Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, Volume 23, No.1, ISSN 2304-1382
ISBN 978-3-901509-91-9, CDROM version, Ed. B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2012
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2012

ASPECTS OF MULTIDIMENSIONAL OPTIMIZATION

MATEIA, N[icolae] A[drian] & CECHIN - CRISTA, P[ersida]

Abstract: *The complexity and dynamics of present-day problems lead to a new stage in research and in seeking the optimal solutions in technical, economic and social environments. Therefore, advanced multidimensional optimization techniques are based not only on mathematic concepts, but also on elements of artificial intelligence. In this paper we present a modified version of the Nelder & Mead algorithm, as well as the trends of Genetic Algorithms and Swarm Intelligence.*
Keywords: *multidimensional optimization, Nelder & Mead algorithm (simplex), evolutionary algorithms, hybrid algorithm*

1. INTRODUCTION

Under the present circumstances of uncertainty and rapid changes, a series of techniques which might offer a solution to management issues and add the necessary information for choosing the appropriate solution are to be mentioned.

Economic theory and practice proves direct correlations between the management and the economic efficiency of any company. In order to achieve an increase in productivity, there is a need for ensuring correlation between the capacity of equipment, on the one hand, and the capacity of workload quotas, on the other. Measures are to be taken in order achieve the technical, quality and economic parameters mentioned in the technical documentation [10].

The present management is mainly dealing with the reduction of the information-decision-action-control cycle and with the complex result evaluation by the managers of the companies [10].

The general framework for optimization models, both from the theoretical point of view, and from the practical one, lies in the models and methods of mathematical optimization. Many phenomena from various domains (economy, technical) can be described by means of linear and especially nonlinear mathematical optimization models [15],[17].

In recent years, specialized literature developed several so-called *"penalty function"* models. These models solve nonlinear optimization problems by solving a sequence of minimization problems of a function with no restrictions [1], [14], [15].

The Nelder & Mead (*simplex*) algorithm is a search direct method, where the shape of the simplex changes as the algorithm progresses [10], [11].

A problem encountered when dealing with Evolutionary Algorithms is the lack of efficient methods to handle constraints. The penalty function approach is applicable to any type of constraint (linear or nonlinear) [17].

Genetic Algorithms (GA) are stochastic optimization methods based on concepts of natural selection and genetics. GAs typically work by iteratively generating and evaluating by means of an evaluation function.

Particle Swarm Optimization (PSO) is based on the idea of collaborative behavior and swarming in biological populations. PSO is more computationally efficient than the GA, due to the use of less function evaluations.

More recently, algorithms based on biology and swarm intelligence present the following advantages [7], [8], [9],[16], [18]:

Are derivate – free;

Are not trapped in a local minimum;

Can be hybridized with other algorithms in order to achieve improved performance.

2. MULTIDIMENSIONAL OPTIMIZATION

The main difficulty of mathematical optimization is the *scale* of the analyzed problems. In the field of mathematical optimization, the scale of a problem depends on several factors [1],[14]:

The number of variables and of restrictions. The complexity of the expressions of the objective function and of the restrictions;

The performance of the equipment, both hardware and software.

Basically, the methods for solving large problems can be classified into two categories:

Direct methods, specializing a general procedure by adapting it to the characteristics of a certain class of optimization problems.

Indirect methods, splitting the large problem into smaller, interconnected sub-problems. The sub-problems can be solved independently (if possible, simultaneously), but the fact that they interact involves a coordination mechanism (problem). Therefore, the original large problem is solved on *two levels*.

Given the mathematical optimization problem:

$$\begin{cases} \min f(x), \\ g_j(x) \le 0, \quad j = 1,2,...m \end{cases} \quad (1)$$

$f, g_j : \Re^n \to \Re$, ($j=1,...,m$) have different properties which are to be specified.

The notation

$$M = \{x \in \Re^n \mid g_j(x) \le 0, \quad j = 1,2,...m\}$$

is used for the set of admissible solutions for the given problem.

The principle of the penalty function method involves replacing the solving of a mathematical optimization problem with restrictions with the solving of a sequence of minimization problems with no restrictions [10], [12].

The principle of the method lies in solving a sequence of minimization problems with no restrictions instead of solving a mathematical optimization problem with restrictions (PR).

The nonlinear optimization problem (P) can also be presented as:

$$\min_{x \in M} f(x) = f(x^*) = m$$

or, in a more concise way, as:

$$\text{(PR)} \quad \min\{ f(x) \mid x \in M \}$$

Besides the penalty functions method, barrier functions can be used. These functions are based on the principle that adding a new constraint (barrier) to the existing restrictions prevents the solutions of optimization problems with no restrictions from exiting the domain of admissible solutions *M*. While penalty methods can start from an inadmissible solution and in the end lead to the optimal (admissible) solution, barrier function methods use techniques which prevent the solution from becoming inadmissible [10], [17].

For several algorithms used to minimize a function with more variables $f : \Re^n \to \Re$ (the Cauchy algorithm, the Fletcher-Reeves algorithm, the Davidon-Fletcher-Powell algorithm, the cyclic optimization algorithm on coordinate axes, the Powell algorithm, etc.) it is indispensable to determine the minimum of the function *f* on a given direction, i.e. at any given iteration *k*, the minimum of the function *f* is searched starting from a given point x_k, on a given direction $d_k\left(x_k, d_k \in \Re^n\right)$, i [10],[13],[14]:

$$\min_{p>0} f\left(x_k + p d_k\right) = f\left(x_k + p_k d_k\right).$$

It is important that most solving algorithms require an initial value, this leading to an additional analysis.

Apparently, the problem of minimizing a function with one variable [25], [36], [68] is quite simple. Therefore, should function *f* be derivable, the determination of the minimum x^* in a given interval [*a,b*] can be performed using the traditional results of mathematical analysis.

Given $x_1, x_2, ..., x_r$ the solutions of the equation

$$f'(x) = 0,$$

which belong to the interval [*a,b*]. The value of the function is calculated in the points $a, x_1, ..., x_r, b$ and the point x^* is immediately chosen.

Optimization algorithms which do not use the data provided by the gradient are based on the main idea described below. Starting at the given point $x^0 \in \Re^n$, a new point $x^0 + p d^j$ is to be calculated on each direction of the coordinate axes. In this case, d^j *(j=1,...,n)* represents the directions of the coordinate axes, while $p>0$ represents the initial exploiting step. Should the value of the function in the considered point be better (i.e. towards the optimal value), a new point is to be determined using the same direction and step. Should the value of the function be worse (i.e. further from the optimal value), research is to be performed in point $x^0 - p d^j$. Should the value in this point still not be better than in point x^0, exploration is to be continued in the other directions *(j=2,...,n)*.

Should the function *f* not be derivable or should its values be obtained following experiments, special algorithms which do not use the derived information are to be used in order to determine the optimal level of the function of a given direction.

The one-dimensional optimization problem is $\min_{x \in [a,b]} f(x)$, where $f:[a,b] \to \Re$, is a continuous function.

Besides the conceptual description of algorithms, the computer-based description was attempted, with the detailed presentation in [11],[14].

Since the number of peaks of the polyhedral set M is *finite*, in case of linear program (P) the problem of determining an optimal solution x^* from the generally infinite set of all admissible solutions is reduced to identifying x^* in the finite set of peaks.

The *simplex* method (Nelder & Mead) *systematically* performs this search, offering the optimal solution x^* in a finite number of steps (iterations).

We propose a modification of the simplex algorithm (Nelder & Mead) based on the idea that a continuous function $f : \Re^n \to \Re$ *can often be accurately approximated around relative optimal points by means of a square function* [13].

Should $n \ge 2$, then $x_1, x_2, x_3 \in \Re^n$ are the first three points of the simplex, so that $f(x_1) \le f(x_2) \le f(x_3)$.

By means of interpolation, the function $y = ax^2 + bx + c$ is determined. The interpolation conditions provide the system:

$$a x_i^2 + b x_i + c = f_i, \quad i = 1,2,3$$

(Linear system of three equations with three unknown values *a*, *b* şi *c*) where $f_i = f(x_i)$, *i* =1,2,3.

This leads to:

$$a = \frac{f_1(x_2 - x_1) + f_2(x_3 - x_1) + f_3(x_1 - x_2)}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)}$$

$$b = \frac{f_1\left[(x_2)^2 - (x_1)^2\right] + f_2\left[(x_3)^2 - (x_1)^2\right] + f_3\left[(x_1)^2 - (x_2)^2\right]}{(x_2 - x_1)(x_2 - x_3)(x_3 - x_1)}$$

and, for $a > 0$, the peak of parabola *y* is in point

$$x_{\min} = -\frac{1}{2}\frac{b}{a} \quad (1)$$

The value of the function *f* in point $x_{\min}$ is calculated.

Should $f(x_{\min}) < f(x_{n+1})$ then $x_{n+1} := x_{\min}$.

Should $f(x_{\min}) \geq f(x_{n+1})$, the algorithm continues with the previous simplex.

A *second modification* is represented by the determination of the initial simplex. (For this purpose, we introduced Step 0 in the description). We suggest the following procedure:

Starting from a random given point $\bar{x} \in \Re^n$ and given $d^i = e_i,\ (i = 1,2,\ldots,n)$ the directions of the coordinate axes.

By means of an optimization algorithm of a function on a given direction, the following is determined:

$$f(\bar{x} + p_i d_i) = \min_p f(\bar{x} + p d_i), \quad i = 1,\ldots,n$$

We chose $x_i := \bar{x} + p_i d_i, \quad i = 1,\ldots,n$ and $x_{n+1} := \bar{x}$. The points of the simplex are to be renumbered following the growing order of the values of function *f*.

Based on these modifications, the steps of the new algorithm are the following:

Step 0 (the determination of the simplex):

Given $\bar{x} \in \Re^n$ (possibly $\bar{x} = (0,\ldots,0)^T$).

$d_i = e^i \quad (i = 1,2,\ldots,n)$ is calculated.

Points $x_i := \bar{x} + p_i d_i \quad (i = 1,2,\ldots,n)$ are determined, where (by means of a minimization algorithm of function *f* on the directions d_i)

$f(\bar{x} + p_i d_i) = \min_p f(\bar{x} + p d_i)$ and point $x_{n+1} := \bar{x}$.

In this case, go to Step 1.

Step 1:

The peaks of the simplex are renumbered so that

$$f(x_1) \leq f(x_2) \leq \ldots \leq f(x_n) \leq f(x_{n+1})$$

Using points x_1, x_2 and x_3 and considering $f_i = f(x_i),\ i = 1,2,3$, we calculate *a* and *b* with, the new point x_{min} with and $f(x_{min})$:

Should $f(x_{\min}) < f(x_{n+1})$ then $x_{n+1} := x_{\min}$; $f(x_{n+1}) := f(x_{\min})$, renumber the peaks of the simplex according to (1) and go to Step 2.

Should $f(x_{\min}) \geq f(x_{n+1})$, go to Step 2.

The other steps of the simplex remain the same.

We mention that the presented modifications preserve the convergence conditions [10].

The presented algorithm and the suggested modifications were tested with the Rosenbrock test function and produced the following values:

Precision	"Simplex" algorithm	Modified algorithm
$\varepsilon_1 = 0{,}1$	$x_1 = 0{,}988$; $x_2 = 0{,}902$; $NF = 131$	$x_1 = 0{,}991$; $x_2 = 0{,}911$; $NF = 119$
$\varepsilon_2 = 0{,}0$	$x_1 = 1{,}012$; $x_2 = 1{,}009$; $NF = 219$	$x_1 = 9{,}998$; $x_2 = 1{,}010$; $NF = 200$

Tab. 1. Function and its values

Specialized literature reveals other modifications of the Nelder & Mead algorithm, as well as comparisons and even the transposition of this algorithm into Genetic Algorithms (GA) and Particle Swarm Optimization algorithms (PSO) [2],[4], [7].

While genetic algorithms successively operate with generations of any given number of individuals, traditional optimization methods iteratively switch from one solution (individual) to another better solution. This characteristic of genetic algorithms leads to a more effective exploration of the possible solutions and avoids the risk of generating a local optimal value. Genetic algorithms use probability to select, form the individuals of one particular generation, those individuals which have a better environmental adaptability in order to create a genetic base for the creation of a new generation [3], [5].

From a different point of view, out of the solutions of one original set, the solutions with the most appropriate values of the performance function are selected and combined in order to generate a new set of solutions with better values of the function. After the process is repeated several times, all solutions in the set have close values when compared to each other and to the optimal solution, leading to the best solution being accepted as a satisfactory sub-optimal solution.

Particle Swarm Optimization Algorithm (PSO) is an EEA introduced by Kennedy and Eberhart in 1995. The aim of this algorithm, similar to all EEAs, is to improve the performance of random searches in optimization, by identifying and exploring the spaces with a high probability of containing more effective solutions [7].

From a conceptual point of view, PSO is a combination of genetic algorithms (GA) and evolutionary programming (EP). The advantage of PSO lies in its dependence on stochastic processes, rather than on evolutionary operations. The particles, characterized by position and speed, are initially distributed randomly in the search area and are meant to converge towards the global optimal solution [5], [7].

The use of PSO as a stochastic search algorithm in case of a large scale problem faces the disadvantage of a premature convergence (caused by the loss of diversity in the intermediate stages)[2], [7].

A genetic algorithm deals with three classes of specific operators (*selection, crossover, mutation*) which can be implemented in various stages. Although PSO does not label the stages the same way as a GA, there is an analogy. PSO does not contain the operator crossover, but the concept is present; each particle is stochastically accelerated towards its best previous position, and ultimately towards the best global position [3].

The mutation of Genetic Algorithms determines the possibility for one individual to jump to any available direction. The evolution of one individual-chromosome during the crossover operation involves information exchanges with other randomly selected individuals. This guarantees that the next generation has individuals with the best above average *fitness value.* The fitness value represents the objective function we want to minimize [3], [9].In the case of PSO, this information exchange with other individuals is not present, but the trajectory of one individual is influenced by the other individuals.

Besides the parameter dependency and the loss of diversity of EA and PSO, another major drawback is that they can only be applied to a search solution space with a fixed dimensionality [18]. The major difference between PSO and other Evolutionary Algorithms is that PSO does not implement the survival of the fitness, since all particles in a PSO are kept as members of the population through the course of the searching process [4].In recent years, there are sufficient studies which present the idea of a *hybrid algorithm*, formed from the Nelder & Mead algorithm (NM) and the Particle Swarm Optimization Algorithm (PSO) or Genetic Algorithm (GA). The hybrid algorithm is based on the idea of combining the advantages of these algorithms and avoiding their disadvantages. The hybrid algorithm can appear in two forms: tandem and cascade. According to experiments, the cascade composition of the two algorithms offers the best results [7], [16].The hybrid algorithm formed from the Nelder & Mead algorithm and Genetic Algorithm combines the advantages of both methods:

The convergence to the global minimum (GA).The high accuracy of the Nelder & Mead algorithm.

Another proposal for multidimensional optimization refers to not visiting the points already visited by genetic algorithms (non-revisiting GA). This method has the following advantages[9], [12]:

It avoids premature convergence. The best individuals converge faster and according to the convergence conditions. To avoid the problem of premature convergence and to improve the rate of PSO convergence to global optimal value, the Fractional Global Best Formation (FGBF) technique is used. FGBF collects all the best dimensional components and fractionally creates an artificial Global Best particle (aGB) which has the potential to be a better guide than the the *gbest* particle [6], [9].

In order to find an optimal value in multi-dimensional dynamic environments, the use a combined multi-swarm algorithm with FGBF technique is recommended so that each swarm can apply the FGBF distinctively [6].

3. CONCLUSION

Traditional optimization algorithms have disadvantages in the case of large scale problems, while evolutionary algorithms offer effective methods for global optimization. Due to the fact that any algorithm has both advantages and disadvantages, the recommended strategy and the future endeavor is to combine traditional and evolutionary algorithms.

4. REFERENCES

[1] Dumitru, V., Luban, F., Moga, S., Şerban, R., (1973)- ”Programare neliniară”, Appendix P., Editura Academiei R.S.R., Bucureşti, 1973

[2] Durand N., Alliot J-M., - A combined Nelder-Mead Simplex and Genetic algorithm, Genetic and Evolutionary Computation Conference, 1999, Orlando, USA

[3] Ebehart C. R., SHI Y., - Comparison between Genetic Algorithms and Particle Swarm Optimization, EP '98 Proceedings of the 7th International Conference on Evolutionary Programming VII, Pages 611 – 616, Springer-Verlag London, UK ©1998, ISBN:3-540-64891-7

[4] Fan S. K., Liang Y. C., Zahara E., - A genetic algoritm and a particle swarm optimizer hybridizer with Nelder – Mead sipmlex search, Computers &Industrial Engineering 50, 401-425, 2006

[5] Hassan R., Cohamin B., Weck O. De, Venter G., A comparision of particle swarm optimization and the genetic algorithm, In 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2005), pp. 1-13

[6] Kiranyaz S., Pulkinnen J., Gabbouj M., - Multi-dimensional Particle Swarm Optimization in Dynamic Enviroments Expert Systems with Applications, Vol 38, Issue 3, 2212- 2223, 2011

[7] Koduru, P., Das, S., Welch, S.: A Particle Swarm Optimization-Nelder Mead Hybrid Algorithm for Balanced Exploration and Exploitation in Multidimensional Search Space. ;In IC-AI(2006), 457-464

[8] Lagarias C. J., Poonen B., Wright H. M., - Convergence of the restricted Nelder Mead algorithm in two dimensions, 2011

[9] Mastorakis E. N., - Genetic Algoritm with Nelder Mead Optimization in the variational methods of Boundary Value Problems, WSEAS Transaction on Mathematics, vol.8, march 2009, ISSN 1109 - 2769

[10] Mateia, N. A., (2009) – Models and algorithm for operative programming production, Phd Thesis

[11] Nelder J. A., Mead R., - A simplex method for function minimization, Computer Journal. 7,308 -313, 1965.

[12] Saroj, Devraj, - A non-revisiting Genetic Algorithm for optimizing numeric multi- dimensional functions, International Journal on Copmutational sciences & Applications (IJCSA), ISSN: 2200-0011 vol2, no 1, 2012

[13] Şerban, R., Şerban, R. R., Mateia, N. A.,(2009) – ”O modificare a algoritmului „SIMPLEX” pentru optimizarea multidimensională”; The Fourt International Conference on Economic Cybernetic Analysis: Global Crisis Effects on Developing Economies, Bucharest Academy of Economic Studies, 22-23 may, 2009, ISBN:979-606-505-219-2, 358-367

[14] Serban, R., Albu, C., Şerban, R: R., (2004).- ”Cercetări operaţionale cu aplicaţii în economie”, Editura Dacia Europa Nova, Lugoj

[15] Şerban, R., (1987) – ”Algorithms of unidimensional optimization”, Economic Computation and Economic Cybernetics Studies and Research, nr1, 1987, 41-56

[16] Zahara E., Kao Y. T.,:- Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst. Appl. 36(2): 3880-3886 (2009)

[17] Yeniay O., - Penalty function methods for constrained optimization with Genetic Algoritms, mathematical and Computational Applications, vol10, No 1, 2005, 45 – 56

[18] Yang C. H., Tu C. J., Chang J. Y., Liu H. H., Po-Chang Ko - Dimensionality Reduction using GA-PSO, In proceeding of: Proceedings of the 2006 Joint Conference on Information Sciences, JCIS 2006, Kaohsiung, Taiwan, ROC, October 8-11, 2006