



SOLUTION OF THE NONLINEAR LEAST SQUARES PROBLEM USING A NEW GRADIENT BASED GENETIC ALGORITHM

D. Geraldine Bessie Amali¹ and Dinakaran M.²

¹School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India

²School of Information Technology, VIT University, Vellore, Tamil Nadu, India

E-Mail: geraldine.mithra@gmail.com

ABSTRACT

In this paper a new Gradient Descent based Genetic Algorithm (GDGA) is proposed and used to solve benchmark Nonlinear Least Squares (NLS) problems. GDGA uses theoretically calculated gradient to perform a gradient descent around the best solution found by the Genetic Algorithm (GA). This approach employs the GA to escape from local minima and estimate a solution in the neighborhood of the global minima. Once an approximation to the global minimum is found, gradient descent is done with the solution found by the GA as the starting point. Stochastic search algorithms like GA can easily compute a solution in the vicinity of the global minimum but take a long time to converge to the exact minimum due to the random nature of the search. Thus GDGA synergistically combines the advantages of deterministic local search and heuristic random global search to compute an accurate solution efficiently. Simulation results indicate that GDGA performs well on benchmark NLS problems.

Keywords: genetic algorithm, gradient descent, nonlinear least squares.

1. INTRODUCTION

Fitting a model to a set of experimental observations is a problem of long standing importance. In many important applications the model is nonlinear with respect to the parameters and well-developed linear regression methods cannot be applied [18]. Consider a set of experimental observations $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), \dots, (x_{n-1}, y_{n-1}), (x_n, y_n)\}$ which relate an input variable $x \in \mathbb{R}^n$ and output variable $y \in \mathbb{R}$. The unknown relationship between x and y can be modelled a functional relationship $y = f(x, \theta)$. Thus the problem of modelling the unknown relationship between x and y can be reduced to finding a value of θ that minimizes an error measure between predictions [18] $f(x_i, \theta)$ and experimental observations y_i .

Alternative error measures such as sum of absolute errors or maximum prediction error over the data can be chosen but have the disadvantage of resulting in a non-differentiable cost function. In the Nonlinear Least Squares (NLS) formulation the error measure is taken to be the sum of squares of the prediction errors [18]. In NLS the error measure or cost function is in general a nonlinear function of the parameters of the model. Also the resulting cost function can be highly multimodal. However the problem of finding a globally optimal solution to a highly nonlinear multimodal function is a NP complete problem and hence only an approximate solution can be computed using heuristic random search algorithms like GAs.

In "A genetic algorithm based technique for computing the nonlinear least squares estimates of the parameters of sum of exponentials model" [2] Mitra *et al.* propose an iterative algorithm based on a generational genetic algorithm for calculating the nonlinear parameters of sum of exponential models. They have tested their algorithm in real life data and their simulation results prove that their proposed procedure is highly efficient. They add that since GA has been employed as a search technique it is not required to calculate the derivatives of

the cost function with respect to the parameters instead the algorithm searches based on the fitness levels and converges faster.

O Kocadagli in "A novel nonlinear programming approach for estimating CAPM beta of an asset using fuzzy regression" [3] has taken the Capital Asset Pricing Models for study. The existing techniques such as the least squares method (LSM) and the Robust Regression Technique (RRT) which are used for evaluating the beta coefficient remove the extreme values. He argues that by removing these extreme values the accuracy of the beta coefficient is reduced as these values might contain very important information. So, he proposes a fuzzy regression approach which takes into account these values as well. He has tested his approach in calculating the beta coefficients of three assets in Istanbul Stock Exchange and compared his results with the traditional approaches.

In the paper "Nonlinear conjugate gradient methods with structured secant condition for nonlinear least squares problems" [4] Kobayashi *et al.* study conjugate gradient methods that utilize the Hessian structure of the objective function for solving nonlinear least squares problems. The structured secant condition provides estimation for the Hessian and given specific assumptions regarding the bound and neighbourhood, provide favourable global convergence.

N Sagara in "A hybrid method for the nonlinear least squares problem with simple bounds" [5] utilizes a trust region technique which imposes upper and lower bounds on the variables of nonlinear least squares problems to construct feasible regions of solutions. The result is a finer convergence devoid of the combinatorial complexities normally encountered in alternative approaches. Z Chen *et al.* in their paper "A Parallel Iterative Method for Solving Nonlinear Least-Squares Problems" [6] employ a parallelized approach to solution of nonlinear least squares problems using modified ordering of variables which are distributed among non-



disjoint Jacobian groups. The technique has good speed up and efficiency.

M Aciet *al.* in their paper "A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm" [7] propose a hybrid approach to machine learning using the said three techniques with the aim of elimination of training sets that are not conducive to learning. They observed that the new technique resulted in upto 75% improvement in accuracy for certain datasets and found that it works well even for small datasets. GR Liu et al. in their paper "A combined genetic algorithm and nonlinear least squares method for material characterization using elastic waves" [8] make use of a material characterization problem in which parameters are determined by minimizing error functions and with a forward solver. Genetic algorithm is used to find favourable solutions which then act as the initial seeds for nonlinear least squares method. Comparison of error function values results in the identification of the required parameters [8].

Two hybrid genetic algorithms are discussed in [14]. One uses GA with hill climbing and the other uses GA with Quasi Newton [14]. Whereas in [17] a hybrid simulated annealing technique is used for estimating the nonlinear parameters. An average differential evolution algorithm is proposed in [16]. S Biswas et al. in their paper "An artificial bee colony-least square algorithm for solving harmonic estimation problems", approach the solution of harmonic estimation problems in power quality signals using a stochastic optimization technique called the artificial bee colony which they hybridize using least squares. The results demonstrate that the new technique outperforms previous techniques.

Rest of the paper is organized as follows. Section 2 contains the detailed explanation of the working of the GA. GD algorithm is discussed in section 3. The proposed GDGA is presented in section 4. Results obtained are discussed in section 5. Section 6 concludes the paper with future enhancements.

2. GENETIC ALGORITHM

Genetic Algorithm is a global search heuristic and has been successfully used to solve optimization problems [1][15][14]. It employs the techniques which are inspired by evolution such as inheritance, mutation, crossover and selection. The algorithm searches for the best solution in the search space that optimizes the cost function. The algorithm starts with randomly generated individuals. Then the individuals are tested against a fitness value which is generally the cost function in the optimization problem. The fittest ones are selected probabilistically from the current population. Some of the selected ones are not altered and are just carried over to the next generation [1][14][15]. A percentage of the other individuals are altered using the genetic operators such as mutation or crossover. In order to perform crossover, two parents are selected from the pool for breeding. The child which is formed will share the characteristics of both the parents. By performing crossover on two fit parents, the child will have high quality chromosomes when compared

to its parents [10]. Mutation is next applied to the individuals where the symbols of the chromosomes are randomly altered with a given probability p . The value of p is set to be very small so that only a few chromosomes undergo change due to mutation. After applying crossover and mutation operations to the pool the next generation of solutions is obtained. Since the new generation is formed by the selected fittest individuals, the new generation will be superior to the previous generation. Iteration terminates when the desired fitness value is obtained or maximum number of iterations have been reached.

GA however has its limitations. Using GA finding the solution to a highly multimodal and high dimensional optimization problem will require repeated calculations of the fitness function which can be very expensive [11]. Therefore an approach that combines the explorative ability of the GA but reduces the number of function evaluations would be very effective.

3. GRADIENT DESCENT ALGORITHM

Gradient descent algorithm is a first order optimization algorithm that is used to determine the local minimum of a cost function [4]. The algorithm requires the computation of the gradient of the function and in order to find the local minimum of the cost function, small steps η are taken proportional to the negative gradient of the function.

This algorithm starts at a random point x_0 and the cost function $F(x_0)$ will reduce the maximum if it moves in the direction of the negative gradient of F at x_0 [13].

Starting from the random point X_0 , we move in the sequence x_1, x_2, x_3, \dots such that $F(x_i) > F(x_{i+1})$. The equation is given by

$$\vec{X}_{n+1} = \vec{X}_n - \eta \nabla F(\vec{X}_n)$$

Where

$$\begin{aligned} \vec{X}_n &= \text{current approximation to the minimum} \\ \eta &= \text{rate of descent} \\ \nabla F &= \text{gradient of the function} \end{aligned}$$

Gradient Descent Algorithm is proven to converge to the local minimum with a good guess of the starting position and value of η [12]. In the case of a convex optimization function the local minimum will be the global minimum thereby the algorithm converges to the global best solution. However, since most cost functions are non-convex, the algorithm might converge to a local minimum. Therefore combining the exploitation ability of the gradient descent algorithm with the explorative ability of the GA is of interest.

4. PROPOSED GRADIENT DESCENT GENETIC ALGORITHM

In this paper a new gradient descent genetic algorithmic (GDGA) approach is applied to minimize the sum of squares error. GA is a stochastic algorithm therefore it is good at exploring multiple minima. Due to



its stochastic nature it explores the whole search space however; it provides only an approximate solution. GD on the other hand is a local search heuristic which does accurate local search.

The algorithm proposed in the paper first uses GA to estimate the parameters of the nonlinear equation. The Genetic algorithm starts with random values for the parameters of the nonlinear function and with every iteration the algorithm approximates the nonlinear equation. Since Genetic algorithm is a global optimization algorithm it is computationally expensive. Therefore we introduce the gradient descent approach which minimizes the number of iterations of the genetic algorithm.

The proposed gradient based Genetic Algorithm (GDGA) uses theoretically calculated gradient to perform a gradient descent around the best solution found by the Genetic Algorithm. This approach employs a GA to

escape from local minima and estimate a solution in the neighbourhood of the global minima. Once an approximation to the global minimum is found, gradient descent is done with the solution found by the GA as the starting point. By using the gradient descent around the best solution determined by the GA it will lead to faster convergence and produce more accurate results.

By obtaining only an approximate solution using the computationally expensive GA and refining it using the efficient GD local search algorithm the advantages of the GA and GD method are effectively combined.

The cost function f is the input to the GA. The GA returns the best solution X . This solution is the starting point for GD. GD returns the optimal solution X^* such that $f(X^*) < f(X)$.

A block diagram of the working of the proposed GDGA is given in Figure-1.

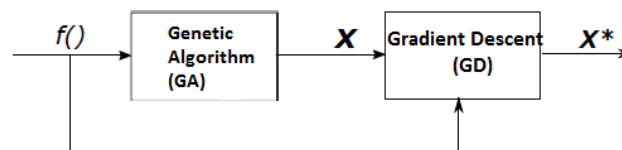


Figure-1. Block diagram of the GDGA

Table-1. Nomenclature used in GDGA

Symbol	Description
N	Size of the population for GA
$x^{(i)}$	Individual 'i' in the population of solutions for GA
$P(k)$	Population of solutions at iteration 'k'
$M(k)$	Mating population of solutions at iteration 'k'
f	Cost function
γ	Mutation probability for GA
η	Learning rate for the gradient descent
∇	Gradient operator
ε	Threshold value for GD

**Table-2.** Gradient descent genetic algorithm (GDGA).

```

Level 1: GA:
    k := 0
    Initialize P(k) with randomly generated 'N' feasible solutions  $x^{(1)}$  to  $x^{(N)}$ 
    for i=1 to N do
        Compute  $f(x^{(i)})$ 
        Done
        Repeat until stopping criterion is met

    Selection: Select best solutions from P(k)
                $M_k := \text{Selected}(P(k))$ 
    Crossover: Randomly select  $x^{(i)}, x^{(j)} \in M(k)$  where  $1 \leq i, j \leq N$ 
                $x^{(i')}, x^{(j')} := \text{Crossover}(x^{(i)}, x^{(j)})$ 
               Replace  $x^{(i)}, x^{(j)}$  with  $x^{(i')}, x^{(j')}$  in M(k)

    do
        Mutation: Randomly select  $x^{(i)} \in M(k)$  where  $1 \leq i \leq N$  with probability  $\gamma x^{(i')} := \text{Mutation}(x^{(i)})$ 
               Replace  $x^{(i)}$  with  $x^{(i')}$  in M(k)

    P(k+1) := M(k)

    For i=1 to N do
        Compute  $f(x^{(i)})$ 
        done

    Return  $x^{(\text{best})}$  such that  $f(x^{(\text{best})}) < f(x^{(i)}) \forall 1 \leq i \leq N$ 

Level 2: GD :
     $x^{(0)} := x^{(\text{best})}$ 
    k := 0
    Repeat until  $|f(x^{(k+1)}) - f(x^{(k)})| < \varepsilon$ 
    do
         $x^{(k+1)} := x^{(k)} - \eta \nabla f(x^{(k)})$ 
         $x^{(\text{best})} := x^{(k+1)}$ 
        k := k + 1

    Return  $x^{(\text{best})}$ 

```

5. RESULTS AND DISCUSSIONS

Two benchmark problems have been taken for testing the proposed GDGA. In benchmark problem 1 estimating the amplitude, angular frequency and phase shift of a sinusoid from noisy data is considered. In benchmark problem 2 estimating the amplitude, angular frequency, phase shift and decay constant of a decaying sinusoid from a noisy data is considered. Theoretical calculation of the gradients is shown below.

5.1 Benchmark problem 1

The cost function for benchmark problem 1 is given in equation (1). Partial derivatives with respect to x_1, x_2, x_3 are given in equations (2), (3) & (4). The gradient descent is given in (5).

$$C(x) = \sum_{i=1}^N (x_1 \sin(x_2 t_i + x_3) - y_i)^2 \quad (1)$$

$$\frac{\partial C}{\partial x_1} = \sum_{i=1}^N 2[x_1 \sin(x_2 t_i + x_3) - y_i] \sin(x_2 t_i + x_3) \quad (2)$$

$$\frac{\partial C}{\partial x_2} = \sum_{i=1}^N 2[x_1 \sin(x_2 t_i + x_3) - y_i] x_1 \cos(x_2 t_i + x_3) t_i \quad (3)$$

$$\frac{\partial C}{\partial x_3} = \sum_{i=1}^N 2[x_1 \sin(x_2 t_i + x_3) - y_i] x_1 \cos(x_2 t_i + x_3) \quad (4)$$

$$\nabla C = \left[\frac{\partial C}{\partial x_1} \frac{\partial C}{\partial x_2} \frac{\partial C}{\partial x_3} \right]^T \quad (5)$$

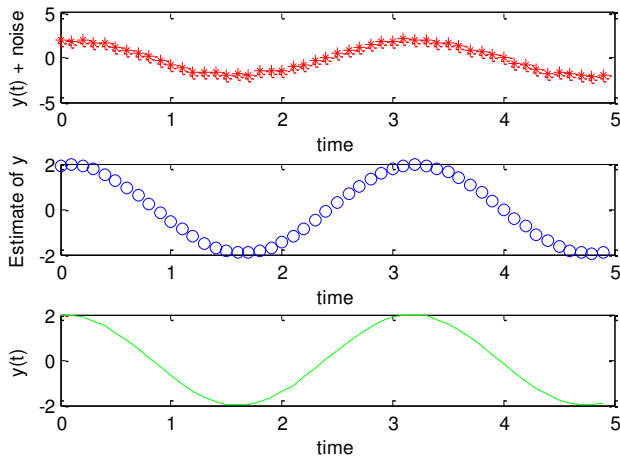


Figure-2. Illustrates the plot of the nonlinear function against time for benchmark problem 1.

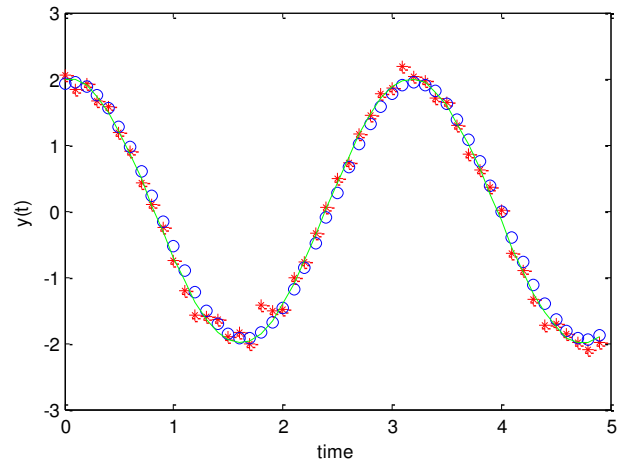


Figure-3. Displays the plot of the nonlinear function against time for benchmark problem 1. Solid line - is the ideal $y(t)$, 'o' - $y(t)$ with parameters estimated with GDGA, '*' - noisy $y(t)$ used for parameter estimation.

5.2 Benchmark problem 2

$$C(x) = \sum_{i=1}^N [x_1 e^{x_2 t_i} \sin(x_3 t_i + x_4) - y_i]^2 \quad (6)$$

$$\frac{\partial C}{\partial x_1} = \sum_{i=1}^N 2[x_1 e^{x_2 t_i} \sin(x_3 t_i + x_4) - y_i] e^{x_2 t_i} \sin(x_3 t_i + x_4) \quad (7)$$

$$\frac{\partial C}{\partial x_2} = \sum_{i=1}^N 2[x_1 e^{x_2 t_i} \sin(x_3 t_i + x_4) - y_i] x_1 \sin(x_3 t_i + x_4) e^{x_2 t_i} t_i \quad (8)$$

$$\frac{\partial C}{\partial x_3} = \sum_{i=1}^N 2[x_1 e^{x_2 t_i} \sin(x_3 t_i + x_4) - y_i] x_1 e^{x_2 t_i} \cos(x_3 t_i + x_4) t_i \quad (9)$$

$$\frac{\partial C}{\partial x_4} = \sum_{i=1}^N 2[x_1 e^{x_2 t_i} \sin(x_3 t_i + x_4) - y_i] x_1 e^{x_2 t_i} \cos(x_3 t_i + x_4) \quad (10)$$

$$\nabla C = \left[\frac{\partial C}{\partial x_1} \frac{\partial C}{\partial x_2} \frac{\partial C}{\partial x_3} \frac{\partial C}{\partial x_4} \right]^T \quad (11)$$

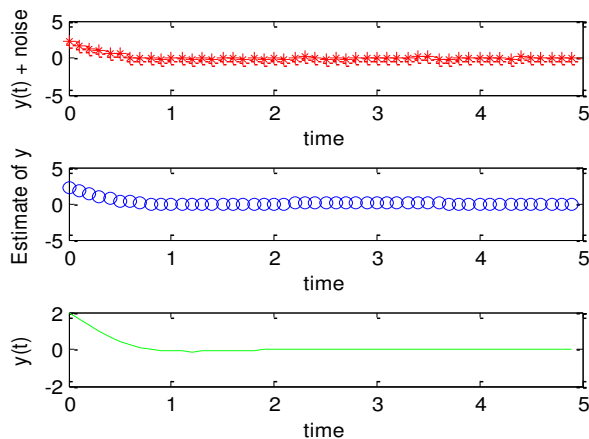


Figure-4. Illustrates the plot of the nonlinear function against time for benchmark problem 2.

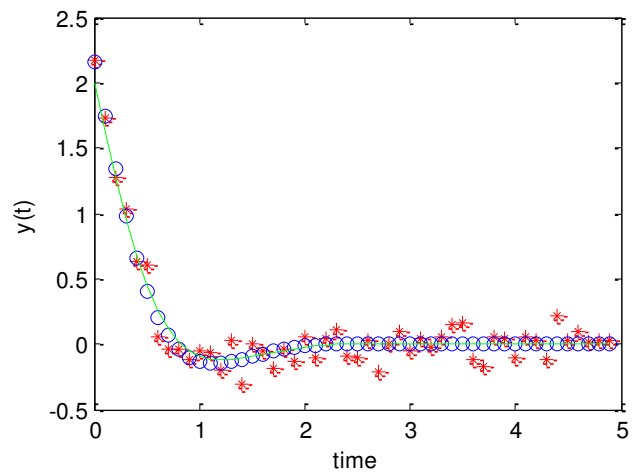


Figure-5. Displays the plot of the nonlinear function against time for benchmark problem 2. Solid line - is the ideal $y(t)$, 'o' - $y(t)$ with parameters estimated with GDGA, '*' - noisy $y(t)$ used for parameter estimation.



The Figures 3 and 5 show the performance of the proposed GDGA which was run for 100 iterations with a population size of 20. The values of the amplitude, angular frequency and the phase shift estimated by the GDGA are

tabulated for benchmark problem 1 and 2. Results indicate that the observed values nearly approximate the target values.

Table-3. The values of the parameters estimated by GDGA.

Benchmark problems	Target output	Output generated by GDGA
Benchmark Problem no 1:		
Amplitude A	2	1.9419
Angular Frequency ω	2	2.0025
Phase Shift Θ	1.5	1.4216
Benchmark Problem no 2:		
Amplitude A	2	2.0087
Angular Frequency ω	2	1.8323
Phase Shift Θ	1.5	1.6498
Decay constant α	-2	-2.2922

6. CONCLUSIONS

In this paper a novel Gradient based Genetic Algorithm (GDGA) that synergistically combines the global random search ability of the genetic algorithm and computationally efficient local search ability of the Nelder-Mead algorithm was proposed. The proposed algorithm was used to compute an accurate solution of benchmark Nonlinear Least Squares (NLS) problems. In this approach computationally expensive random search with the GA is only used to compute an approximate solution in the neighbourhood of the global minimum. Once an approximate solution is computed, the exact minimum is computed with an efficient local search algorithm avoiding the need for locating the exact minimum by random search. Simulation results indicate that GDGA performs well on benchmark NLS problems. Future work can explore the performance of combinations of alternate global and local search algorithms. Parallel implementations of GDGA for solution of large scale NLS problems can also be considered.

REFERENCES

- [1] Tom Mitchell. 1997. Machine Learning. 1st edition. MIT Press and Mc-Graw Hill Companies, Inc.
- [2] S Mitra, Amitra. 2012. A genetic algorithm based technique for computing the nonlinear least squares estimates of the parameters of sum of exponentials model. *Expert Systems with Applications*. 39: 6370-6379.
- [3] Ozan Kocadagli. 2013. A novel nonlinear programming approach for estimating CAPM beta of an asset using fuzzy regression. *Expert Systems with Application*. 40: 858-865.
- [4] M Kobayashi, Y Narushima., H Yabe. 2010. Nonlinear conjugate gradient methods with structured secant condition for nonlinear least square problems. *Journal of Computational and Applied Mathematics*. 234: 375-397.
- [5] N Sagara, M Fukushima. 1991. A hybrid method for the nonlinear least squares problem with simple bounds. *Journal of Computational and Applied Mathematics*. 36: 149-157.
- [6] Z Chen. Y Lv., C Liu. 2009. A Parallel Iterative Method for Solving Nonlinear Least-Squares Problems. *International Conference on Computational Intelligence and Software Engineering*. pp. 1-3.
- [7] M Aci, C Inan, M Avci. 2010. A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm. *Expert Systems with Applications*. 37: 5061-5067.
- [8] GR Liu., X Han., KY Lam. 2002. A combined genetic algorithm and nonlinear least squares method for material characterization using elastic waves. *Computer Methods in Applied Mechanics and Engineering*. 191: 1909-1921.
- [9] S Biswas, A Chatterjee, SK Goswami. 2013. An artificial bee colony-least square algorithm for solving harmonic estimation problems. *Applied Soft Computing*. 13: 2343-2355.
- [10] Tsai J. T., Ho, W. H. and Chou J. H. 2009. Design of Two-Dimensional IIR Digital Structure-Specified Filters by Using an Improved Genetic Algorithm. *Expert Systems with Applications*. 36: 6928-6934.
- [11] Karaboga N., Cetinkaya B. 2004. Design of minimum phase digital IIR filters by using genetic algorithm.



Proceedings of the 6th Nordic Signal Processing Symposium. pp. 29-32.

- [12] Kiwiel Krzysztof C. 2001. Convergence and efficiency of subgradient methods for quasiconvex minimization. Mathematical Programming Series A, Berlin, Heidelberg: Springer. pp. 1-25.
- [13] Arfken G. 1985. The Method of Steepest Descents. Mathematical Methods for Physicists, 3rd ed. Orlando, FL: Academic Press. pp. 428-436.
- [14] Renders J.M., Flasse S.P. 1996. Hybrid methods using genetic algorithms for global optimization. IEEE Transactions on Systems, Man, and Cybernetics. 26: 243-258.
- [15] Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. New York: Addison-Wesley.
- [16] Liao F., Mao M.H., Zhao W. 2013. Nonlinear parameter estimation based on average information differential evolution algorithm. Math Practical Theory. 43: 123-128.
- [17] Guo D.L., Xia H.M., Zhou Y.Q. 2010. Hybrid simulated annealing and evolution strategy algorithm in non-linear parameter estimation application. Math Practical Theory. 40: 91-98.
- [18] Edwin K P Chong., Stanislaw H Zak. 2103. An Introduction to Optimization. 3rd edition. Wiley Series in Discrete Mathematics and Optimization.