International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 4 Issue 5 ǁ May. 2016 ǁ PP. 48-58

A Hybrid of Particle Swarm Optimization And Genetic Algorithm for Training Back-Propagation Neural Network

NIE Shan-Kun[1], WANG Yu-Jia[1], XIAO Shanli[1],Chen Ke[1]*
School Of Electronic And Electrical Engineering, Shanghai University Of Engineering Science, Shanghai

***Abstract**:* An evolutionary back-propagation neural network (BP) which automates the design of BP networks using a new evolutionary learning algorithm is proposed in this paper. This new evolutionary learning algorithm is based on a hybrid of genetic algorithm(GA) and particle swarm optimization (PSO), and is thus called HGAPSO. In HGAPSO, the position and velocity in a new generation are updated, not only by PSO, but also by the crossover and mutation operation as in GA. The concept of select strategy is adopted in HGAPSO, where upper-half of the best-performing particles in a population are selected to be updated by PSO. However, instead of discard the other particles, these particles not selected will be updated by performing crossover and mutation operation to generate the next generation. What's more, a series of experiment is performed on six benchmark functions to confirm its performance. It is found that the proposed algorithm can get better solution quality in solving the global optimization problems and avoiding the local minimal. Based on it, the proposed algorithm is applied to train the BP neural network for the fault diagnosis. The performances of the trained neural network are compared with other neural networks trained by different inspired algorithms and these results show that the neural network trained by the proposed algorithm is more robust and efficient when it is used to predict the fault diagnosis.

Keywords: neural network, particle swarm optimization, genetic algorithm, elite strategy, fault diagnosis

I. INTRODUCTION

Among all kind of optimization methods, the particle swarm optimization (PSO) has become very popular and it was applied in many fields, such as manufacturing and robotics [1–3], electrical power systems [4–8], engineering [9–11] and others[12–18].

The PSO was introduced for unconstrained continuous problems [19-20]. The initialization of PSO is done with a population of random particles, and each particle adjusts its position according to its own experience and the experience of neighboring particles. Moreover, the PSO algorithm was improved by many researchers in recent years because the PSO is a very simple theoretical framework [19] and easy to be coded and implemented. In addition, the PSO technique can obtain a high quality solution within shorter calculation time and it is stable convergence characteristics than other stochastic methods [21].

Although PSO has fast convergence behavior, there is some deficiency in the performance of PSO. The PSO algorithm was improved by many researchers in recent years to resolve the premature convergence. In [22], a new approach to particle swarm optimization algorithm is presented. The proposed algorithm concern the particle swarm optimization algorithm as a model of predator and prey. In[23], an efficient global particle swarm optimization (GPSO) algorithm is proposed which is based on a new updated strategy of the particle position. The strategy can enhance the exploration capability of the GPSO algorithm and avoid the local minimal. In [24], the authors try to improve the performance of the PSO by incorporating the linkage concept, which is an essential mechanism in genetic algorithms, and design a new linkage identification technique that is used to address the linkage problem in real-parameter optimization problems.

During the past decades, artificial neural network (ANN) has been widely used in many different domains [25–27], such as load forecasting [28,29] and fault diagnosis. Compared with other methods, ANN methods are advantageous in terms of high data error tolerance, easy adaptability to online measurements, no need for excess information. A back Propagation Neural Network (BPNN) [30] is a typical artificial neural network. It is essentially a mapping function from input vector(s) to output vector(s) without knowing the correlation between the data. It can implement any complex nonlinear mapping function proved by mathematical theories, and approximate an arbitrary nonlinear function with satisfactory accuracy [31].

But the BP neural network easy to get local optimal, and the convergence speed slower in late training, so some researchers tried to introduce inspired algorithm to optimize the BP neural network to find the optimal weights and biases of the network before training, thus speeding up the convergence speed of training and effectively avoiding algorithm into local optimal at the same time. In[32], a new evolutionary artificial neural network (ANN) algorithm based on an improved PSO is presented. The improved PSO employs parameter automation strategy, velocity resetting, and crossover and mutations to significantly improve the performance of the original PSO algorithm in global search and fine-tuning of the solutions. In[33], a method that combines PSO-BP with comprehensive parameter selection is proposed. The IS-PSO-BP is short for Input parameter Selection IS-PSO-BP, where IS stands for Input parameter Selection. To evaluate the forecast performance of proposed approach. In[34], a model is established by combining GA with BP neural network. The model makes use of GA to solve global optimization problems, and construct neural network model. In[35], a hybrid algorithm is proposed for training ANN. The ability of metaheuristics and greedy gradient based algorithms are combined to obtain a hybrid improved opposition based particle swarm optimization and a back propagation algorithm with the momentum term.

Although these researchers have shown that evolutionary algorithms performs well for training neural network because it is capable of quickly finding and exploring promising regions in the search space, they take relative inefficiency in fine tuning solutions and may fall in the local optimal or cannot find the global best solution. Moreover, a potentially dangerous property in PSO still exists: stagnation due to the lack of momentum, which makes it impossible to arrive at the global optimum. By embedding of the genetic operators in the PSO, the balance between the exploration and exploitation ability is improved further. Thus, the idea of the proposed algorithm is to combine the ability of social thinking in PSO with the global search capability of GA to improve the ability of the hybrid algorithm.

The rest of the paper is organized as follows. The related works for algorithm is developed in Section 2. In section 3, a hybrid of genetic algorithm(GA) and particle swarm optimization (PSO) is presented. In section 4, a series of experiment is performed on six benchmark functions to confirm the performance of the proposed algorithm. Section 5 describes the BP neural work and the model of the BP neural network optimized by the HGAPSO. And the application of the model used in machinery fault diagnosis is developed in section 6. Finally, the conclusions are given in Section 7.

II. RELATED WORKS

The proposed algorithm combines GA with PSO to form a hybrid PSO. This hybrid of a PSO with existing algorithms can always produce a better algorithm than either the GA or the existing algorithms alone. In this section, basic concepts of GA and PSO are introduced, followed by a detailed introduction of hybrid PSO-GA algorithm in the next section.

2.1 Particle Swarm Optimization

The PSO is inspired from the strategy that is to search the surrounding area of the bird which is the closest to the food and it is used to settle optimization problems [19-20]. The bird has been assumed as the particle with no quality and volume and it has been extended to *N*-dimensional space where the position of particle *i* is expressed as vector $x_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]$ and its flying velocity as vector $V_i = [V_{i1}, V_{i2}, \ldots, V_{iN}]$. For the *kth* iteration, every particle in the PSO changes according to the equation as follow:

$$v_i^{k+1} = \omega v_i^k + c_1 rand() \times (P_i - x_i^k) + c_2 rand() \times (P_g - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Particle *i* decides its next position according to equation (1) and (2). And the movement of these particle in the swarm is shown in Fig.1 .

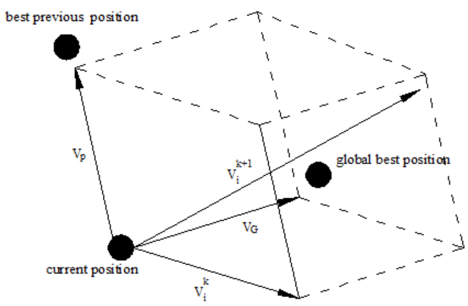


Fig.1 Particle movement in the swarm

Where $\upsilon_p = c_1 rand() \times (P_i - x_i^k)$ and $\upsilon_G = c_2 rand() \times (P_g - x_i^k)$

2.2 Genetic Algorithm

In GA[36], a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. Each individual represents a point in the search space, and hence a possible solution to the problem. A population consists of a finite number of individuals. Each individual is decided by an evaluating mechanism to obtain its fitness value. Based on this fitness value and undergoing genetic operators, a new population is generated with each successive population referred to as a generation. The GA use three basic operators (selection, crossover, and mutation) to manipulate the genetic composition of a population. Selection is a process by which the most highly rated individuals in the current generation are selection to produce the offspring in the new generation. The crossover operator produces two off-springs (new candidate solutions) by recombining the information from two parents. There are two processing steps in this operation. In the first step, a given number of crossing sites are selected uniformly, along with the parent individual at random. In the second step, two new individuals are formed by exchanging alternate pairs of selection between the selected sites. Mutation is a random alteration of some gene values in an individual. The allele of each gene is a candidate for mutation, and its function is determined by the mutation probability.

III. HYBRID OF PSO AND GA

Inspired by previous studies[37-39], the new offspring generated by GA and their parents are combined and sorted. Thereafter, the best *M* members are chosen to form the population of the next generation. It provides the chance of survival for parents in the next generation, hence the overall rapid convergence are enhanced. In addition, in the later process of the evolution, a large variance of variables will result in the long convergence history.

The proposed hybrid method is done in a two-step process. A generation has *M* particles and their values are evaluated. The first step is named PSO update step. It will be done on the best *n* particles and produce n offspring to be used in the next generation. The number of the best particles will be discussed in this paper, then the proportion will be decided according to the performance. The rest particles will not be discarded to make room for the new offspring generated by the GA step. Second, in the GA step, the rest particles will be generated by using genetic algorithm to improve the diversity of the swarm. And the genetic variants are taken by crossing over with the rest particles and dynamic mutation to generate other members. As a result, the new generation will have *M* particles whose values will be evaluated again. The flowchart showing the hybrid PSO-GA is presented in Fig. 2.

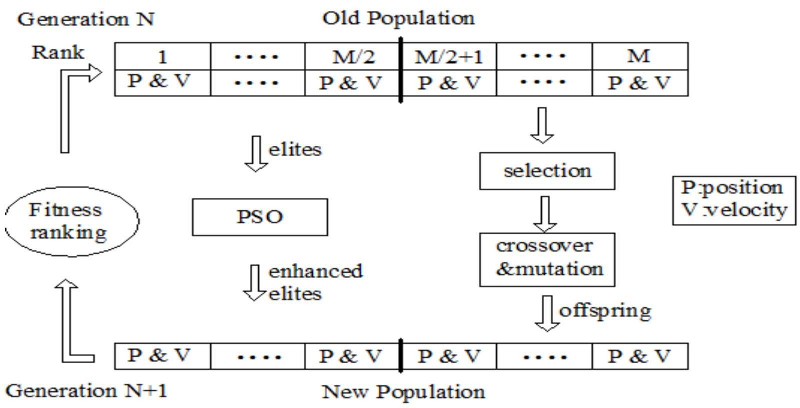


Fig.2: Flow of the HGAPSO

IV. TEST FUNCTIONS AND EXPERIMENTAL RESULTS

Real problems nowadays are more and more complex. Their objective functions are often multimodal with peaks, valleys, channels, and flat hyper-planes of different heights. Solving these types of problems, which are classified as global optimization problems, to optimal these problems undoubtedly becomes a true challenge. Test functions have many characteristics for simulating the complexity of most real applications. For example; multimodal functions are used to test the ability of an algorithm to escape from any local minimum.

4.1 Benchmark And Experimental Settings

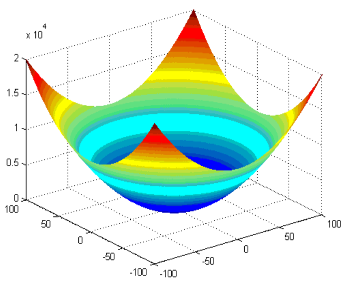
Twelve well-known classical benchmark functions [40] that consist of unimodal, multimodal, separable, and non-separable types are used to evaluate the performance of some algorithms. In this paper, the HGAPSO is tested on six standard benchmark functions to confirm its performance. The equations of these functions are shown in Table 1 and the figures of these functions are shown in Fig.3.

Table 1 Test functions

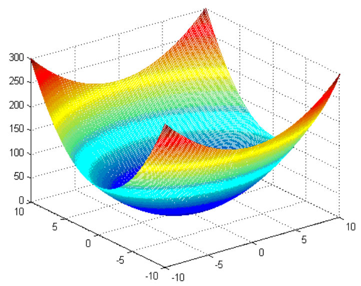
Test function	Dimension	Range of search

$f_1(x) = \sum_{i=1}^{n} x_i^2$	20	[-100,100]
$f_2(\mathrm{x}) = \sum_{i=1}^{n} i x_i^2$	20	[-10,10]
$f_3(\mathrm{x}) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	20	[-10,10]
$f_4(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20$	20	[-32,32]
$f_5(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$	20	[-5,5]
$f_6(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	20	[-10,10]

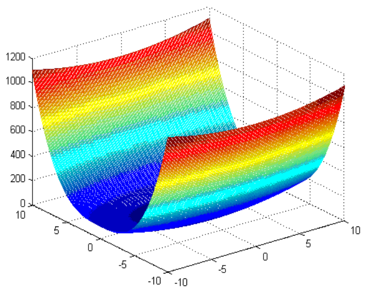
In the Fig.3, the figures of six benchmark functions when the dimension is 2 is shown. And the first three functions are unimodal with only one optimal, which is used to confirm the ability of the HGAPSO in finding the global best solution. The others are multimodal with some local optima, which can be used to confirm that the HGAPSO can avoid the local minimal. The HGAPSO has also been tested against with GA and improved PSO algorithm in [41-43].



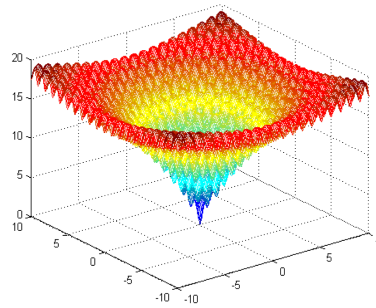
(a)



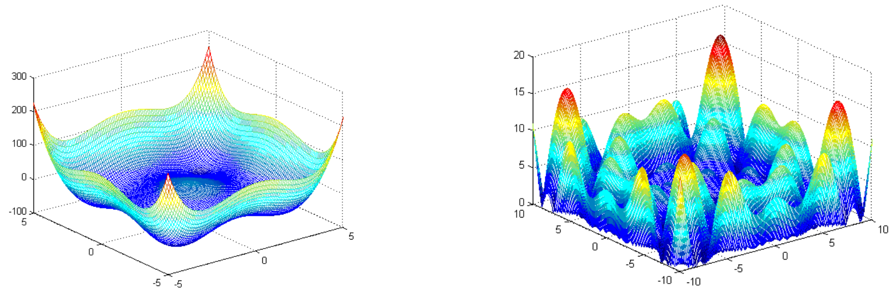
(b)



(c)



(d)



(e) (f)

Fig.3: The figure of benchmark function

(a) test function 1 (b) test function 2 (c) test function 3 (d) test function 4 (e) test function 5 (f) test function 6

4.2 Comparison Results

The comparison among different algorithms is based on four types of performance indexes that is the average value, the best value, the worst value and the standard deviation over 50 runs. The computation results of these algorithms are demonstrated in Table 2.

Tables 2 shows the results of the best, worst, average and the standard deviation values that are achieved by the PSO, IPSO and the HGAPSO for six benchmark functions in the same dimensional sizes. The standard optimal value for function 1 to 4 and function 6 in this analysis is zero. Whereas, The standard optimal value for function 5 in this analysis is smaller. From Table 2, it can be seen that the proposed algorithm is superior to other algorithms not only in the average value, the best value and the worst value, the standard deviation of the proposed algorithm tested in six benchmark functions is also far less than other algorithm. What's more, since the dimension of each test function is high, the PSO and IPSO algorithms could not provide the global optimal value in the same manner that the HGAPSO could. Thus, the HGAPSO algorithm can overcome the problem of falling in the local optima but others can't. From the experimental result, it is clear that the mean fitness and standard deviation of PSO are big of the first two functions, therefore, this algorithm also has bigger fluctuations when solving some simple functions. Meanwhile, from the performance of these improved PSO algorithms, it can be seen that these improved PSO algorithms can't solve these functions all with good performances, so there are some flaws of these improved PSO algorithms. Therefore, it can be seen from the above experimental result that the HGAPSO is a suitable tool to solve the global optimization problems of complicated functions.

Table 2: The computation results of different algorithms

Function		PSO	In [41]	In [42]	In [43]	HGAPSO
f_1	Best	5.5877E-02	1.3431E-02	1.4784E-03	7.9793E-02	1.5015E-17
	Worst	6.7045E+00	2.8484E+00	4.3045E-01	3.5472E+00	2.1862E-16
	Average	1.0249E+00	3.9260E-01	1.1762E-01	9.8227E-01	8.7670E-17
	St.Dev	1.2202E+00	5.3264E-01	1.0454E-01	8.8799E-01	5.0651E-17
f_2	Best	1.9766E-02	2.1550E-02	2.8987E-03	3.0987E-01	6.8852E-15
	Worst	1.1894E+01	4.4995E+00	2.6397E+00	1.0157E+01	7.0592E-11
	Average	1.3062E+00	5.0634E-01	4.0032E-01	2.0803E+00	6.2126E-12
	St.Dev	1.9546E+00	7.7339E-01	5.2785E-01	1.6967E+00	1.2899E-11
f_3	Best	3.8404E-06	1.9732E-06	1.4765E-06	1.3954E-05	1.7164E-11
	Worst	2.7290E+01	1.1016E+03	1.0639E+02	3.8092E+00	1.9888E-07
	Average	1.5786E+00	4.1515E+01	3.9266E+00	1.5014E-01	9.8060E-09
	St.Dev	5.0268E+00	1.9025E+02	1.9237E+01	5.7379E-01	3.0401E-08

f_4	Best	1.8567E+00	2.0292E+00	1.8557E+00	1.8864E+00	1.0489E-08
	Worst	5.5275E+00	5.4745E+00	5.8222E+00	5.3541E+00	2.0133E+00
	Average	3.7516E+00	3.6462E+00	3.4811E+00	3.4858E+00	4.0266E-02
	St.Dev	7.7320E-01	8.7155E-01	9.1665E-01	8.2545E-01	2.8473E-01
f_5	Best	-7.2596E+01	-7.4090E+01	-7.1264E+01	-7.4197E+01	-7.8332E+01
	Worst	-5.9886E+01	-6.1363E+01	-5.9953E+01	-5.7742E+01	-7.2678E+01
	Average	-6.6605E+01	-6.7325E+01	-6.6566E+01	-6.4299E+01	-7.6229E+01
	St.Dev	3.1008E+00	3.0139E+00	2.4808E+00	3.5653E+00	1.5104E+00
f_6	Best	2.3919E-02	4.0090E-02	3.7182E-02	4.2645E-01	8.7759E-08
	Worst	8.4541E+00	5.1556E+00	5.1195E+00	1.4823E+01	1.5293E-05
	Average	2.2566E+00	1.7906E+00	1.3084E+00	3.6955E+00	1.5081E-06
	St.Dev	1.7580E+00	1.3546E+00	1.1070E+00	2.7761E+00	3.2340E-06

V. HGAPSO-BP NEURAL NETWORK

Artificial neural network (ANN) is established by referring to the structure and characteristics of human brain, which is interconnected with a large number of simple processing units. And ANN is a nonlinear dynamical system to realize large-scale parallel distributed information processing. Compared with conventional information processing methods, ANN has some characteristics including structure variability, high nonlinearity, self-learning and self-organization ,etc. As a kind of ANN, BP neural network has been one of the most widely used network nowadays.

5.1 BP neural network

The BP neural network is used as a baseline method and it is described briefly in this section. The structure of a BP neural network is shown in Fig. 4. Each node in the network is a neuron whose function is to calculate the inner product of the input vector and weight vector by a nonlinear transfer function to get a scalar result. This particular network is a three-layer network: the input layer, the hidden layer and the output layer.

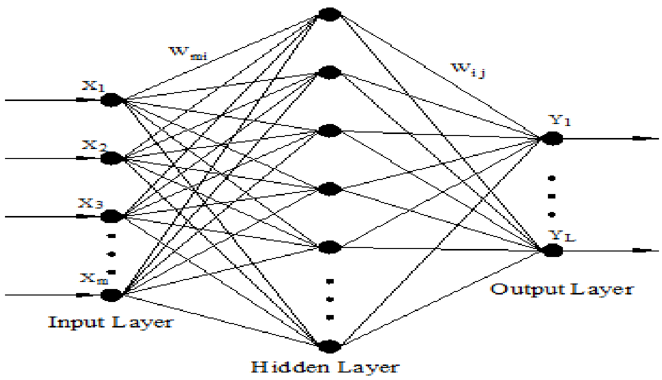
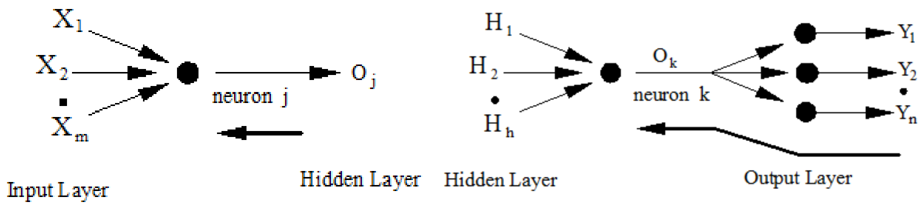


Fig.4: The structure of the BP neural network

In the Fig.4 , the number of the input neuron is *Q*, the number of the output neuron is *J* and the number of the hidden neuron is *I*. Before the design of a neural network, the number of the input neuron and the number of the output neuron are decided by the set of the sample used to train the neural network. The number of the hidden neuron is decided by some empirical equations [44].

5.2 The back-propagation learning algorithm

Error back-propagation divides the learning process to two stages. The first stage is forward propagation. Via the input layer, the input is processed by the hidden layer to obtain the actual output of each neuron. The second stage is back propagation. If the output layer can't get the expected value, the error that is the difference between the actual output and the expected value, is layer-by-layer calculated recursively. Then the error is used to adjust the weight between different layers. And nodal bias using the chain rule as shown in Fig.5.



(a) Input Layer (b) Hidden Layer

Fig.5: Error back-propagation

It is proved that a BP neural network with hidden layers can approximate any nonlinear continuous function with any accuracy. Then define a BP neural network with a hidden layers and h neurons. Each neuron just receives the outputs of previous layer, and outputs to the neurons of the next layer.

5.3 The HGAPSO-BP learning algorithm

The BP neural network optimized by the HGAPSO is also called HGAPSO-BP algorithm, it takes the weights and biases of neurons trained as one particle for the proposed algorithm. The fundamental idea of HGAPSO-BP algorithm can be described in Fig.6.

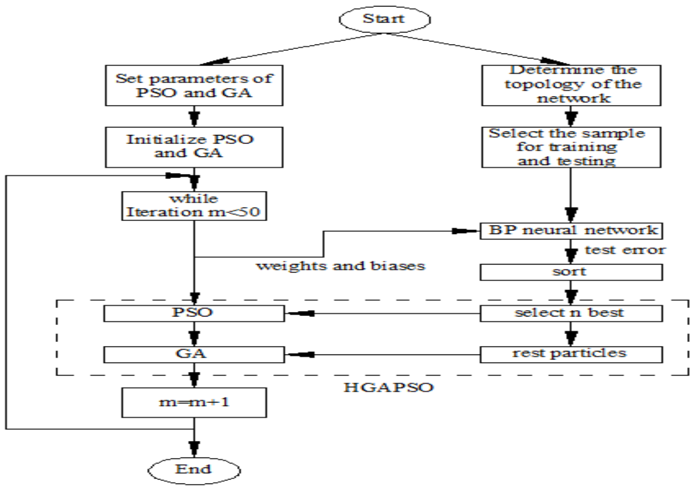
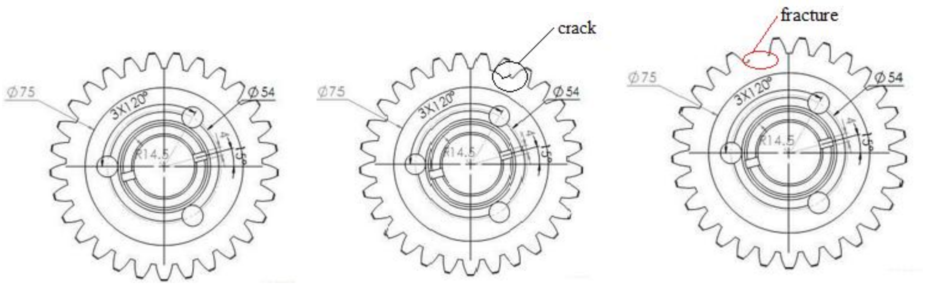


Fig.6: The flowchart of HGAPSO-BP

VI. SIMULATION AND RESULTS

Statistics show that about 60% of all failures for a gearbox failure is caused by the gear, so the failure of the gear is studied in this paper. For the study of the failure of gear, the two mainly failures for the gear are the crack in the bottom of gear and the fracture of the gear. And there are figures used to show the status of the gear (Fig.7). Furthermore, the status of the gear is decided by fifth feature values and these values are the sample data used to be trained and tested in the neural network.

In this paper, the three kinds status of the gear is defined as vector to be recognized by the neural network and the normal status is (1,0,0), the fault of the crack in the bottom of gear is (0,1,0) and the fault of the fracture of the gear is (0,0,1).



(a) (b) (c)

Fig.7: The status of gear
(a) normal (b) the crack in the bottom of gear (c) the fracture of the gear

In the design of the neural network, the most important task is to decide the number of neuron in each layer. From the description of the problem mentioned above, it can be seen the output is decided by fifth feature values and the output is defined as a vector, so the number of neurons in input layer is 15 and the number of neurons in output layer is 3. There are many empirical equations can be used to compute the number of neurons in hidden layer. So, the number of neurons in hidden layer will be decided by the performance of these empirical equations and the performance is demonstrated in the error of the neural network. From equation (3), it can be computed the number of neurons in hidden layer shown in Table 3 and the performance is shown in Table 4.

Table 4 shows the results of the best, worst, average and the standard deviation values that are obtained when the neurons in hidden layer is different. In the criterion shown in Table 4, the range of the performance can be seen from the best value and the worst value. And the stability can be seen from the average value and the standard deviation. From Table 4, it can be seen when the number of neurons is 12, the best value is prior to other and when the neurons is 14 and 16, the worst value is superior to other neurons. Meanwhile, the average and the standard deviation is the best when the number of neurons is 16. Therefore, the number of neurons in hidden layer is 16 to get the best performance.

Table 3: The number of neurons computed by empirical equations

Empirical equation	Range
$I = \sqrt{Q + J} + c, c \in [1,10]$	[6,15]
$I = \log_2 Q$	4
$I = \sqrt{QJ}$	7
$I = 2Q + 1$	31
$I \le Q + 1$	[0,16]

Table 4: The performance of neural network with different neurons in hidden layer

Error	4	6	7	8	9	10	11
Best	4.3615E-03	1.4077E-03	3.3410E-03	5.1981E-03	4.5023E-03	1.6927E-03	3.5295E-03
Worst	1.4143E+00	1.4142E+00	1.7320E+00	1.7321E+00	1.7321E+00	1.4132E+00	1.4127E+00
Average	4.0187E-01	2.2391E-01	3.3449E-01	3.6695E-01	2.9966E-01	2.4718E-01	1.6033E-01
St.Dev	4.9922E-01	4.2101E-01	5.0586E-01	5.2362E-01	4.8530E-01	4.2841E-01	3.5823E-01

Error	12	13	14	15	16	31

Best	1.0813E-03	1.1500E-03	6.1742E-03	1.5408E-03	1.3599E-03	2.2614E-03
Worst	1.6180E+00	1.0015E+00	1.0001E+00	1.4084E+00	1.0001E+00	1.4142E+00
Average	1.7515E-01	2.0371E-01	1.8071E-01	1.4479E-01	1.2742E-01	2.2281E-01
St.Dev	3.8181E-01	3.7926E-01	3.6150E-01	3.4878E-01	2.9535E-01	4.1773E-01

In order to compare the performance of the HGAPSO-BP algorithm with other evolutionary algorithms, the GA-BP, PSO-BP, improved PSO-BP with nonlinear weights and a hybrid GA-PSO based on GA[45]are implemented. Table 5 shows the numerical results of the best value, the worst value, the average and the standard deviation of the tested error back from BP neural network of the 50 independent runs produced by both learning algorithms. The four integral criteria can describe the performance of learning algorithm. The range of the performance can be seen from the best value and the worst value. And the stability can be seen from the average value and the standard deviation.

Table 5 shows the values of best, worst, average and the standard deviation that are achieved by the PSO, GA, improved PSO with nonlinear weight, hybrid GA and PSO based on GA and the proposed algorithm when these algorithms are used as learning algorithm for BP neural network and applied on the fault diagnosis of the gear. The ideal optimal value for the tested error produced from the test sample of the gear's diagnosis in this analysis is that the fitness value is zero.

From the value of best value, worst value, the average and the standard deviation obtained from the proposed algorithm, it can be seen that the proposed algorithm has the minimum value between these algorithms, and the hybrid GA and PSO based on GA algorithm is second to the proposed algorithm on the worst value, average and the standard deviation and the BP learning algorithm get the worst value in all aspects. And since the population size of each algorithms is high, the PSO-BP and improved PSO with nonlinear weight algorithms could not provide the global optimal value in the same manner that the proposed algorithm could. From these results, it can be concluded that the evolutionary algorithm is useful to train neural network and the hybrid algorithm can improved the performance compared to these algorithm alone. Thus, the proposed algorithm has a higher accuracy and stability than other algorithms.

Table 5: The results of the error for both learning algorithm

Algorithms	BP	GA-BP	PSO-BP	Improved PSO with nonlinear weight	Hybrid GA and PSO based on GA	HGAPSO-BP
Best	1.3599E-03	1.2510E-03	2.4143E-06	6.9256E-13	8.4927E-08	2.1843E-141
Worst	1.0001E+00	5.3918E-03	8.6337E-03	3.2219E-03	9.3893E-04	1.2921E-04
Average	1.2742E-01	3.2141E-03	9.2315E-04	5.4572E-04	3.2621E-04	3.7925E-05
St.Dev	2.9535E-01	1.0860E-03	1.4093E-03	7.7790E-04	2.9709E-04	3.6730E-05

VII. CONCLUSION

An evolutionary back-propagation neural network (BP) which automates the design of BP networks using a new evolutionary learning algorithm is proposed in this paper. This new evolutionary learning algorithm is based on a hybrid of genetic algorithm(GA) and particle swarm optimization (PSO), and is thus called HGAPSO. The performance of proposed algorithm is tested on 6 classical benchmark functions with high dimensional size and compared with other inspired algorithm. The simulation results show that the proposed algorithm method performed well in solving high dimensional problems. The update by performing crossover and mutation operation to generate the next generation in the proposed algorithm did not only help the algorithm reach the global optimal results in a few iterations, but it also prevented the proposed algorithm from being trapped in local optimal results.

Based on the proposed algorithm, the proposed algorithm is applied to train the BP neural work for the fault diagnosis. And the proposed algorithm is mainly used in adjusting the weight and bias of the neural network to test the sample data using to diagnose which type fault is produced. The results showed that the proposed algorithm is effective and has more excellent performance compared to other algorithms. It can be concluded that the proposed algorithm is not only a suitable tool to solve the global optimization problems of complicated functions, but also a superior method to adjust the weight and bias of the neural network compared to other evolutionary algorithms.

REFERENCES

[1]. T. Navalertporn and N. V. Afzulpurkar, "Optimization of tile manufacturing process using particle swarm optimization," Swarm and Evolutionary Computation, vol. 1, no. 2, pp. 97–109,2011.
[2]. G. S. Chyan and S. G. Ponnambalam, "Obstacle avoidance control of redundant robots using variants of particle swarm optimization," Robotics and Computer-Integrated Manufacturing, vol. 28, no. 2, pp. 147–153, 2012.
[3]. Q. Tang and P. Eberhard, "A PSO-based algorithm designed for a swarm of mobile robots," Structural and Multidisciplinary Optimization, vol. 44, no. 4, pp. 483–498, 2011.
[4]. D. N. Jeyakumar, T. Jayabarathi and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," International Journal of Electrical Power and Energy Systems, vol. 28, no. 1,
[5]. pp.36–42, 2006.
[6]. Q. Kang, T. Lan, Y. Yan, L. Wang and Q. Wu, "Group search optimizer based optimal location and capacity of distributed generations," Neurocomputing, vol.78, no. 1, pp. 55–63, 2012.
[7]. J. Y. Kim, K. J. Mun, H.S. Kim and J. H. Park, "Optimal power system operation using parallel processing system and PSO algorithm," International Journal of Electrical Power and Energy Systems, vol. 33, no. 8, pp. 1457–1461, 2011.
[8]. Y. W. Ying, J. Zhou, C. Z. Chao Zhou, Y. W. Wang, H. Q. Qin and Y. L. Lu, "An improved self-adaptive PSO technique for short-term hydrothermal scheduling," Expert Systems with Applications, vol. 39, no. 3,
[9]. pp. 2288–2295, 2012.
[10]. G. Isazadeh, R. A. Hooshmand and A. Khodabakhshian, "Modeling and optimization of an adaptive dynamic load shedding using the ANFIS-PSO algorithm," Simulation, vol. 88, no. 2, pp. 181–196, 2012.
[11]. H. Liu, Z. Cai and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," Applied Soft Computing Journal, vol. 10, no. 2,
[12]. pp. 629–640, 2010.
[13]. A. Farshidianfar, A. Saghafi, S. M. Kalami and I. Saghafi, "Active vibration isolation of machinery and sensitive equipment using $H\infty$ control criterion and particle swarm optimization method," Meccanica, vol. 47, no. 2, pp. 437–453, 2012.
[14]. S. M. Seyedpoor, J. Salajegheh and E. Salajegheh, "Shape optimal design of materially nonlinear arch dams including dam-water-foundation rock interaction using an improved PSO algorithm," Optimization and Engineering, vol. 13, no. 1, pp. 79–100, 2012.
[15]. A. M. El-Zonkoly, "Optimal placement of multi-distributed generation units including different load models using particle swarm optimization," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 50–59, 2011.
[16]. R. J. Kuo and C. Y. Yang, "Simulation optimization using particle swarm optimization algorithm with application to assembly line design," Applied Soft Computing Journal, vol. 11, no. 1, pp. 605–613, 2011.
[17]. E. Peksen, T. Yas, A. Y. Kayman and C. Ozkan, "Application of particle swarm optimization on self-potential data," Journal of Applied Geophysics, vol. 75, no. 2, pp. 305–318, 2011.
[18]. W.T. Pan, "Combining PSO cluster and nonlinear mapping algorithm to perform clustering performance analysis: take the enterprise financial alarming as example," Quality and Quantity, vol. 45, no. 6,
[19]. pp. 1291–1302, 2011.
[20]. L. Goel, D. Gupta and V. K. Panchal, "Hybrid bio-inspired techniques for land cover feature extraction: a remote sensing perspective," Applied Soft Computing Journal, vol. 12, no. 2, pp.832–849, 2012.
[21]. H. M. Jiang, C. K. Kwong, W. H. Ip and T. C. Wong, "Modeling customer satisfaction for new product development using a PSO-based ANFIS approach," Applied Soft Computing Journal, vol. 12, no. 2,
[22]. pp. 726–734, 2012.
[23]. J. W. Li, Y. C. Chang, C. P. Chu and C. C. Tsai, "A self-adjusting e-course generation process for personalized learning," Expert Systems with Applications, vol. 39, no. 3, pp. 3223–3232, 2012.
[24]. J. Kennedy and R.C. Eberhart, "Particle swarm optimization," Proceedings of the International Conference on Neural Networks, pp. 1942-1948,1995.
[25]. H. Fan, "A modification to particle swarm optimization algorithm, "Engineering Computations, vol.19,
[26]. pp.970-989, 2002.
[27]. C.C. Kao, C.W. Chuang and R.F. Fung, "The self-tuning PID Control in a slider-crank mechanism system by applying particle swarm optimization approach," Mechatronics, vol.16, pp.513-522, 2006.
[28]. I. Gosciniak, "A new approach to particle swarm optimization algorithm," Expert Systems with Applications,vol.42,pp.844-854,2015.
[29]. J. J. Jamian, M. N. Abdullah, H. Mokhlis, M. W. Mustafa and A. H. A. Bakar, "Global Particle Swarm Optimization for High Dimension Numerical Functions Analysis," Hindawi Publishing Corporation Journal of Applied Mathematics, 2014.
[30]. Y. P. Chen, W. C. Peng and M. C. Jian, "Particle Swarm Optimization With Recombination and Dynamic Linkage Discovery," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.37, pp.1460-1470,2007.

[31]. E. Hadavandi, H. Shavandi, A. Ghanbari, "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting," Knowledge-Based Systems, vol.23,pp. 800–808, 2010.

[32]. A. Khosravi, S. Nahavandi, D. Creighton, "Quantifying uncertainties of neural network-based electricity price forecasts," Applied Energy,vol.112,pp. 120–129, 2013.

[33]. A. Khotanzad, H. Elragal, T.-L. Lu, "Combination of artificial neural-network forecasters for prediction of natural gas consumption," IEEE Transactions on Neural Networks vol.11,pp. 464–473, 2000.

[34]. Z. Bashir, M. El-Hawary, "Applying wavelets to short-term load forecasting using PSO-based neural networks," IEEE Transactions on Power Systems,vol.24, pp.20–27, 2009.

[35]. Z. Xiao, S. J. Ye, B. Zhong, C.-X. Sun, "BP neural network with rough set for shortterm load forecasting," Expert Systems with Applications, vol.36, pp.273–279, 2009.

[36]. D. E. Rumelhart, G. E. Hintont, R. J. Williams, "Learning representations by back propagating errors," Nature vol.323, pp.533–536, 1986.

[37]. L. Zhang, K. Wu, Y. Zhong, P. Li, "A new sub-pixel mapping algorithm based on a BP neural network with an observation model," Neurocomputing, vol.71, pp. 2046–2054, 2008.

[38]. J. B. Yu, L. F. Xi and S. J. Wang, "An Improved Particle Swarm Optimization for Evolving Feedforward Artificial Neural Networks," Neural Process Lett, vol.26, pp.217-231, 2007.

[39]. C. Ren, N. An, J. Z. Wang, L. Li, B. Hu, D. Shang, "Optimal parameters selection for BP neural network based on particle swarm optimization: A case study for wind speed forecasting," Knowledge-Based Systems,

[40]. vol.56, pp.226-239, 2014.

[41]. H. Li, C. X. Hu, Y. Li, "Application of the purification of Materials Based on GA-BP," Energy Procedia,

[42]. vol.17, pp.762-769, 2012.

[43]. M. Yaghini, M. M. Khoshraftar, M. Fallahi, "A hybrid algorithm for artificial neural network training,"

[44]. Engineering Application of Artificial Intelligence, vol.26, pp. 293-301, 2013.

[45]. D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning," MA:Addison-Wesley,

[46]. 1989.

[47]. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II,"

[48]. IEEE Trans Evolut Comput, vol.2, pp.182-197, 2002.

[49]. M. Basu, "Fuel constrained economic emission dispatch using non-dominated sorting genetic algorithm-II,"

[50]. Energy, vol.78, pp. 649-664, 2014.

[51]. G. D'Errico, T. Cerri , T. Pertusi, "Multi-objective optimization of internal combustion engine by means of

[52]. 1D fluid-dynamic models," Appl Energy, vol.88, pp. 767-777, 2011.

[53]. A. M. El-Zonkoly, "Optimal placement of multi-distributed generation units including different load models

[54]. using particle swarm optimization," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 50–59, 2011.

[55]. A. Taeib, A. Ltaief and A. Chaari, "PID control based on modified particle swarm optimization for nonlinear

[56]. process," 2013 World Congress on Computer and Information Technology (WCCIT), pp. 1-5, June. 2013.

[57]. N. Chaudhary, R. Raj, K. Kiran and S. Nema, "Improved asynchronous PSO based design of multivariable

[58]. PID controller", International Conference on Control, Automation Robotics and Embedded Systems (CARE),

[59]. pp. 1-4, Dec. 2013.

[60]. J. S. Chiou, S. H. Tsai and M. T. Liu, "A PSO-based adaptive fuzzy PID-controllers," Simulation Modelling

[61]. Practice and Theory, vol. 26, pp. 49-59, 2012.

[62]. L. W. Wang, "Determination of Number of ANNs' Hidden Layer Neurons[D]," Chongqing University, 2012.

[63]. H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problem," Applied Mathematics and

[64]. Computation, vol.274, pp. 292-305, 2016.