



# Hybridization of Metaheuristic and Population-Based Algorithms with Neural Network Learning for Function Approximation

Zhen-Yao Chen<sup>(✉)</sup>

Department of Business Administration, Hungkuo Delin University of Technology,  
No. 1, Ln 380, Qingyun Road, New Taipei City 23654, Taiwan  
keyzyc@gmail.com

**Abstract.** This paper attempts to improve the learning representation of radial basis function neural network (RBFNN) through metaheuristic algorithm (MHA) and evolutionary algorithm (EA). Next, the ant colony optimization (ACO)-based and genetic algorithm (GA)-based approaches are employed to train RBFNN. The proposed hybridization of ACO-based and GA-based approaches (HAG) algorithm incorporates the complementarity of exploration and exploitation abilities to reach resolution optimization. The property of population diversity has higher chance to search the global optimal instead of being restricted to local optimal extremely in two benchmark problems. The experimental results have shown that ACO-based and GA-based approaches can be integrated intelligently and develop into a hybrid algorithm which aims for receiving the best precise learning expression among relevant algorithms in this paper. Additionally, method assessment results for two benchmark continuous test function experiments and show that the proposed HAG algorithm outperforms relevant algorithms in term of preciseness for learning of function approximation.

**Keywords:** Hybrid algorithm · Radial basis function neural network · Artificial colony optimization · Genetic algorithm · Function approximation

## 1 Introduction

The typical optimization approaches such as linear and nonlinear programming [29], which are based on optimization formulation, would not be the best option for the intelligent forecasting models due to their nature of inherent limitations and complexity of obtained objective function [38]. Studies have also shown that the metaheuristic (MH) algorithms (MHAs), which primarily include evolutionary and swarm intelligence (SI) based algorithms, have been applied to search the feasible solutions for these complicated predictive models in the past two decades [38].

Further, researchers have devoted to improving the precision of models through modifying and developing them with new approaches [3]. One possible approach to improve precision is to integrate the model with robust optimization MHAs such as genetic programming (GP), genetic algorithm (GA), and ant colony optimization (ACO)

to minimize deviation and increase the accuracy of the model [27, 34]. However, evolutionary algorithms (EAs) mimetic the evolutionary procedure in nature where the global area optimal are obtained through generating new offspring that inherit the properties from the parents' population. The set of candidate solution is improved gradually until satisfying the termination criteria. As such, throughout generation, the probability of achieving advanced results near the global area optimal will raise, however obtaining approximation of the global optima with high preciseness is not guaranteed [32].

On the other hand, the common mechanism of artificial neural networks (ANNs) is built based on the behavior within the neurons of human brains. This biological behavior was first modeled in several mathematical equations by McCulloch & Pitts [26] in 1943. Their paper has become an innovative work and it opens a new era of computational intelligence [2]. Moreover, the radial basis function (RBF) neural network (RBFNN) was proposed by Duda & Hart [9] in 1973. It exists a number of dominances over other types of NNs: excellent approximation capabilities, easier network constitutions and algorithms with faster learning capability [28]. Further, Zhang & Liao [39] in 2014 examined the prediction performance of RBFNN and hybrid fuzzy clustering (HFC) algorithm. The HFC algorithm has shown better performance over the former [39]. In addition, the representation of RBFNN depends upon the relevant parameters of nonlinear kernel functions for RBFNN. At the meantime, not many efforts have been made to hybrid several MH and population-based algorithms with trained RBFNN where exists gaps to improve in term of the fitting preciseness for function approximation. Accordingly, this study intends to propose a HAG algorithm for training RBFNN and make adequate performance verification and comparison. The proposed HAG algorithm incorporates the local and global area search abilities for problem resolution. Next, the HAG algorithm utilizes two benchmark continuous test functions, which are usually adopted in the literature to be the contrast of algorithm expression in the experiment.

The remainder of this study is organized as follows. Section 2 presents the literature review related to this paper, while the proposed HAG algorithm is introduced and illustrated in Sect. 3. Sections 4 discussed the experimental and assessment results. The study ends by the concluding remarks in Sect. 5.

## 2 Literature Review

A number of soft computing (SC) techniques normally named as MHAs have been emerged as the outputs for this research field that considered to imitate the biological processes, the group behavior of agents and survival of the fittest and so on, for the optimization problems [14]. Further, swarm-based algorithms exist the features of information sharing among multiple individuals, capability of being collaboration, self-organized, and learning during generations to fulfill efficient seeking procedures [37]. This section presents general background associated to this study, including MH and population-based algorithms for RBFNN learning.

MHAs are inspired from nature and ACO is one of the instances, have been well applied to numerous different optimization problems [10]. In Savsani et al. [31] in 2014, ACO algorithm developed by Dorigo [8] in 1996 was utilized on the simulation of collaboration mechanism of real-world ant colonies. Ants have the inclination to seek

the shortest route between their nest and sources of food. Consider an optimization problem as a multi-layer topology. In which, the count of nodes within specific layer is equivalent to the count of detachment values corresponding to the design variables and the count of layers is equivalent to the number of design variables. Next, ACO has fascinated much concentration in the domains of discrete problems due to its population-based search ability as well as robustness and simplicity. ACO uses heuristic technique to generate a well initial solution and decides an appropriate search tendency according to the experience. While it is deserving to note that this strategy often brings a well solution for ACO, it leads to ACO trapped in local optimal as trade-off [19].

EAs imitate rules in natural evolution operators. GA is inspired through Darwinian evolutionary principle, which is the most broadly used EA [11]. One of beforehand works to imitate the appearance to seek the global area extreme values of an optimization task was done in 1975 by Holland [15] in 1992 when he introduced his GA, which simulates the evolutionary theory proposed through Darwin [12]. Next, GA is a random search approach through the Darwinian evolutionary principle, adopting procedures such as natural selection, reproduction, mutation and crossover. Early, an initial stochastic population was established, after that several solutions are sorted through their objective function and then, the first probability of them are transmitted to the next generation. Later, any two solution sets are chosen adopting the Roulette wheel (RW) selection and are merged to establish new offsprings. The procedure employed in establishing the new population is the mutation step. Lastly, objective value estimation of the new population should be fulfilled [30]. Further, gene choice permits us to comprehend the situations of a cell influenced through an illness. Especially, gene choice is a principle through selecting the most dominant genes that can valid forecast the class to which a cell specimen belongs to [30].

The ACO is a probabilistic approach to solve computational tasks. It provides the optimal solution by the paths of graph although it perhaps be fall into a local area optimal solution and is different from a global area optimal one [21]. On the other hand, Holland (1975) developed the GA, which is a population-based and stochastic-based optimization approach. The model was built based on nature-inspired evolutionary procedures such as natural selection, inherit, crossover and mutation [1]. Next, a hybrid algorithm of GA and ACO (i.e., GA-ACO algorithm) was proposed by Luan et al. [25] in 2019 and was adopted to resolve the linear programming model for supplier extract task. The GA-ACO algorithm applies the superiorities of GA with high initial accelerate convergence and the advantages of ACO with valid and parallelism feedback. As for the GA-ACO algorithm, the solutions generated through GA will be utilized to determine the initial produced pheromones for ACO [25].

### 3 Methodology

RBFNN is normally premeditated as a three-layered construction composed by input, hidden, and output layers [36], in which the RBF interpolation is formulated as [17]:

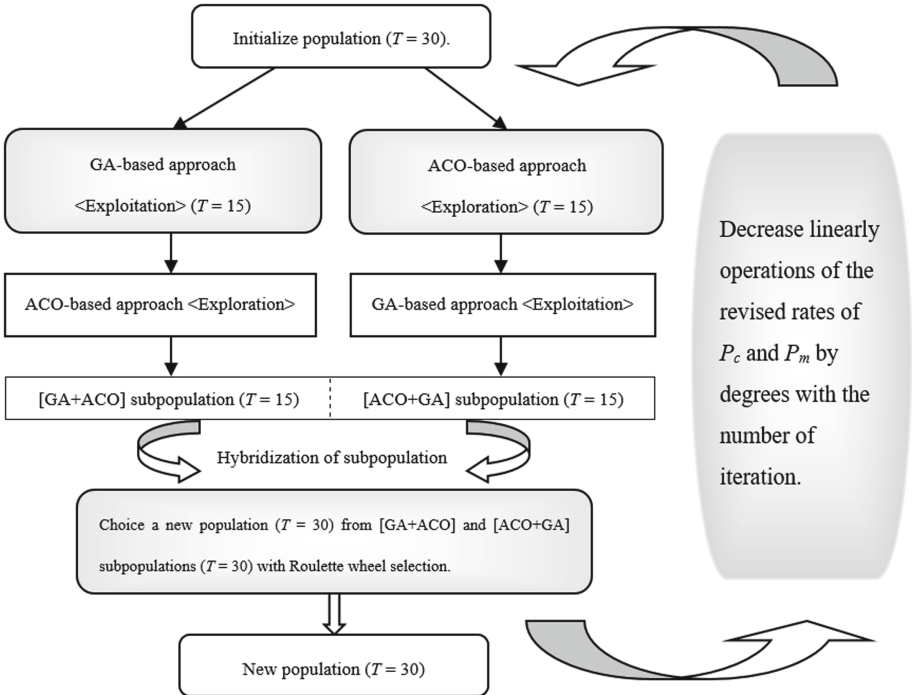
$$u(x) = \sum_{i=1}^N w_i \xi_i(\|x - x_i\|_2), \quad (1)$$

where  $w_i$  are the weights value,  $\xi_i$  is the RBF, and  $\|x - x_i\|_2$  denotes the Euclidean distance between the new point  $x$  and a sample point  $x_i$ . The RBFs  $\xi_i$ , used in this work is the Gaussian basis function:

$$\xi(r) = e^{-(\varepsilon r)^2}, \tag{2}$$

where  $r$  is the Euclidean distance between a vector of RBFNN input layer and a center of RBFNN hidden layer, and  $\varepsilon$  is the width factor determining the size of the RBF [36].

Further, as one class of topical kernel function (KF), the parameter  $\varepsilon$  of RBF resolves the width of the KF. Merely the selected point in sight of the trial point may influence on the yield of the function. In other words, RBF function has partial features and capability on interpolation [35]. Thus, the nonlinear function of the RBFNN hidden layer adopted is the Gaussian basis function shown in Eq. (2). Additionally, a typical hidden node in a RBFNN is characterized through its center, which is a vector where its number of dimensions is the number of inputs to the node. Then, the framework for the proposed HAG algorithm is illustrated in Fig. 1.



**Fig. 1.** The framework for the proposed HAG algorithm

### 3.1 The Detailed Description of the Proposed HAG Algorithm

This paper focused on training and tuning the corresponding parameters for RBFNN. The best solution of parameter values set can be received and adopted in the proposed HAG algorithm with the RBFNN to solve the problem for function approximation. The purpose is to receive the maximum of a fitness function regarding the parameters of the RBFNN (i.e., the hidden node center, width, and weight between the hidden and output layers). The inverse of mean absolute error (MAE) (i.e.,  $MAE^{-1}$ ) is adopted as fitness function. The fitness values for the HAG algorithm in the experiment are calculated by maximizing the  $MAE^{-1}$  defined as Eq. (3):

$$Fitness = MAE^{-1} = N \cdot \left( \sum_{i=1}^N |y_i - \hat{y}_i| \right)^{-1}, \quad (3)$$

where  $y_i$  is the actual output;  $\hat{y}_i$  is the predicted output of the learned RBFNN for the  $i^{th}$  testing pattern;  $N$  is the number of the testing set. Therefore, RBFNN can be trained to approximate two benchmark functions to a higher degree of precision. Next, the progress procedures for the HAG algorithm was then executed and summarized as follows.

- (1) **Initialization:** The initialization corresponding to nature random selection assures the diversity among units (i.e., ants in ACO-based approach; chromosomes in GA-based approach) and benefits the evolutionary procedure hereafter. An initial population with a number of units is produced and the initializing procedures are as follows.
  - (a) Each unit in the initial population is the set of positions for neuron (i.e.,  $c'_{i,j}$ ) and width (i.e.,  $d'_i$ ) on RBFNN, defined in a matrix form. The results are adopted as the number of neurons in RBFNN.
  - (b) The weights  $w_i$  are obtained by resolving the linear relationship [17]: where  $A = A_{ij} = \xi_i(\|x - x_i\|_2)$  and  $u = u(x_i)$  are the investigated function values at the specimen points. The chosen RBFs will generate a positive-definite matrix  $\mathfrak{R}$ , thus assuring a sole solution to Eq. (4) [17].
  - (c) The fitness value of unit matrix in population is calculated through Eq. (3) (i.e.,  $MAE^{-1}$ ).

- (2) **ACO-based approach** [8, 31]:

Let the ant nest include  $K$  ants. In the origination of the optimization procedure, all paths are initialized with an equivalent quantity of pheromone. In each generation, ants begin at the nest node, traverse across the varying layers from the first to the last layer, and finish at the target node [31]. Through following Eq. (5), each ant may determine only one node in every layer [8].

In which,  $P_{ij}^k$  indicates the probability of selecting  $j$  as the next intent aim for ant  $k$  located at node  $i$ ,  $\tau_{ij}$  is the pheromone trail and  $\alpha$  is the pheromone sensitiveness.

In case the route is finish, the ant precipitates some pheromone on the route based on the locally trial updating rule given through Eq. (6):

$$Aw = u \quad (4)$$

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum \tau_{ij}^\alpha} & \text{if } j \in K_i^k \\ 0 & \text{if } j \notin K_i^k \end{cases} \quad (5)$$

$$\tau_{ij} = \tau_{ij} + \Delta\tau^k, \quad (6)$$

where  $\Delta\tau^k$  is the pheromone accumulation via  $k$ th ant on the route it has transited.

When all the ants fulfill their routes, the pheromones on the global area best route are revised utilizing the globally trial updating rule given through Eq. (7):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^K \Delta\tau_{ij}^k, \quad (7)$$

where  $\rho$  is the pheromone attenuating (exhalation) rate,  $\Delta\tau_{ij}^k$  is the pheromone precipitated via the best ant  $k$  on the route  $ij$  estimated as  $Q \cdot (\text{fitness}^k)^{-1}$ , and  $Q$  is a constant [8]. Furthermore, for the population in ACO-based approach, the ant concludes better solution by referencing itself and other ants, determines the proceeding direction and therefore is able to explore in a global search space.

- (3) **Duplication:** The population promoted via the learning of ACO-based approach [8, 20] is replicated and is named as ACO population.
- (4) **GA-based approach:** The approach of GA evolution that includes crossover and mutation operators in the population of ACO-based approach learning is called [GA+ACO] subpopulation. The operators used in GA-based approach are as follows.
  - (a) Further, each row of the chosen paired  $C_t$  will execute crossover operator with  $P_c$ .
  - (b) Through the mutation operator, the values are substituted via randomly chosen values from the range of the search domain in each dimension, which keeps the diversity and produces new solutions.
- (5) **Reproduction:** The [ACO+GA] and [GA+ACO] subpopulations are hybridization after the refined evolution. Units with same quantity from the initial population are randomly chosen via the proportional RW selection [13] for the evolution hereafter. Thus, by applying ACO-based and GA-based approaches to conduct exploration and exploitation in the solving space respectively, it is expected to obtain the optimal solution with their best complementary properties.

Additionally, owing to the feature of local search with GA-based approach, whether what the fitness values of the units in population are, they entirely have the chances to make progress with some genetic operators and enter into the next iteration of population to perform. GA-based approach then is able to exploit the potential solution.

- (6) **Termination:** The HAG algorithm will not stop returning to step (2) unless a specific number of iterations has been reached.

In summary, executing an evolution program via the ACO-based approach would receive a promoted progress population, which is better than the initial population. Moreover, the advantage of the feature of global search in ACO-based approach allows wider exploration on dimension domain among different experiments and the solving space is able to be expanded. On the other hand, as the HAG algorithm progresses, the members of the population evolve gradually. In this way, the HAG algorithm accords the essence of GA-based approach, assures the genetic diversity in the refinement of future evolution, and makes progresses to obtain a new promoted population. Besides, through the GA-based approach within the HAG algorithm to calculate the fitness values of unit parameters solution in the population, the better solutions will be received gradually. Accordingly, the solution space in population could be improved progressively and converge regarding to the global optimal solution.

## 4 Experimental Results

This section focused on training and tuning the corresponding parameters in RBFNN for function approximation problem. The objective is to receive the maximum of a fitness function concerning the parameters of the RBFNN. The intention is then to solve the suitable values of the parameters from the setting domain in the experiment. The proposed HAG algorithm will gradually be able to train and thus receive a set of solutions for parameter values.

### 4.1 Benchmark Problems Experiment

Continuous test function induces excellent approximation to recompense RBFNN for the outcome of nonlinear mapping relation. This paper applies two continuous test functions that are usually used in the literature to be the comparative benchmark of estimated algorithms. As such, the experiment contains the following two benchmark problems, including Rosenbrock and Griewank [4] continuous test functions are listed in Table 1.

**Table 1.** Two benchmark continuous test functions [4] adopted in this experiment

Benchmark continuous test function	Equation	Initial range
Rosenbrock	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-100, 100]

## 4.2 Parameter Setup

In the proposed HAG algorithm, four parameters (i.e., pheromone trail, pheromone decay rate, crossover probability, and mutation probability), which have major impact on calculation results are analyzed. Besides, this paper also referred to the related literature for the range of the parameters' value setting. Next, the setup of the parameters for HAG algorithm is tuned by referring to the Taguchi experimental design [33] with analysis mode substitute for using trial and error procedure [38]. After, the maximum number of iterations is set at 1,000 to set as termination condition in the experiment. Finally, the appraisal of the parameters setting for the proposed HAG algorithm was conducted with the content listed in Table 2.

**Table 2.** Parameter setup for the proposed HAG algorithm

Parameter	Description	Value
$E$	The maximum number of iterations	1000
$g_t$	The number of the RBFNN hidden node centers	[1, 50]
$d_i^t$	The width of RBFNN hidden layer	[1000, 40000]
$\delta$	The learning rate of the RBFNN	0.3
$T$	Population size	30
$K$	The number of ant	100
$\tau$	The pheromone trail	0.5
$\alpha$	The pheromone sensitiveness	0.8
$\rho$	The pheromone attenuating rate	0.3
$Q$	A constant	0.65
$P_c$	Crossover probability	[0.4, 0.5]
$P_m$	Mutation probability	[0.01, 0.05]

## 4.3 Performance Assessment and Comparison

The learning of all algorithms on several solutions of parameter values set (i.e., hidden node center, width, and weight) for RBFNN that are generated by the population during



the operation of the progress procedure in the experiment are discussed in this section. Consequently, 1000 randomly generated data sets are divided into three parts to train RBFNN which are training set (65%), testing set (25%), and validation set (10%) [24] respectively, and in which we can assess the learning status and tune the parameters' setting. Next, this study uses these algorithms to resolve the best solution of parameter values set for RBFNN, and it randomly generates unrepeatable 65% training set from 1000 generated data and input the set to RBFNN for training. With the same approach, it randomly generates unrepeatable 25% testing set to inspect unit parameters' solution in population and further calculates the fitness value. RBFNN has adopted 90% dataset in learning stage at this point. After 1000 iterations in the evolution process, the best solution of parameter values set for RBFNN are received. Finally, it randomly generates unrepeatable validation set (10% dataset) to certify how the unit parameters solution approximates the two benchmark problems and record the RMSE values to justify the learning status of RBFNN. Once the data refining steps presented above have completed, all algorithms are ready to execute. The learning and validation stages mentioned above were carried out 50 times before the average RMSE (i.e.,  $RMSE_{avg}$ ) values were calculated. The values of the  $RMSE_{avg}$  and standard deviation (SD) for all algorithms calculated from the experiment are listed in Table 3.

**Table 3.** Result comparison among relevant algorithms adopted in this experiment

Benchmark problem Competitive algorithm	Rosenbrock		Griewank	
	Training set	Validation set	Training set	Validation set
RBFNN [5]	12046.37 ± 1361.53	12971.61 ± 2624.59	27.87 ± 2.79	28.21 ± 3.82
GA-based [15]	1672.38 ± 378.17	2592.16 ± 253.21	19.59 ± 16.47	54.27 ± 17.38
ACO-based [8]	1235.05 ± 147.61	1986.38 ± 362.53	7.34 ± 13.27	51.16 ± 10.37
GA-ACO [25]	914.52 ± 21.18	627.72 ± 29.02	6.38 ± 3.27	15.06 ± 4.15
<b>HAG</b>	<b>707.36 ± 13.28</b>	<b>590.02 ± 23.59</b>	<b>5.47 ± 2.83</b>	<b>13.51 ± 2.65</b>

In Table 3, the results demonstrate that HAG algorithm obtains the smallest values with stable expression during the entire training process of the experiment. Thus, RBFNN is able to receive the single parameters' solution from the evolution learning process in population, which has realized the situation with optimal function approximation. When the training of RBFNN by the HAG algorithm is accomplished, the unit with the best solution of parameter values set in learning stage is the RBFNN setting in certain. Additionally, the HAG algorithm shows robust learning within two benchmark problems and shows remarkable approximation results.

## 5 Conclusions

This paper proposed the HAG algorithm through incorporating ACO-based and GA-based approaches, which offers the settings of RBFNN parameters. The experimental results shown that ACO and GA algorithms can be integrated intelligently and develop into a hybrid algorithm which is designed for receiving the best precise learning expression among all algorithms in this study. Additionally, method assessment results for two benchmark continuous test function experiments and show that the proposed HAG algorithm outperformed other algorithms in preciseness of function approximation.

## References

1. Al-Obaidi, M.A., Li, J.P., Kara-Zaïtri, C., Mujtaba, I.M.: Optimisation of reverse osmosis based wastewater treatment system for the removal of chlorophenol using genetic algorithms. *Chem. Eng. J.* **316**, 91–100 (2017)
2. Al-Roomi, A.R., El-Hawary, M.E.: Universal functions originator. *Appl. Soft Comput.* **94**, 106417–106448 (2020)
3. Ansari, M., Othman, F., El-Shafie, A.: Optimized fuzzy inference system to enhance prediction accuracy for influent characteristics of a sewage treatment plant. *Sci. Total Environ.* **722**, 137878–137890 (2020)
4. Bilal, M.P., Zaheer, H., Garcia-Hernandez, L., Abraham, A.: Differential evolution: a review of more than two decades of research. *Eng. Appl. Artif. Intell.* **90**, 103479–1034502 (2020)
5. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Netw.* **2**(2), 302–309 (1991)
6. Chen, S., Wu, Y., Luk, B.L.: Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. *IEEE Trans. Neural Netw.* **10**(5), 1239–1243 (1999)
7. Denker, J.S.: Neural network models of learning and adaptation. *Physica D.* **22**, 216–232 (1986)
8. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **26**(1), 29–41 (1996). <https://doi.org/10.1109/3477.484436>
9. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York (1973)
10. Dzalbs, I., Kalganova, T.: Accelerating supply chains with Ant Colony Optimization across a range of hardware solutions. *Comput. Ind. Eng.* **147**, 106610–106623 (2020)
11. Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S.: Equilibrium optimizer: a novel optimization algorithm. *Knowl. Based Syst.* **191**, 105190–105210 (2020)
12. Ghafil, H.N., Jarmai, K.: Dynamic differential annealed optimization: new metaheuristic optimization algorithm for engineering applications. *Appl. Soft Comput.* **93**, 106392–106410 (2020)
13. Goldberg, D.E.: *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, Boston (1989)
14. Hamzaday, A., Baykasoglu, A., Akpınar, S.: Solving combinatorial optimization problems with single seekers society algorithm. *Knowl. Based Syst.* **201–202**, 106036–106065 (2020)
15. Holland, J.H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge (1992)

16. Islam, J., Vasant, P.M., Negash, B.M., Laruccia, M.B., Myint, M., Watada, J.: A holistic review on artificial intelligence techniques for well placement optimization problem. *Adv. Eng. Softw.* **141**, 102767–102786 (2020)
17. Jakobsson, S., Andersson, B., Edelvik, F.: Rational radial basis function interpolation with applications to antenna design. *J. Comput. Appl. Math.* **233**(4), 889–904 (2009)
18. Jugulum, R., Taguchi, S., et al.: *Computer-Based Robust Engineering: Essentials for DFSS*. ASQ Quality Press, Milwaukee (2004)
19. Kefayat, M., Ara, A.I., Niaki, S.A.N.: A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources. *Energ. Convers. Manage.* **92**, 149–161 (2015)
20. Kozak, J., Boryczka, U.: Multiple boosting in the Ant colony decision forest meta-classifier. *Knowl. Based Syst.* **75**, 141–151 (2015)
21. Lin, Y.K., Yeh, C.T., Huang, P.S.: A hybrid ant-tabu algorithm for solving a multistate flow network reliability maximization problem. *Appl. Soft Comput.* **13**, 3529–3543 (2013)
22. Lin, C.F., Wu, C.C., Yang, P.H., Kuo, T.Y.: Application of Taguchi method in lightemitting diode backlight design for wide color gamut displays. *J. Disp. Technol.* **5**(8), 323–330 (2009)
23. Liu, B., Aliakbarian, H., Ma, Z., Vandenbosch, G.A.E., Gielen, G., Excell, P.: An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques. *IEEE Trans. Antennas Propag.* **62**(1), 7–18 (2014)
24. Looney, C.G.: Advances in feedforward neural networks: demystifying knowledge acquiring black boxes. *IEEE Trans. Knowl. Data Eng.* **8**(2), 211–226 (1996)
25. Luan, J., Yao, Z., Zhao, F., Song, X.: A novel method to solve supplier selection problem: hybrid algorithm of genetic algorithm and ant colony optimization. *Math. Comput. Simul.* **156**, 294–309 (2019)
26. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943)
27. Naghibi, S.A., Ahmadi, K., Daneshi, A.: Application of support vector machine, random forest, and genetic algorithm optimized random forest models in groundwater potential mapping. *Water Resour. Manage.* **31**(9), 2761–2775 (2017)
28. Qasem, S.N., Shamsuddin, S.M., Zain, A.M.: Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. *Knowl. Based Syst.* **27**, 475–497 (2012)
29. Rao, R.V., Rai, D.P., Balic, J.: A multi-objective algorithm for optimization of modern machining processes. *Eng. Appl. Artif. Intell.* **61**, 103–125 (2017)
30. Salehpoor, I.B., Molla-Alizadeh-Zavardehi, S.: A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms. *Appl. Soft Comput.* **75**, 233–253 (2019)
31. Savsani, P., Jhala, R.L., Savsani, V.: Effect of hybridizing biogeography-based optimization (BBO) technique with artificial immune algorithm (AIA) and Ant colony optimization (ACO). *Appl. Soft Comput.* **21**, 542–553 (2014)
32. Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H.: Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **87**, 103330–103342 (2020)
33. Taguchi, G., Chowdhury, S., Wu, Y.: *Taguchi's Quality Engineering Handbook*. Wiley, Hoboken (2005)
34. Tayfur, G.: Modern optimization methods in water resources planning, engineering and management. *Water Resour. Manage.* **31**(10), 3205–3233 (2017)
35. Tian, Z.: Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM. *Eng. Appl. Artif. Intell.* **91**, 103573–103596 (2020)
36. Urquhart, M., Ljungskog, E., Sebben, S.: Surrogate-based optimisation using adaptively scaled radial basis functions. *Appl. Soft Comput.* **88**, 106050–106066 (2020)

37. Yang, Z., Li, K., Guo, Y., Ma, H., Zheng, M.: Compact real-valued teaching-learning based optimization with the applications to neural network training. *Knowl. Based Syst.* **159**, 51–62 (2018)
38. Yin, X., Niu, Z., He, Z., Li, Z.S., Lee, D.: An integrated computational intelligence technique based operating parameters optimization scheme for quality improvement oriented process-manufacturing system. *Comput. Ind. Eng.* **140**, 106284–106298 (2020)
39. Zhang, F., Liao, Z.: Gold price forecasting based on RBF neural network and hybrid fuzzy clustering algorithm. In: Jiuping, X., Fry, J.A., Lev, B., Hajiyev, A. (eds.) *Proceedings of the Seventh International Conference on Management Science and Engineering Management*, pp. 73–84. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-642-40078-0\\_6](https://doi.org/10.1007/978-3-642-40078-0_6)