function, two additional cases on the present two-dimensional function have been studied in detail; in these particular studies, the initial population has been randomly generated in a restricted area far from the optimum.

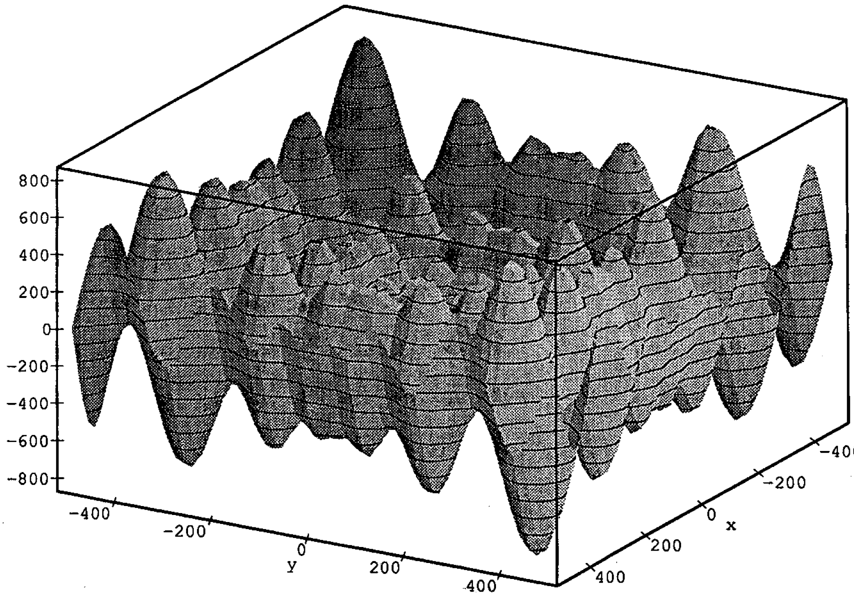

Figure 5. – Two-dimensional version of function of example 6.

Case a: the generating domain for the initial population was $-500 \leq x_i \leq -400$ to be compared to the search domain: $-500 \leq x_i \leq 500$. At each generation, the centroid for the population has been represented on figure 6 (contours of the function). The numbers on figure 6 indicate the generation in question. It can be seen that during the evolution towards the optimum which is reached after 42 generations, the centroid undergoes phases of "acceleration" and "deceleration": sometimes, it seems to be trapped in a local minimum before emerging towards a new position. The population size was $N=500$ and $G=0.2$.

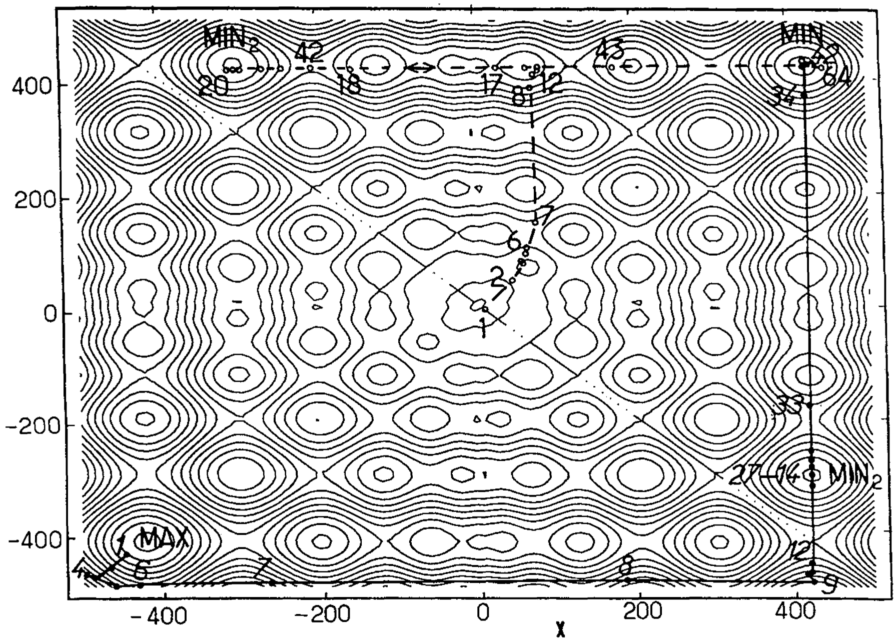

Figure 6. – Contours for function of example 6 and different ways to reach the optimum.

Case b: the generating domain for the initial population was $-100 \leq x_i \leq 100$. Near the generation 10, the centroid evoluates slowly around the point (80, 420) (*fig.* 6), then near the generation 20 around the point (–302, 420) which is a second-order local minimum (Min_2). It extirpates itself only at the generation 40 where it evoluates quickly towards the searched minimum reached after 64 generations. The population size was only $N=50$, with $G=0.2$.

4. CONCLUSION

In this paper, a new and original method for function optimization problems is proposed. It uses concepts of genetics to perform the search. Its main originality is the use of actual values of parameters instead of coded ones as usually used in genetic algorithms. The main operating parameters to be defined for the algorithm in a given optimum search are the population size, the number of genes, the mutation rate, the generation patrimony. The algorithm strategy is elitist, *i. e.* only the best points are retained. The method has been tested on several cases of nonlinear unconstrained and constrained optimization problems. A fitness index has been defined to serve as an indicator for convergence. Even when the test function presented pronounced extrema and when the initial population was confined near a local extremum, at the end, the global optimum was always located. Though the global convergence cannot be proved, the experience gained through many complex optimization problems makes us think that the algorithm is both robust and powerful.

Further work may be done along the following main research directions: first development of additional properties that can increase the computational efficiency and secondly application of this method in various domains as an optimization tool.

REFERENCES

1. C. A. Floudas and P. M. Pardalos, *Recent Advances in Global Optimization,* Princeton University Press, 1992.
2. C. A. Floudas and V. Visweswaran, A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs. I. Theory. II. Application of Theory and Test Problems, *Comp. Chem. Engng,* 1990, *14,* (12), pp. 1397-1434.
3. S. Kirkpatrick, C. D. Gelalt and M. P. Vecchi, Optimization by Simulated Annealing, *Science,* 1983, *220,* 4598, pp. 671-680.
4. A. H. G. Rinnoy Kan and G. T. Timmer, Stochastic Global Optimization Methods. Part I: Clustering Methods, *Math. Program.,* 1987, *39,* p. 27.
5. A. Zilinskas, Global Optimization. Axiomatics of Statistical Models, Algorithms and Their Applications, *Moklas.,* Vilnius, 1986.
6. A. A. Torn and A. Zilinskas, Global Optimization, *Lecture Notes in Comput. Sci.,* *350,* Springer-Verlag, Berlin, 1989.
7. A. H. G. Rinnooy Kan and G. T. Timmer, Global Optimization, Handsbooks in OR & MS, *1,* G. L. Nemhauser *et al.* Eds., Elsevier, Amsterdam, 1989.
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, Mass., 1987.
9. J. H. Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Presss, Ann Arbor, 1975.

10. K. A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, *Ph. D. thesis*, University of Michigan, Dissertation Abstracts International 36 (10) 5140B, University Microfilms No. 76-9381, 1975.
11. A. O. Griewangk, Generalized Descent for Global Optimization, *J.O.T.A.*, 1981, *34*, pp. 11-39.
12. H. P. Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, *26*, *Interdisciplinary System Research*, Birkhäuser, Basel, 1977.