Exchange Rate Forecasting with Combined Genetic Algorithms

Jui-Fang Chang
Department of International Business
National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan
rose@cc.kuas.edu.tw,

Abstract

In recent years, Artificial Intelligence (AI) methods have proven to be successful tools for forecasting in the sectors of business, finance, medical science and engineering. In this study, we utilize a Genetic Algorithm (GA) to select the optimal variable weights in order to predict exchange rates using Genetic Algorithms, Particle Swam Optimization (PSO) and Back Propagation Network (BPN). Then, we construct three models: GA_GA, GA_PSO, and GA_BPN. Fundamentally, we expect enhanced variable selection to provide improved forecasting performance. The results of our experiments indicate that the GA_GA model achieves the best forecasting performance and is highly consistent with the actual data.

Keywords: Genetic Algorithm (GA), Particle Swam Optimization (PSO), Back Propagation Network (BPN)

1. Introduction

Managers and investors rely on exchange rates as indicators of international market forces, as well as monetary policies of individual countries. However, accurate forecasting is difficult; moreover, many nations have experienced dramatic changes in the exchange values of their currencies (Branson, 1997) [1].

Nelly and Weller (2002) [2] adopted exchange rates such as US/DM and US/JPY with GA, GARCH and Risk Metrics model to predict the volatility of foreign currency markets. The results show that GA has better performance than GARCH and Risk Metrics, based on standard criteria of performance evaluation, mean square error (MSE), mean absolute error (MAE) and R^2. Brandl et al. (2006) [3] used the genetic algorithm methodology to select explanatory variables used in dynamic regression models to generate the forecasts. Using genetic algorithms in the variable selection process is new and overcomes typical problems associated with traditional variable selection procedures. They do not need assumptions about the kind of explanatory variables, the number of explanatory variables, and the length of time series.

Recently, the artificial neural network (ANN) has been successfully applied to forecasting time series in many areas such as business, finance, medical science, engineering, etc. ANN works by imitating biological neural networks; moreover, the Back Propagation Network (BPN) is one of the most representative ANNs. In 1997, Rauscher [4] used the long-term equilibrium function and macroeconomic factors of the Monetary School as input variables of BPN and produced better experimental results than the error correction model (ECM) in predicting exchange rates. Ashok and Amit (2002) [5] state that a solution obtained through standard ANN algorithms like BPN suffers from serious drawbacks regarding neuron selection. A hybrid artificial intelligence method was therefore adopted and combined with the GA approach. The

results indicate superior performance of the proposed method compared to the traditional non-linear time series techniques and also fixed-geometry ANN models.
Natarajan, Saravanan & Periasamy (2006)[6] claimed that the estimation of the tool life generally requires considerable time and material; hence, it is a relatively expensive procedure. Using PSO and BPN to train a neural network, their study compared PSO with BPN. Their results showed that PSO outperforms BPN. Geethanjali, Slochanal & Bhavani (2008)[7] proposed model monitors and discriminates among the different operating conditions of power transformers. PSO technique is used to train the multi-layered feed forward neural networks to discriminate different operating conditions. These two ANNs were trained using back propagation neural network algorithm (BPN) and the PSO technique; next, the simulated results are compared. A comparison of the simulated results of the above two cases indicates that training the neural network by PSO technique gives more accurate (in terms of sum square error) and also faster (in terms of number of iterations and simulation time) results than BPN. The PSO trained ANN-based differential protection scheme provides faster, more accurate, more secure and reliable results for power transformers.
Some research has taken account of the macroeconomic factors to forecast exchange rates by some given models that include the Purchasing Power Parity (Frenkel, 1981 [8]), Monetary model (MacDonald and Taylor, 1994 [9]), Interest Rate Parity, Balance of Payment, Portfolio Balance Model (Branson, 1997 [1]; Bisignano and Hoover, 1982 [10]).
In this paper, hybrid models GA_GA, GA-PSO, and GA_PBN have been separately utilized to forecast monthly foreign NT dollar/dollar exchange rates. The GA model is first adopted to provide the optimal variable weights. After selecting the optimal set of variables as the input factors, we use a GA, PSO and BPN model to forecast the future (*t* period) NT dollar/dollar exchange rate. The detailed comparisons of the proposed method with the GA_GA, GA-PSO, and GA_PBN models are also comprised, with criteria of MSE, MAE, and root mean square error (RMSE).

2. Methodology and Data Description

In this paper, the combinations of GA, PSO, BPN were used to forecast the NT dollar/USdollar exchange rate. In the first step, the GA started with the random initial population, and the chromosomal representation was established. The GA gives an optimal set of beta values when achieving the stopping condition. We determined the top ten variables by sorting the beta β, and achieved similar or superior forecast performance when it became an input factor.

2.1 Genetic Algorithms

Genetic Algorithms (GA) (Holland, 1975 [11]) work on a population of potential solutions in the form of chromosomes, attempting to locate the best solution through the process of artificial evolution. It consists of the following repeated artificial genetic operations: evaluation, selection, crossover, and mutation.
We use GA to formulate operations in the basic GA algorithm, as follows:

(1) Encoding procedure: the parameters of solution denoted as

$$\beta = (\beta_1, \beta_2, ..., \beta_n), lb_j \le \beta_j \le ub_j,\ j = 1....n \qquad (1)$$

where the *lb* and *ub* indicate the lower and upper bounds for each parameter, this allows each parameter β_j to be transformed by function *L(x)* to be an integer y_j satisfying its boundaries, which can be encoded as n*D bits of binary code.
$y_j = L(x_j), c_j = u_1^j u_2^j \ldots u_{D_j}^j$, $j = 1 \ldots D$, $u_D^j \in \{0,1\}$ A chromosome is defined as a binary string combining binary codes of all parameters. There are 27*D bits in a chromosome.

(2) Random crossover and mutation:

The mutation and crossover operations occur at random. They exchange the parts of chromosome in order to randomly inverse the selected single bit of chromosome.

(3) Fitness function

In this study, we define the minimal error as the fitness function:

$$Minimize\ \ MAPE = \frac{\sum_1^n \left| (\hat{St} - St) / St \right|}{n} \tag{2}$$

where MAPE is the mean absolute percentage error between real and forecasted exchange rates. It was adopted as the fitness function to minimize forecasting errors.

(4) Parameters selection of GA

- *Population size*: 200
- *Total generation*:2000
- *Crossover rate*: *0.5*
- *Mutation Rate*:0.05
- *Stopping condition*: 0.02

2.2 Particle Swarm Optimization

PSO is an evolutional algorithm that might be used to find optimal solutions (Eberhart, & Kennedy, 1995)[12].The primary forecasting model of PSO is shown in (3).

$$\hat{St} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_{D-1} x_{D-1} \tag{3}$$

PSO algorithm explores the optimal target through local history and global communications in populations that consist of potential solutions, defined as particles. For achieving a high efficiency of searching the optimal solution in high dimensions, PSO drives the particle swarm to move toward the higher-object-value region based on the best experiences of each particle and the entire population. Thus, PSO converges to

the optimal point significantly faster than evolutionary optimization. Note that each particle indicates a candidate solution; if we have D-dimensional parameters in solution, the *i*-th particle for the *t*-th iteration can be represented as

$$B_i^t = \left(\beta_{i,0}^t, \beta_{i,1}^t, \beta_{i,2}^t, \beta_{i,3}^t, \beta_{i,4}^t ..., \beta_{i,D-1}^t\right) \tag{4}$$

Assume that the best previous position of the *i*-th particle at the *t*-th iteration is represented as

$$P_i^t = (p_{i,1}^t, p_{i,2}^t, ..., p_{i,D}^t) \tag{5}$$

Then, the values of cost function $MAPE$ (.) are expected to decrease by iterations.

$$Min \quad MAPE\ (.) = \frac{\sum_1^n \left|(\hat{St} - St) / St\right|}{n} \tag{6}$$

Where $\varepsilon(.)$ is the $\left|(\hat{St} - St)\right|$, to minimize the forecasting errors and is adopted as the fitness function.

$$LB \le B_i^t \le UB \tag{7}$$

$$\varepsilon(P_i^t) \le \varepsilon(P_i^{t-1}) \le ... \le \varepsilon(P_i^1) \tag{8}$$

The velocity of the i-th particle at the t-th iteration, V_i^t can be represented as

$$V_i^t = (v_{i,1}^t, v_{i,2}^t, ..., v_{i,D}^t) \tag{9}$$

The best position among all particles is G^t, until the t-th iteration,

$$G^t = \arg\min_{p_i}\left(\varepsilon\left(p_{i,j}^t\right)\right) \tag{10}$$

which is denoted as $G^t = (g_1^t, g_2^t, ..., g_D^t)$.

The modified version of PSO algorithms has been applied by an inertia weight W^t multiplying the velocity term in numerous other applications. This version can be expressed as:

$$V_i^{t+1} = W^t \cdot V_i^t + c_1 \cdot r_1 \cdot (P_i^t - B_i^t) + c_2 \cdot r_2 \cdot (G^t - B_i^t) \qquad r_1, r_2 \sim U(0,1) \tag{11}$$

$$B_i^{t+1} = B_i^t + V_i^{t+1}, i = 0,1,..., N-1 \tag{12}$$

where the updating velocity is bounded, and r_1 and r_2 are random variables with uniform distributions. Here, c_1 and c_2 are factors used to control any influences from

local and global velocity terms. The parameters are: Population size: 200; Iteration: 2000; Vmax: 0.2; c_1: 2; c_2: 2; Initial weight: 0.9; Final weight: 0.4.

After training by PSO algorithm, we obtain the optimal solution shown below:

$$B_i^t = \left(\beta_{i,0}^t, \beta_{i,1}^t, \beta_{i,2}^t, \beta_{i,3}^t, \beta_{i,4}^t ..., \beta_{i,D-1}^t\right) \tag{13}$$

2.3 Back Propagation Network

The feed-forward neural network with back propagation learning is the most conventional model of ANN that updates weight and bias values. The proposed BPN has 10 input layer neurons; in this study, we propose an appropriate transfer multiple-hidden-layer (at most 2 hidden layers), with a Hyperbolictangent transfer function and normalized data (between -1 and 1). We also select the gradient descent with momentum back propagation as the learning function; the parameters of BPN are:

- *Input layer neurons: n (selected from GA)*
- *Hidden layer neurons: (5,5)*
- *Learning rule : traingdm*
- *Transfers function: TanH*
- *Hidden layer: 2*
- *Stopcriteria: error< 1e-5*

2.4 Variables and data normalization

The experimental data in this paper is to collect the historic exchange rate NT dollar / US dollar, and eight macroeconomic factors. The period is from Jan.1997 to Dec.2007 and a total of 132 monthly data is collected. We collect three lagged times from each factor. So, we have 27 factors for 132 months. We divide the whole data into six periods of our sliding window. Through this process, the thesis can get more regressions so that the prediction result will be more accurate.

Recently, some research has considered the macroeconomic factors as variables when forecasting exchange rate; a linear model is formed to describe the fluctuation of exchange rate. The main variables are displayed in Table 1. Each variable will be divided into three data periods, specifically t_{-3} period, t_{-2} period, and t_{-1} period, to forecast the exchange rate by considering the lagged period. Normalization is necessary because the variables have different units. In (1), data is normalized by function premnmx in Matlab with a range of -1 to 1.

$$\begin{aligned} &[pn, Min\ p, Max\ p] = premnmx(p) \\ &pn = \frac{2 \cdot (p - Min\ p)}{Max\ p - Min\ p} - 1 \end{aligned} \tag{14}$$

Where p is the original data, and pn is the normalized data.

Since macroeconomic factors could affect exchange rates, the variables which we propose are summarized as follows:

Table 1 Variables

Variables	Description
Constant	1 : Beta 0
CPI	2~4 : CPI index (t_{-1})~ (t_{-3}) CPI index as an an independent variable
M1 and M1B	5~7: US M1 (t_{-1})~ (t_{-3}) 8~10:TW M1B (t_{-1})~ (t_{-3})
Commercial paper rate, Federal fund rate	11~13: Commercial paper rate (t_{-1})~ (t_{-3}), 14~16: Federal fund rate (t_{-1})~ (t_{-3})
Balance of trade	17~19: Balance of trade (t_{-1}) ~(t_{-3})
Exchange rate: USD/NTD	20~22: USD/NTD(t-1)~ (t-3)
Stock return	23~25: Stock return (t-1)~ (t-3)
Foreign investment	26~28: Foreign investment (t-1)~ (t-3)

3. Experiment Design

In this paper, we apply the GA method to select the optimal variables. This method also replaces the traditional trial and error method that determines the input factors, layer number or input neurons.

Step 1 Variable selection

We sort the weight obtained from GA, and adopt the top 5, top 10, top 15, top 20, and top 25 variables from 27 variables with each sliding windows as input factors for BPN; we adopted the smallest value obtained after repeating the experiment three times. In most of the sliding windows, the set with 10 variables has the optimum forecast performance. For instance, the sliding window 1 results illustrate that the set with 10 variables has the best forecasting performance (See Table 2).

Table 2 Comparison of different variables

Sliding windows	Variables	RMSE	MSE	MAE
Sliding window 1	5	0.7936	0.6298	0.7121
	10	***0.6382***	***0.4073***	***0.4946***
	15	0.6844	0.4683	0.6690
	20	1.0971	1.2037	0.5458
	25	1.2915	1.6680	0.6882

Therefore, in the second stage of the experiment, we will select the top 10 variables of smallest values in each sliding window data. And the final top 10 variables were decided to be inputs of our model, as shown in Table 3.

Table 3 The optimal 10 variables

Variables	Variables
Exchange rate (t_{-1})	Foreign investment(t_{-1})
Exchange rate (t_{-2})	Commercial paper rate(t_{-2})
Stock return (t_{-1})	M1(t_{-3})
Federal fund rate(t_{-1})	M1B(t_{-1})
Stock return (t_{-2})	Stock return (t_{-3})

Step 2 Problem Formulation

In the following equations, $f(x)$ has 27 variables $(x_1, x_2, ..., x_{27})$ with weights of $(\beta_0, \beta_1, \beta_2, ..., \beta_n)$. $f(x)$ is the linear model in which we formulate the exchange rate (i.e. $f(x) = \hat{S}t$). $\hat{S}t$ is t period for forecasting exchange rates.

Where *f*(x) is the forecast exchange rate,

β_0 is the constant,

X_n is the weight of each variables, and is the set of constant and variables,

β_n is the set of constant and variables,

B is the set of weight, and is the error between real and forecast exchange rate,

ε is the error between real and forecast exchange rate,

S_t is real exchange rate,

$\hat{S}t$ is the forecast exchange rate

Step 3 Forecast criteria

In consideration of several earlier empirical studies (Neely and Weeler 2001; Heckerling, Gerber 2004), we will evaluate the forecasting results based on three criteria: MSE (Equation 20), MAE (Equation 21), and RMSE (Equation 22).

(1) Mean square error ($MSE = \sum \frac{(\hat{S}_t - S_t)^2}{n}$) (20)

(2) Mean absolute error ($MAE = \sum \frac{\left| \hat{S}_t - S_t \right|}{n}$) (21)

(3) Root mean square error ($RMSE = \sqrt{\sum \frac{(\hat{S}_t - S_t)^2}{n}}$) (22)

4. Experiment results

4.1 Comparing the forecasting results by three criteria

To test which selected variables will have better forecast performance, we randomly select 10 variables for GA to select. The out-of-sample forecasting performance of the GA_GA model, the GA_PSO model, and the GA_BPN model are illustrated in Table 4, Table 5, Table 6, respectively.

As shown in Table 4, the average forecasting performance of the GA_GA model, which is measured with the average RMSE, is 0.5147. The average of MSE is 0.2991, and the average MAE is around 0.4040.

Table 4 Forecasting results of GA_GA model

Sliding Window 1~6	RMSE	MSE	MAE
GA_GA 1	0.8187	0.6702	0.6818
GA_GA 2	0.6720	0.4515	0.5268
GA_GA 3	0.3870	0.1498	0.2789
GA_GA 4	0.2526	0.0638	0.1607
GA_GA 5	0.4636	0.2150	0.3403
GA_GA 6	0.4941	0.2441	0.4352
Average	**0.5147**	**0.2991**	**0.4040**

Table 5 presents the average forecasting performance of the GA_PSO model, which is measured with the average of RMSE, is 2.1188. The average of MSE is 1.0423, and the average of MAE is around 0.9690. The results (in all RMSE, MSE, MAE) of the GA_GA model are lower than the GA_ PSO model with no outlier values. This indicates that the GA_GA model offers better performance than the GA_ PSO model in forecasting exchange rates.

Table 5 Forecasting results of GA_PSO **model**

Sliding Window 1~6	RMSE	MSE	MAE
GA_PSO 1	0.9430	0.9711	0.8606
GA_PSO 2	0.3071	0.5542	0.4599
GA_PSO 3	0.0991	0.3148	0.2689
GA_PSO 4	0.1417	0.3765	0.3221
GA_PSO 5	0.6076	0.7795	0.6713
GA_PSO 6	10.6145	3.2580	3.2314
Average	**2.1188**	**1.0423**	**0.9690**

As presented in Table 6, the forecasting performance of GA_BPN, which is measured with the average of RMSE is 0.7661, and the average of MSE is 0.6092. The average of MAE is 0.6732. The results (in all RMSE, MSE, MAE) of GA_GA model are lower than the GA_ BPN model, with no outlier values. This indicates that the GA_GA model has better performance than the GA_ BPN model in forecasting exchange rates.

Table 6 Forecasting results of GA_BPN **model**

Sliding Window1~6	RMSE	MSE	MAE
GA_BPN 1	0.5503	0.3028	0.3858
GA_BPN 2	0.8918	0.7952	0.7372
GA_BPN 3	0.6978	0.4869	0.6505
GA_BPN 4	0.6363	0.4049	0.5469
GA_BPN 5	0.9744	0.9494	0.9248
GA_BPN 6	0.8460	0.7156	0.7940
Average	**0.7661**	**0.6092**	**0.6732**

4.2 Comparison of the forecast results by monthly exchange rate

There are sixty-month for training data and twelve-month for testing data in this study. The testing data will have six-month overlap with the training data. For instance, the training data start from Jan. 1997, which is the 1st data to the 60th data which is Dec.2001; and the testing data start from the 55th collected in data Jun. 2001 to the 66th data collected in May 2002. The monthly forecast results are shown in Table 7. In Figure 1, the dotted line with the open square represents the forecast exchange rate by GA_GA, and the dotted line with the star represents the forecast exchange rate by GA_BPN. The line with the triangle represents the forecast exchange rate by GA_PSO, and the line with diamond represents the real exchange rate. It is evident that the forecast exchange rates with GA_GA model have only a slight disparity with the real exchange rate, with the exception of Months 2, 7 and 9. Based on these figures, we observe that the forecasting exchange rate of GA_GA has the least error in relation to the actual exchange rate. Therefore, the forecast performance of these three models, the

GA_GA model has the best forecast performance, followed by the GA_BPN model and the GA_PSO model produced the least accurate results.

Table 7 Comparison of the forecasting results by monthly exchange rate

ITEM	Real Exchange rate	GA_GA	GA_BPN	GA_PSO
Month1	0.0208	-0.1330	-0.2029	-0.5135
Month2	0.0581	0.0919	0.0515	-0.4627
Month3	0.1357	0.1029	-0.0788	-0.4621
Month4	0.1564	0.2700	-0.0614	-0.4705
Month5	0.1030	0.0161	-0.1769	-0.5575
Month6	0.1146	0.2387	-0.3657	-0.3703
Month7	0.1846	0.0140	-0.1678	-0.3297
Month8	0.0448	0.1999	-0.1744	-0.2269
Month9	0.0134	0.4179	0.0076	-0.3275
Month10	0.0603	0.0644	0.0299	-0.3582
Month11	-0.0152	0.0918	-0.2719	-0.4708
Month12	-0.1329	-0.0573	-0.3098	-0.5307

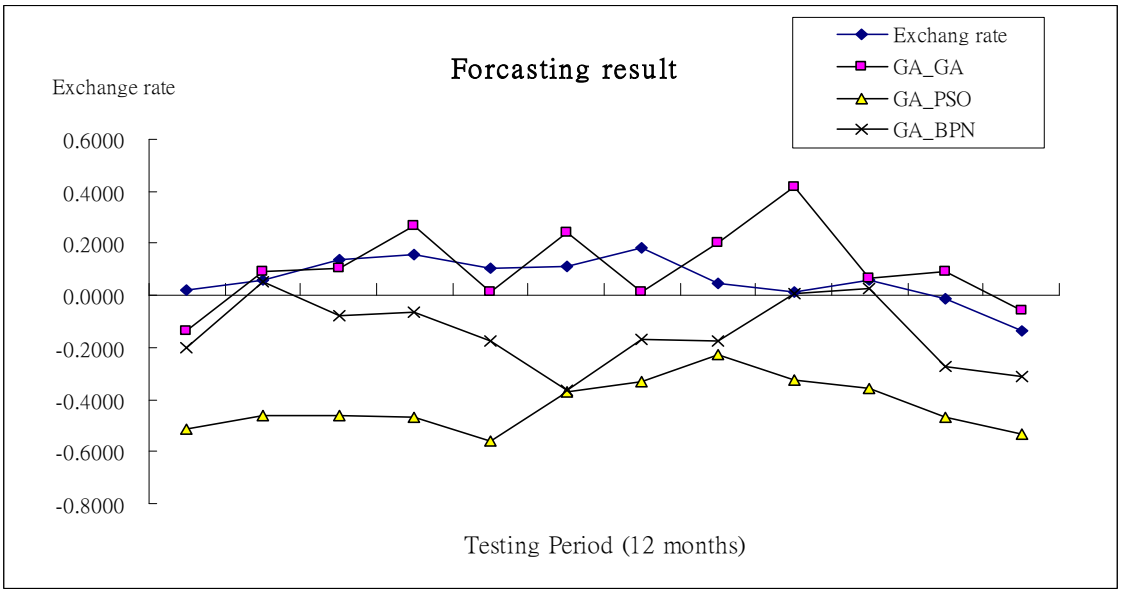

Figure 1 Comparison of the forecasting results by monthly exchange rate

5. Conclusion and Suggestions

In this study, we proposed the GA_GA, GA_PSO, and GA_BPN models, which are applied to the forecasting of the NT dollar/US dollar exchange rate. In addition to the

measurement by RMSE, MSE, and MAE, we also compared the forecasting results by monthly exchange rate. The results show that GA_GA has superior forecast ability. Moreover, we found that its forecast of NT dollar/USD dollar exchange rates is virtually identical with the real exchange rates in this study's experiments. For instance, the GA_GA (represented by a square sign) is matched to the trend of real exchange rates (diamond sign). Therefore, regarding the forecast performance of these three models, the GA_GA model has superior forecast performance, followed by the GA_BPN model, and results of the GA_PSO model were last. Additionally, future studies could adopt the nonlinear model in GA, whereas future researchers may compare the results by applying other optimal methods, (i.e. Ant algorithm, or Cat algorithm, etc). Also, we suggest applying these models to other currencies over a longer period of time.

6. References

[1] Branson, W.H., "Assets Markets and Relative Prices in Exchange Rate Determination" *Sozialassenschaftliche Annalen,* 1997, pp.68-89

[2] Nelly, C.J. & Weller, "Predicting Exchange Rate Volatility: Genetic Programming Versus GARCH and Risk Metrics", *The Federal Reserve Bank os St. Louis*, May/June, 2002, Volume 84, no.3, pp.43-54.

[3] Brandl B, Keber C, Schuster M. G., "An automated econometric decision support system: forecasts for foreign exchange trades"*Central European Journal of Operations Research. Heidelberg,* 2006. Vol. 14, Iss. 4; pg. 401, 15 pgs

[4] Rauscher, F.A., "Exchange Rates Forecasting: A Neural VEC Approach to Non-Linear Time Series Analysis", *Journal of Time Series Analysis*, 1997, pp.461-471.

[5] Ashok Nag, K and Amit, Mitra, "Forecasting Daily Foreign Exchange Rates Using Genetically Optimized Neural Networks", *Journal of Forecasting*, 2002, Volume 21 Issue 7, pp.501 – 511.

[6] Natarajan, Saravanan, & Periasamy, 2006. Application of particle swarm optimisation in artificial neural network for the prediction of tool life. *The International Journal of Advanced Manufacturing Technology*, 28: 1084-1088.

[7] Geethanjali, Slochanal & Bhavani, 2008. PSO trained ANN-based differential protection scheme for power transformers. *Neurocomputing*, 71(4-6):904-918.

[8] Frenkel, J.A., "Flexible Exchange Rates, Prices, and the Role of "News" : Lessons from the 1970s" *Journal of Political Economic*, 1981, pp.665-703

[9] MacDonald, R. &Taylor M.P., "The Monetary Model of the Exchange Rate: Long-run Relationships, Short-run Dynamics and how to Beat a Random Walk", *Journal of International Money and Finance*, June 1994, pp.276-290

[10] Bisignano, J., and Hoover K., "Some Suggested Improvements to a Simple Protfolio Balance Model of Exchange Determination with Special Reference to the U.S. Dollar/ Canadian Dollar Rate", *Weltwirtschaftiches Archiv*, 1982, pp.19-37

[11] Holland J.H., Adaptation in natural and artificial systems, Ann Arbor, MI: The university of Michigan Press, 1975.

[12] Eberhart, R., & Kennedy, J., 1995. A new optimizer using particle swarm theory. *Sixth International Symposium on Micro Machine and Human Science*: 39-43.