BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXI No. 3/2009	211 - 216	Seria Tehnică

The Possibility of Hybrid Genetic Algorithms and Standard Methods Optimization

Marius Olteanu, Nicolae Paraschiv

Petroleum – Gas University of Ploiești, 39 Bucureşti Blvd., Ploiești, ROMÂNIA
e-mail: molteanu@upg-ploiesti.ro, nparaschiv@upg-ploiesti.ro

Abstract

This article presents the possibility of designing a hybrid optimization method using the Artificial Intelligence technique of Genetic Algorithms and a standard derivative-free optimization method. By taking into account the time restriction, usually found in real-time application, a rough approximation of the solution is computed, solution that can be further improved by applying the standard method. The results of applying this hybrid method to a well known test function for optimization methods are presented in detail.

Key words: *genetic algorithms, optimization, control.*

Introduction

A genetic algorithm optimization approach is different from that of the traditional methods because it is using a population of points, not a single point, also probabilistic transition rules, not deterministic ones and they do not require derivative information or other auxiliary knowledge. As a consequence, we can state that the Genetic Algorithms are applicable to a wide class of optimization problems. Because of this stochastic approach, genetic algorithms do not guarantee the finding of the optimum, especially when we deal with highly non-linear, constrained and multimodal functions in contrast with traditional optimization techniques applied in special contexts (e.g. the starting point for the algorithm is relative close to the optimum) [1].

The branch of derivative-free optimization methods has undergone continuous research motivated by the increasing complexity in mathematical modeling and higher sophistication and performance of scientific computing. The Nelder-Mead algorithm is such an example in which the solution found is the result of a repeated evaluation of the objective function on a finite number of points, procedure that does not require any explicit or implicit derivative approximation or model building.

Using the advantage of a very wide area of searching for the optimum solution provided by the genetic algorithm and the robustness of the Nelder-Mead algorithm, it is possible to design a hybrid method that can benefit from both approaches.

Genetic Algorithms in optimization. Schwefel function example

The beginnings of genetic algorithms go back to 1960's, when John Holland from the University of Michigan, together with colleagues and students developed its basic ideas that emerged from their research on the phenomenon of adaptation as it occurs in nature and how the mechanisms of natural adaptation might be imported into computer systems, ideas presented in his 1975 book, *Adaptation in Natural and Artificial Systems* [2].

If we formulate the optimization problem as follows:

$$\begin{aligned} &\textit{minimize} && f(x) \\ &\text{subject to} && x \in \Omega, \end{aligned}$$

the algorithms starts with an initial set of points in Ω, denoted P(0), which represents the initial population and after evaluating the initial population and applying a set of genetic operators (crossover, mutation), a new set of points, P(1) is created. The procedure described is repeated, generating populations P(2), P(3), ... until a stopping criterion is reached. The algorithm can be summarized as follows:

1. set k = 0; form initial population P(0)
2. evaluate P(k)
3. if stopping criterion satisfied → stop
4. select M(k) from P(k)
5. evolve M(k) to form P(k+1)
6. set k = k+1, go to 2

where M(k) is the mating pool that results after the selection process [3].

For testing the performance of genetic algorithms, multivariable, nonlinear, multimodal test function are frequently used. Examples of such functions are Schwefel, Rosenbrock, Rastrigin, Ackley, and Powell. In the following, the case of Schwefel's function for two variables is used to demonstrate the efficiency of the genetic algorithms, but in the same time, the weaknesses that result from the fact that it is a stochastic method:

$$f(x) = \sum_i -x(i) \cdot \sin\left(\sqrt{|x(i)|}\right), \tag{1}$$

where $x(i) \in [-500, 500]$.

The global minimum for this function is $f(x)$ = -837.9658, value obtained for $x(i)$ = 420.9687, $i \in (1,n)$. In Figure 1 it is presented the function's plot in the case i=2 (two independent variables).

Using the Matlab® toolbox *gatool* having the following main parameters: population size 200, initialized with a uniform distribution function in the [-500, 500] domain, *real codification* of the solution, *Stochastic Uniform* selection, *Scattered* function for the *crossover* operator, *mutation* operator having a uniform probability distribution with a 0.2 value and the stopping criterion being the time limit, an approximate solution to the problem has been obtained [6].

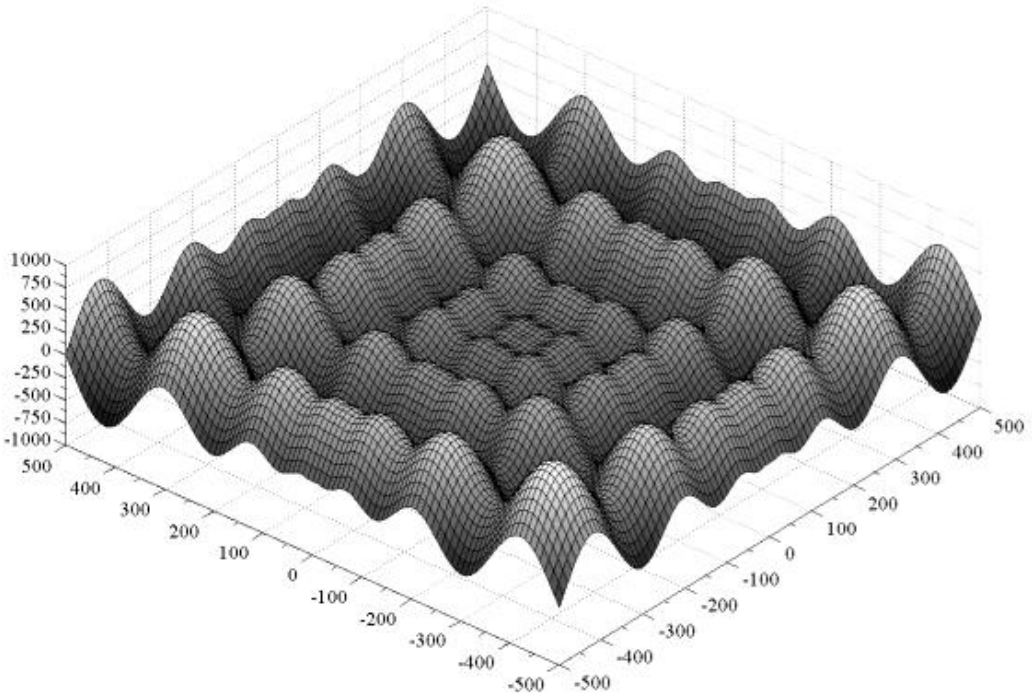

Fig. 1. Plot of the Schwefel function for two independent variables.

In the case of real-time control, time is a critical resource which set important constraints to the algorithms used in computing the command values. In this case a time limit of one second T_{run} = 1s is imposed. Such a restriction causes the algorithm to perform in a hardware dependent manner. The results plotted in Figure 2 reveal the progress of the genetic algorithm to the global optimum. Having insufficient time to improve the solution found, the algorithm provides a rough approximation of the solution, in this case $f(x)$ = -815.1946, for x_1= 431.40065, x_2 = 412.46638.

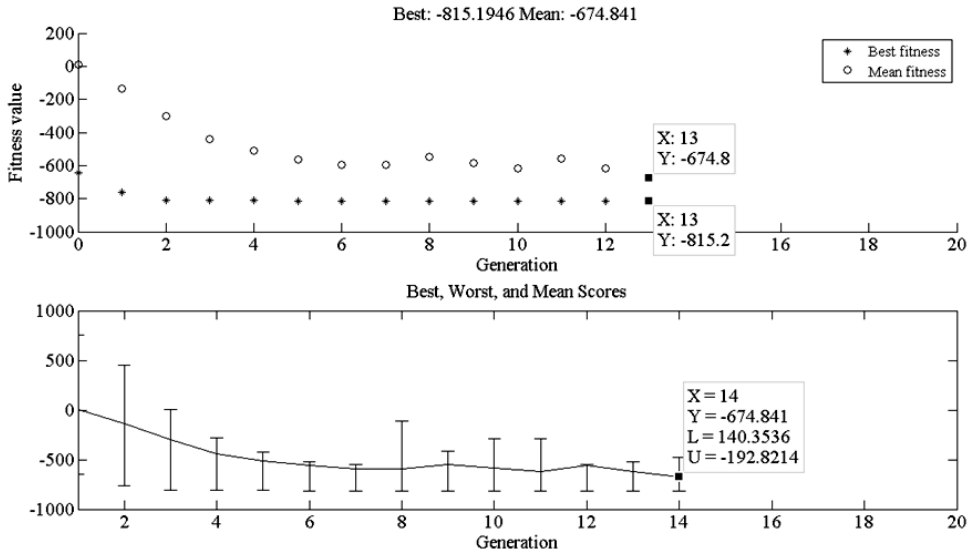

Fig. 2. Results for the genetic algorithm run with a population size of 200.

Nelder-Mead free derivative optimization method

Derivative-free optimization methods are frequently used in practice due to the unavailability or high cost of obtaining the derivative. Examples of such domains are tuning parameters of nonlinear optimization methods, engineering and circuit design, molecular geometry etc. [4].

Such a method is represented by the Nelder-Mead algorithm also called *downhill simplex method*. A simplex is defined as the elementary geometrical figure that can be formed in a *N* dimension space, having *N*+1 sides. Every iteration in $\Re^n$ is based on a simplex of *n*+1 vertices $Y=\{y^0, y^1, y^2, \ldots y^n\}$ ordered by increasing value of *f*. Only one point P_0 of the simplex is specified by the user, the other *N* points are found by

$$P_n = P_0 + c_s e_n \,, \tag{2}$$

where e_n are *N* unit vectors and c_s is a scaling constant [5].

The goal of the method is to move the simplex until it surrounds the minimum and then to contract the simplex around the minimum until it is within an acceptable error. To attain this goal, in the case of a two-dimensional space a series of steps are performed [4]. The most common Nelder-Mead iterations perform a reflection, an expansion or a contraction in which the algorithm replaces the worst vertex y^n by a point in the line that connects y^n and y^c:

$$y = y^c + \delta(y^c - y^n),\ \delta \in \mathbf{R}, \tag{3}$$

where $y^c = \sum_{i=0}^{n-1} y^i/n$ is the centroid of the best *n* vertices, also the value of δ indicates the type of iteration (see Figure 3):

- $\delta = 1$ isometric reflection;
- $\delta = 1$ expansion;
- $\delta = 1/2$ outside contraction;
- $\delta = -1/2$ inside contraction.

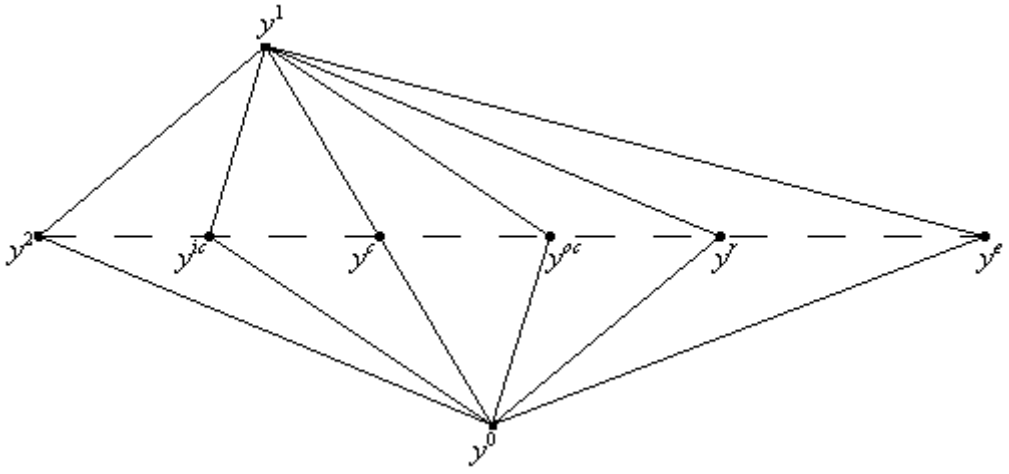

Fig. 3. Reflection, expansion, outside contraction and inside contraction of a simplex.

Hybrid optimization using Nelder-Mead algorithm and Genetic Algorithms

By combining the rough solution obtained using Genetic Algorithms with the Nelder-Mead method, a very good result is obtained for the Schwefel function. The approximate solution x_1 = 431.40065, x_2 = 412.46638 for which f(*x*) = -810.0454 is used as the starting point P_0 for

the Nelder-Mead algorithm. In the Matlab® programming environment *fminsearch* function have been used for calculating the minimum of the specified function for which a starting point is provided and also other parameters as the level of display, maximum number of function evaluations or iterations allowed and also the termination tolerance having a value of 10^{-5}. After a number of 48 iterations, the exact solution is successfully found. A two-dimensional representation of the steps performed by the algorithm is presented in Figure 4. The running time for the algorithm is negligible.

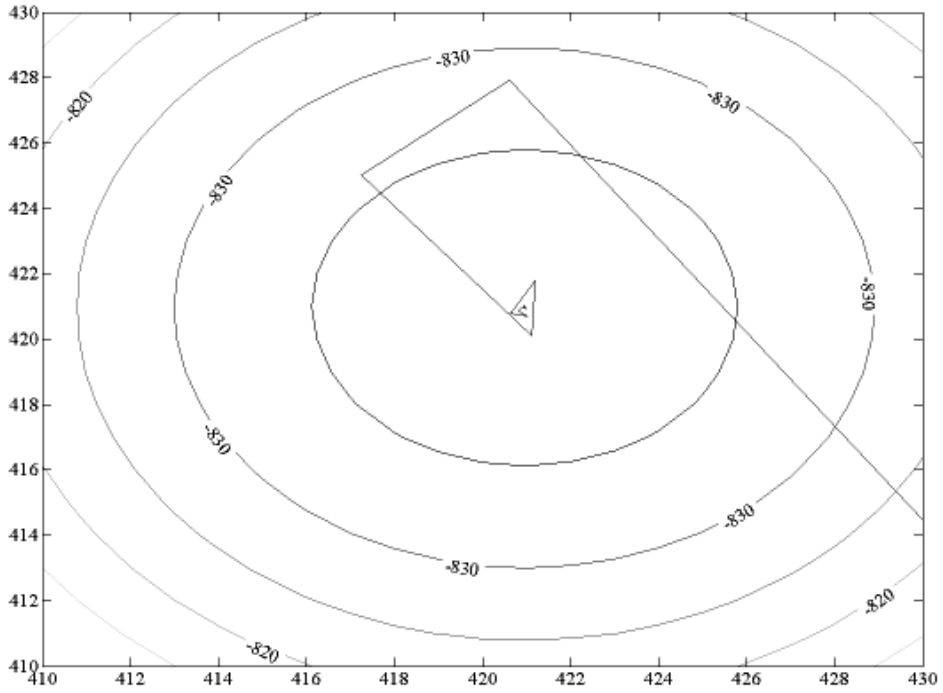

Fig. 4. Two-dimensional representation of the algorithm steps.

Conclusions

Although the genetic algorithms constitute a stochastic optimization method, by tuning the parameters of the algorithm is possible to obtain an approximation or a good solution to many difficult and practical problems even in the case of other restriction, like the running time. By limiting the time of computing for the genetic algorithm, the number of generations usually is not satisfiable and the solution will be an approximation. Using the solution obtained this way as input for a standard method for which the running time can be negligible we can obtain a very good solution and in the same time satisfying the important restriction of time in the case of real-time applications.

References

1. Zalzala, A.M.S., Fleming, P.J. – *Genetic Algorithms in Engineering Systems*. IEEE London, UK, 1997.
2. Mitchell, M. – *An introduction to genetic algorithms*. Cambridge, MIT Press, 1996.
3. Chong, E.K.P., Żak, S.H. – *An introduction to optimization*. 2nd edition, John Wiley & Sons, 2001.

4. Conn A.R., Scheinberg, K., Vicente, L.N. – *Introduction to derivative free optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 2009.
5. Haupt, R.L., Haupt, S.E. – *Practical genetic algorithms*. 2nd edition, Wiley Interscience, 2004.
6. *** – Online documentation for Mathworks products (Genetic Algorithm and Direct Search Toolbox), http://www.mathworks.com/.

Posibilitatea optimizării hibride folosind algoritmi genetici şi metode standard

Rezumat

În acest articol este prezentată posibilitatea dezvoltării unei metode hibride de optimizare ce cuprinde tehnica algoritmilor genetici şi o metodă standard de optimizare fără derivate. În cazul aplicaţiilor de timp real, timpul impune importante restricţii asupra aplicaţiilor, ceea ce în cazul algoritmilor genetici produce o micşorare a performanţei, soluţia găsită nefiind foarte exactă, însă fiind un bun punct de plecare pentru o metodă standard de optimizare. Sunt prezentate de asemenea rezultatele unei astfel de metode hibride aplicate unei funcţii des utilizate pentru a testa metodele de optimizare.