*Gazi University Journal of Science*
*GU J Sci*
*29(1):187-199 (2016)*

# Estimating the Parameters of Nonlinear Regression Models Through Particle Swarm Optimization

Volkan Soner ÖZSOY[1], H. Hasan ÖRKÇÜ[1,♠]

[1] *Gazi University, Faculty of Sciences, Department of Statistics, Ankara*

*Received: 10/11/2015* *Revised: 13/01/2016* *Accepted: 14/01/2016*

**ABSTRACT**

Nonlinear regression models are widely used for modeling of stochastic phenomena and the estimating parameters problem plays a central role in the inference in nonlinear regression models. In this paper, this problem has been briefly discussed and an effective approach based on the Particle Swarm Optimization (PSO) algorithm is proposed in order to enhance the estimation accuracy. The PSO algorithm is tested on the well-known 28 nonlinear regression tasks of various level of difficulty. The results show that PSO approach which exhibits a rapid convergence to the minimum value of the sum of squared error function in less iterations, provides accurate estimates and is satisfactory for the parameter estimation of the nonlinear regression models.

**Keywords:** *Particle swarm optimization; nonlinear regression; parameter estimation.*

## 1. INTRODUCTION

Regression analysis is a statistical procedure explaining the relationship between two or more variables. It describes the relationship between two kinds of measurements: the independent or predictor measurements, denoted $x = (x_1, x_2, \dots, x_k)$ and the dependent or response measurements, denoted $y$. The general form of a regression model is $y = f(x, \beta) + \varepsilon$. The response $y$ is composed of two parts: the systematic part $f(x)$ depends on $x$, while the random part $\varepsilon$ is independent from predictors.

In a linear regression model the regression function is a linear function of the unknown parameters, whereas in a nonlinear regression model the regression function is

♠Corresponding author, e-mail: *hhorkcu@gazi.edu.tr*

not a linear function of the unknown parameters. Nonlinear regression models are formally defined as models in which at least one of the model parameters occurs nonlinearly in the model expression. Nonlinear models are used to model complex interrelationships among variables and play an important role in various scientific disciplines and engineering. Common examples on nonlinear models include growth, yield density and dose-response models and various models that are used to describe physical, biological, industrial and econometric processes [1-3].

Whereas the statistical theory of parameter estimation in linear models is almost completely developed, many problems are unsolved in the nonlinear case. The basic idea of nonlinear regression is the same as that of linear regression, namely to relate a response $y$ to a vector of predictor variables $x = (x_1, x_2, \dots, x_k)$. Nonlinear regression is a useful statistical tool, relating observed data and a nonlinear function of unknown parameters. Nonlinear regression is characterized by the fact that the prediction equation depends nonlinearly on one or more unknown parameters. Whereas linear regression is often used for building a purely empirical model, nonlinear regression usually arises when there are physical reasons for believing that the relationship between the response and the predictors follows a particular functional form. For a pair of $(x_i, y_i)$ including $n$ observations, a nonlinear regression model has the form $y_i = f(x_i, \beta) + \varepsilon; i = 1, \dots, n$. The $\varepsilon_i$ are usually assumed to be uncorrelated with mean zero and constant variance.

In the parameter estimation problem, the form of the nonlinear regression function is known but it contains unknown parameters $\beta_1, \dots, \beta_p$. A popular method for estimating the unknown parameters in a nonlinear regression function is the method of ordinary least squares [3]. According to this method, the estimates of $\beta_1, \dots, \beta_p$ are obtained by minimizing the quantity $S(\beta) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - f(x_i, \beta)\right)^2$, the sum of squared of errors prediction. A nonlinear parameter estimation problem is an optimization whereby the objective function $S(\beta)$ is minimized. Since $S(\beta)$ is nonlinear, it has various local minimum and it might be a better alternative to classical (analytical) methods to use meta-heuristic methods.

Parameter estimation procedures are very important in the many scientific fields for development of mathematical models, since all of the process depend on model parameter values obtained from experimental data. Nonlinearity model makes the estimation of parameter and the statistical analysis of parameter estimates more difficult and more challenging. Difficulties are arise due to the large number of parameters and multi modal nature the objective function. It is very difficult to minimize sum of squared of errors function ($S(\beta)$), using ordinary optimization techniques [4]. In order to overcome these difficulties, the use of a powerful meta-heuristic method such as Particle Swarm Optimization (PSO) algorithm may be considered. The PSO has many advantages including simplicity of implementation, is reliable, robust, and in general is considered an effective meta-heuristic optimization algorithm introduced by Kennedy and Eberhart [5] and Eberhart and Kennedy [6]. The PSO is inspired of the behaviors of social models like bird flocking or fish schooling and is based on individual improvement and social cooperation. In this study, the PSO was used for minimum sum of squared of errors function.

There are numerous articles about the parameter estimation of nonlinear regression models. Křivý et al. [7] proposed the controlled random search algorithm for the estimating the parameters of nonlinear regression models. Li et al. [4] proposed a hybrid optimization strategy by incorporating the jumping property of simulated annealing (SA) into the PSO, namely PSOSA, for estimating parameters of non-linear systems, which is an important issue in control fields and essentially is a hard multi-dimensional numerical optimization problem. Kapanoğlu et al. [8] examined the genetic algorithms (GAs) for parameter estimation of nonlinear regression models over a large set of test problems with three difficulty levels. Tvrdík et al. [9] proposed two adaptive population-based search algorithms are proposed for parameter estimation problem of nonlinear regression models. Aşıkgil and Erar [10] examined the efficiency of nonlinear parameter estimation under the problem of autocorrelated errors. In this paper, the nonlinear parameter estimation problem has been briefly discussed and an effective approach based on the PSO algorithm is proposed in order to enhance the estimation accuracy. The PSO algorithm is tested on the well-known 28 nonlinear regression models of various level of difficulty. The experimental results show that the PSO algorithm is significantly reliable for the parameter estimation problem in nonlinear regression model

The remaining contents are organized as follows. In Section 2, some brief information about what the PSO algorithm is explained. In Section 3, how to use the PSO method in the parameter estimation of the nonlinear regression models is introduced. Numerical experiments on well-known 28 nonlinear regression benchmark tests are presented in Section 4. Finally, Section 5 concludes the study.

## 2. OVERVIEW OF THE PARTICLE SWARM OPTIMIZATION

The PSO is biologically inspired technique derived from the collective behavior of bird flocks, first introduced by Kennedy and Eberhart [5] and Eberhart and Kennedy [6]. The PSO, known as an optimizer, is a population-based, self-adaptive search optimization technique [11]. The PSO consists of a set of solutions (particles) called population. Each solution consists of a set of parameters and represents a point in multidimensional space. All the particles in the swarm act individually under the same governing principle: accelerate toward the best personal and best overall location while constantly checking the value of its current location. Each particle has a memory that helps it in keeping the track of its previous best position. The positions of the particles are distinguished as personal

best (pbest) and global best (gbest). Each particle remembers the location where it personally encountered the most flowers. This location with the highest fitness value personally discovered by a particle is known as the personal best or pbest. Each particle has its own pbest determined by the path that it has flown. At each point along its path the particle compares the fitness value of its current location to that of pbest. If the current location has a higher fitness value, pbest is replaced with its current location. Each particle also had some way of knowing the highest concentration of flowers discovered by the entire swarm. This location of highest fitness encountered is known as the global best or gbest. For the entire swarm there is one gbest to which each particle is attracted. At each point along their path every particle compares the fitness of their current location to that of gbest. If any particle is at a location of higher fitness, gbest is replaced by that particles' current position [26-32].

In a *n*-dimensional search space, the position and velocity of individual (particle or solution) *i* are represented as the vectors $X_i = (x_{i1}, \dots, x_{in})$ denote a particle's position (coordinate) and $V_i = (v_{i1}, \dots, v_{in})$ denote the particle's flight velocity over a solution space in the PSO algorithm. Each individual *x* in the swarm is scored using a scoring function that obtains a score (fitness value) representing how good it solves the problem. Let $pbest_i = (x_{i1}^{pbest}, \dots, x_{in}^{pbest})$ and $gbest = (x_1^{gbest}, \dots, x_n^{gbest})$ be the position of individual *i* and its neighbours' best position so far, respectively. Each particle records its own personal best position (pbest), and knows the best positions found by all particles in the swarm (gbest). Then, all particles that fly over the *n*-dimensional solution space are subject to updated rules for new positions, until the global optimal position is found. The modified velocity and position of each individual can be calculated using the current velocity and the distance from $pbest_i$ to $gbest$ as follows [12]:

$$V_i^{k+1} = \omega V_i^k + c_1 R_{Rand_1}\left(pbest_i^k - X_i^k\right) + c_2 R_{Rand_2}\left(gbest^k - X_i^k\right) \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2)$$

where,

$V_i^k$ velocity of individual *i* at iteration *k*,

$\omega$ weigh parameter (inertia weight),

$c_1, c_2$ acceleration coefficients,

$R_{Rand_1}, R_{Rand_2}$ random numbers uniformly distributed between 0 and 1,

$X_i^k$ position of individual *i* at iteration *k*,

$pbest_i^k$ best position of individual *i* until iteration *k*,

$gbest^k$ best position of the group until iteration *k*.

Equation (1) indicates that the velocity of a particle is modified according to three components. The first component is its previous velocity, $V_i^k$, scaled by an inertia, $\omega$. This component is often known as "habitual behavior." The second component is a linear attraction toward its previous best position, $pbest_i^k$, scaled by the product of an acceleration constant, $c_1$, and a random number. Note that a different random number is assigned for each dimension. This component is often known as "memory" or "self-knowledge." The third component is a linear attraction toward the global best position, $gbest^k$ scaled by the product of an acceleration constant, $c_2$, and a random number. This component is often known as "team work" or "social knowledge."

Acceleration constants $c_1$ and $c_2$, personal and social learning factors, represent the weights of the stochastic acceleration terms that push a particle toward pbest and gbest, respectively. Small values allow a particle to roam far from target regions. Conversely, large values result in the abrupt movement of particles toward target regions. A usual choice for the accelaration coefficients $c_1$ and $c_2$ is usually $c_1$ equals to $c_2$ and range between 0 and 4. In this work, constants $c_1$ and $c_2$ are both set at 2, following the typical practice in Eberhart and Shi [13, 14].

Suitable selection of inertia weight provides a balance between global and local exploration, thus requiring less iteration on average to find a sufficiently optimal solution. In general, the inertia weight $\omega$ has a linearly decreasing dynamic parameter framework descending from $\omega_{max}$ to $\omega_{min}$ as follows

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iter_{max}}.Iter \quad (3)$$

where, $\omega_{max}$ and $\omega_{min}$ are the initial and final inertia weights, $Iter_{max}$ is maximum iteration number and $Iter$ is current iteration number [14-16].

The fundamental structure and pseudo-code of the PSO algorithm is given in Table.1

**Table 1:** Pseudo code of the PSO algorithm

```
for each particle
        generate an initial particle
end
do
        for each particle
                Calculate fitness value
                If the fitness value is better than the best fitness value (pbest) in history
                        set current value as the new pbest
                end
        end
        Choose the particle with the best fitness value of all the particles as the gbest
        for each particle
                Calculate particle velocity according equation (1)
                Update particle position according equation (2)
        end
while maximum iterations or minimum error criteria is not attained.
```

## 3. IMPLEMENTATION OF THE PSO TO PARAMETER ESTIMATION OF NONLINEAR REGRESSION MODEL

Since the sum of squared error function estimation is used in this study, in order to obtain a solution in the real parameter neighborhood, the sum of squared error function $S(\beta)$ should be minimized. Hence, the cost (fitness) function in the PSO search engine is selected as $S(\beta)$, specifically: $S(\beta) = \sum_{i=1}^{n}\left(y_i - f(x_i,\beta)\right)^2$. For instance, for *Meyer1* model in the Table 3, $S(\beta)$ is $\sum_{i=1}^{5}\left(y_i - \frac{\beta_1\beta_3 x_{1i}}{1+\beta_1 x_{1i}+\beta_2 x_{2i}}\right)^2$. In the *Meyer1* model, the observation number is $n = 5$.

The main parameters of the PSO method are $\omega, c_1, c_2$ and the swarm size. The settings of these parameters determine how it optimizes the search space. The role of the inertia weight $\omega$ is considered important for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. Thus, the parameter $\omega$ regulates the trade-off between the global and the local exploration abilities of the swarm. A proper value for the inertia weight $\omega$ provides balance between the global and local exploration ability of the swarm and thus results in better solution. If the $\omega \ll 1$, only little momentum is preserved from the previous time-step; thus quick changes of direction are possible with the setting. High settings near 1 facilitate global search, and lower settings in the range [0.2, 0.5] facilitate rapid local search [11]. Eberhart and Shi [17] have studied $\omega$ in several papers and found that an inertia-weight of 0.8 is a good choice. Many researchers have also applied an annealing scheme for the $\omega$ setting of the PSO, where $\omega$ decreases linearly from $\omega = 0.9$ to $\omega = 0.4$ over the whole run. In general, the inertia weight $\omega$ has a linearly decreasing dynamic parameter framework descending from $\omega_{max}$ to $\omega_{\min}$ as shown in equation (3). According to Das et al. [11], for inertia weight, $\omega_{max}$ and $\omega_{min}$ are 0.9 and 0.4, respectively produces satisfactory results and also taking $\omega_{min} = 0.4$ and $\omega_{max} = 0.9$ are appropriate choices for these parameter.

A usual choice for the accelaration coefficients $c_1$ and $c_2$, personal and social learning factors, is usually $c_1$ equals to $c_2$ and range between 0 and 4. Swarm size plays a very important role in the PSO, robustness and complexity of algorithm are also affected by it. Small population size may result in local convergence; large size will increase computational efforts and may make slow convergence. In this paper, swarm size is taken 20, 50 or 100 according to structure of the nonlinear models, searching space and number of estimated parameters.

Hence, the algorithm parameters $\omega_{max}$, $\omega_{min}$, $c_1$ and $c_2$ are selected as 0.9, 0.4, 2 and 2, respectively and for inertia weight, equation (3) is used. The termination criterion is selected as the iteration limit, specifically the algorithm is set to stop after 100 iterations and 50 independent experiments are conducted in order to check the robustness of the estimation strategy. Additionally, all algorithm evaluations are performed on standard commercial processing unit of *2.50 GHz Intel (R) Core (TM) i5-2520 M type CPU with 4.00 GB of RAM*. Moreover, the PSO implementation steps are given in Table 2.

**Table 2:** Pseudo code of the PSO implementation to parameter estimation of the nonlinear regression models

Initialize the PSO parameters, $\omega_{max}$, $\omega_{min}$, $c_1$, $c_2$ swarm size and define the iteration number

Take the values predictor variable $y$ and the values of explanatory variable(s) $x$

Calculate cost (fitness) of initial population, $S(\beta)$

*do*

*for* each particle

Calculate fitness value

*If* the fitness value is better than the best fitness value (pbest) in history

set current value as the new pbest

*end*

*end*

Choose the particle with the best fitness value of all the particles as the gbest

*for* each particle

Calculate particle velocity according equation

$$V_i^{k+1} = \omega V_i^k + c_1 R_{rand_1}\left(pbest_i^k - X_i^k\right) + c_2 R_{rand_2}(gbest^k - X_i^k)$$

Update particle position according equation

$$X_i^{k+1} = X_i^k + V_i^{k+1}$$

*end*

*while* (maximum iterations are not attained)

## 4. NUMERICAL COMPUTATIONS

In order to show the improvement in the nonlinear regression parameter estimation for the PSO algorithm, we have used well-known 28 nonlinear regression models whose list is given in Table 3. The data sets for the all models and the original data whose references are summarized in the supplementary data file. Data sets were taken Lanczos [18], Jennrich and Sampson [19], Meyer and Roth [20], Box et al. [21], Kowalik and Osborne [22], Daniel and Wood [23], Nelson [24], Ratkowsky [1], Kahaner et al. [25] and *NIST* data set collection. In the models, the number of parameters is ranging from 2 to 9 and the number of observations is ranging from 5 to 200. For instance, *Chwirut1* model has the 3 parameters and for this model, 214 observations are used.

The 1-3 columns of Table 3 show name of model, function of model, and the searching space, respectively. In the last four columns, optimal (true) and estimated parameter values are involved. $\hat{\beta}$ values indicate the estimated parameter values for the real $\beta$ parameters obtained by the PSO and $S(\hat{\beta})$ shows the estimated sum of squared error function value.

The estimate of parameters of these nonlinear regression models is a difficult task for classical algorithms of optimization. The starting values of parameters were chosen at random from searching spaces, for each model 50 independent attempts were performed.

In order to see how the PSO algorithm approaches to minimum of the sum of squared error function, and finally the estimations, their performances are illustrated on 28 test examples. From Table 3, for all the 28 nonlinear regression model examples, it can be observed that the ‘‘best’’ results obtained by the PSO are reasonably very close to the true parameter values, which demonstrates the high searching quality of the PSO.

In this study, comparing criteria are constructed on the principle of whether the technique provides a desired and suitable solution (a close estimation) or not. Additionally, since the convergence behaviors of the methods are observed that they are in a rapid convergence tendency, iteration number is limited to 100 iterations.

**Table 3:** List of nonlinear regression models and the results of the PSO algorithm

| ***Name*** | ***Regression Model*** | ***Searching Space*** | ***Parameters*** | | | |
|---|---|---|---|---|---|---|
| | | | ***Optimal*** | | ***Estimated*** | |
| *Meyer1* | $\frac{\beta_1\beta_3 x_1}{1+\beta_1 x_1+\beta_2 x_2}$ | 0 - 100 | $\beta_1$ | 3.131500 | $\hat{\beta}_1$ | 3.131500 |
| | | 0 - 100 | $\beta_2$ | 15.159000 | $\hat{\beta}_2$ | 15.159400 |
| | | 0 - 100 | $\beta_3$ | 0.780100 | $\hat{\beta}_3$ | 0.780063 |
| | | | | | $\hat{S}(\beta)$ | *0.000043553* |
| *Meyer4* | $\beta_3(\exp(-\beta_1 x_1)+\exp(-\beta_2 x_2))$ | 0 - 100 | $\beta_1$ | 13.241000 | $\hat{\beta}_1$ | 13.240900 |
| | | 0 - 100 | $\beta_2$ | 1.500700 | $\hat{\beta}_2$ | 1.500700 |
| | | 0 - 100 | $\beta_3$ | 20.10000 | $\hat{\beta}_3$ | 20.09900 |
| | | | | | $\hat{S}(\beta)$ | *0.000074712* |
| *Meyer5* | $\beta_3(\exp(-\beta_1 x_1)+\exp(-\beta_2 x_2))$ | 0 - 2000 | $\beta_1$ | 814.9700 | $\hat{\beta}_1$ | 906.7232 |
| | | 0 - 100 | $\beta_2$ | 1.5076 | $\hat{\beta}_2$ | 1.5076 |
| | | 0 - 100 | $\beta_3$ | 19.9200 | $\hat{\beta}_3$ | 19.9204 |
| | | | | | $\hat{S}(\beta)$ | *1.2519* |
| *Meyer7* | $\beta_1+\beta_2\exp(\beta_3 x)$ | 0 - 1000 | $\beta_1$ | 16.6730 | $\hat{\beta}_1$ | 16.0118 |
| | | 0 - 2 | $\beta_2$ | 0.9994 | $\hat{\beta}_2$ | 0.6999 |
| | | 0 - 2 | $\beta_3$ | 0.0222 | $\hat{\beta}_3$ | 0.0272 |
| | | | | | $\hat{S}(\beta)$ | *0.010941* |
| *Militky4* | $\beta_1\exp(\beta_3 x)+\beta_2\exp(\beta_4 x)$ | 0 - 1000 | $\beta_1$ | 1655.2 | $\hat{\beta}_1$ | 1408.5358292 |
| | | 0 - 1000 | $\beta_2$ | 3.4E+07 | $\hat{\beta}_2$ | 2.1423E+07 |
| | | -2 - 0 | $\beta_3$ | -0.6740 | $\hat{\beta}_3$ | -0.6710 |
| | | -5 - 0 | $\beta_4$ | -1.8160 | $\hat{\beta}_4$ | -1.7501 |
| | | | | | $\hat{S}(\beta)$ | *129.2111* |
| *Militky5* | $\beta_1 x^{\beta_2}+\beta_3^{\beta_2/x}$ | 0 - 5 | $\beta_1$ | 0.055890 | $\hat{\beta}_1$ | 0.055887 |
| | | 0 - 5 | $\beta_2$ | 3.548900 | $\hat{\beta}_2$ | 3.548900 |
| | | 0 - 5 | $\beta_3$ | 1.482200 | $\hat{\beta}_3$ | 1.482200 |
| | | | | | $\hat{S}(\beta)$ | *0.0043753* |
| *Gompertz* | $\beta_1\exp(-\exp(\beta_2-\beta_3 x))$ | 0 - 1000 | $\beta_1$ | 722.7500000 | $\hat{\beta}_1$ | 723.1086000 |
| | | 0 - 100 | $\beta_2$ | 2.5030000 | $\hat{\beta}_2$ | 2.5001840 |
| | | 0 - 100 | $\beta_3$ | 0.4510000 | $\hat{\beta}_3$ | 0.4501031 |
| | | | | | $\hat{S}(\beta)$ | *13606.1427* |
| *Logistic* | $\frac{\beta_1}{1+\exp(\beta_2-\beta_3 x)}$ | 0 - 1000 | $\beta_1$ | 702.9 | $\hat{\beta}_1$ | 702.8714 |
| | | 0 - 100 | $\beta_2$ | 4.443 | $\hat{\beta}_2$ | 4.4426 |
| | | 0 - 100 | $\beta_3$ | 0.689 | $\hat{\beta}_3$ | 0.6886 |
| | | | | | $\hat{S}(\beta)$ | *8929.883* |
| *Richards* | $\frac{\beta_1}{(1+\exp(\beta_2-\beta_3 x))^{1/\beta_4}}$ | 0 - 1000 | $\beta_1$ | 699.6 | $\hat{\beta}_1$ | 699.6484 |
| | | 0 - 10 | $\beta_2$ | 5.277 | $\hat{\beta}_2$ | 5.2765 |
| | | 0 - 10 | $\beta_3$ | 0.760 | $\hat{\beta}_3$ | 0.7596 |
| | | 0 - 10 | $\beta_4$ | 1.279 | $\hat{\beta}_4$ | 1.2790 |
| | | | | | $\hat{S}(\beta)$ | *8786.4051* |
| *Jennrich* | $\exp(\beta_1 x)+\exp(\beta_2 x)$ | 0 - 100 | $\beta_1$ | 0.2578 | $\hat{\beta}_1$ | 0.25783 |
| | | 0 - 100 | $\beta_2$ | 0.2578 | $\hat{\beta}_2$ | 0.25783 |
| | | | | | $\hat{S}(\beta)$ | *124.3622* |

**Table 3:** List of nonlinear regression models and the results of the PSO algorithm (continue)

| *Name* | *Regression Model* | *Searching Space* | *Parameters* | | | |
|---|---|---|---|---|---|---|
| | | | *Optimal* | | *Estimated* | |
| *Militkty2* | $\exp(\beta_1 x) + \exp(\beta_2 x)$ | -50 - 50 | $\beta_1$ | 0.2807 | $\hat{\beta}_1$ | 0.28067 |
| | | -50 - 50 | $\beta_2$ | 0.4064 | $\hat{\beta}_2$ | 0.40638 |
| | | | | | $\hat{S}(\beta)$ | *0.0088963* |
| *Ratkowsky2* | $\frac{\beta_1}{1 + \exp(\beta_2 - \beta_3 x)}$ | 0 - 100 | $\beta_1$ | 72.4622 | $\hat{\beta}_1$ | 72.4622 |
| | | 0 - 10 | $\beta_2$ | 2.6181 | $\hat{\beta}_2$ | 2.6181 |
| | | 0 - 10 | $\beta_3$ | 0.0674 | $\hat{\beta}_3$ | 0.0674 |
| | | | | | $\hat{S}(\beta)$ | *8.0565* |
| *Eckerle4* | $\frac{\beta_1}{\beta_2}\exp\left(\frac{-(x-\beta_3)^2}{2\beta_2^2}\right)$ | 0 - 1000 | $\beta_1$ | 1.5544 | $\hat{\beta}_1$ | 1.5544 |
| | | 0 - 1000 | $\beta_2$ | 4.0888 | $\hat{\beta}_2$ | 4.0888 |
| | | 0 - 1000 | $\beta_3$ | 451.5412 | $\hat{\beta}_3$ | 451.5412 |
| | | | | | $\hat{S}(\beta)$ | *0.0014636* |
| *Ratkowsky3* | $\frac{\beta_1}{(1 + \exp(\beta_2 - \beta_3 x))^{1/\beta_4}}$ | 0 - 1000 | $\beta_1$ | 699.6415 | $\hat{\beta}_1$ | 699.3725 |
| | | 0 - 10 | $\beta_2$ | 5.2771 | $\hat{\beta}_2$ | 5.3408 |
| | | 0 - 10 | $\beta_3$ | 0.7596 | $\hat{\beta}_3$ | 0.7655 |
| | | 0 - 10 | $\beta_4$ | 1.2792 | $\hat{\beta}_4$ | 1.2999 |
| | | | | | $\hat{S}(\beta)$ | *8787.1522* |
| *BoxBOD* | $\beta_1(1 - \exp(-\beta_2 x))$ | 0 - 1000 | $\beta_1$ | 213.8094 | $\hat{\beta}_1$ | 213.8094 |
| | | 0 - 100 | $\beta_2$ | 0.5472 | $\hat{\beta}_2$ | 0.5472 |
| | | | | | $\hat{S}(\beta)$ | *1168.0089* |
| *Thurber* | $\frac{\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3}{1 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3}$ | 0 - 1500 | $\beta_1$ | 1288.1397 | $\hat{\beta}_1$ | 1288.1098 |
| | | 0 - 1500 | $\beta_2$ | 1491.0793 | $\hat{\beta}_2$ | 1498.1622 |
| | | 0 - 1000 | $\beta_3$ | 583.2384 | $\hat{\beta}_3$ | 588.4335 |
| | | 0 - 100 | $\beta_4$ | 75.4166 | $\hat{\beta}_4$ | 76.4298 |
| | | 0 - 1 | $\beta_5$ | 0.9663 | $\hat{\beta}_5$ | 0.9717 |
| | | 0 - 1 | $\beta_6$ | 0.3980 | $\hat{\beta}_6$ | 0.4006 |
| | | 0 - 1 | $\beta_7$ | 0.0497 | $\hat{\beta}_7$ | 0.0508 |
| | | | | | $\hat{S}(\beta)$ | *5652.0481* |
| *MGH09* | $\frac{\beta_1(x^2 + x\beta_2)}{x^2 + x\beta_3 + \beta_4}$ | 0 - 1 | $\beta_1$ | 0.1928 | $\hat{\beta}_1$ | 0.19299 |
| | | 0 - 1 | $\beta_2$ | 0.1913 | $\hat{\beta}_2$ | 0.2147 |
| | | 0 - 1 | $\beta_3$ | 0.1231 | $\hat{\beta}_3$ | 0.13987 |
| | | 0 - 1 | $\beta_4$ | 0.1361 | $\hat{\beta}_4$ | 0.14535 |
| | | | | | $\hat{S}(\beta)$ | *0.00030946* |
| *ENSO* | $\beta_1 + \beta_2\cos\frac{2\pi x}{12} + \beta_3\sin\frac{2\pi x}{12} + \beta_5\cos\frac{2\pi x}{\beta_4} + \beta_6\sin\frac{2\pi x}{\beta_4} + \beta_8\cos\frac{2\pi x}{\beta_7} + \beta_9\sin\frac{2\pi x}{\beta_7}$ | 0 - 100 | $\beta_1$ | 10.5107 | $\hat{\beta}_1$ | 10.5107 |
| | | 0 - 100 | $\beta_2$ | 3.0762 | $\hat{\beta}_2$ | 3.0762 |
| | | 0 - 100 | $\beta_3$ | 0.5328 | $\hat{\beta}_3$ | 0.5328 |
| | | 0 - 100 | $\beta_4$ | 44.3111 | $\hat{\beta}_4$ | 44.3122 |
| | | -100 - 100 | $\beta_5$ | -1.6231 | $\hat{\beta}_5$ | -1.6226 |
| | | 0 - 100 | $\beta_6$ | 0.5255 | $\hat{\beta}_6$ | 0.5261 |
| | | 0 - 100 | $\beta_7$ | 26.8876 | $\hat{\beta}_7$ | 26.8886 |
| | | 0 - 100 | $\beta_8$ | 0.2123 | $\hat{\beta}_8$ | 0.2129 |
| | | 0 - 100 | $\beta_9$ | 1.4967 | $\hat{\beta}_9$ | 1.4964 |
| | | | | | $\hat{S}(\beta)$ | *788.5398* |

**Table 3:** List of nonlinear regression models and the results of the PSO algorithm (continue)

| ***Name*** | ***Regression Model*** | ***Searching Space*** | ***Parameters*** | | | |
|---|---|---|---|---|---|---|
| | | | ***Optimal*** | | ***Estimated*** | |
| *Roszman1* | $\beta_1 - \beta_2 x - \frac{\arctan\frac{\beta_3}{x-\beta_4}}{\pi}$ | 0 - 100 | $\beta_1$ | 0.2020 | $\hat{\beta}_1$ | 0.20198 |
| | | -100 - 100 | $\beta_2$ | -6.195E-06 | $\hat{\beta}_2$ | -6.197E-06 |
| | | 0 - $10^5$ | $\beta_3$ | 1204.4556 | $\hat{\beta}_3$ | 1204.4145 |
| | | -$10^4$ - $10^4$ | $\beta_4$ | -181.3427 | $\hat{\beta}_4$ | -181.3132 |
| | | | | | $\hat{S}(\beta)$ | *0.00049485* |
| *Misra1d* | $\frac{\beta_1\beta_2 x}{1+\beta_2 x}$ | 0 - 1000 | $\beta_1$ | 437.3737 | $\hat{\beta}_1$ | 437.3697 |
| | | -$10^3$ - $10^3$ | $\beta_2$ | 0.0003022732 | $\hat{\beta}_2$ | 0.000302273 |
| | | | | | $\hat{S}(\beta)$ | *0.056419* |
| *Misra1c* | $\beta_1\left(1-\frac{1}{\sqrt{1+2\beta_2 x}}\right)$ | 0 - 1000 | $\beta_1$ | 636.4273 | $\hat{\beta}_1$ | 636.4273 |
| | | 0 - 1000 | $\beta_2$ | 0.000208136 | $\hat{\beta}_2$ | 0.000208136 |
| | | | | | $\hat{S}(\beta)$ | *0.040967* |
| *Lanczos2* | $\beta_1\exp(-\beta_2 x)+\beta_3\exp(-\beta_4 x)+\beta_5\exp(-\beta_6 x)$ | 0 - 10 | $\beta_1$ | 0.09625103 | $\hat{\beta}_1$ | 0.89696 |
| | | 0 - 10 | $\beta_2$ | 1.00573329 | $\hat{\beta}_2$ | 4.98390 |
| | | 0 - 10 | $\beta_3$ | 0.86424690 | $\hat{\beta}_3$ | 0.98516 |
| | | 0 - 10 | $\beta_4$ | 3.00782839 | $\hat{\beta}_4$ | 2.57480 |
| | | 0 - 10 | $\beta_5$ | 1.55290169 | $\hat{\beta}_5$ | 0.63676 |
| | | 0 - 10 | $\beta_6$ | 5.00287981 | $\hat{\beta}_6$ | 5.76930 |
| | | | | | $\hat{S}(\beta)$ | *0.00036413* |
| *Nelson* | $\exp(\beta_1 - \beta_2 x_1\exp(-\beta_3 x_2))$ | 0 - 10 | $\beta_1$ | 2.5907 | $\hat{\beta}_1$ | 2.5907 |
| | | 0 - 1 | $\beta_2$ | 5.6178E-09 | $\hat{\beta}_2$ | 5.6178E-09 |
| | | -1 - 1 | $\beta_3$ | -0.0577 | $\hat{\beta}_3$ | -0.0577 |
| | | | | | $\hat{S}(\beta)$ | *3.7977* |
| *Misra1b* | $\beta_1\left(1-\frac{1}{\left(1+\frac{\beta_2 x}{2}\right)^2}\right)$ | 0 - 1000 | $\beta_1$ | 337.99746163 | $\hat{\beta}_1$ | 337.9975 |
| | | 0 - 1000 | $\beta_2$ | 0.0003903909 | $\hat{\beta}_2$ | 0.000390391 |
| | | | | | $\hat{S}(\beta)$ | *0.075465* |
| *DanWood* | $\beta_1 x^{\beta_2}$ | 0 - 1000 | $\beta_1$ | 0.76886226 | $\hat{\beta}_1$ | 0.76886 |
| | | 0 - 1000 | $\beta_2$ | 3.86040559 | $\hat{\beta}_2$ | 3.86040 |
| | | | | | $\hat{S}(\beta)$ | *0.0043173* |
| *Chwirut1* | $\frac{\exp(-\beta_1 x)}{\beta_2+\beta_3 x}$ | 0 - 1000 | $\beta_1$ | 0.19027 | $\hat{\beta}_1$ | 0.19028 |
| | | 0 - 1000 | $\beta_2$ | 0.0061314 | $\hat{\beta}_2$ | 0.0061314 |
| | | 0 - 1000 | $\beta_3$ | 0.010530 | $\hat{\beta}_3$ | 0.010531 |
| | | | | | $\hat{S}(\beta)$ | *2384.4771* |
| *Chwirut2* | $\frac{\exp(-\beta_1 x)}{\beta_2+\beta_3 x}$ | 0 - 1000 | $\beta_1$ | 0.16657 | $\hat{\beta}_1$ | 0.16658 |
| | | 0 - 1000 | $\beta_2$ | 0.0051653 | $\hat{\beta}_2$ | 0.0051653 |
| | | 0 - 1000 | $\beta_3$ | 0.012150 | $\hat{\beta}_3$ | 0.01215 |
| | | | | | $\hat{S}(\beta)$ | *513.048* |
| *Misra1a* | $\beta_1(1-\exp(-\beta_2 x))$ | 0 - 1000 | $\beta_1$ | 238.94212918 | $\hat{\beta}_1$ | 238.9421 |
| | | 0 - 10 | $\beta_2$ | 0.0005501564 | $\hat{\beta}_2$ | 0.000550156 |
| | | | | | $\hat{S}(\beta)$ | *0.12455* |

As the PSO algorithm is random in nature, the convergence behavior and final estimated values can be of interest. For *Meyer1* and *Thurber* nonlinear models, by way of example, Figure 1 and Figure 2 illustrate the error function behavior of the PSO approach which consists of the values that have been evaluated during the process of minimization, respectively. It is quite clear that the PSO algorithm converge after at most 20 iterations (especially for *Meyer4*, *Meyer5*, *Meyer7*, *Gompertz*, *Logistic*, *Richards*, *Militky2*, *Jennrich*, *Ratkowsky2*, *Ratkowsky3*, *BoxBoD*, *Rozsman1*, *ENSO*, *Lanczos2*, and *DanWood* models, in fact the convergence is realized only about 10 iterations) to their minimum sum of squared error value at the estimates of the parameters. For all models, figures are summarized in the supplementary figure file.

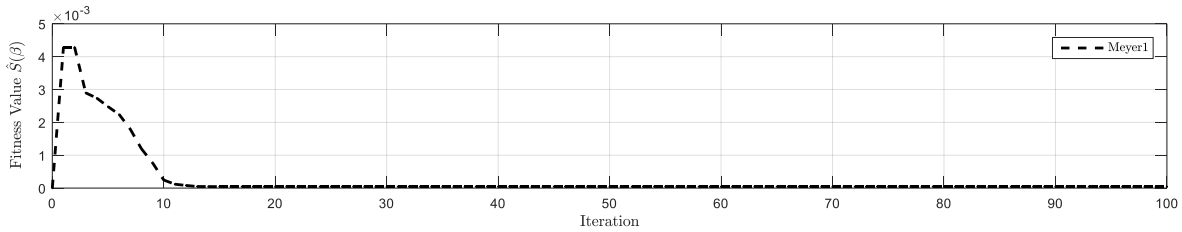


**Figure 1:** The sum of squared error function behavior of the PSO approach for *Meyer1* model

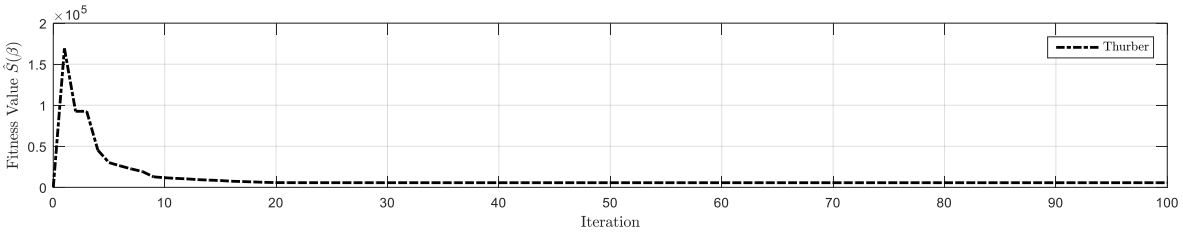


**Figure 2:** The sum of squared error function behavior of the PSO approach for *Thurber* model

If the Figures 1 and 2 are examined closely, the superiority variation of the method without any obvious decisive factor is due from the instantaneous values of the random initial population and random based operators of the evolutionary techniques.

As examples to estimation accuracy, for *Meyer1* model; for real parameter values (3.1315, 15.159, 0.7801), the PSO estimated values are (3.1315, 15.1594, 0.780063) and estimated error value is 0.000043553, for *Thurber* model; for real parameter values (1288.1397, 1491.0793, 583.2384, 75.4166, 0.9663, 0.3980, 0.0497), the PSO estimated values are (1288.1098, 1498.1622, 588.43352, 76.429786, 0.9717212, 0.4006428, 0.0507479) and estimated error value is 5652.0481. The obtained results are reasonably very close to the true parameter values.

Since *Jennrich* and *Militky2* models have the 2 parameters, detailed examination of the sum of squared error function of these models is provided by the Figures 3 and 4. They exhibit behavior of the particles for the *Jennrich* and *Militky2* model, respectively. From Figure 3, for *Jennrich* model, it is seen easily that the particles heavily intensify the (0.2578, 0.2578) point at the approximately 50th iteration. From Figure 4, for *Militky2* model, the particles heavily intensify the (0.28, 0.40) point at the approximately 50th iteration. For both models, it is easily seen that while particles are making several searches in the first iterations, they are taking the form of almost a single point in the last iterations and in the each iteration; the searches are diversified to reduce the impact of a local solution. It is noted that, the convergence is provided in the first few iterations for the *Jennrich* model, it is provided about 10 iterations for the *Militky2* model. Moreover, according to Table 3 and all Figures, it can be seen that the PSO estimates are very close to the real parameter values and it may be concluded that the PSO algorithm seems available and may be considered as an effective parameter estimation method for nonlinear regression models.

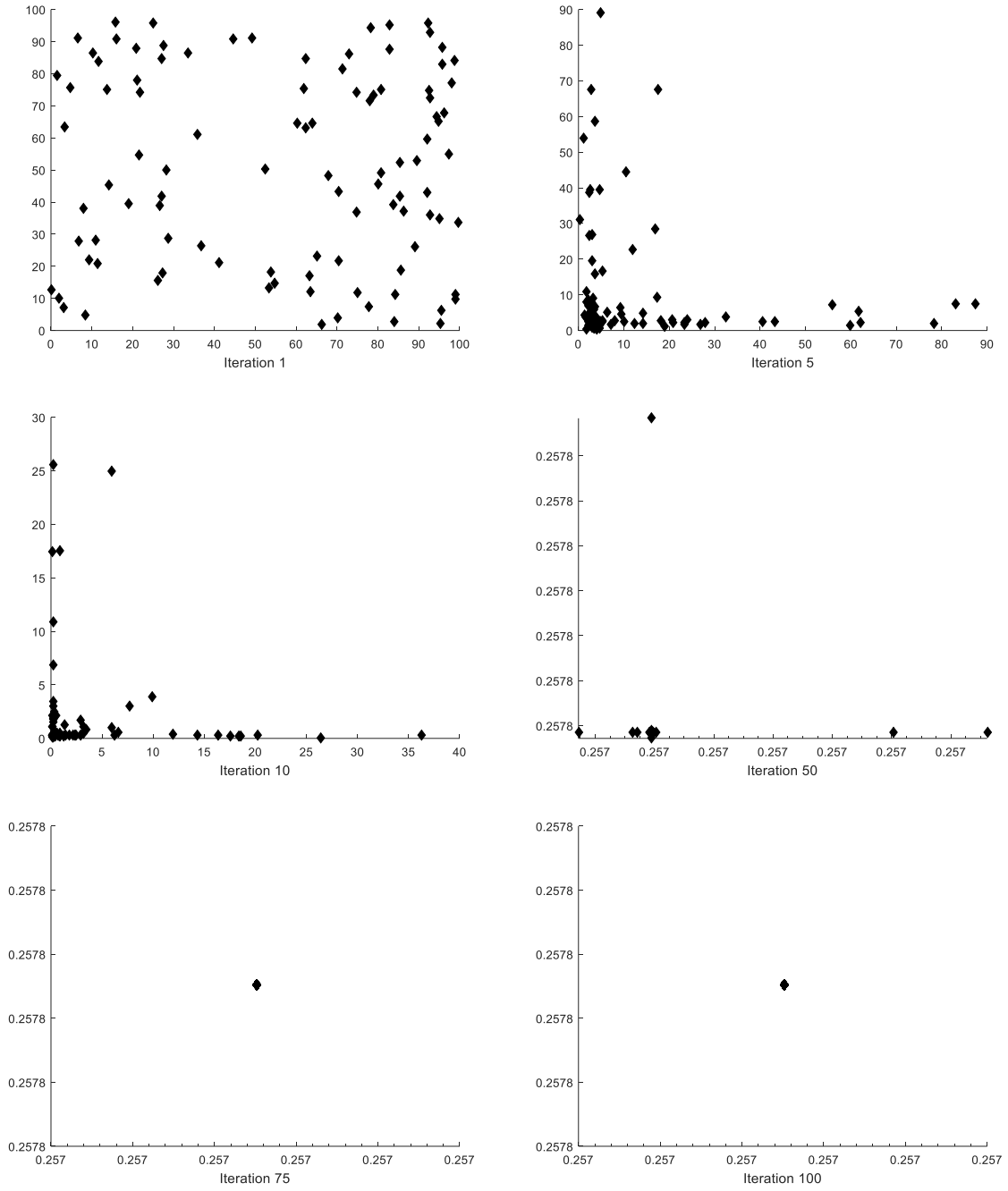


**Figure 3:** Scatter plot of particles in different iterations for *Jennrich* model

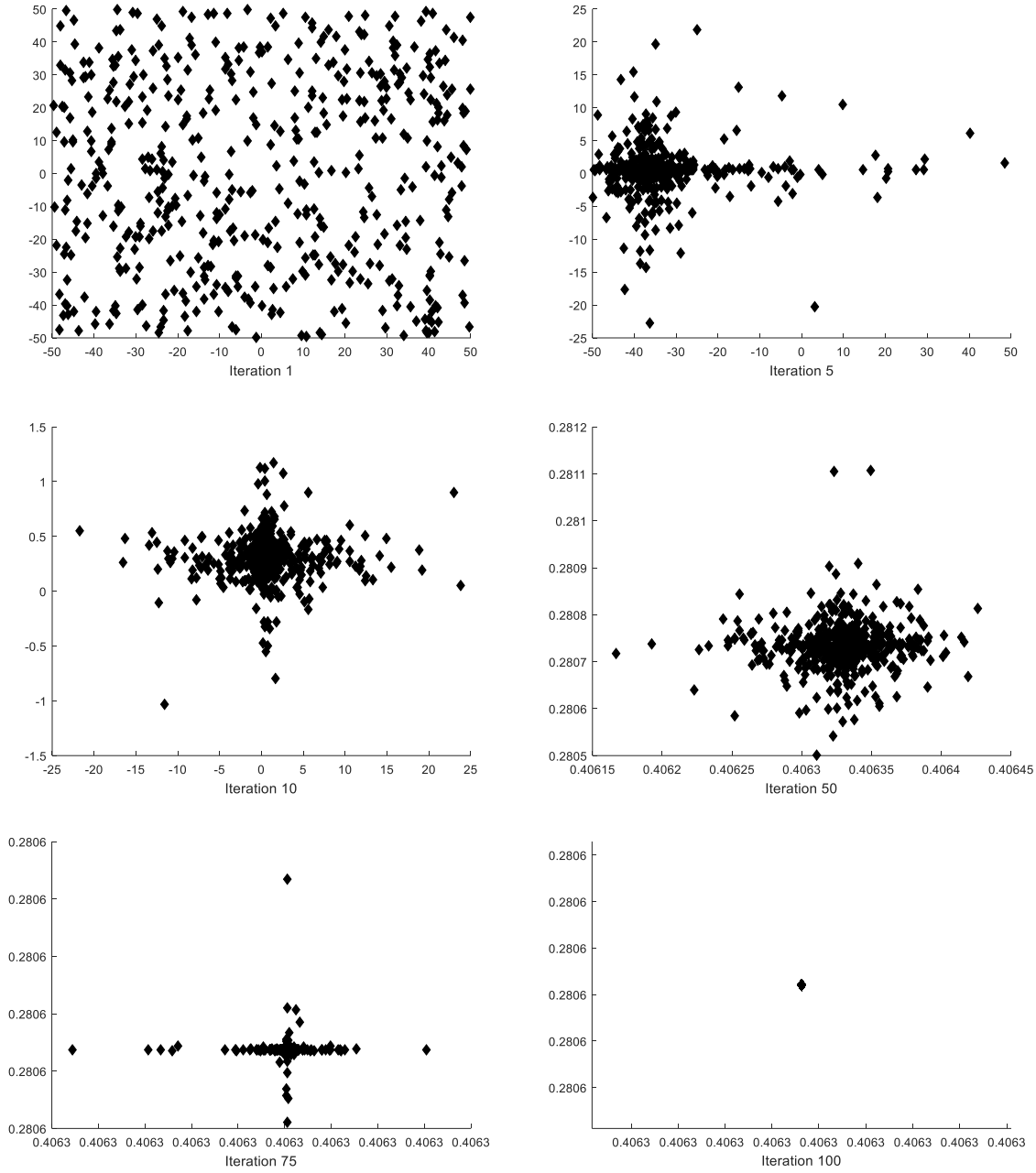


**Figure 4:** Scatter plot of particles in different iterations for *Militky2* model

## 5. CONCLUSIONS

The main aim of this article is to develop a reliable alternative parameter estimation approach based on the PSO algorithm in nonlinear regression model. When the PSO method is used for estimation of nonlinear regression model parameters, the approach presented here does not require any additional calculations to be performed. It is only necessary to select the points evaluated by the PSO. Also, it must be noted that the PSO algorithm exhibit a rapid convergence tendency, specifically the algorithm converged after at most 20-25 iterations for all the models which the number of parameters ranging from 2 to 9 and the number of observations ranging from 5 to 214. It can be concluded that all estimated values are in the neighborhood of the real parameter to be estimated. The numerical examples indicate that the PSO is the efficient method for handling the problems of parameter estimation of the nonlinear regression models.

## CONFLICT OF INTEREST

No conflict of interest was declared by the authors.

## REFERENCES


[1] Ratkowsky, D.A., Nonlinear regression modeling: a unified practical approach. Statistics: Textbooks and Monographs, Marcel Dekker, New York, (1983).

[2] Nash, J.C., & Walker-Smith, M., Nonlinear Parameter Estimation: An Integrated System in BASIC. Marcel Dekker, New York, (1987).

[3] Seber, G. A. F., & Wild, C. J., Nonlinear Regression, Wiley Series in Probability and Mathematical Statistics, Wiley, New York, (1989).

[4] Li, L.L., Wang, L., and Liu. L.H., "An effective hybrid PSOSA strategy for optimization and its application to parameter estimation", Applied Mathematics and Computation, 179(1): 135-146, (2006).

[5] Kennedy, J., and Eberhart, R., "Particle swarm optimization", Neural Networks, Proceedings., IEEE International Conference on Neural Networks (Path, Australia), IEEE Service Center, Piscataway, NJ, Vol. 4, 1942-1948, (1995).

[6] Eberhart, R., and Kennedy, J., "A New Optimizer Using Particle Swarm Theory", Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan. IEEE Service Center Piscataway NJ, 39-43, (1995).

[7] Křivý, I., Tvrdík, J., & Krpec, R., "Stochastic algorithms in nonlinear regression", *Computational Statistics & Data Analysis*, 33(3):277-290, (2000).

[8] Kapanoglu, M., Koc, I.O., & Erdogmus, S., "Genetic algorithms in parameter estimation for nonlinear regression models: an experimental approach", *Journal of Statistical Computation and Simulation*, *77*(10):851-867, (2007).

[9] Tvrdík, J., Křivý, I., & Mišík, L., "Adaptive population-based search: application to estimation of nonlinear regression parameters", *Computational Statistics & Data Analysis*, *52*(2):713-724, (2007).

[10] Aşıkgil, B., & Erar, A., "Polynomial tapered two-stage least squares method in nonlinear regression", *Applied Mathematics and Computation*, *219*(18):9743-9754, (2013).

[11] Das, S., Abraham, A., & Konar, A., "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives", Advances of Computational Intelligence in Industrial Systems, Studies in Computational Intelligence, Springer Verlag, Germany,1-38, (2008).

[12] Fukuyama, Y., "Fundamentals of particle swarm optimization techniques", Edited by K.Y.Lee and M.A. El-Sharkawi, Institute of Electrical and Electronics Engineers, *Modern heuristic optimization techniques: theory and applications to power systems*, 71-87, (2008).

[13] Eberhart, R. C., & Shi, Y., "Particle swarm optimization: developments, applications and resources", *Proceedings of IEEE International Congress on Evolutionary Computation*, 1, 81–86, (2001).

[14] Shi, Y., & Eberhart, R., "A modified particle swarm optimizer", *Proceedings of IEEE International Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 69-73, (1998).

[15] Lee, K. Y., & El-Sharkawi, M. A., "Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems", John Wiley & Sons, Inc., New Jersey, (2008).

[16] Lee, W. N., & Park, J. B., "Educational Simulator for Particle Swarm Optimization and Economic Dispatch Applications", MATLAB – A Ubiquitous Tool for the Practical Engineer, Edited by Clara M. Ionescu, INTECH Open Access Publisher, 81-110, (2011).

[17] Eberhart, R. C., & Shi, Y., "Comparing inertia weights and constriction factors in particle swarm optimization", *Proceedings of IEEE International Congress on Evolutionary Computation*, 1:84-88, (2000).

[18] Lanczos, C., Applied Analysis, Prentice-Hall, Old Tappan, N. J., 272-280, (1956).

[19] Jennrich, R. I., & Sampson, P. F., "Application of stepwise regression to non-linear estimation", *Technometrics*", *10*(1):63-72, (1968).

[20] Meyer, R. R., & Roth, P. M., "Modified damped least squares: an algorithm for non-linear estimation", *IMA Journal of Applied Mathematics*, *9*(2):218-233, (1972).

[21] Box, G. E., Hunter, W. G., & Hunter, J. S., "Statistics for Experimenters", Wiley, 483-487, New York, (1978).

[22] Kowalik, J. S., & Osborne, M. R., "Methods for Unconstrained Optimization Problems", Elsevier North-Holland, New York, (1968).

[23] Daniel, C., & Wood, F. S., "Fitting Equations to Data: Computer Analysis of Multifactor Data", John Wiley and Sons, New York, 428-431, (1980).

[24] Nelson, W., "Analysis of performance-degradation data from accelerated tests", *IEEE Transactions on Reliability*, *30*(2), 149-155, (1981).

[25] Kahaner, D., Moler, C., & Nash, S., "Numerical methods and software", *Englewood Cliffs: Prentice Hall*, 441-445, (1989).

[26] Kennedy, J. (1997, April). "The particle swarm: social adaptation of knowledge", IEEE International Conference on Evolutionary Computation, 303 -308, (1997).

[27] Shi, Y., & Eberhart, R. C. (1999). "Empirical study of particle swarm optimization", *Proceedings of IEEE Congress on Evolutionary Computation*, 1945 -1949, (1999).

[28] Shi, Y., & Eberhart, R. C., "Fuzzy adaptive particle swarm optimization", *Proceedings of IEEE Congress on Evolutionary Computation*,1:101-106, (2001).

[29] Hu, X., Shi, Y., & Eberhart, R. C., "Recent advances in particle swarm" , Proc. IEEE Congr. Evol. Comput., 1:90 -97, (2004).

[30] Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G., "Particle swarm optimization: basic concepts, variants and applications in power systems", IEEE Trans. Evol. Comput., 12(2):171-195, (2008).

[31] Coello, C. A. C., Pulido, G. T., & Lechuga, M. S., "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, *8*(3), 256-279, (2004).

[32] Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R., "Feature selection based on rough sets and particle swarm optimization", *Pattern Recognition Letters*, *28*(4), 459-471, (2007).