Global Hybrid Ant Bee Colony Algorithm for Training Artificial Neural Networks

Habib Shah, Rozaida Ghazali, Nazri Mohd Nawi, and Mustafa Mat Deris

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia (UTHM)
Parit Raja, 86400 Batu Pahat. Johor, Malaysia
Habibshah.uthm@gmail.com,
{rozaida,nazri,mmustafa}@uthm.edu.my

Abstract. Learning problems for Neural Networks (NNs) has widely been explored from last two decades. Population based algorithms become more focus by researchers because of its nature behavior processing with optimal solution. The population-based algorithms are Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), and recently Hybrid Ant Bee Colony (HABC) Algorithms produced easy way for training NNs. These social based techniques mostly used for finding optimal weight values and over trapping local minima in NNs learning. Typically, NNs trained by a traditional and recognized algorithm called Backpropagation (BP) has difficulties such as trapping in local minima, slow convergence or might fail sometimes. In this research, the new method named Global Hybrid Ant Bee Colony (GHABC) algorithm used to train NNs to recover the BP gaps. The simulation result of a hybrid algorithm evaluates with ordinary ABC, Levenberg-Marquardt (LM) training algorithms. From the investigated results, the proposed GHABC algorithm did get better the learning efficiency for NNs using Boolean Function classification task.

Keywords: Swarm Intelligence, Ant Colony Optimization, Artificial Bee Colony, Hybrid Ant Bee Colony Algorithm, Global Hybrid Ant Bee Colony.

1 Introduction

Nowadays, neural networks are widely used in different functions such as linear and nonlinear modeling, prediction, forecasting and classification, mostly due to their property of generality [1-3]. It has powerful and flexible applications that have been successfully used in various applications such as statistical, biological, medical, industrial, mathematical, and software engineering. [4-6]. ANN learned their training techniques by parallel processing. NNs are capable of achieving many scientific research applications by providing best network architecture, activation function, input pre processing and optimal weight values.

NNs tools are the most interested and understandable to mathematical problems, statistical modeling by using different background of various data types. The accuracy makes this particular use of NNs as attractive to scientist analysts in various areas for

B. Murgante et al. (Eds.): ICCSA 2012, Part I, LNCS 7333, pp. 87–100, 2012.

different task such as image processing, prediction, classification and other desired combinatorial task.

Many training techniques and NNs architectures for learning parity problem and other difficult boolean function's classification were accessible in [7–10]. These techniques are appropriate only for the parity problem and will not work for other complex problems. Biological NNs are able to solve quite complex learning problems inherent in optimization of intelligent actions. Finding general algorithm capable of solving a larger set of problems of similar complexity such as the XOR, Encoder Decoder and parity problems are still a challenge to the scientific scientists.

Traditionally, NNs models learn by changing the interconnection weights of their associated neurons. It can be trained by different approaches such as: BP, Improved BP algorithm, Evolutionary Algorithms (EA), Swarm Intelligence (SI), Differential Evolution (DE) and Hybrid Bee Ant Colony (HBAC), IABC-MLP and recently HABC Algorithms [11-16]. However, BP algorithm results in long training time and insufficient performance for the binary classification task. However, a BP learning algorithm has some difficulties; especially, it's getting trapped in local minima, where it can affect the NNs performance [17].

In order to overcome the weaknesses of standard BP, many approaches used, which are based on mathematical approach, local and global optimization and population techniques for training the NNs, These are: Particle Swarm Optimization (PSO), ACO, (ABC-LM), ABC-MLP, HABC, HBAC recently population-based and Evolutionary algorithms having reliable performance on training NNs [13-16,18,22,24,26].

In this study, the new hybrid populations based algorithm Global Hybrid Ant Bee Colony (GHABC) is used for training NNs for recovering the BP crack. The experimentation test is done by a Boolean Function while the result compared with different approaches such as ABC and BP (LM) algorithms.

The rest paper is organized as follows: Boolean function classification problems given in Section 2. A brief review on ANN and BP training is given in Section 3. Section 4 contains on swarm intelligence briefing with subsection ABC, ACO and HABC algorithms. The proposed hybrid technique GHABC algorithm is detailed in Section 5. Section 6 contains the simulation result with discussion. Finally, the paper concludes in Section 7.

2 Boolean Function Classification

A Boolean function "f" is a map $f: f\{0;\ 1\}^N \rightarrow f\{0;\ 1\}$, where N is the number of input bits. The function f is completely defined when the corresponding outputs for each of all the 2^N inputs are determined. Classification of data concern with the use of computers in order to create a structure that learns how automatically chooses to which of a predefined set of classes, a given object belongs. Boolean function classification is the most important issue until today how to decide the 0 and 1 or on or off separate classes by different techniques. These are Boolean functions such as XOR, 3-Bit Parity and Encoder Decoder problems. These are non-linear benchmark classification tasks consisting of 2^N patterns with N inputs and one output. Each input or output is either a 0 or a 1.

Definition (Function) If *A* and *B* are sets, a function from *A* to *B* is a rule that tells us how to find a unique $b \in B$ for each **a** ∈ A. We write $f(a) = b$ and say that f maps *a* to *b*. We also say the value of f at a is *b*.

We write f: $A \rightarrow B$ to indicate that *f* is a function from *A* to *B*. We call the set A the domain of f and the set *B* the range or, equivalently, co domain of *f.* To specify a function completely you must give its domain, range and rule.

Problem 1: XOR is a difficult classification problem mapping two binary inputs to a single binary output as *(0 0;0 1;1 0;1 1) → (0;1;1;0).*

Table 1. XOR classification table

P	**q**	**f**
F	F	F
F	T	T
T	F	T
T	T	F

The problem is taking the modulus 2 of summation of three inputs. In other words, if the number of binary inputs is odd, the output is 1, otherwise it is 0.

Table 2. 3-Bit Parity Problem

p	**q**	**r**	**f**
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	F
T	T	F	F
T	T	T	T

Table 3. 4-Bit Encoder/Decoder

p	**q**	**r**	**s**	f_1	f_2	f_3	f_4
F	F	F	T	F	F	F	T
F	F	T	F	F	F	T	F
F	T	F	F	F	T	F	F
T	F	F	F	T	F	F	F

4-Bit Encoder/Decoder is quite close to real world pattern classification tasks, where small changes in the input pattern cause small changes in the output pattern [19].

3 Learning Algorithms of Artificial Neural Networks

MLP is a universal approximate and mathematical model that contains a set of processing elements known as artificial neurons [20]. The network which is also known as feed forward neural network was introduced in 1957 to solve a non-linear XOR, and was then successfully applied to different combinatorial problems [20]. The basic building of MLP is constructed by neurons, which have some categories as input, hidden and output layers, as shown in Figure 1. The weight values between input and hidden nodes and between hidden and output, nodes are randomly initialized. The network is highly used and tested on a different job. Figure 1 shows the basic architecture Multilayer Perceptron [21].

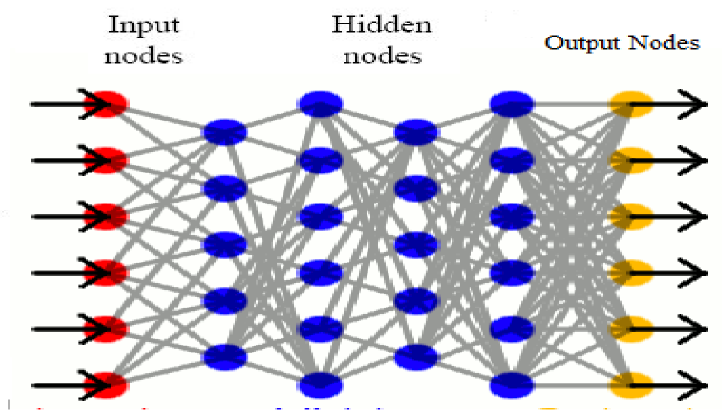

Fig. 1. Multilayer Perceptron Model

The output value of the MLP can be obtained by the following formula:

$$Y = f_i\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) \tag{1}$$

Where Y is the output of the node x is the j_{th} input to the node, w is the connection weight between the input node and output node, b_i is the threshold (or bias) of the node, and f_i is the node transfer function. Usually, the node transfer function is a non-linear function such as: a sigmoid function, a Gaussian functions. Network error function E will be minimized as

$$E(w(t)) = \frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{k}\left(d_k - o_t\right)^2 \tag{2}$$

Where $E(w(t))$ is the error at the t_{th} iteration; $w(t)$ is the weight in the connections at the t_{th} iteration; j shows training set, d_k is the desired output; o_t is the actual value of the k-th output node; K is the number of output nodes; and n is the no of patterns.

NN learning is a process of obtaining new knowledge or adjusting the existing knowledge through the training process. BP is currently the most widely and well known used algorithm for training MLP developed by Rumelhart [3]. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. This step by step mathematical procedure adjusts the weights according to the error function. So, the adjustment of weights which decrease the error function is considered to be the optimal solution of the problem. In the input layer only inputs propagate through weights and passing through hidden layers and get output by some local information. For the BP error, each hidden unit is responsible for some part of the error.

Although the BP algorithm is a powerful technique applied to classification, combinatorial problems and for training, MLP. However, as the problem complexity increases, the performance of BP falls off rapidly because gradient search techniques tend to get trapped at local minima. When the nearly global minima are well hidden among the local minima, BP can end up bouncing between local minima, especially for those non-linearly separable pattern classification problems or complex function approximation problem [17]. A second shortcoming is that the convergence of the algorithm is very sensitive to the initial value. So, it often converges to an inferior solution and gets trapped in a long training time.

4 Swarm Intelligence (SI)

Since the last two decades, SI has been the focus of many researches because of its unique behavior inherent from the swarm insects [12]. Bonabeau has defined the SI as “any attempt to design algorithm or distributed problem-solving devices inspired by the collective behavior of nature insect colonies and other animal societies." He mainly focused on the behavior of social insects alone such as termites, bees, wasps, and different ant species. However, swarms can be considered as any collection of interacting agents or individuals. Ants are individual agents of ACO [22].

4.1 Ant Colony Optimization (ACO)

ACO is a meta-heuristic procedure for the solution of a combinatorial optimization and discrete problems that has been inspired by the social insect’s foraging behaviour of real ant decision developed in 1990s [22]. Real ants are capable of finding Food Source (FS) by a short way through exploiting pheromone information, because ants leave pheromone on the ground, and have a probabilistic preference for trajectory with larger quantity of pheromone. Ants appear at a critical point in which they have to choose to get food, whether to turn right or left. Initially, they have no information about which is the best way for getting the FS.

Ants move from the nest to the FS blindly for discovering the shortest path. The above behavior of real ants has inspired ACO, an algorithm in which a set of artificial ants cooperate in the solution of a problem by sharing information. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds FS, it evaluates the quantity and the quality of the food and carries some of it back to the nest. The following is the ACO pseudo code.

```
Initialize Trail
Do While (Stopping Criteria Not Satisfied) – Cycle Loop
Do Until (Each Ant Completes a Tour) – Tour Loop
Local Trail Update
End Do
Analyze Tours
Global Trail Update
End Do
```

4.2 Artificial Bee Colony (ABC) Algorithm

ABC was proposed for optimization, classification, and NNs problem solution based on the intelligent foraging behavior of honey bee [23-24]. The three bees determine the objects of problems by sharing information to other's bees. The employed bees use multidirectional search space for FS with initialization of the area. They get news and all possibilities to find FS and solution space. Sharing of information with onlooker bees is performed by employed bees. Onlooker bees: Onlooker bees evaluate the nectar quantity obtained by employed bees and choose a FS depending on the probability values calculated using the fitness values. Onlooker bees watch the dance of hive bees and select the best FS according to the probability proportional to the quality of that FS. Scout bees: Scout bees select the FS randomly without experience. If the nectar quantity of a FS is higher than that of the old source in their memory, they memories the new position and forget the previous position. Whenever employed bees get a FS and use the FS very well again, they become scout bees to find a new FS by memorizing the best path.

5 Hybrid Ant Bee Colony Algorithm (HABC)

Hybrid Ant Bee Colony (HABC) technique was proposed for training NNs based on the intelligent foraging behavior of honey bee and ant swarms [14, 16]. The ABC algorithm has a strong ability to find the global optimistic results through most favorable weight's values by bee's representatives [23]. It is successfully trained the NNs for classification and prediction task [24].

HABC algorithm combines the ACO properties in the ABC algorithm which may accelerate the evolving speed of ANN and improve the classification precision of the well-trained networks. The hybrid algorithm is easily understandable, using an ABC algorithm to search the optimal combination of all the network parameters, and ACO used for selection best FS to find the accurate value of each parameter. HABC algorithm provides a solution in an organized form by dividing the agents into different tasks such as, employed bee, onlooker ant and scout bees. The algorithm of HBAC algorithm is shown as follows:

1. Load colony size and food Number

$$FN = \frac{SN}{2} \tag{3}$$

King Bee {

If

$$FN = SN\,\%\,2 = 0 \tag{4}$$

Then

$$FN = \frac{SN+1}{2} \tag{5}$$

2. Initialize of solutions x_i
3. Evaluate the fitness of the population
4. Cycle =1
5. Repeat
6. Produce a new solution v_i by using equation

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \tag{6}$$

Calculate τ_i

$$\tau_{(i)} = \begin{cases} \tau \geq 0 & for \dfrac{Q}{1+f_i} \\ \tau < 0 & 1+abs(f_i) \end{cases} \tag{7}$$

$$Q = \frac{1+FN}{\sum_{i=1}^{SN}(E+O)} \tag{8}$$

Apply greedy selection process

7. Calculate the probability values $P_{(i)}$for the solution xi by

$$P(i) = \frac{(0.09)*\tau(i)}{\sum_{1\leq j\leq m}^{SN}\tau(j)} \tag{9}$$

8. FOR each onlooker ant {
   Select a solution xi depending on P_i
   Produce new solution v_i

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \tag{10}$$

9. Calculate the value τ_i by eq (7)
10. Apply greedy selection process}
11. Determine the abandoned solution (source), if exists, replace it with a new randomly produced solution xi for the scout using the following equation.

$$x_{ij}^{rand} = x_{ij}^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \tag{11}$$

12. Memories the best FS position (solution) achieved so far.
13. Cycle=cycle+1
14. Until the cycle= Maximum Cycle Number (MCN)

Where *Q, E* and *O* shows Numbers of Queen, Employed and Onlookers bees respectively. and x_i represents a solution is the fitness solution o ,v_i indicates a neighbour solution of x_i, and p_i value of x_i. Also τ represents the fitness solution of trial, *i* which is not improved and j represents the improved solution. In the algorithm, first half of the colony consists of employed ant, and the second half constitutes the onlooker ant.

The Scout bee will be deciding the best values between onlookers and employed ant. In HABC algorithm, the position of a FS represents a possible solution to the optimization problem, and the nectar total of a FS corresponds to the fitness solution of the associated solution by King Bee. The King Bee initialized the colony size for employed and onlooker's ant. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2... Maximum Cycle Number (MCN), of the search processes of the employed, onlooker and Scout Bee.

An employed ant produces a modification on the position in her memory depending on the local information (visual information) and tests the nectar total (accurate solution) of the new source. The King Bee will gather employed and onlooker ant for decision of fitness solution. After all employed ants around complete the search process; they combine the nectar information of the FS and their position information with the onlooker ant on the food's area. An onlooker ant evaluates the nectar amount taken from all employed ants and chooses a FS with a probability related to its nectar quantity. The onlooker ant chooses a FS depending on the probability value associated with that FS, and P(i), calculated by the eq (9).

The King Bee initialized the colony size for bee. The FS will be divided into same quantity. King's bee will update the FS for equal division on employed and onlooker's bee. The number of FS equals the half of the colony size and after division on employed and onlooker ant the will start searching for finding optimal FS. The information sharing with onlooker ant is performed by employed. An employed ant produces a modification of the source position in her memory and discovers the unused FS position, which provided that the nectar quantity of the new source is higher than that of the previous source, the employed ant memorizes the new source position and forgets the old one.

The onlooker ants evaluate the nectar quantity obtained by employed ants and choose a FS depending on the probability values. For this purpose, a fitness-based selection technique can be used. Onlooker ants watch the dance of hive ants and select the best FS according to the probability proportional to the quality of that FS. Scout bees: Scout bees select the food source randomly without experience. If the nectar amount of a food source is higher than that of the previous source in their memory, they memorize the new position and forget the previous position. Whenever employed ants get. a food source and use the food source very well again, they become scout bees to find new food source by memorizing the best path.

6 Global Hybrid Ant Bee Colony (GHABC) Algorithm

Global Artificial Bee Colony Search (GABCS) Algorithm used for Numerical Function Optimization outperforms other algorithms in almost all of these experiments, especially multimodal benchmark functions [25]. Hybrid Ant Bee Colony (HABC) Algorithm successfully applied for training NNs using Boolean Function and Time-series data prediction task [14, 16]. HABC has powerful ability for finding best weight for NNs model. Here Global Hybrid Ant Bee Colony (GHABC) will combine the properties of HABC with intelligent behavior of GABCS algorithm [25, 16].

The GABCS will update the solution step and will convert to best solution based on neighborhood values. These modified steps will be in employed, Onlooker and Scout Section. Furthermore, in HABC algorithm, the employed ant and onlookers exploit their solutions based on the neighbor information of each individual with intensity of ant technique.

Usually, in bee swarm, the experienced foragers can use previous knowledge of position and nectar quantity of food source to regulate their group directions in the search space. Furthermore, in social insect's technique, the best foods can be find through experience or neighbor cooperation. So HABC agents employed, scout and onlookers can be improved by their best food source. The GABCS will merge their best finding approaches with HABC by the following steps.

Step 1: Modified the employed section as

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) + y \tag{11}$$

$$y = c_1 rand\,(0,1)(x_j^{best} - x_{ij}) + c_2 rand\,(0,1)(y_j^{best} - x_{ij}) \tag{12}$$

Step 2: Repeat the above formula with onlookers section

Where y shows Best_Food_Source, C_1 and C_2 are two constant, x_j^{best} is the j_{th} element of the global best solution found so far, y_j^{best} is the j_{th} element of the best solution in the current iteration, ϕ_{ij} is a uniformly distributed real random number in the range [-1, 1]. The best values of c_1 and c_2 are 2.5 and -3.5 selected for the best performance for given problems.

Step 3: Modified the scout section as

$$x_{ij}^{rand} = x_{ij}^{\min} + rand\,(0,1)(x_j^{\max} - x_j^{\min}) \tag{13}$$

If rand (0, 1) <=0.5,

$$x_{ij}^{\text{mutation}} = x_{ij} + rand\,(0,1) * (1 - \frac{iter}{iter_{\max}})^b + (x_j^{best} - x_{ij}) \tag{14}$$

else

$$x_{ij}^{\text{mutation}} = x_{ij} + rand\,(0,1) * (1 - \frac{iter}{iter_{\max}})^b + (y_j^{best} - x_{ij}) \tag{15}$$

Then comparing the fitness value of random generated solution x_i^{rand} and mutation solution $x_i^{\ mutation}$ the better one is chose as a new food source, where b is a scaling parameter which generally is a positive integer within the range of [2,5].

7 Simulation Results and Discussion

In this work, swarms intelligent-based combine technique based on Global Hybrid Ant Bee Colony algorithm is used to train feed-forward artificial neural networks. In order to calculate the performance of the GHABC with ABC and LM algorithms in terms of Mean Square Error (MSE), Standard Deviation (S.D) of Mean Square Error and success rate using boolean function for classification, where simulation experiments performed by Matlab 2010a software.

The Boolean function classifications are categorized in three phases as XOR, 3 bit parity and 4 bit Encoder Decoder problems. The stopping criteria minimum error is set to 0.0001 LM while ABC and GHABC stopped on MCN. During the experimentation, 30 trials were performed for training. The sigmoid function is used as activation function for network output. During the simulation, when the number of inputs, hidden and output nodes of the NNs and running time varies, the performance of training algorithms were stable, which is important for the designation of NNs in the current state. The value of c_1 and c_2 were selected 2.5 and -3.5 respectively. From the simulation experiment, the GHABC performance can be affected by c_1 and c_2. So the best values selected for these two constant values.

Table 4. Parameters of the problems considered in the experiments. D: a Dimension of the problem, MCN: Maximum Cycle Numbers, NNs structure.

Problem	**Colony Range**	**NN structure**	**D**	**MCN**	**Epoch**
XOR6	[100,-100]	2-2-1	6	7500	32000
XOR9	[10,-10]	2-2-1+ Bias(3)	9	100	500
XOR13	[-10,10]	2-3-1+ Bias(4)	13	75	250
3-Bit	[-10,10]	3-3-1+bias(4)	16	1000	1600
Enc/Dec	[10,-10]	4-2-4+Bias(6)	22	1000	2100

From table4 the 2-2-1 NNs structure for XOR6 has six parameters including weights and without bias. The colony range is [100,-100] with 7500 MCN and dimension 6. The XOR9 problem has the same NNs structure included three biases with colony size [10,-10], where MCN and D are 100 and 9 respectively. The NNs 2-3-1+4 (bias) is set with MCN 75 and D 13 for the XOR13 classification task. The NNs 3-3-1+4 (bias) is set with MCN 1000 and D 16 for 3-bit parity problem with parameter [10,-10].

The 4-bit encoder/decoder problem was defined for 4-2-4 NNs shape with dimension 22, where 1000 MCN. The simulation runs thirty times with randomly foods while the above information from table 4 is used for training NNs. The average of MSE for training NNs by the GHABC, ABC and LM algorithms are shown in table 5. From fig 3, the averages of MSE for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder are 0.0436906, 0.00122154, 0.00149476, 0.00016009, 0.000285367 respectively using GHABC algorithm.

The LM algorithm has averages of MSE are 0.1107, 0.0491, 0.0078, 0.0209 and 0.0243 for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder problems. The ABC has an average of MSE 0.007051, 0.006956, 0.006079, 0.006679 and 0.008191 for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder respectively.

The GHABC has outstanding MSE for XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder from ABC and LM, where ABC is less than GHABC for XOR6 only in terms of MSE.

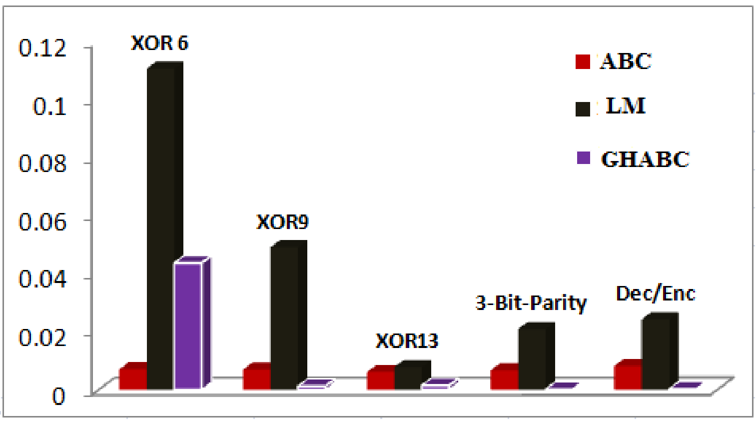

Fig. 2. Average of MSE for Training NNs by ABC, GHABC and LM

The S.D of MSE for GHABC, ABC and LM algorithms are given in fig 4.

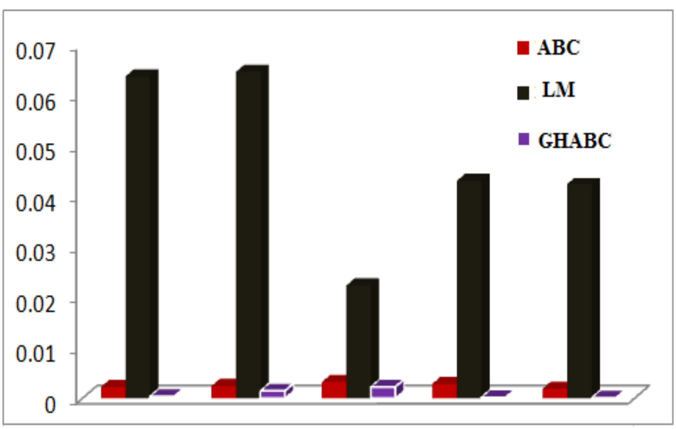

Fig. 3. Average of S.D for Training NNs by ABC, GHABC and LM

From fig 3, the SD of MSE for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder are 0.000391805, 0.00146855, 0.00223964, 0.00016729 and 9.91015E-05 respectively using GHABC algorithm. T

The LM algorithm has SD of MSE are 0.0637, 0.0646, 0.0223, 0.043 and 0.0424 for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder problems respectively. The ABC has the average of MSE 0.00223, 0.002402, 0.003182, 0.00282 and 0.001864 for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder respectively. The GHABC has outstanding S.D MSE for XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder from ABC and LM.

The success rate of GHABC, ABC and LM algorithms are given in fig 4. The GHABC has 100% success rate for XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder where LM has maximum 96.66 and minimum 6% only for Boolean function classification task.

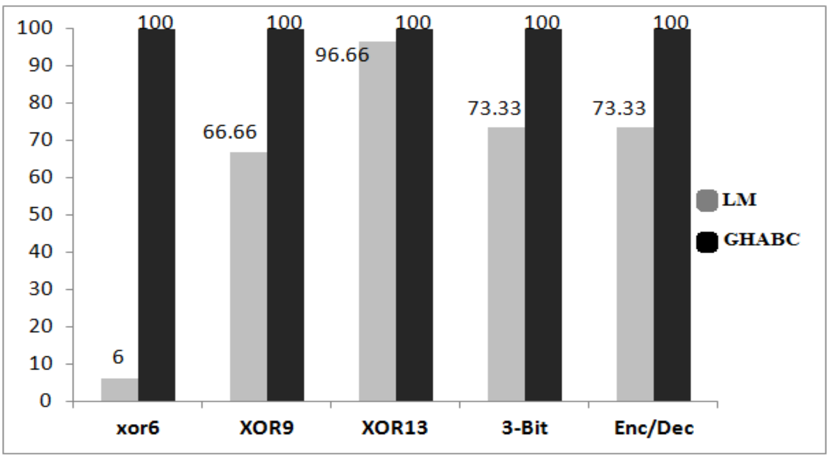

Fig. 4. Success Rates of BP (LM) and GHABC Algorithms

8 Conclusion

The GHABC algorithm combines the goods of three nature swarm's techniques, are ACO, HABC and GABCS algorithms effectively, which are finding best solution so far so best, exploration and exploitation. The optimum possible weight values prove the high performance of the training MLP for boolean function classification using GHABC algorithm. GHABC has the great ability of searching global optimum solution in solution space. The optimal weights set of the algorithm may speed up the training process and improve the classification accuracy. The success rate of GHABC algorithm shows its high excellent association in training NN. The above simulation results show that the proposed GHABC algorithm can outstandingly learn the Boolean data, which further extends the quality of the given approach with lower prediction error. The GHABC approach will use for prediction and classification time-series data in the next project.

Acknowledgment. The authors would like to thank University Tun Hussein Onn Malaysia (UTHM) and Ministry of High Education (MOHE) for supporting this research under the FRGS.

References

1. Romano, M., et al.: Artificial neural network for tsunami forecasting. Journal of Asian Earth Sciences 36, 29–37 (2009)
2. Ghazali, R., Hussain, A.J., Liatsis, P.: Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals. Expert Systems with Applications 38(4), 3765–3776 (2011)
3. Rumelhart, D.E., McClelland, J.L., The PDP Research Group: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, 2. MIT Press, Cambridge (1986)
4. Osamu, F.: Statistical estimation of the number of hidden units for feedforward neural networks. Neural Networks 11(5), 851–859 (1998)
5. Shokri, A., Hatami, T., et al.: Near critical carbon dioxide extraction of Anise (Pimpinella Anisum L.) seed: Mathematical and artificial neural network modeling. The Journal of Supercritical Fluids 58(1), 49–57 (2011)
6. Thwin, M.M.T., Quah, T.-S.: Application of neural networks for software quality prediction using object-oriented metrics. Journal of Systems and Software 76(2), 147–156 (2005)
7. Iyoda, E., Nobuhara, H., Hirota, K.: A solution for the n-bit parity problem using a single translated multiplicative neuron. Neural Processing Letters 18, 233–238 (2003)
8. Stork, D., Allen, J.: How to solve the n-bit parity problem with two hidden units. Neural Networks 5, 923–926 (1992)
9. Wilamowski, B., Hunter, D.: Solving parity-n problems with feedforward neural network. In: Proc. of the Int. Joint Conf. on Neural Networks (IJCNN 2003), Portland, Oregon, vol. I, pp. 2546–2551. IEEE Computer Society Press, Los Alamitos (2003)
10. Liu, D., Hohil, M., Smith, S.: N-bit parity neural networks: new solutions based on linear programming. Neurocomputing 48, 477–488 (2002)
11. Ilonen, J., Kamarainen, J.-K., et al.: Differential Evolution Training Algorithm for Feed-Forward Neural Networks. Neural Processing Letters 17(1), 93–105 (2003)
12. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, NY (1999)
13. Mendes, R., Cortez, P., Rocha, M., Neves, J.: Particle swarm for feedforward neural network training. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 1895–1899 (2002)
14. Shah, H., Ghazali, R., Nawi, N.M.: Hybrid Bee Ant Colony Algorithm (HBAC) for Classification and Prediction Mission. In: 2nd World IT Conference Turkey (Presented in Turkey, Procedia Computer Science Journal) (2011)
15. Shah, H., Ghazali, R.: Prediction of Earthquake Magnitude by an Improved ABC-MLP, DESE, UAE (Presented in UAE, DESE 2011, to be published soon in IEEE) (2011)
16. Shah, H., Ghazali, R., Nawi, N.M.: Hybrid Ant Bee Colony Algorithm for Volcano Temperature Prediction. In: Chowdhry, B.S., Shaikh, F.K., Hussain, D.M.A., Uqaili, M.A. (eds.) IMTIC 2012. CCIS, vol. 281, pp. 453–465. Springer, Heidelberg (2012)
17. Gori, M., Tesi, A.: On the problem of local minima in back-propagation. IEEE Trans. Pattern Anal. Mach. Intell. 14(1), 76–86 (1992)

18. Yao, X.: Evolutionary artificial neural networks. International Journal of Neural Systems 4(3), 203–222 (1993)
19. Fahlman, S.: An empirical study of learning speed in back-propagation networks, Technical Report CMU-CS-88-162, Carnegie Mellon University, Pittsburgh, PA 15213 (September 1988)
20. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
21. Rosenblatt, F.: A Probabilistic Model for Information Storage and Organization in the Brain. Cornell Aeronautical Laboratory 65, 386–408 (1958)
22. Dorigo, M., Di Caro, G., Corne, D., Dorigo, M., Glover, F.: The ant colony optimization meta-heuristic. New Ideas in Optimization, 11–32 (1999)
23. Karaboga, D., Akay, B., Ozturk, C.: Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 318–329. Springer, Heidelberg (2007)
24. Ozturk, C., Karaboga, D.: Hybrid Artificial Bee Colony algorithm for neural network training. In: 2011 IEEE Congress on Evolutionary Computation, CEC (2011)
25. Peng, G., Wenming, C., et al.: Global artificial bee colony search algorithm for numerical function optimization. In: 2011 Seventh International Conference on Natural Computation, ICNC (2011)
26. Shah, H., Ghazali, R., Nawi, N.M.: Using Artificial Bee Colony Algorithm for MLP Training on Earthquake Time Series Data Prediction. Journal of Computing 3(6), 135–142 (2011)