

Feedforward neural network training using intelligent global harmony search

Saeed Tavakoli · Ehsan Valian · Shahram Mohanna

Received: 8 May 2011 / Accepted: 12 March 2012 / Published online: 28 March 2012
© Springer-Verlag 2012

Abstract Harmony search algorithm is a meta-heuristic optimization method imitating the music improvisation process, where musicians improvise their instruments' pitches searching for a perfect state of harmony. First, an improved harmony search algorithm is presented using the concept of swarm intelligence. Next, it is employed for training feedforward neural networks for three benchmark classification problems. Then, the performance of the proposed algorithm is compared with that of three methods. Simulation results demonstrate the effectiveness of the proposed algorithm.

Keywords Artificial neural network · Classification · Harmony search · Optimization

1 Introduction

A typical artificial neural network (ANN) has two types of basic components. They are neurons which are processing elements and links which are interconnections between neurons. Each link has a weighting parameter. Each neuron receives stimulus from other neurons, processes the information, and produces an output. Neurons are categorized to input, output and hidden neurons. The first and the last layers are called input and output layers, respectively, and the remaining layers are called hidden layers (Zhang and Gupta 2003).

Consider N_k as the number of neurons in the k th layer. Let w_{ij}^k , represents the weight of the link between the j th neuron of the $(k - 1)$ th layer and the i th neuron of the k th layer. Suppose w_{i0}^k is an additional weighting parameter for each neuron, representing the bias for the i th neuron of the k th layer. The weighting parameters are initialized before training the neural network. During training, they are updated iteratively in a systematic manner (Wang et al. 1999). Once the neural network training is completed, the weighting parameters remain fixed throughout the usage of the neural network as a model.

The process of training an ANN is to adjust weights and biases. The back-propagation (BP) learning has become the most popular method to train feedforward ANNs in many domains (Chronopoulos and Sarangapani 2002). However, one limitation of this technique, which is a gradient-descent technique, is that it requires a differentiable neuron transfer function. Also, as neural networks generate complex error surfaces with multiple local minima, the BP tends to converge into local minima instead of a global minimum (Gupta and Sexton 1999).

In recent years, many improved learning algorithms have been proposed to overcome the handicaps of gradient-based techniques. These algorithms include a direct optimization method using a polytope algorithm (Curry and Morgan 1997), global search techniques such as evolutionary programming (Salchenberger et al. 1992), genetic algorithm (GA) (Sexton et al. 1998; Kim et al. 2005), ant colony optimization (ACO) (Socha and Blum 2007), particle swarm optimization (PSO) (Yu et al. 2008; Garro et al. 2009; Zamani and Sadeghian 2010), differential evolution (DE) (Slowik and Bialko 2008; Garro et al. 2010, 2011), and artificial bee colony algorithm (ABC) (Karaboga and Ozturk 2009). The standard gradient-descent BP is not trajectory-driven, but population driven. However, the

S. Tavakoli (✉) · E. Valian · S. Mohanna
Faculty of Electrical and Computer Engineering,
University of Sistan and Baluchestan, Zahedan, Iran
e-mail: tavakoli@ece.usb.ac.ir

improved learning algorithms have explorative search features. Consequently, these methods are expected to avoid local minima frequently by promoting exploration of the search space.

Harmony search (HS) is a meta-heuristic optimization method imitating the music improvisation process where musicians improvise their instruments' pitches searching for a perfect state of harmony (Lee and Geem 2004, 2005a; Lee et al. 2005b). HS algorithms have been successfully applied to a wide range of practical optimization problems, such as structural optimization (Lee and Geem 2004, 2005a; Lee et al. 2005b), parameter estimation of the nonlinear Muskingum model (Kim et al. 2001), design optimization of water distribution networks (Geem et al. 2002; Geem 2006, 2009a, b), vehicle routing (Geem et al. 2005), combined heat and power economic dispatch (Vasebi et al. 2007), design of steel frames (Degertekin 2008a, b), bandwidth-delay-constrained least-cost multicast routing (Forsati et al. 2008), transport energy modeling (Ceylan et al. 2008) and quadratic programming (Fesanghary et al. 2008). Also, several algorithms, such as novel global harmony search (NGHS) (Zou et al. 2010), have recently been developed based on the adoption of search behavior of the HS. In this paper, the swarm intelligence technique is employed to enhance the performance of the NGHS.

The paper is organized as follows. In Sect. 2, the procedure of HS and NGHS algorithms are briefly presented. The improved NGHS algorithm is presented in Sect. 3. In Sect. 4, this algorithm is employed to train a feedforward ANN. We end this paper with some conclusions in Sect. 5.

2 Harmony search and novel global harmony search algorithms

In this section, harmony search and NGHS algorithms are reviewed.

2.1 HS algorithm

In basic HS algorithm (Lee and Geem 2005a), each solution, called a harmony, is represented by an n -dimension real vector. First, an initial population of harmony vectors is randomly generated and stored in a harmony memory (HM). Then, a new candidate harmony is generated from all of solutions in the HM by performing a memory consideration rule, a pitch adjustment rule and a random re-initialization. Next, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. Finally, the worst harmony vector is replaced by the new candidate vector

if the latter is better. The above process is repeated until a certain termination criterion is met. The basic HS consists of three phases, which are initialization, improvisation of a harmony vector and updating the HM. These phases are described as follows.

2.1.1 Initialization of the algorithm parameters

In general, the global optimization problem can be written as

$$\begin{aligned} & \min f(X) \\ \text{s.t. : } & x(j) \in [LB(j), UB(j)], \quad j = 1, 2, \dots, n \end{aligned} \quad (1)$$

where $f(X)$ is the objective function, n is the number of decision variables, $X = [x(1), x(2), \dots, x(n)]$ is the set of decision variables, and $LB(j)$ and $UB(j)$ are the lower and upper bounds for the decision variable $x(j)$, respectively.

The parameters of the HS algorithm are harmony memory size (HMS), i.e., the number of solution vectors in harmony memory, harmony memory consideration rate (HMCR), pitch adjusting rate (PAR), distance bandwidth (BW), and number of improvisations (NI), i.e., the total number of function evaluations. Obviously, a good set of parameters can enhance the algorithm's ability to search for the global optimum or near optimum region with a high convergence rate.

2.1.2 Initialization of the harmony memory

The number of harmony vectors in the HM is HMS. Let $X_i = [x_i(1), x_i(2), \dots, x_i(n)]$, which has randomly been generated, represents the i th harmony vector, as follows

$$x_i(j) = LB(j) + (UB(j) - LB(j)) \times r \quad (2)$$

$$i = 1, 2, \dots, HMS, j = 1, \dots, n$$

where r is a uniform random number between 0 and 1. The HM matrix is then filled with harmony vectors as follows.

$$HM = [X_1, X_2, \dots, X_{HMS}]^T. \quad (3)$$

2.1.3 Improvise a new harmony

A new decision variable $x_{\text{new}}(j)$ is improvised by applying three rules, a memory consideration, a pitch adjustment and a random selection. First, a uniform random number r_1 is generated in the range $[0, 1]$. If r_1 is less than the value of HMCR, the decision variable $x_{\text{new}}(j)$ is generated by the memory consideration. Otherwise, $x_{\text{new}}(j)$ is obtained by a random selection, i.e., random re-initialization between the search bounds. In the memory consideration, $x_{\text{new}}(j)$ is

substituted by a random selection of the j th harmony vector in the HM. Next, if each decision variable $x_{new}(j)$ is updated by the memory consideration, it will undergo a pitch adjustment with a probability of PAR. The pitch adjustment rule is given by

$$x_{new}(j) = x_{new}(j) \pm r \times BW \tag{4}$$

where r is a uniform random number between 0 and 1.

2.1.4 Updating the harmony memory

Once a new harmony vector X_{new} is generated, the HM must be updated by the survival of the fitter competition between X_{new} and the worst harmony vector, X_w . If the fitness value of X_{new} is better than that of X_w , the latter will be replaced by X_{new} and become a new member of the HM.

2.1.5 Computational procedure

The computational procedure of the basic HS algorithm for a minimization problem can be summarized as follows (Lee and Geem 2005a).

- Step 1 Set the parameters HMS, HMCR, PAR, BW, and NI
- Step 2 Initialize the HM and calculate the objective function values for each harmony vector
- Step 3 Improvise a new harmony X_{new} as follows

```

for each  $j \in [1, n]$  do
  if ( $r_1 < \text{HMCR}$ ) then
     $x_{new}(j) = x_a(j)$  where  $a \in (1, 2, \dots, \text{HMS})$ 
  if ( $r_2 < \text{PAR}$ ) then
     $x_{new}(j) = x_{new}(j) \pm r \times \text{BW}$  where  $r \in (0, 1)$ 
  endif
else
   $x_{new}(j) = \text{LB}_j + r_3 \times (\text{UB}_j - \text{LB}_j)$  where  $r_1, r_2, r_3 \in (0, 1)$ 
endif
endfor
    
```

- Step 4 Update the HM as $X_{worst} = X_{new}$ if $f(X_{new}) < f(X_{worst})$
- Step 5 If NI is completed, return the best harmony vector X_{best} from the HM; otherwise go back to step 3.

2.2 Novel global harmony search algorithm

A recent proposed approach, called NGHS (Zou et al. 2010), modifies the improvisation step of the HS by substituting HMCR and PAR by genetic mutation probability p_m . The NGHS improvisation step is as follows.

```

for each  $j \in [1, n]$  do
   $x_R = 2 \times x_j^{best} - x_j^{worst}$ 
  if  $x_R > x_{ju}$ 
     $x_R = x_{ju}$ 
  elseif  $x_R < x_{jl}$ 
     $x_R = x_{jl}$ 
  endif
   $x_{new}(j) = x_j^{worst} + r_1 \times (x_R - x_j^{worst})$ 
  if  $r_2 < p_m$  then
     $x_{new}(j) = \text{LB}_j + r_3 \times (\text{UB}_j - \text{LB}_j)$  where  $r_1, r_2, r_3 \in (0, 1)$ 
  endif
endfor
    
```

where x_{jl} and x_{ju} refer to the lower and upper bands of decision variables, respectively. Here, ‘best’ and ‘worst’ are the indexes of the best and worst harmonies in HM, respectively. Figure 1 illustrates the principle of position updating. The adaptive step for the j th decision variable is defined as $\text{Step}_j = |x_j^{best} - x_j^{worst}|$. The region between P and R, which is actually a region near the global-best harmony, is defined as the trust region for the j th decision variable.

A balance between the global search and the local search is kept by dynamically adjusted step_j . To enhance the proposed algorithm’s capacity of escaping from the local optimum, a genetic mutation operation with a small probability is used. The worst harmony X_{worst} in HM is always replaced with the new harmony X_{new} .

3 Intelligent global harmony search (IGHS) algorithm

To solve continuous optimization problems, this paper presents a new optimization algorithm, named IGHS. The main motivation for presenting the IGHS is to enhance the ability of HS algorithms in dealing with such optimization problems. In this method, the concept of swarm intelligence, as proposed in PSO technique (Kennedy and Eberhart 1995), is employed to improve the performance of the third step. In PSO technique, a swarm of particles could

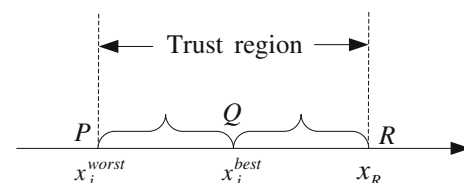


Fig. 1 The schematic diagram of position updating

fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle can be influenced by the best position visited by itself, i.e., its own experience, and the position of the best particle in the swarm, i.e., the experience of swarm.

To enhance the performance of the NGHS, this idea is used in IGHS to modify the improvisation step of the NGHS, as follows. The modification is done in such a way that the new harmony imitates one dimension of the best harmony in the HM.

In comparison with HS algorithms, the IGHS has the following differences.

1. The new harmony is chosen from the range between the values of worst and best harmonies.
2. To improve the performance of this algorithm, the best harmony in HM is used in the third step of the IGHS.
3. To have a suitable population diversity and a good search in the search space, the worst harmony in HM is always replaced with a new harmony.

The computational procedure of the basic IGHS algorithm for a minimization problem can be summarized as follows.

- Step 1* Set the parameters HMS, HMCR, PAR and NI
Step 2 Initialize the HM and calculate the objective function values for each harmony vector
Step 3 Improvise a new harmony X_{new} as follows

```

for each  $j \in [1, n]$  do
  if  $(r_1 < HMCR)$  then
    if  $(r_2 < PAR)$  then
       $x_{new}(j) = x_j^{best}$ 
    else
       $x_R = 2 \times x_j^{best} - x_j^{worst}$ 
      if  $x_R > x_{ju}$ 
         $x_R = x_{ju}$ 
      elseif  $x_R < x_{jl}$ 
         $x_R = x_{jl}$ 
      endif
       $x_{new}(j) = x_j^{worst} + r \times (x_R - x_j^{worst})$  where  $r \in (0, 1)$ 
    endif
  else
     $x_{new}(j) = LB_j + r_3 \times (UB_j - LB_j)$  where  $r_1, r_2, r_3 \in (0, 1)$ 
  endif
endfor

```

- Step 4* Update the HM as $X_{worst} = X_{new}$
Step 5 If NI is completed, return the best harmony vector X_{best} from the HM; otherwise go back to step 3.

4 Simulation results, analysis and discussion

When ANN training is initiated, the iterative process of presenting the training patterns of the dataset to the network's input continues until a given termination condition is satisfied. This usually happens based on a criterion indicating that the current achieved solution is presumably good enough to stop training. For instance, one common termination criterion in the BP is the difference between the current value of sum of squared errors (SSE) and that obtained in the previous iteration (Fausett 1994).

Figure 2 illustrates a small scale sample ANN. A harmony vector in the HM represents the decision variables in the HS. Each vector represents a complete set of ANN weights including biases. The objective function is to minimize SSE (Hassoun 1995). The squared difference between the target output and actual output determines the amount of error. This is represented by $(t - z)^2$ for each pattern and each output unit as shown in Fig. 2. Since the values of ANN weights are usually within the same range we simplify the HS model by setting $x_L = -1$, $x_U = 1$.

In this study, three benchmark classification problems, which are classification of Iris, Breast Cancer, and Glass, are studied. According to four properties of one flower, the categories of Iris are identified in the Iris problem. This problem consists of 150 examples; all of them can be classified into three categories whilst each category accounts for one-third of the examples. 101 examples are used for training and the rest are used for testing. In the Breast Cancer problem, we correctly diagnose breast lumps as either benign or malignant based on data from examination of cells. This problem consists of 699 examples. Each example includes nine inputs and two outputs. 500 examples are used for training and the rest are used for testing. The Glass dataset predicts the type of glass, either window glass or non-window glass, based on measurements of the chemical content. This problem consists of 214 examples. Each example includes 9 inputs and 2 outputs. 154 examples are used for training and 60 examples are used for testing. To evaluate the performance of the proposed algorithm, we compare its computational results with those provided by the BP, HS, and NGHS.

A benchmark dataset is partitioned into training, validation and testing sets. To prevent over-fitting and, therefore, to improve the classification performance in the test data, training and validation sets may be used individually (Hassoun 1995). However, since evolutionary-based

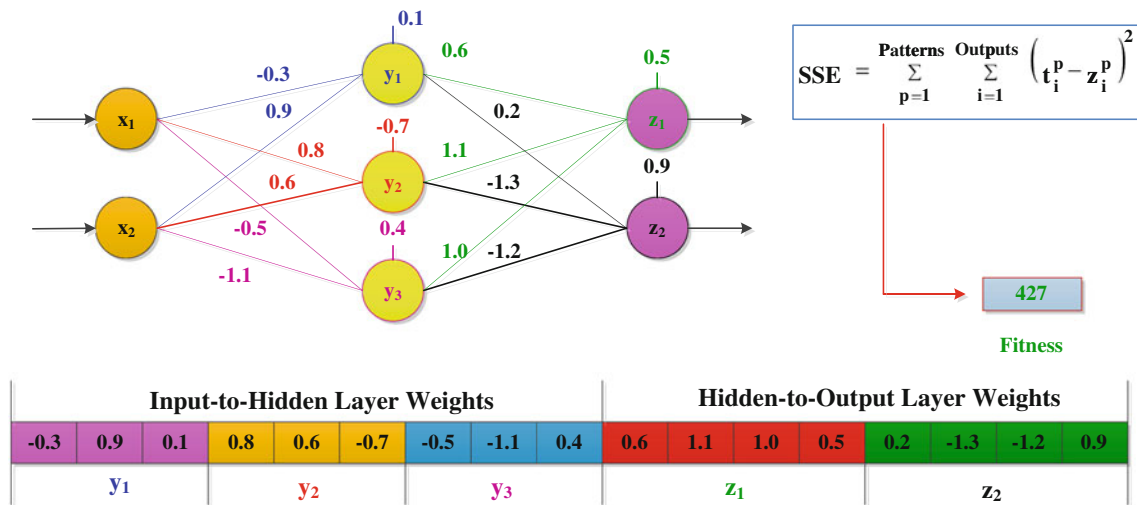


Fig. 2 A small scale sample ANN

techniques search globally for a solution the use of validation set is not required for them (Sexton and Dorsey 2000; Kiranyaz et al. 2009).

In the BP, the learning rate η is 0.7 whilst 5 % of training data is used for validation check. Table 1 shows the parameter settings of all HS algorithms including IGHS algorithm. Here, ‘NI’ represents the number of iterations. The parameters have been chosen according to the recommendations given by related references (for example, see Lee and Geem 2005a; Zou et al. 2010) and also based on results of simulations.

For all benchmark problems, the number of hidden neurons for all algorithms is 8. The number of neurons has been chosen to be less than the number of inputs. This increases the speeds of the algorithms. Also, the number of tuning parameters (i.e., neurons and biases) is set to be less than that of training data. This results in a good generalization for the network.

After 20 iterations, the results of classification are demonstrated in Table 2. In this table, the parameter ‘Std’ represents the standard deviation. The values in bold refer to the best results. The results show that the NGHS and IGHS perform better than the BP and HS in terms of classification accuracy and computational expenses. Furthermore, Table 2 shows that the IGHS has a better overall performance in comparison with the NGHS. Moreover, the simulation studies reveal that among all these algorithms the IGHS has a higher chance to find a good solution for a given number of iterations.

The lowest SSE values are found by the BP for iris dataset and by the IGHS for Breast Cancer and Glass datasets. It can be seen from Table 2 that the proposed IGHS algorithm finds higher testing accuracies than BP and

Table 1 Parameter settings of HS algorithms

Algorithm	HMS	HMCR	PAR	BW	p_m	NI
HS	5	0.9	0.3	0.01	–	5,000
NGHS	5	–	–	–	0.005	5,000
IGHS	5	0.995	[0.1, 0.2]	–	–	5,000

other HS algorithms. This means that IGHS algorithm can classify datasets better than the other algorithms considered in this paper.

5 Conclusions

Using the concept of swarm intelligence, this paper presented an intelligent global harmony search algorithm, called IGHS. It was used for training feedforward neural networks for three benchmark classification problems, namely Iris, Breast Cancer, and Glass problems. The performance of the proposed algorithm was compared with that of the BP, HS and NGHS.

Taking classification accuracy and computational expenses into account, simulation results revealed that NGHS and IGHS algorithms performed better than the BP and HS. In comparison with the NGHS, it was also shown that the IGHS had a better overall performance. Moreover, the simulation studies proved that the IGHS had a higher chance to find a good solution for a given number of iterations in comparison with other algorithms. The lowest SSE values were found by the BP for Iris dataset and by the IGHS for Breast Cancer and Glass datasets. Furthermore, the IGHS found higher testing accuracies than BP and

Table 2 Simulation results for benchmark problems

Algorithm	Breast Cancer dataset			Iris dataset			Glass dataset		
	SSE	Training accuracy	Testing accuracy	SSE	Training accuracy	Testing accuracy	SSE	Training accuracy	Testing accuracy
BP									
Min.	47.15	96.99	87.14	9.49	55.76	53.33	19.70	76.67	75.97
Average	64.99	97.68	96.00	19.75	92.63	91.33	47.85	80.75	81.46
Max.	84.36	98.56	7.10	33.65	97.52	98.70	65.40	91.67	94.16
Std.	15.48	0.47	3.49	11.77	13.94	10.76	15.76	6.02	7.74
HS									
Min.	57.00	94.80	95.98	30.23	73.27	67.35	19.85	86.64	81.67
Average	67.46	95.81	96.98	34.52	88.12	83.67	27.01	91.27	89.99
Max.	75.09	96.20	97.49	36.05	97.03	97.96	35.22	93.51	93.33
Std.	7.23	0.17	0.36	1.89	8.91	11.66	4.00	6.49	6.67
NGHS									
Min.	41.92	95.80	97.99	22.85	95.05	95.92	17.56	88.32	85.00
Average	43.79	96.07	98.30	26.06	96.55	97.10	22.55	92.11	92.89
Max.	48.27	96.40	98.49	30.05	98.02	97.96	32.37	94.16	98.33
Std.	1.57	0.23	0.06	2.03	1.22	0.59	4.31	1.83	1.67
IGHS									
Min.	41.84	96.40	98.49	23.69	96.04	97.96	16.30	89.97	89.14
Average	43.10	97.88	98.82	25.73	97.03	98.78	21.76	92.24	93.37
Max.	45.33	98.82	99.29	28.63	98.02	100	27.88	94.81	100
Std.	2.10	1.01	0.13	1.71	0.69	0.87	3.77	1.52	3.04

other HS algorithms. Therefore, it was concluded that the IGHS performed better in classifying datasets in comparison with other algorithms considered in this paper.

References

- Ceylan H, Haldenbilen S et al (2008) Transport energy modelling with meta-heuristic harmony search algorithm, an application to Turkey. *Energ Policy* 36(7):2527–2535
- Chronopoulos AT, Sarangapani J (2002) A distributed discrete time neural network architecture for pattern allocation and control. In: *Proceedings of the international parallel and distributed processing symposium (IPDPS'02)*, FL, USA, pp 204–211
- Curry B, Morgan P (1997) Neural networks: a need for caution. *OMEGA. Int J Manag Sci* 25:123–133
- Degertekin SO (2008a) Harmony search algorithm for optimum design of steel frame structures: a comparative study with other optimization methods. *Struct Eng Mech* 29(4):391–410
- Degertekin SO (2008b) Optimum design of steel frames using harmony search algorithm. *Struct Multidiscip Optim* 36(4):393–401
- Fausett L (1994) *Fundamentals of neural networks architectures, algorithms, and applications*. Prentice Hall, New Jersey
- Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y (2008) Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Comput Method Appl Mech Eng* 197(33–40):3080–3091
- Forsati R, Haghghat AT, Mahdavi M (2008) Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing. *Comput Commun* 31(10):2505–2519
- Garro BA, Sossa H, Vázquez RA (2009) Design of artificial neural networks using a modified particle swarm optimization algorithm. *International joint conference on neural networks, 2009 (IJCNN 2009)*, pp 938–945
- Garro BA, Sossa H, Vázquez RA (2010) Design of artificial neural networks using differential evolution algorithm. In: *ICONIP'10 Proceedings of the 17th international conference on neural information processing: models and applications, vol Part II*, pp 201–208
- Garro BA, Sossa H, Vázquez RA (2011) Evolving neural networks: a comparison between differential evolution and particle swarm optimization. *ICSI 2011:447–454*
- Geem ZW (2006) Optimal cost design of water distribution networks using harmony search. *Eng Optim* 38:259–280
- Geem ZW (2009a) Harmony search optimization to the pump-included water distribution network design. *Civ Eng Environ Syst* 26(3):211–221
- Geem ZW (2009b) Particle-swarm harmony search for water network design. *Eng Optim* 41(4):297–311
- Geem ZW, Kim J, Loganathan G (2002) Harmony search optimization: application to pipe network design. *Int J Model Simul* 22(2):125–133
- Geem ZW, Lee KS, Park YJ (2005) Application of harmony search to vehicle routing. *Am J Appl Sci* 2(12):1552–1557
- Gupta JND, Sexton RS (1999) Comparing backpropagation with a genetic algorithm for neural network training. *Omega Int J Manag Sci* 27:679–684
- Hassoun MH (1995) *Fundamentals of artificial neural networks*. MIT Press, Massachusetts
- Karaboga D, Ozturk C (2009) Neural networks training by artificial bee colony algorithm on pattern classification. *Neural Netw World* 19:279–292

- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
- Kim JH, Geem ZW, Kim ES (2001) Parameter estimation of the nonlinear Muskingum model using harmony search. *J Am Water Resour Assoc* 37:1131–1138
- Kim D, Kim H, Chung D (2005) A modified genetic algorithm for fast training neural networks. In: Advances in neural networks—ISSN 2005, vol 3496/2005. Springer, Berlin, pp 660–665
- Kiranyaz S, Ince T, Yildirim A, Gabbouj M (2009) Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. *Neural Netw* 22(10):1448–1462
- Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. *Comput Struct* 82:781–798
- Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Comput Method Appl Mech Eng* 194:3902–3933
- Lee KS, Geem ZW, Lee SH, Bae KW (2005) The harmony search heuristic algorithm for discrete structural optimization. *Eng Optim* 37:663–684
- Salchenberger LM, Cinar EM, Lash NA (1992) Neural networks: a new tool for predicting thrift failures. *Decis Sci* 23:899–916
- Sexton RS, Dorsey RE (2000) Reliable classification using neural networks: a genetic algorithm and back propagation comparison. *Decis Support Syst* 30:11–22
- Sexton RS, Dorsey RE, Johnson JD (1998) Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. *Decis Support Syst* 22:171–186
- Slowik A, Bialko M (2008) Training of artificial neural networks using differential evolution algorithm. In: Proceedings of the IEEE conference on human system interaction, Sracow, Poland, pp 60–65
- Socha K, Blum C (2007) An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Comput Appl* 16:235–247
- Vasebi A, Fesanghary M, Bathaee SMT (2007) Combined heat and power economic dispatch by harmony search algorithm. *Int J Electr Power* 29:713–719
- Wang F et al (1999) Neural network structures and training algorithms for microwave applications. *Int J RF Microw C E* 9:216–240
- Yu J, Wang S, Xi L (2008) Evolving artificial neural networks using an improved PSO and DPSO. *Neurocomputing* 71:1054–1060
- Zamani M, Sadeghian A (2010) A variation of particle swarm optimization for training of artificial neural networks. In: Al-Dahoud Ali (ed) Computational intelligence and modern heuristics, INTECH. ISBN: 978-953-7619-28-2
- Zhang QJ, Gupta KC (2003) Neural networks for RF and microwave design—from theory to practice. *IEEE Trans Microw Theory* 51(4):1339–1350
- Zou DX, Gao L, Wu J, Li S (2010) Novel global harmony search algorithm for unconstrained problems. *Neurocomputing* 73:3308–3318