An ensemble multi-objective particle swarm optimization approach for exchange rates forecasting problem

Thi Thu Huong DINH
Sai Gon University
Ho Chi Minh, Vietnam
huongdtt2011@gmail.com

Van Truong VU
Le Quy Don Technical University
Ha Noi, Viet Nam
Truongvv@mta.edu.vn

Lam Thu BUI
Le Quy Don Technical University
Ha Noi, Viet Nam
Lam.Bui07@gmail.com

ABSTRACT
In this paper, the authors propose an ensemble multi-objective particle swarm optimisation approach (named EMPSO) for forecasting the currency exchange rate chain. The proposed algorithm consists of two main phases. The first phase uses a multi-objective particle swarm optimisation algorithm to find a set of the best optimal particles (named leaders). The second phase then uses these leaders to jointly calculate the final results by using the soft voting ensemble method. The two objective functions used here are predictive error and particle diversity. The empirical data used in this study are six different sets of currency exchange rates. Through comparison results with other evolutionary algorithms and other multi-objective PSO algorithms, the proposed algorithm shows that it can achieve better as well as more stability results on experimental data sets.

CCS Concepts
• Computing methodologies

Keywords
Time series forecasting; PSO; multi-objective PSO; ensemble learning.

1. INTRODUCTION
Currently, the prediction problem in general and the forecast of currency exchange rates, in particular, have become a popular topic in the field of machine learning and practical applications. Over many years of development, many algorithms have been developed to solve this problem. Which may include linear models such as Moving Average (ARMA), Autoregressive Integrated Moving Average Model (ARIMA) [1][2] or nonlinear models like ANNs [3], SVM [4], ... Some models combine many algorithms, ensemble learning models [5] developed to improve the accuracy of the model.

In general, among the machine learning algorithms that have been used to solve this problem, ANNs is the algorithm that has been proved to be suitable and most commonly used. In ANNs, weight plays the most crucial role. This weight is usually optimised through a learning algorithm, in which the backpropagation algorithm (BP) [6] is most commonly utilized. This algorithm has the advantage of fast convergence speed, but its weak point is that it is easy to get stuck at the local optimal locations. To overcome this weakness, one of the most common strategies is to combine BP with an evolutionary computation algorithm (EC). Because EC algorithms are characterized by population-approach, they are therefore capable of finding global extremes [7].

In the field of evolutionary computation, it is often divided into two main branches: the evolutionary algorithm and swarm intelligence. Some studies use evolutionary algorithms such as Genetic Algorithm (GA) [8], Differential Evolution (DE) [9], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10] for time series forecasting problems. In this study, the authors propose a swarm intelligence algorithm (e.g. multi-objective particle swarm optimisation) to optimise the weight of ANNs.

The main idea of the PSO algorithm is to simulate the movement of groups of animals (e.g. fish) in nature. PSO uses a group (called a swarm) of objects (called particles) to move in D-dimensional space. During this movement, the particles will have to determine the direction and magnitude of the velocity to focus on the global maximum. Velocity calculations play the most important role in PSO algorithms. In general, the speed of a particle is determined based on three factors: the current velocity ($V_{current}$), the best position that an individual find out from the beginning of its movement (named *localBest*) and the best location that either a swarm or a group of particles can find (named *globalBest*). The influence coefficients of these parameters are not the same, the degree of influence is represented through the coefficients W (with $V_{current}$), C1 (with *localBest*) and C2 (with *globalBest*). Depending on how these values are set, different versions of PSO can be created.

Originally the PSO algorithms used were single-objective algorithms. Then there are some extended versions of the PSO algorithm that solve multi-objective problems such as OMOPSO [11], SMPSO [12], DMOPSO [13]. The difference of multi-objective algorithms in comparison with the single-objective ones is the selection of best particles (named leaders). If in the single-objective version, the algorithms select an individual having the best results, in the multi-objective version, a set of equally good particles is selected instead of only one. Therefore, depending on the way to find out the leaders set and to choose which one is the *globalBest* will create different algorithm versions. In the OMOPSO algorithm, the authors used the ϵ-Dominance sorting to find the leader set (or *bestArchive*). The idea of selecting new particles is borrowed from the NSGA-II [14]. In which the author uses two criteria to choose. They are Pareto dominance and crowding distance. In which criterion 1 is preferred, criterion 2 is only used to remove particles when the number has exceeded the permitted size.

Currently, the research on the use of swarm intelligence in the forecasting problem mainly uses the single-objective PSO

ICMLSC 2020, January 17–19, 2020, Haiphong City, Viet Nam

ACM ISBN 978-1-4503-7631-0/20/01…$15.00
https://doi.org/10.1145/ 3380688.3380717

algorithm such as [15-17]. It is quite rare studies related to using a multi-objective approach to solve this problem [18]. In this paper, the authors will improve the OMOPSO algorithm with some changes in the velocity calculation step and *globalBest* selection mechanism for selecting a leader set. We will then use an ensemble learning method to make the outcome. Details of the algorithm will be described in the next sections.

The organisation of this paper is as follows: Section 2 presents the proposed method. Section 3 shows the experimental results. Finally, conclusions are presented in Section 4.

2. PROPOSED METHOD

The pseudocode is shown in Algorithm 1. The general idea of the algorithm is as follows: From the time series data, use a sliding window to make sample dataset. The size of the sliding window is a predetermined constant. The training dataset will be used as input to the proposed algorithm. To calculate predictive results, we use the neural networks (ANNs) as a base learner. The weight of each ANNs will be optimized through a swarm optimisation algorithm.

Algorithm 1: The ensemble multi-objective particle swarm optimisation algorithm (EMPSO)

```
Input: T: the training dataset
       N: the swarm size
       L: the length of a particle.
Output: The mean squared error (MSE) value
1:  /*Phase 1: Using the multi-objective
2:  particle swarm optimisation algorithm to
3:  get the best archive*/
4:  swarm ← createInitialSwarm(N)
5:  swarm ← evaluateSwarm(swarm,T)
6:  speed[N][L]← initializeVelocity(N,L)
7:  localBest[N]← initializeLocalBest(swarm)
8:  bestArchive[]← initializeLeader(swarm)
9:  // The update process of the swarm.
10: While (Stop condition) Do
11:     speed[N][L] ← updateVelocity(swarm);
12:     swarm ← updatePosition(swarm);
13:     swarm ← evaluateSwarm(swarm)
14:     bestArchive[]← updateLeaders(swarm)
15:     localBest[]← updateLocalBest(swarm)
16: End While
17: /*Phase 2: Using a soft voting ensemble
18: method to make a final result*/
19: Sort(bestArchive); //Ascending of MSE value
20: If(sizeOf(bestArchive)>3) then
21:  bestArchive←getFirstHalfOf(bestArchive);
22: End If
23: MSE ← SoftVoting(bestArchive,T)
24: Return MSE;
```

In general, there are two main phases:

Phase 1: Using the multi-objective PSO algorithm to find the best particles (or *bestArchive*).

Phase 2: The selected particles will work together to calculate the output value for each sample dataset through a soft voting mechanism.

Details of the steps will be described below:

2.1 Encoding

As mentioned above, in this study, we use ANNs as a base learner to calculate predictive results. In particular, the weight of ANNs will be optimized through the PSO algorithm. To do that, we encode each particle as a weight. Specifically, a particle is an array of real numbers representing the values of the weight. The order of each element in the array corresponding to each edge of the ANNs is shown in Figure 1.

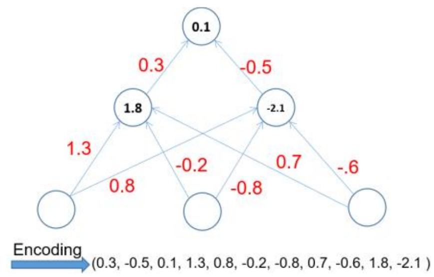

Figure 1. Encoding each particle as an ANN's weight matric.

2.2 Objective functions

In this study, the two objective functions we selected here are a predictive error (MSE) and particle diversity (DIV). Specifically, the calculation formulas for the two objectives are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(x_{real} - x_{output}\right)^2 \tag{1}$$

Where: n - the number of data samples; x_{real}- real value; x_{output}- output value

$$DIV_i = \frac{1}{(N-1)*\text{delta}}\left(1 - \sum_{j=1, j\neq i}^{N} d(i,j)\right) \tag{2}$$

Where N is swarm size; d(i,j)> 0 (Eq.3) is the distance between the particle$_i$ and the particle$_j$; delta (Eq.4) is a distance between the particles having the maximum and minimum of MSE values.

$$d(i,j) = |\,\text{MSE}_i - MSE_j\,| \tag{3}$$

$$delta = |\,\text{MSE}_{max} - MSE_{min}\,| \tag{4}$$

2.3 Phase 1: Using the multi-objective particle swarm optimisation algorithm to get the best archive.

Phase 1 of the algorithm is shown from *Line 4* to *Line 17* in Algorithm 1. The proposed algorithm is developed based on the OMOPSO algorithm [11]. Some of the changes will be covered in the sections below.

In general, this phase can be divided into two main steps: initialising the swarm and the movement process of the swarm.

a. Initialisation

The first step is to randomly initialize the position of each particle. In this study, we randomly generated an array of real values in the range [-0.5, 0.5] for each particle. After determining the location, evaluate the fitness value (or objective functions) of each particle. One difference is that the objective function 2 (DIV) is dependent on the objective function 1 (MSE). Therefore, it is necessary to calculate the objective function 1 for all particles before computing the objective function 2 for these particles.

Next, the velocity of each particle (*speed*) is initially set to 0. The *localBest* array contains the best positions of each particle that it finds in motion. Initially, these best values are assigned by the original location of each particle. Another procedure that is also important is the initialisation of leaders (*bestArchive*). Based on the distribution in the initial space of each particle, the list of leaders will be calculated and selected particles that meet both criteria *Pareto dominance* and *Crowding distance*.

b. The update process

This loop is a process of updating the following four parameters: velocity, position, local best and leaders (or *bestArchive*).

In which, the process of updating location and local best is done like the regular PSO versions; The Leaders update procedure we use is the same as in the OMOPSO algorithm. The difference in the proposed algorithm lies in the speed update step, which is the most essential step of any swarm optimisation algorithm.

The usual speed update formula is as follows:

$$\begin{aligned} v_i(t+1) &= \mathrm{W} * v_i(t) + \\ & rand(0,1) * \mathrm{C}_1 * (\mathrm{localBest}_i - x_i(t)) + \\ & rand(0,1) * \mathrm{C}_2 * (\mathrm{globalBest}_g - x_i(t)) \end{aligned} \tag{5}$$

Where W is an inertia constant that is important in balancing exploration and exploitation or between global and local search. C1, C2 are acceleration factors. These two parameters control the displacement distance.

In the OMOPSO algorithm, the parameters C1 and C2 are randomly generated within [1.5, 2.0]. The parameter W is a random value in the range [0.1, 0.5]. In this study, we calculate W according to an algorithm in [19]:

$$\mathrm{w} = 0.9 - \frac{0.5}{iter_{\max}} * \mathrm{iter}_{current} \tag{6}$$

Where $iter_{current}$ is the current loop index; $iter_{max}$ is is the total number of iterations.

One point to note here is how to choose a global best value. OMOPSO uses a binary tournament mechanism, which randomly selects two individuals in the *bestArchive* array to compare based on crowding distance. The winner will be chosen as the global best. In our study, two selected objective functions are MSE and DIV. In which we determine MSE is more critical than DIV. DIV is only a factor that helps prevent premature particles from converging to local extremes. We have therefore modified the criteria to select global best, which is based on MSE instead of diversity.

2.4 Phase 2: Using a soft voting ensemble method to make a final result.

After the update process has ended, a list of the best particles found and stored in the *bestArchive* array. The question now is which particle will be selected to calculate the result. It is usually not easy to find a single particle that gives excellent and stable results with many different data sets. Because each individual is only best suited to certain types of data, this is the reason why we choose the ensemble learning solution. That is, we will select a set of particles to make a decision instead of just a single particle.

Algorithm 2: SoftVoting

```
Input: T: the training dataset
       bestArchive: the leaders archive.
Output: the MSE value
1:   MSE =0.0;
2:   Foreach Ti in T Do
3:       Outputi=0;
4:       Foreach Pi in bestArchive Do
5:           ANNi = new ANN(Pi).
6:           OutPuti += ANNi.ComputeOutputs(Ti);
7:       EndFor
8:       OutPuti = OutPuti/SizeOf(bestArchive)
9:       MSE+= SquareError(OutPuti,Ti)
10:  EndFor
11:  MSE = MSE/SizeOf(T);
12:  Return MSE;
```

However, one thing to note here, if we select all the particles in the *bestArchive* list, the size can reach N (swarm size). The calculation will take a lot of time. Therefore, we only select ½ particles from that list (when the number of particles is greater than or equal to 4). However, as mentioned earlier, MSE is still our priority. Therefore, the *bestArchive* list will be sorted in ascending order by MSE and the first half of *bestArchive* will be selected.

The ensemble learning method we use here is soft voting. The idea of this method is quite simple. From a selected particle we will have a corresponding ANNs. For each sample data in the training dataset, each ANNs will produce a result, and the final result will be averaged from these values. Details of this step are shown in Algorithm 2.

3. EXPERIMENTAL RESULTS

3.1 Data description

Experimental data includes 6 different currency rates which are converted into AUD (Australian dollar). They are CAD (Canadian Dollar); EUR (Euro); GBP (Pound sterling); HKD (Hong Kong Dollar); USD (United States Dollar); VND (Vietnamese Dong). We collect 900 data for each currency exchange rate, from 07/06/2017 to 23/11/2019 on the website http://fx-rate.net/.

The data is divided into two sets of train data and test data with the ratio of 80:20 respectively.

3.2 Parameter setting

The value of the algorithm parameters is shown in Table II.

TABLE II. THE PARAMETERS SETTING

Method	Parameters	Value
ANN	Input number	5
	Output number	1
	Number hidden nodes	4
EMPSO	Swarm size	100
	Probability of mutation	1/(Particle length)
	Max Iterations	250
	Eta	0.0075

3.3 Test scenarios

To evaluate the performance of the proposed algorithm, we conducted empirical scenarios on the test data set as follows:

Scenario 1: Performance comparison with single-objective PSO.

To see the effectiveness of the multi-objective approach, we compared the proposed algorithm with the single-objective version. The result is shown in Figure 2.

It is easy to see that EMPSO is superior to the PSO algorithm in all test data sets. The smallest difference between these two methods is in the case of GBP; the figures for the two algorithms are 7.83E-04 and 2.07E-03 respectively. Other currencies have very different results, especially VND when EMPSO is nearly ten times better. Data for EMPSO and PSO are 1.3E-03 and 9.37E-02 respectively. Through this experiment, we can see that the effect of multi-objective algorithm is much better than using only one objective function.

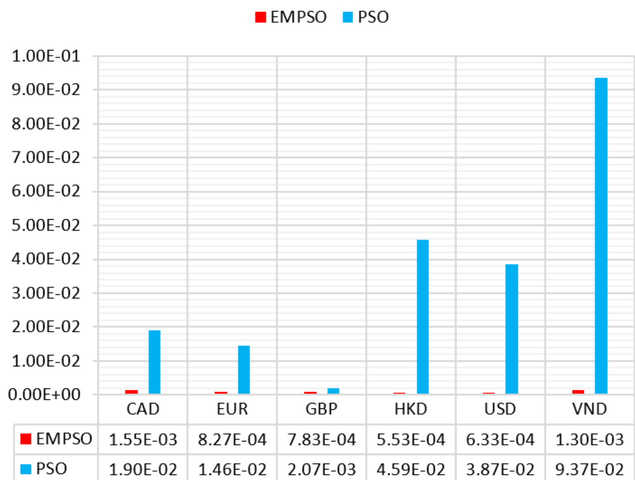

Figure 2. Performance comparisons between the multi-objective PSO (EMPSO) and single-objective one (PSO).

Scenario 2: Performance comparison with other evolutionary algorithms.

To better understand the performance of the proposed algorithm, we compare it with EAs algorithms, including both single-objective (GA, DE) and multi-objective algorithms (NSGA-II). Experimental results are shown in Figure 3.

From the results, it can be seen that the GA algorithm gives the worst results in all cases. DE algorithm and NSGA-II algorithm provide relatively similar results. While in the CAD and GBP data sets, NSGA-II gave better results, in the remaining data sets DE was better than NSGA-II. With the proposed algorithm, the results on the data sets are better than other algorithms, especially in the EUR, GBP, HKD and USD sets. Through this result, we can see that compared with NSGA-II multi-objective algorithm, the proposed algorithm always proved to be superior. DE has been shown to have a high-speed convergence. However, it can be seen in this experiment when considering only one criterion (MSE). This algorithm will not give better results in comparison with the proposed algorithm. The reason is that when using only one objective, individuals tend to converge quickly into the same region and in some cases, this region is far away from the global extreme zone or maybe local extreme regions. By using an additional DIV target function to always ensure a certain distance between the parties. The proposed algorithm has somewhat avoided early convergence and thereby making the particles more accessible to global extremes.

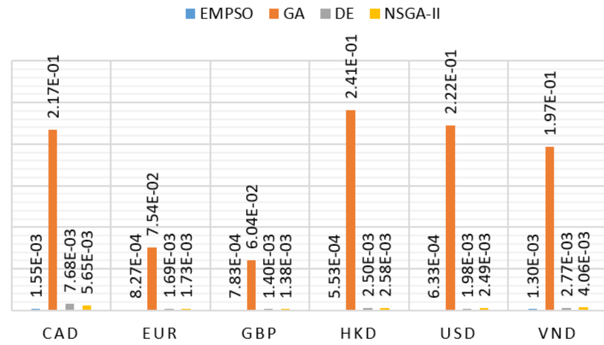

Figure 3. Performance comparisons between the proposed method (EMPSO) with some evolutionary algorithms.

Scenario 3: Performance comparison with the original version (OMOPSO).

In this experiment, we want to see how effective the changes on the original algorithm (OMOPSO) are. We compare the proposed algorithm with the OMOPSO algorithm. Experimental results are shown in Figure 4.

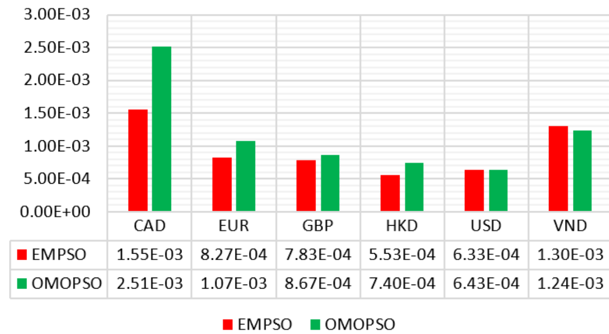

Figure 4. Performance comparisons between the proposed method (EMPSO) with original algorithm version.

Similar to the results of the previous experiments, the proposed algorithm produces better results than the OMOPSO algorithm in all cases. Especially with the EUR data set, the difference between the two methods is more pronounced. The figures for the proposed method and the OMOPSO are 8.27E-04 and 1.07E-03, respectively. Through this experiment, it can be seen that with the change of W coefficient as well as the selection mechanism of Global best also helps the algorithm to give better and more stable results.

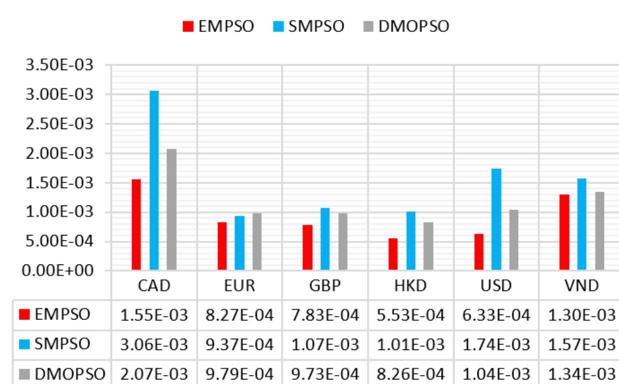

Figure 5. Performance comparisons between the proposed method (EMPSO) with other multi-objective PSO algorithms.

Scenario 4: Performance comparison with other multi-objective PSO algorithms.

The purpose of this scenario is to test how the proposed model compares with other PSO multi-objective models. To do this, we set up the objective functions and ensemble calculation into SMPSO and DMOPSO algorithms like the proposed algorithm. Experimental results are illustrated in Figure 5.

One easily observed point is that the experimental data sets of the SMPSO algorithm give the worst results, followed by the DMOPSO algorithm and finally, the proposed algorithm EMPSO. With the VND data set, the difference between the three algorithms is not much; the corresponding figures for EMPSO, SMPSO and DMOPSO are 1.3E-03, 1.57E-03 and 1.34E-03, respectively. With the data sets of EUR, CAD, GBP, the difference is similar. With the HKD data set, the difference becomes more pronounced when SMPSO produces worse results than the other two algorithms. The most significant difference lies in the USD data set. While EMPSO gives the best results, at 6.33E-04, the different two algorithms produced worse results at 1.74E-03 and 1.04E-03, respectively. These results show the advantages of the proposed algorithm.

4. CONCLUSION

In this paper, the authors have presented an ensemble multi-objective PSO. In which, we introduced the idea of designing the swarm optimisation algorithm for currency forecasting problem. We also show the steps of the algorithm, its changes, improvements and differences from the original algorithm. In the empirical part, we have conducted four scenarios comparing the proposed algorithm with single-objective, multi-objective evolutionary and swarm optimisation algorithms. Experimental results on six sets of real datasets show that the proposed algorithm gives better results with high stability. This paper is the initial study in our series of multi-objective swarm optimisation research. In subsequent studies, we will focus on building a co-evolution model among multiple swarms to achieve even more impressive results.

5. ACKNOWLEDGMENTS

This research is funded by Sai Gon university under the project "*Design forecasting model based on the principle of swarm intelligence and applications*" (Grant no. 900/HĐ-ĐHSG).

6. REFERENCES

[1] Xiong, Li, and Yue Lu. "Hybrid ARIMA-BPNN model for time series prediction of the Chinese stock market." In *2017 3rd International Conference on Information Management (ICIM)*, pp. 93-97. IEEE, 2017.

[2] Grigonytė, Ernesta, and Eglė Butkevičiūtė. "Short-term wind speed forecasting using ARIMA model." *Energetika* 62, no. 1-2 (2016).

[3] P.G. Harrald, M. Kamstra, Evolving artificial neural networks to combine financial forecasts, IEEE Trans. Evolut. Comput. 1 (1997) 40–52.

[4] Kim, Kyoung-jae. "Financial time series forecasting using support vector machines." *Neurocomputing* 55, no. 1-2 (2003): 307-319.

[5] Lam Thu BUI, Van Truong VU, Thi Thu Huong DINH. "A novel evolutionary multi-objective ensemble learning approach for forecasting currency exchange rates." *Data & Knowledge Engineering* 114 (2018): 40-66.

[6] D. Rumelhart, G. Hinton, R. Williams, Learning internal representation by backpropagating errors, in: D.E. Rumelhart, J.L. McCleland (Eds.), Parallel.

[7] Weise, Thomas. "Global optimization algorithms-theory and application." *Self-Published Thomas Weise* (2009).

[8] Szpiro, George G. "Forecasting chaotic time series with genetic algorithms." *Physical Review E* 55, no. 3 (1997): 2557.

[9] Donate, Juan Peralta, Xiaodong Li, Germán Gutiérrez Sánchez, and Araceli Sanchis de Miguel. "Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm." *Neural Computing and Applications* 22, no. 1 (2013): 11-20.

[10] Thi Thu Huong Dinh, Van Truong VU, Lam Thu BUI , A multi-objective ensemble learning approach based on the non-dominated sorting differential evolution for forecasting currency exchange rates, in: Proceedings of the Knowledge and Systems Engineering (KSE), 2016 Eighth International Conference on, 2016, pp. 96–102.

[11] Sierra, Margarita Reyes, and Carlos A. Coello Coello. "Improving PSO-based multi-objective optimization using crowding, mutation and∈-dominance." In *International conference on evolutionary multi-criterion optimization*, pp. 505-519. Springer, Berlin, Heidelberg, 2005.

[12] Nebro, Antonio J., Juan José Durillo, Jose Garcia-Nieto, CA Coello Coello, Francisco Luna, and Enrique Alba. "SMPSO: A new PSO-based metaheuristic for multi-objective optimization." In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pp. 66-73. IEEE, 2009.

[13] Zapotecas Martínez, Saúl, and Carlos A. Coello Coello. "A multi-objective particle swarm optimizer based on decomposition." In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 69-76. ACM, 2011.

[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2002) 182–197.

[15] Mahnam, Mehdi, and GHOMI SMT FATEMI. "A particle swarm optimization algorithm for forecasting based on time variant fuzzy time series." (2012): 269-276.

[16] Adhikari, Ratnadip, and Rajiv Kumar Agrawal. "Effectiveness of PSO Based Neural Network for Seasonal Time Series Forecasting." In *IICAI*, vol. 3, pp. 231-244. 2011.

[17] Jha, Girish K., Parimala Thulasiraman, and Ruppa K. Thulasiram. "PSO based neural network for time series forecasting." In *2009 International Joint Conference on Neural Networks*, pp. 1422-1427. IEEE, 2009.

[18] Taormina, Riccardo, and Kwok-wing Chau. "Neural network river forecasting with multi-objective fully informed particle swarm optimization." *Journal of Hydroinformatics* 17, no. 1 (2014): 99-113.

[19] Shi, Yuhui, and Russell Eberhart. "A modified particle swarm optimizer." In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), pp. 69-73. IEEE, 1998.