AN ALGORITHM FOR TIME SERIES PREDICTION USING PARTICLE SWARM OPTIMIZATION (PSO)

HAYDER M. ALBEHADILI, ABDURRAHMAN, AND N. E. ISLAM

1 ELECTRICAL ENGINEERING DEPARTMENT, UNIVERSITY OF MISSOURI, COLUMBIA, MISSOURI 65211 USA
2 NUCLEAR ENGINEERING DEPARTMENT, KING ABDUL AZIZ UNIVERSITY, JEDDAH, KSA 21432
3 CURRENTLY ON SABBATICAL, NUCLEAR ENGINEERING DEPARTMENT, KING ABDUL AZIZ UNIVERSITY, JEDDAH, KSA 21432

Abstract—Although Particle Swarm Optimization (PSO) is used in a variety of applications, it has limitations in the training phase. In this work, a new enhancement for PSO is proposed to overcome such limitations. The proposed PSO optimization consists of two stages. In the first stage, a Gaussian Maximum Likelihood (GML) is added to PSO to update the last 25% of swarm particles, while in the second stage, a Genetic Algorithm is applied whenever there is lethargy or no change in the fitness evaluation for two consecutive iterations. Finally, the proposed PSO is applied in time series predictions using Local Linear Wavelet Neural Network (LLWNN). The work is evaluated with three different data sets. Implementation of the proposed PSO shows better results than conventional PSO and many other hybrid PSOs proposed by others.

Key Words — PSO, GML, GA, and Time Series Prediction.

I. INTRODUCTION

Time-series prediction is an important research tool utilized in modeling problems in many areas, including decision-making [1], economic predictions [2], dynamic allocation [3], signal processing, and finance [4], solid isotropic material with penalization [5], hybrid wireless sensor networks coverage [6], and gold price forecasting [7]. The prediction algorithm initially used was based on Particle Swarm Optimization (PSO), but the recent trend in time-series prediction is a hybrid of different algorithms including PSO [8].

PSO based prediction for gold price forecasting was proposed by Esmaeil [7]. The method was superior to other algorithms available at that time and satisfied dramatic convergence. However, the analysis was limited to only 40 observations, 35 of them for training and the rest for testing. Also, it was pointed out by Jun Tang et al. [9], that the PSO algorithm usually suffers occurring in local minimum. This is because all particles of a swarm entirely depend on its history and the history of the best particle to update their positions. It is therefore possible for the best particle to attract the whole swarm to a local minimum, which is difficult to avoid. As a result of the local maximum, convergence begins to slow down and computations saturate in the final stages. The computations are effectively idle and reside in local optimum due to the fact that all particles exist in a local optimum.

To overcome the limitations of PSO algorithm, researchers introduced technique involving PSO and another algorithm to develop time-series prediction for a given situation. Huang, for example [1], presented a hybrid model for time-series prediction based on PSO and adaptive fuzzy time series and PSO, while Ernawati proposed combining PSO and similar sequence matching (SSM) algorithm [2]. Tang and Zhao [9] proposed hybrid PSO called LSPSO to solve falling in local minima. Zhang et al. used Local Linear Wavelet Neural Network (LLWNN) and perturbation technique like Simulated Annealing (SA) for their predictions. Gustavo H. T. Ribeiro et al. depicted Frankenstein’s Particle Swarm Optimization (FPSO) to find proper lags for time series forecasting [10]. In [1], Yao-Lin Hung et al. showed hybrid forecasting model based on adaptive fuzzy and PSO. H et al mentioned that PSO suffers premature convergence and they used Local Optima Avoidable PSO (LOAPSO) [11]. Other prominent models suggested include [12, 13, 14, 15].

Thus, in the realm of time-series predictions, the choice of an optimum combination of algorithms for the best prediction is an area of research interest to many. In this research we propose a technique to optimize time-series prediction and to avoid sluggishness in the computation process (like local minimum introduced in PSO algorithm). The proposed technique is then applied for training LLWNN. Specifically, we propose the addition of the Particle Swarm Optimization and Gaussian Maximum
Likelihood GA-GML in the solutions for training LLWNN. Two collaborative stages are used to introduce a robust algorithm. In the first stage, a Gaussian distribution and Maximum Likelihood (GML) and PSO are combined to implement the GML-PSO model. It will be shown that the GML-PSO hybridization will overcome the lethargy inherent by the PSO algorithm. However, the GML-PSO staggers to overcome local minima in evaluating some datasets, thus affecting computations. To counter this problem, we propose the addition of a Genetic algorithm (GA), which is applied to PSO whenever errors of two successive iterations are identical. GA is added to boost the information sharing and broadcasting among particles. The proposed GML-PSO-GA heterogeneous algorithm will utilize the strengths of both GML and GA to reproduce new individuals. Simulation results show that GA and GML play important role for error and epochs degradation and prevents particles’ stagnation. The GA maintains vital population because it uses crossover to exchange particles’ parameters with each other so that particles remain active along training period. Each particle has several chromosomes randomly aligned with another particle to reproduce new particles that inherit genetic traits from their ancestors. According to our knowledge, the hybrid GML-GA-PSO for time-series prediction and for training LLWNN has not been proposed earlier. Our work has been experimentally evaluated using various benchmarks and it is shown that GA-GML-PSO is superior compared with standard particle swarm optimization, and other techniques.

The rest of this paper is organized as follows. Section II depicts fast and brief review for PSO, GA, and LLWNN. Section III describes hybrid GML-PSO and how GML can use for training LLWNN with PSO and exhibits hybrid GA combined with GML-PSO obtained the limitations of applying GA to GA. In section IV, simulation setup is explained and an explanation for data sets. Section V explains results and discussions. Finally, in section VI, the conclusion is illustrated.

II. PROPOSED ALGORITHM

It has been pointed out by Mohd et al. [16] and Yi-Xiong Jin et al. [17] that the PSO alone always occurs in local optimum and cannot contribute significantly to time series optimization of a given problem. This is principally due to the fact that the PSO algorithm entirely depends on social behavior of swarm, where each particle in a swarm explores its new position according to its old best location and the location of the best particle in the swarm. The reason is inherent in the equation of updating positions and velocities of particles described in (1) and (2) respectively [9,11].

$$v_{k}^{l+1} = w^{l}v_{k}^{l} + c_{1}r_{1}(p_{best_{k}} - x_{k}^{l}) + c_{2}r_{2}(g_{best} - x_{k}^{l})$$  
(1)

$$x_{k}^{l+1} = x_{k}^{l} + v_{k}^{l}$$  
(2)

where, $v_{k}^{l}$ and $x_{k}^{l}$ represent velocity and position of a particle k at moment t, respectively; $c_{1}$ and $c_{2}$ are accelerators, $r_{1}$ and $r_{2}$ are random numbers between [0,1], pbest is the best position for a particle, gbest is the best particle for the whole swarm, T is the total number of iterations, and

$$w_{t} = \frac{(w_{final} - w_{init})t}{T} + w_{init}, \quad t = 1, 2, ..., T.$$  

The PSO has remarkable convergence in the initial stages, but it quickly traps to local optimum. In addition, PSO has difficulty to overcome local optimum if the search space contains only optimal solution [18]. Moreover, unlike conventional PSO, which requires a long time to reach the potent area, hybrid PSO provides fast and enhanced optimization [19, 20]. The PSO predominantly experiences premature convergence and search in region adjacent to global minimum as training progresses chronologically [17, 19] causing PSO to be permanently trapped in local optimum region.

The GA is an evolutionary algorithm used in solving problems in various fields [21, 22, 23, and 24]. It defines an initial generation that searches in domain space of the problem and generates a new population based mechanisms of reproduction, crossover, and mutation, which is frequently applied to produce new offspring. Usually, new descendants have higher quality and better fitness than ancestors. The GA contributes to enhancing PSO by merging particles in an intelligent approach, to produce new generations. Merging it with POS contributes in sharing information among particles, increasing the diversity of search space, allowing the training along vital computation steps, and preventing PSO to occur in local optimum. To maintain a smooth transition in PSO along computation steps, GA is applied to PSO whenever there are i) premature convergence, ii) no progressive in fitness function, or iii) it remains steady for two consecutive steps. GA’s crossover is applied to the particles as shown below

$$A^{X} = [a_{11}^{X}, a_{12}^{X}, ..., a_{1n}^{X}, a_{21}^{X}, a_{22}^{X}, ..., a_{2n}^{X}, ..., a_{m1}^{X}, a_{m2}^{X}, ..., a_{mn}^{X}]$$

$$A^{Y} = [a_{11}^{Y}, a_{12}^{Y}, ..., a_{1n}^{Y}, a_{21}^{Y}, a_{22}^{Y}, ..., a_{2n}^{Y}, ..., a_{m1}^{Y}, a_{m2}^{Y}, ..., a_{mn}^{Y}]$$
where \( A^x \) and \( A^y \) are dilations of two independent particles. In addition, we have also added a GML distribution, which massively affects PSO convergence because it has variety and explores new regions. The PDF is used to increase the diversity in the search scope [25]. In our approach, the Gaussian, shown in Fig. 1, distribution is used, which is one of the most important probability distributions for continuous variables [26]. Gaussian distribution is given by (3):

\[
\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} \tag{3}
\]

where, \( \mu \) and \( \sigma^2 \), which control the diversification search, is called mean and covariance.

### Likelihood function

Likelihood function is given by eqn. 4.

\[
p(x|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x|\mu, \sigma^2) \tag{4}
\]

In order to find unknown parameters \( \mu \) and \( \sigma^2 \), we need to maximize (4) by taking natural (log) for both sides leading to eqn.5 [26].

\[
\ln p(x|\mu, \sigma^2) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 = \frac{N}{2} \ln(2\pi)
\]

\[
\mathbf{\mu_{ML}} = \frac{1}{\text{total number of chromosomes}} \sum \text{chromosomes}
\]

where chromosomes: are wavelet translations. To find Gaussian’s covariance, the same formula has been used for wavelet dilations. The new spawned generation is descended from the best particles in the swarm and it gives chance to expand and diverse search space. Fig. 2 shows entire process for the proposed PSO. In our work, the PSO-GA-GML algorithm is implemented to train the LLWNN. LLWNN is chosen because it provides advantages over NN and WNN [6, 7], since it requires less neurons in hidden layer and has higher efficiency [6], requiring less training period. More details for LLWNN are available in [6, 7, and 25].

![Fig. 1. Gaussian distribution](image)

![Fig. 2. Procedures of training PSO with GA-GML](image)

### III. SIMULATION SETUP AND DATA SETS

To evaluate the proposed algorithm, three data sets are used in our simulations as benchmarks for time series forecasting. The first two, Mackey-Glass and Box-Jenkins, are detailed in [25]. Box-Jenkins consists of 296 samples. The first 200 samples are used for training LLWNN and the rest samples are used for testing the model. The second data set, Mackey-Glass described in (6), consists of 1000 observations half of them are used for training and the others for testing.

\[
\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^10(t-\tau)} - b \quad ; \quad \tau > 17 \tag{6}
\]

The third data set Santa Fe Competition was used in [27] to produce from 1 up to 25 steps ahead forecasting. We choose one-step ahead forecasting to compare results. There are various measurements used to evaluate time series forecasting [10]. Mean Square Error (MSE), Root Mean Square Error
(RMSE), and Mean Absolute Percentages Error (MAPE) are described in (7), (8), and (9) respectively [10].

\[
RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (T_j - Y_j)^2}
\]

\[
MAPE = \frac{1}{N} \sum_{j=1}^{N} \left| \frac{T_j - Y_j}{T_j} \right|
\]

where \( T_j \) is the expected value, \( Y_j \) is the LLWNN output, and \( N \) is the total number of samples. In these experiments, we use the same lag’s setting as [25], and the experiment parameters are described in table I.

### Table I. PSO parameters

<table>
<thead>
<tr>
<th>Parameter’s name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>2.05</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>2.05</td>
</tr>
<tr>
<td>( r_1 )</td>
<td>0.6</td>
</tr>
<tr>
<td>( r_2 )</td>
<td>0.5</td>
</tr>
<tr>
<td>( w_{\text{final}} )</td>
<td>0.89</td>
</tr>
<tr>
<td>( w_{\text{init}} )</td>
<td>0.45</td>
</tr>
<tr>
<td>( a_{\text{in}} )</td>
<td>Randomly initialized</td>
</tr>
<tr>
<td>( b_{\text{in}} )</td>
<td></td>
</tr>
<tr>
<td>( w_{\text{om}} )</td>
<td></td>
</tr>
</tbody>
</table>

The structure of the LLWNN for Mackey-Glass time series is 4-10-1, matched to [25], where there are four inputs in the input layer, 8 neurons in the hidden layer, and 1 input in the output. The structure of LLWNN for Box-Jenkins time series is 2-8-1, which is also identical to [25].

### I. RESULTS AND DISCUSSIONS

Table II shows data sets partitions, training and testing parts. In addition, it shows the number of iterations, in terms of Mackey-Glass data set; there are 1258 and 2000 iterations consumed by both PSO and GA-GML-PSO respectively. For Box-Jenkins data set, the number of iterations is approximately 2000 and 2600, respectively. It is obvious that there is dramatic difference between both tuning algorithms.

The experiments are contacted over 20 runs; the average of RMSE for both PSO and GA-GML-PSO is shown in Table III. From Table III, the results achieved by GA-GML-PSO approach are clearly superior to PSO. In Mackey-Glass data set, the best obtained result in terms of MAPE and RMSE are 0.0028 and 0.0032, respectively. However, the best results performed by PSO for the same data set in terms MAPE and RMSE are 0.0063 and 0.0101, respectively. The results of other data sets are given in Table III. Regarding Box-Jenkins data set, GA-GML-PSO surpasses PSO for both data sets. In Mackey-Glass data set and in terms of RMSE and MAPE, GA-GML-PSO achieves preferable results comparing with PSO. Fig. 3 shows the results of PSO and GA-GML-PSO approaches for all data sets Mackey-Glass, Box-Jenkins, and Santa Fe competition respectively.

To analyze the GA strength and obtain whether it consolidates decreasing error or not, we recorded the history of applying GA to PSO along training history as marked points shown in Fig.4. It is obvious that GA has essential role to minimize error along iterations period. As we can see in Fig.4, decreasing RMSE approximately saturates at iteration 158 with RMSE equal to 52.61 until it reaches iteration 297 with RMSE equal to 52.56 along this iterations there is no progress in dropping RMSE.

### Table II. Data sets’ samples for training and testing

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training samples</th>
<th>Testing samples</th>
<th>No. of iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GA-GML-PSO</td>
</tr>
<tr>
<td>Mackey-Glass</td>
<td>500</td>
<td>500</td>
<td>1258</td>
</tr>
<tr>
<td>Box-Jenkins</td>
<td>200</td>
<td>96</td>
<td>2000</td>
</tr>
<tr>
<td>Santa Fe</td>
<td>500</td>
<td>500</td>
<td>3000</td>
</tr>
</tbody>
</table>
Table III. Experimental Results of PSO and GA-GML-PSO.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>GA-GML-PSO</th>
<th></th>
<th></th>
<th>PSO</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>MSE</td>
<td>MAPE</td>
<td>RMSE</td>
<td>MSE</td>
<td>MAPE</td>
</tr>
<tr>
<td>Mackey-Glass</td>
<td>0.00324396</td>
<td>0.00524294</td>
<td>0.00283378</td>
<td>0.01013162</td>
<td>0.05132492</td>
<td>0.00632105</td>
</tr>
<tr>
<td>Box-Jenkins</td>
<td>0.00997412</td>
<td>0.93605479</td>
<td>9.87929e-04</td>
<td>0.01356950</td>
<td>1.08253025</td>
<td>0.00732132</td>
</tr>
<tr>
<td>Santa Fe</td>
<td>17.9986313</td>
<td>1.6197e+05</td>
<td>0.38602576</td>
<td>22.0006679</td>
<td>2.3051e+05</td>
<td>0.46416187</td>
</tr>
</tbody>
</table>

Fig. 3. The actual and predicted for all data sets (a, b) Mackey-Glass, (c, d) Box-Jenkins, and (e, f) Santa Fe, using GA-GML-PSO and PSO respectively.
However, after applying GA at iteration 297, which had consecutive iterations with same RMSE, RMSE eventually drops. There is drastically decreasing in RMSE between the period 297 and 330 with 52.56 and 17.14 respectively. Although applying GA at iteration 330, RMSE quickly jumps to 52.56 at iteration 338, but according to experiments that we have done in this work, RMSE always comes back to the previous lowest point after few iterations as is evident in 405 RMSE becomes 18.61 after approximately 70 iterations. Fig. 5 shows a second sample of applying GA to one of our data sets.

It is also evident that GA has been applied at specific point of training LLWNN using GA-GML-PSO approach. It is applied at iteration 116 with RMSE identical to 0.1139. It is evident that RMSE declines steeply from 0.1139 to 0.05036 for the period 116 to 123, respectively. In iteration 726, RMSE leaps to 0.066 but promptly reverts to 0.0089 roughly at 735.

According to the above two samples, GA has the ability to stimulate particles and deny them from lethargy and sluggishness. Furthermore, it increases search space by create new individuals with divers fitness and allows particles to encompass their information with each other.

I. CONCLUSION

Optimizing PSO alone results in the early occurrence in local optima which requires extended iterations to reach optimal solution. In addition, it is impossible for PSO to escape local minima. In this article, a new technique for PSO optimization is proposed to overcome these drawbacks. GML and GA boost PSO to overcome premature saturation and sluggishness. The two stages of optimization play a vital role in PSO progression during the training period; in the first stage, Gaussian Maximum Likelihood (GML) is added to PSO to update the last 25% of swarm particles which reduces errors significantly and speeds up the algorithm. In the second stage, Genetic Algorithm is added whenever there is no change in the fitness evaluation for two consecutive iterations. This reduces errors and keeps the particles vital for the duration of the training process. In effect, the proposed model surpasses the standard PSO in all selected benchmarks.

BIographies

Hayder M. Albehadili received B.Sc. and M.Sc. in computer engineering from university of Basra. He is a PhD candidate at electrical and computer engineering/University of Missouri-Columbia in the United States. His field of interest is high performance computing, algorithm acceleration, GPUs, Neural Networks, and Image Processing.
Naz E. Islam (M’96–SM’04) received the Ph.D. degree in Engineering Physics from the Department of Nuclear Engineering and Engineering Physics (NEER), Rensselaer Polytechnic Institute, Troy, NY, in 1992. His Ph.D. dissertation was the study of the effects of space and nuclear radiation on circuits and electronics. Following his postdoctoral work at NEER, he was a National Research Council Fellow with the Air Force Research Laboratory, Kirtland Air Force Base, Albuquerque, NM. In July 1997, he was with the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, as a Research Associate Professor. Since 2003, he has been a Faculty Member with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, as a LaPierre Professor. His research interests include the physics and technology of energetic charged particle’s interactions with semiconductor devices, dose rate and total dose effects, study of photonic devices used in the generation of high-power microwaves, and semiconductor-device simulation studies.

References


Christopher M. Bishop” Pattern Recognition and Machine Learning” 2009.

Mazlina Mamat and Salina Abdul Samad, “Comparison between Adaptive and Non-Adaptive HRBF Neural Network in Multiple