Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Integration Learning of Neural Network Training with Swarm Intelligence and Meta-heuristic Algorithms for Spot Gold Price Forecast

Zhen-Yao Chen

To cite this article: Zhen-Yao Chen (2022) Integration Learning of Neural Network Training with Swarm Intelligence and Meta-heuristic Algorithms for Spot Gold Price Forecast, Applied Artificial Intelligence, 36:1, 1994217, DOI: 10.1080/08839514.2021.1994217

To link to this article: https://doi.org/10.1080/08839514.2021.1994217

Published online: 25 Oct 2021.

APPLIED ARTIFICIAL INTELLIGENCE
2022, VOL. 36, NO. 1, e1994217 (163 pages)
https://doi.org/10.1080/08839514.2021.1994217

Taylor & Francis
Taylor & Francis Group

Integration Learning of Neural Network Training with Swarm Intelligence and Meta-heuristic Algorithms for Spot Gold Price Forecast

Zhen-Yao Chen

Department of Business Administration, Hungkuo Delin University of Technology, New Taipei City, Taiwan

ABSTRACT

This research attempts to enhance the learning performance of radial basis function neural network (RBFNuNet) via swarm intelligence (SI) and meta-heuristic algorithms (MHAs). Further, the genetic algorithm (GA) and ant colony optimization (ACO) algorithms are applied for RBFNuNet to learn. The proposed integration of GA and ACO approaches-based (IGACO) algorithm combines the complementarity of exploitation and exploration capabilities to achieve optimization resolve. The feature of population diversification has higher opportunity to pursue the global optimal substitute being constrained to local optimal exceeding in five continuous test functions. The experimental results have illustrated that GA and ACO approaches can be incorporated intelligently and propose an integrated algorithm, which intents for obtaining the optimal accuracy training performance among relevant algorithms in this study. Additionally, method assessment results for five benchmark problems and a practical spot gold price forecast exercise show that the proposed IGACO algorithm outperforms other algorithms and the Box-Jenkins models in terms of forecasting preciseness and execution time.

ARTICLE HISTORY
Received 4 August 2021
Revised 9 October 2021
Accepted 12 October 2021

Introduction

Conventional gradient-based techniques regularly turn into powerless owing to their rigorous adopt conditions and slowly convergence (Wang et al. 2018). In addition, time series method contains autoregressive (AR) moving average (MA) model (Erdem and Shi 2011), AR integrated MA (ARIMA) model (Cadenas et al. 2016), ARIMA with exogenous (ARIMAX) variables model (Yan et al. 2017), etc. (Tian 2020). However, studies have shown that these approaches exist some shortcomings, for instance difficulty in parameter evaluation of high-order model and low forecasting precision of low-order model (Tian 2020).

CONTACT Zhen-Yao Chen keyzyc@gmail.com Department of Business Administration, Hungkuo Delin University of Technology, New Taipei City, Taiwan

This article has been republished with minor changes. These changes do not impact the academic content of the article.

Evolutionary computation (EC) is constituted of a variation of evolution approaches like differential evolution (DE), evolutionary strategies, genetic programming, genetic algorithms (GAs), etc. These approaches are population-based and exercise a global area seek. They incorporate a repetitively revised of experimental solution sets in generations. The solution sets of identical generation are sorted via their fitness values, and the most suitable unit of the recent generation are permitted to generate the inferior generation through means of varying operators (Oprea 2020). Besides, evolutionary algorithms (EAs) have some characters such as nonlinear, nonconvex, nondifferentiable, multimodal, etc., EAs are a major division of derivative-free methods for resolving several challenging optimization tasks. As a type of adaptive and random optimization methods, EAs illustrate afflatus from the group behavior and physical advancement of animal natural collectives or social hexapod colonies (Zhang et al. 2018).

Moreover, swarm intelligence (SI) is a group of methods excited through the group behavior of animal and insect such as fishes, birds, ants, bees, bacteria, etc. (Nanda and Panda 2014). Several representative instances are the artificial colony optimization (ACO), artificial bee colony (ABC), and particle swarm optimization (PSO) algorithms (Jose-Garcia and Gomez-Flores 2016). They are investigated for their effectiveness in resolving optimization instance problems, particularly in continuous resolution spaces (Song, Ma, and Qiao 2017).

Nowadays, optimization problem resolving has turn into a popular topic in engineering and science fields. Also, these optimization problems are acquiring increasingly complicated owing to the features such as nondifferentiable, nonconvex, discontinuous, and nonlinear (Cui et al. 2017). Lately, some metaheuristic (MH) algorithms (MHAs) based on population have fascinated wide advertence to resolving incorporating optimization tasks (Zhao et al. 2018a) with higher quality solving methods and in a rational time (Talbi 2009). Further, MHAs can summarily classify into four primary divisions: human behavior-based, chemistry or physics-based, swarm-based, and evolutionary-based algorithms (Kaur et al. 2020). MH methods are changing more prevalent particularly in engineering relevant problems owing to their capability to escape from the local area optimal with depending on simplicity conceptions that imitate from nature and can be adopted in a broad range of tasks from numerous subjects. Stimulated through nature MH are quite brief and mainly excited via simplicity conceptions (Sulaiman et al. 2020).

Another principal reason that judges the satisfactory application and exercise of MHAs in resolving tasks is a complementarity trade-off between exploitation and exploration strategies. Consequently, both exploitation and exploration can be mentioned to as intensification and diversification procedures (Nasir and Tokhi 2015). For instance, Wang et al. (2017a) implemented a multiobjective algorithm based on gradient and neighborhood mechanism

for realizing the balance of exploitation and exploration strategies. This multi-objective algorithm was applied on wind turbine blade design and trialed on two to four objectives problem (Wang et al., 2017).

EAs like GAs (Abualigah and Hanandeh 2015) and SI-based algorithms such as ant colonies (Dowlatshahi and Derhami 2017), PSO (Abualigah, Khader, and Hanandeh 2018), and epsilon-greedy swarm optimizer (Dowlatshahi, Derhami, and Nezamabadi-pour 2017) belong to this type of SI algorithms. Further, a hybrid algorithms based on two or more MHs may integrate individual algorithm's superiorities and further enhance the optimization performance (Chen, Tianfield, and Li 2019). On the other hand, artificial neural network (ANN) is designed to simulate the function and structure of human brain. It is consisted of a large amount of simple processing units connected in a large scope with a certain topological constitution. The following properties: error tolerance, distributed storage, parallel process, self-adaptation, self-organization (e.g., self-organizing map neural network (SOMNN)), and self-learning, allows ANN to be applied for prediction (Huseyin and Tansu 2019). Further, ANNs are calculating networks that simulate the human brain and the nervous network. Such networks train to fulfill problems through premeditating samples, thus deduce implication relevant to unseen examples. Learning is recognizing the association between the characteristics in the examples and how that connection impacts the objective conception (Day, Iannucci, and Banicescu 2020).

Moreover, owing to radial basis function (RBF) neural network (RBFNuNet) possess a few of superiorities over other models of ANNs and these reveal superior approximation abilities, briefer network constructions, and speedy learning algorithms (Qasem, Shamsuddin, and Zain 2012). In these methods, the training work is to acquire network structures that can react as intently to the system to be imitated as possible. The construction of the RBFNuNet includes three layers: input, hidden (i.e., RBFs), and (linear) output layers (Su et al. 2012). Further, a compact teaching and learning-based optimization was proposed by Yang et al. (2018) to optimize feed-forward NN (FFNN) and RBF model. Rani and Victoire (2018) utilized RBF model optimized through an improved PSO and differential search optimizer in the application of wind speed prediction. Consequently, for the purpose of model the tortuous relation, RBFNuNet is applied to predict the expected colors trained via a SI algorithm (Li et al. 2020). Next, the accumulated value $y(t)$:

$$y(t) = \sum_{i=1}^{n} w_i \cdot \phi_i(X) \tag{1}$$

denotes the RBFNuNet model output at time lag t, whereas w_i correlates with the linear output weight for the i^{th} neuron within the hidden layer. The RBF ϕ_i for input vector X is resolved as standard Gaussian function formula as below,

$$\phi_i(X) = \exp(-\frac{1}{2\sigma_i^2}\|X - c_i\|^2), i = 1, 2, \ldots, n \tag{2}$$

where σ_i and c_i denote the Gaussian distributed width and center of the i^{th} neuron within hidden layer. The count of RBFNuNet hidden layer is denoted as n. It is worth to note that the featured nonlinear function parameters denoted in the Euclidean distance equations and denominator calls for effective optimization approach to determine (Yang et al. 2020).

SI and evolutionary computation (EC) relevant algorithms have been both used in various applications (Del Ser et al. 2019). Thanks to the new machine learning techniques, newly improved intelligence methodologies have been adopted to solve time series prediction problems in several scientific domains (Kouziokas 2020). For example, Moayedi et al. (2019) implemented several NN and evolutionary approaches for predicting the ultimate bearing capacity. Also, Khashei and Hajirahimi (2018) have attempted to act appropriate evaluation of two feasible classes of series models, established with ANN and ARIMA models for stock price prediction (Hajirahimi and Khashei 2019).

Subsequently, Khashei et al. (2009) merged ARIMA model with ANN and fuzzy logic for the prediction of daily price of gold and exchange rate. It conquered the linear and data constraints of ARIMA models and in result produced with higher accuracy. Further, Zhang and Liao (2013) inspected the forecast capability of hybrid fuzzy clustering (HFC) algorithm and RBFNuNet, and employed the HFC algorithm on gold price forecasting. The HFC algorithm has evidenced superior capability over the previous. Wen et al. (2017) applied complete ensemble empirical mode decomposition (CEEMD) along with support vector machine (SVM) and ANN for prediction and analysis of gold price. Next, Kristjanpoller and Hernandez (2017) adopted a hybrid ANN-generalized auto-regressive conditional heteroskedasticity (GARCH) (ANN-GARCH) model with regressors to forecast the price variability of gold, copper, and silver. Contrast experiments evidenced that the integration of ANN raised forecasting accuracy contrasted with traditional GARCH model.

Afterward, the expression of RBFNuNet affected through several parameters of nonlinear RBF functions for RBFNuNet. Simultaneously, not enough efforts have been implemented to integrate some soft computing algorithms and applied on RBFNuNet where retains gaps to enhance in term of the fitting accurateness for function approximation. Thereby, this research expects to propose the IGACO algorithm for training RBFNuNet and realize sufficient performance confirmation and analysis. The proposed IGACO algorithm integrates the local and global search capabilities for task resolve.

Latter, the IGACO algorithm adopts five benchmark continuous test functions, which are frequently utilized in the experiment to be the comparison of algorithm performance. Besides, we apply the inspected IGACO algorithm in term of forecasting accuracy to verify the exercise of spot gold price forecast.

The rest of this paper is structured as follows. In Section 2, the literature review is dissertated. Section 3 illustrates the methodology in detail for the proposed IGACO algorithm. The experimental results and performance evaluation are deliberated in Section 4. The practical exercise for the spot gold price forecast is provided and discoursed in Section 5. Finally, the conclusions are summarized in Section 6.

Literature Review

The recent optimization algorithm was exploited to resolve a broad scope of optimization tasks in distinct implementations of artificial intelligence (AI) such as nonlinear and linear calculations (George Lindfield 2019), resolving nonlinear practices (Truong and Kim 2018), any task where the global maximum or minimum is wished (Ghafil and Jarmai 2020), and constructional optimization (Mortazavi, Toğan, and Moloodpoor 2019). In addition, SI and EAs are stimulated via evolutionary procedures, natural phenomena, and the group behaviors of crowds of bees and ants, and flocks of birds when they look for a finer circumstance or food (Ma et al. 2019).

In the recent period, taking stimulated afflatus from different natural phenomenon, many MHAs have been developed by scholars from all over the relevant domains. Some major MHAs include GA (Wang et al., 2017b; Liu et al. 2018), differential evolution (Xu, Chen, and Tao 2018; Zhu et al. 2018), PSO (Chen et al. 2017; Nagra et al. 2019), ACO (Xiaowei et al. 2014), and artificial bee colony (ABC) (Wang et al. 2019) algorithms. These MHAs have also been broadly applied to resolve related optimization problems and have shown extraordinary performance (Chen et al. 2018; Wang et al. 2018). Consequently, by information sharing among individuals, collaborative operators raise optimize individuals and population diversification (Huang and He 2020). This section dissertates general background correlatives to this research, containing SI and evolutionary MHA for RBFNuNet training.

GA-based Optimization Algorithm for NN Training

EAs imitate biological evolutions such as selection, duplication, crossover, and mutation. Chromosomes in the population act as some candidate solutions for the given task to be optimized and fitness of every chromosome is estimated using the estimation function. The solution for the task to

be optimized is acquired via utilizing the distinction processes. No assumptions are made for the fitness parameters in EAs, and therefore given task will obtain good approximating solutions (Baeck, Fogel, and Michalewicz 2018).

On the other hand, GA is an evolution approach exploited based on the concept of organism species progress and natural selection presented by Holland (2008). Basically, GA may search the global optimal; however, GA revealed poor convergence in several cases (Islam et al. 2020). Owing to the limitation of looking for new spaces, the GA may easier cause premature convergence to the solutions on local area extreme points (Yan et al. 2020). With the Roulette wheel method, GA looks for the global optimal solutions via selecting and assessing the source population from the initialized population and the utilization of progressive manipulators on the parent to generate the next offspring in each generation (Ansari, Othman, and El-Shafie 2020).

Further, Sarimveis et al. (2004) proposed a GA-based algorithm, which targeted to minimize the error function associated to the relevant parameters of the RBFNuNet. However, as the hidden layer of RBFNuNet utilizes the restraint of the thin-plate-spline function (Chen, Cowan, and Grant 1991), the GA-based algorithm invalided to decide a proper width value within RBFNuNet. This can lead to lower preciseness in training throughout function approximation for RBFNuNet. Besides, Deniz et al. (2017) merged multi-objective GA with machine learning techniques and applied it to select features in classification tasks. The opinion is to choose the minimum count of characteristics while enhancing or retaining the classification preciseness. Moreover, Hamida, Azizi, and Saad (2017) integrated 'similarity operator' toward GA to resolve a fine arrangement task in their research and consulted it as genetic similarity algorithm (GSA). Research has indicated that the 'similarity operator' while retaining GA's exploitation and exploration capabilities has produced comprehensive refinements on the solution (Islam et al. 2020). In addition, Zhao et al. (2018b) exploited a hybrid MHA by embedding ANN into Monte-Carlo emulation and GA to choose sea-rail container routes to minimize total traffic cost. Besides, Zhou et al. (2018) devised an ensemble model with attempts to find the global area optimal handle parameters for the laser brazing onto galvanized steel. In their work, the RBF and Kriging are adopted as substitute model, and GA is used to solve the optimization formulation (Yin et al. 2020).

ACO-based Optimization Algorithm for NN Training

ACO algorithms simulate the group action of a living ant when looking for food. In the process of looking, each ant indicates the trace it is movable along via deposing a material called pheromone. Ants on a shorter route

will be faster to backtrack to the lair, so higher density pheromone will be laying on the shorter routes. The quantity of pheromone in a route makes other subsequent ants recognize whether it is favorable (Zhang et al., 2019).

Compared to other algorithms, the ACO algorithm is inspired by biology (Mustaffa, Yusof, and Kamaruddin 2014). The ACO algorithm imitates the behavior of real ants as they travel over different routes between their nest and sources of food. Communication between the real ants arises through a chemical remained by ants called pheromone. When real ants visit different routes to a food source, shorter routes typically end up with higher concentration pheromone sediments (i.e., shorter node-to-node journey time) than longer routes. In result, the majority of ants learns over time and will take the shorter route to seek for food source (Pendharkar 2015).

The ACO is a swarm-based MHA for resolving combinatorial optimization problems and its ability to produce good solutions space within a reasonable computation time has been indicated (Zhang et al., 2019). For example, Tabakhi et al. (2014) presented a novel unsupervised feature selection (FS) method based on ACO algorithm, which applies multivariate approach and possible dependencies among selected features are taken into account to reduce the redundancy (Tabakhi et al., 2014). Next, an ACO with three-level algorithm is proposed by Rais and Mehmood (2018), where ACO is adopted as a FS method. In the proposed task ACO seeks for the optimal characteristics set through iteratively way based on the pheromone trial value of each generation (Rais and Mehmood 2018).

Further, most ACO algorithms involve two obvious phases – solution establishment and pheromone revision to other ants. In general, an ant constructs its solution from the pheromone deposited via former ants, thus permitting communication beyond many generations by a pheromone matrix and converges to a superior solution. The operation of solution construction and pheromone revise is duplicated over numbers of generations until the stopping condition is arrived, which can be either total calculation spend time or total number of generations (Dzalbs and Kalganova 2020). By nature, the solution establishment strategy of ACO is adequate for a discrete seek space (Du and Swamy 2016). Since ACO establish discrete solutions directly, it prevents extra procedures when protraction solutions to the discrete space (Zhao, Zhang, and Zhang 2020).

Hybridization of GA and ACO Approaches for NN Training

Inspired by Darwinian's 'survival of the fittest' theory, the GA approach realize an optimum seeking strategy. After several complicated calculations, the GA receive a (near-) optimal solution. As a result of the preference of practicing fine on searching optimization, the trend is to adopt GA in integration with

other approaches (Day, Iannucci, and Banicescu 2020). In addition, excited by nature MHAs such as ACO approach have been favorably applied to numerous optimization tasks (Dzalbs and Kalganova 2020).

A further hybrid prediction model was developed by Xiao et al. (2017) which incorporates maximum overlapping discrete wavelet convert of time series data with ANN and takes it to forecast container throughput for Shanghai and Tianjin ports. Further, Amar, Zeraibi, and Redouane (2018) implemented time-dependent multi-NN (mNN) and used it as a dynamic substitute model. Through merging constructed proxies with GA and ACO algorithms, the empirical result shown that the proposed proxy can be taken as an alternative numerical emulator (Islam et al. 2020). Additionally, Luan et al. (2019) proposed a hybrid GA-ACO algorithm, it was applied for supplier extract task and further utilized to solve the linear programming model. Furthermore, the solutions produced via GA method will be employed to resolve the initial generated pheromones for ACO method. The hybrid GA-ACO algorithm exploits the advantages of GA method with peak primary quicken convergence and the superiorities of ACO method with parallel and effective feedback (Luan et al. 2019). However, the investigate deepness of exploration and exploitation for GA-ACO algorithm is still extremely insufficient and it affects its performance in resolving. On the other hand, an intelligent optimization approach-based hybrid model is proposed by Zhou et al. (2020) to resolve the optimal solutions with parameters setting and further to accomplish optimal technical and economical indicators for an iron-making plant (Zhou et al., 2020).

Methodology

Since the 1990s, many distinct MHAs for resolving optimization problems that simulate the natural colony behavior of animals have been exploited. Algorithms based on these MHs are usually computationally more efficient than corresponding exact solutions. However, with the exploration in their probabilistic resolving, they are not able to assurance to find the global area optimal solution (Comuzzi 2019). Yet a MH is still a repetitive method and it exploits and explores effectively in the search space to obtain the optimal area solution (Dey, Bhattacharyya, and Maulik 2014).

There are two significant properties in MHAs: exploitation and exploration. Exploitation is the capability to locally seek round prospective solutions in an attempt to improve their quality. Relatively, exploration is the capability to globally seek the solution space. This capability is interrelated with abscond from local area optimal and avoiding local optimal hesitation. Favorable performance is realized through an adequate balance between these two properties. These features are utilized by all population-based algorithms yet with distinct mechanisms and manipulators (Faramarzi et al. 2020).

According Ayala and Coelho (2016), the arithmetic formula which expounds a RBFNuNet is shown as

$$\hat{y}(t) = \sum_{m=1}^{M} w_m \phi[r(t), c_m, \sigma_m] \tag{3}$$

where $\hat{y}(t) \in \Re^+$ is the RBFNuNet forecasted output, $M \in \mathrm{N}^+$ is the number of RBF neurons within the RBFNuNet hidden layer. Next, the weights of the RBFNuNet output layer is given via $w_m \in \Re$, $r(t) \in \Re^{n_r}$ is an input vector at the given instant t, $c_m \in \Re^{n_r}$ is the center point within the m^{th} hidden neuron of the RBFNuNet, $\sigma_m \in \Re^+$ is the width within the m^{th} hidden neuron of the RBFNuNet. Finally, the Gaussian RBF is expounded as:

$$\phi(r, c, \sigma) = \exp\left(-\frac{\|r - c\|^2}{2\sigma^2}\right) = \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n_r}(r_i - c_i)^2\right] \tag{4}$$

The current task treats sole output systems. The expansion to multiplex output systems is blunt through the application of a RBFNuNet for individual system's output (Ayala and Coelho 2016).

When the width parameter is settled and a set of RBF neurons is stipulated, RBFNuNet which has such construction and an algorithm with orthogonal least squares (OLS) (Chen, Cowan, and Grant 1991) method are prepared to construct concise RBFNuNet (Chen, Wu, and Luk 1999). Then, the RBF of hidden layer on the RBFNuNet utilized is the Gaussian function represented in Eq. (4). At the same time, a normal neuron within hidden layer on the RBFNuNet is distinguished via its center vector, where its number of inputs to the neuron is equals to the number of dimensions.

The Detailed Description of the Proposed IGACO Algorithm

This study aimed on training and adjusting relevant solutions of parameter values set on RBFNuNet. The resolved solutions set can be further employed on RBFNuNet with the proposed IGACO algorithm to resolve the function approximation problem. The aim is to achieve the appropriate parameter values set (i.e., the values of the center within hidden neuron, width, and weight parameters) for RBFNuNet. Hence, the fitness function utilized the inverse of mean absolute error (MAE) (i.e., MAE^{-1}) and expound as Eq. (5). Latter, the optimal solutions of parameter values set for the IGACO algorithm in the examination are calculated via maximizing the MAE^{-1} values.

$$Fitness = MAE^{-1} = N \cdot \left(\sum_{i=1}^{N} |y_i - \hat{y}_i|\right)^{-1} \tag{5}$$

where y_i is the realistic output; $\hat{y}_i$ is the estimated output of the learned RBFNuNet for the i^{th} testing specimen; N is the count of the testing set. Additionally, RBFNuNet can be tuned and learned to approximate five nonlinear test functions to the better accuracy.

Moreover, the data are divided into three subsets with individual size Ω_1, Ω_2 and Ω_3, which are the training set: (F_1, Z_1) (65%), testing set: (F_2, Z_2) (25%), and validation set: (F_3, Z_3) (10%) respectively (Looney 1996). Next, the pseudo-code for the proposed IGACO algorithm is illustrated in Figure 1, and the evolutionary sequences for the proposed IGACO algorithm were hereafter enforced and illustrated as follows.

(1) **Initialization**: The initialize sequence according to natural emulation selection insures the variety among all units (i.e., ants (chromosomes) in ACO (GA) approach) and boosts the subsequently progressive sequence. The primary population with a count of units is yielded and the initializing stages are as expounded below.

(a) Individual unit within the initial population is the set of the center within hidden neuron (i.e., $c_{i,j}^t$) and width (i.e., d_i^t) for RBFNuNet, which described as a matrix type. Figure 2 explains the idea of matrix schematically.

$$C_t = \begin{bmatrix} c_{1,1}^t & c_{1,2}^t & \cdots & c_{1,N}^t & d_1^t \\ c_{2,1}^t & c_{2,2}^t & \cdots & c_{2,N}^t & d_2^t \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{g_t,1}^t & c_{g_t,2}^t & \cdots & c_{g_t,N}^t & d_{g_t}^t \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_t$$

The outcomes are utilized as the count of the centers within neurons on RBFNuNet. The $\{1, \ldots., g_t\}$ rows of the C_t are substituted by an equal number of row vectors of size $1 \times (N+1)$ that are the neurons of RBFNuNet associated with this unit. The $\{g_t + 1, \ldots, G\}$ rows retain equal to zero and do not conform to a neuron.

And then, Figure 3 denotes the design of decoding convention for the matrix form.

Meantime, the intrinsic values of C_t are equivalent to RBFNuNet hidden neurons which involve the $c_{i,j}^t (i = \{1, \ldots., G\}, j = \{1, \ldots., N\})$ and $d_i^t (i = \{1, \ldots., G\})$ for solution of parameter values set (i.e., units) such as positions of neuron and width. Tmatrices $C_1, C_2, \cdots, C_T$ (i.e., population size) of size

```
Begin
    Initialize population (T=50);
        Calculate the weights and fitness values of unit (individual);
        Retain the best unit;
        for (i = 0; i < number of iterations; i++ ) {
            GA (First_Half);
            ACO (First_Half);
              Begin
                  Initialization pheromone revise (pheromone) exam on population;
                  While termination criterion not met
                  Establish ant colony solution (pheromone, heuristic) via Eq. (8);
                  Apply the regionally exam improving (heuristic) via Eq. (9);
                  Revise the universally exam improving (pheromone) via Eq. (10);
                  End while
                  Result := the best solution;
              End
            ACO (Last_Half);
            GA (Last_Half);
                  Execute 2-point crossover operator on chromosomes;
                  Execute 2-point mutation operator on chromosomes;
                  Calculate the weights and fitness values of unit;
                  Choose a new population (T=50) from [GA+ACO] and [ACO+GA] subpopulations (T=50) with Roulette wheel selection;
                  Substitute the weakest unit with the former best unit;
                  Retain the best unit;
                  Decrease linearly operations of the improved rates of Pc, Pm, and W_i by degrees with the number of iteration;
End
```

Figure 1. The pseudo-code for the proposed IGACO algorithm.

$$C_t = \begin{bmatrix} c_{1,1}^t & c_{1,2}^t & \cdots & c_{1,N}^t & d_1^t \\ c_{2,1}^t & c_{2,2}^t & \cdots & c_{2,N}^t & d_2^t \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{g_t,1}^t & c_{g_t,2}^t & \cdots & c_{g_t,N}^t & d_{g_t}^t \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_t$$

Figure 2. Illustrative schema of the unit matrix.

$G \times (N+1)$ are created through setting all their instances equal to zero. For each $C_t (t = 1, 2, \cdots, T)$, a random integer $g_t \in \{1, \ldots., G\}$ from the number of centers produced in RBFNuNet is chosen.

(a) The weights w_i within hidden and output layers on RBFNuNet are acquired via deconstructing the linear relationship (Jakobsson, Andersson, and Edelvik 2009):

$$\mathrm{A}w = u \tag{6}$$

where $u = u(x_i)$ and $\mathrm{A} = A_{ij} = \xi_i(\|x - x_i\|_2)$ are the inspected function values at the sample points. The picked neurons will yield a positive-definite matrix $\Re$, thereby insuring a single-handed solution to Eq. (6) (Jakobsson, Andersson, and Edelvik 2009). For every C_t, Eq. (7) is calculated to get the output weights of respective RBFNuNet (Denker 1986):

$$w_t = (\Phi_t^T \cdot \Phi_t)^{-1} \cdot (\Phi_t^T \cdot Z_1) = \Phi_t^{-1} Z_1 \tag{7}$$

where w_t is the pseudo-inverse of the devise matrix Φ_t; Φ_t is the $\Omega_1 \times g_t$ matrix including the reactions of the hidden layer to the F_1 subset of instances; Z_1 is the wished reaction vector in the training set. The count of columns within the Φ_t equals to the count of neurons within the hidden layer and the count of rows equals to the count of training specimens. For all input data,

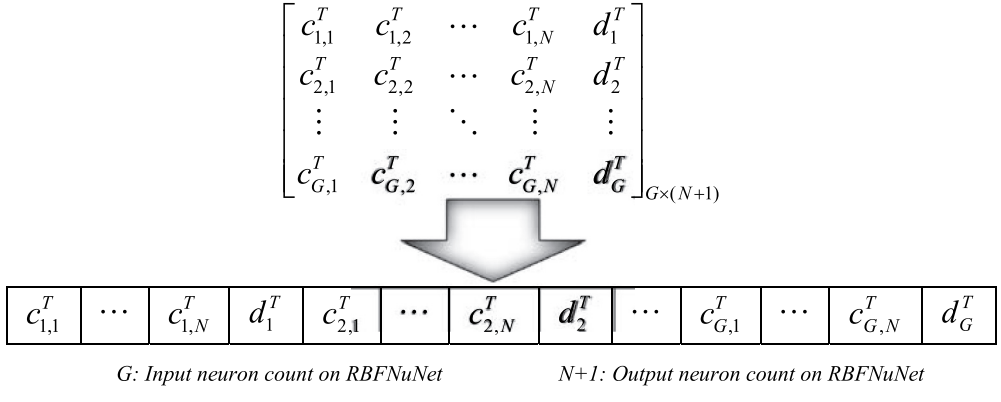

Figure 3. The design of decoding convention for the matrix form.

each column of Φ_t conforms to the reaction of the separate hidden neuron (Barra, Bezerra, and de Castro 2006). For every C_t, the calculation of the output weights fulfill the formulation of g_t RBFNuNets, which can be demonstrated by the pairs $(C_1, w_1), (C_2, w_2), \ldots, (C_{g_t}, w_{g_t})$.

(a) The fitness value of each unit matrix within population is calculated via Eq. (5) (i.e., MAE^{-1}).

(2) **ACO approach** (Dorigo, Maniezzo, and Colorni 1996; Savsani, Jhala, and Savsani 2014):

Assume the ant home comprise K ants. In the primary of the optimization program, all pathways are initialized as an equality amount of pheromone. In per period, ants stride at the home node, journey over the diversified layers from the original to the final layer, and accomplish at the goal node (Savsani, Jhala, and Savsani 2014). Then, according Eq. (8), all ants may choice alternative node in per layer (Dorigo, Maniezzo, and Colorni 1996).

$$P_{ij}^{k} = \begin{cases} \eta_{ij}^{\partial} \cdot \left(\sum \eta_{ij}^{\partial} \right)^{-1} \mathrm{if} j \in K_i^k \\ 0 \mathrm{if} j \notin K_i^k \end{cases} \tag{8}$$

In which, P_{ij}^{k} represents the probability of picking node j as the latter intention goal node for ant k situated at node i, η_{ij} is the pheromone exam and ∂ is the pheromone impressibility.

Supposing the pathway is stop, the ant sediments few pheromones on the pathway based on the regionally exam updating rule given via Eq. (9):

$$\eta_{ij} = \eta_{ij} + \Delta\eta^{k} \tag{9}$$

where $\Delta\eta^{k}$ is the pheromone cumulating by k^{th} ant on the pathway it has passing.

When all ants accomplish their pathways, the pheromones on the universal best pathway are modified taking the universally exam improving rule given via Eq. (10).

$$\eta_{ij} = (1 - \varphi)\eta_{ij} + \sum_{k=1}^{K} \Delta\eta_{ij}^{k} \tag{10}$$

where φ is the pheromone diminish (steam) rate, $\Delta\eta_{ij}^{k}$ is the pheromone sediment through the best ant k on the pathway ij assessed as $H \cdot MAE_k^{-1}$, and H is a constant (Dorigo, Maniezzo, and Colorni 1996). Furthermore, exploration is the capability to seek the global space and is cooperated with absconding from local area optimal while avoiding trapped local optimal

standstill (Faramarzi et al. 2020). Accordingly, for the generated population through ACO approach, the ant establishes superior solution through referring other ants and itself, decides the subsequent direction and thus may exploration in a universal search space.

(3) **Duplication**: The population boosted through progressive learning of ACO approach (Dorigo, Maniezzo, and Colorni 1996; Kozak and Boryczka 2015) is reproduced and is called as ACO population.

(4) **GA approach**: The local space seek technique is established on the vicinity construction and the regulations which determine the method to receive a new solution from the present one. Its essential opinion is to revise the existing solutions in terms of the revision technique determined via the operator from its neighborhood, so a new practical solution with promising performance is generated (Qiu and Lau 2014). On the other hand, exploitation is the capability to seek locally around prospective solutions with attempt to improve their quality (Faramarzi et al. 2020). The procedure of GA progress that comprises two-point mutation and two-point crossover operators within the population of ACO approach progressive learning is named as [GA+ACO] subpopulation. The operators utilized in GA approach are as declarative below.

(a) GA adopts the crossover opinion to generate improved solutions (i.e., offspring). Besides, based on several suited solutions, stipulated as parents. Crossover is a natural phenomenon which helps retain diversity in ecosystem and with this sensation, is to explore the region (Faramarzi et al. 2020). Accordingly, Figure 4 explains the idea of twp-point crossover idea schematically. Next, every row of the picked paired C_t will implement two-point crossover operator with P_c.

$$\begin{bmatrix} c_{1,1}^{t} & c_{1,2}^{t} & \cdots & c_{1,N}^{t} & d_{1}^{t} \\ 0 & 0 & \cdots & 0 & 0 \\ c_{2,1}^{t} & c_{2,2}^{t} & \cdots & c_{2,N}^{t} & d_{2}^{t} \\ c_{3,1}^{t} & c_{3,2}^{t} & \cdots & c_{3,N}^{t} & d_{3}^{t} \\ c_{4,1}^{t} & c_{4,2}^{t} & \cdots & c_{4,N}^{t} & d_{4}^{t} \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_{t} \Leftrightarrow \begin{bmatrix} c_{1,1}^{t+1} & c_{1,2}^{t+1} & \cdots & c_{1,N}^{t+1} & d_{1}^{t+1} \\ c_{2,1}^{t+1} & c_{2,2}^{t+1} & \cdots & c_{2,N}^{t+1} & d_{2}^{t+1} \\ c_{3,1}^{t+1} & c_{3,2}^{t+1} & \cdots & c_{3,N}^{t+1} & d_{3}^{t+1} \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ c_{4,1}^{t+1} & c_{4,2}^{t+1} & \cdots & c_{4,N}^{t+1} & d_{4}^{t+1} \\ c_{5,1}^{t+1} & c_{5,2}^{t+1} & \cdots & c_{5,N}^{t+1} & d_{5}^{t+1} \end{bmatrix}_{t+1}$$

Figure 4. Illustrative schema of two-point crossover between C_t and C_{t+1} through each pair of rows individually alternating their values with P_c.

$$\begin{bmatrix} c_{1,1}^t & c_{1,2}^t & \cdots & c_{1,N}^t & d_1^t \\ 0 & 0 & \cdots & 0 & 0 \\ c_{2,1}^t & c_{2,2}^t & \cdots & c_{2,N}^t & d_2^t \\ c_{3,1}^t & c_{3,2}^t & \cdots & c_{3,N}^t & d_3^t \\ c_{4,1}^t & c_{4,2}^t & \cdots & c_{4,N}^t & d_4^t \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_t \Leftrightarrow \begin{bmatrix} c_{1,1}^{t+1} & c_{1,2}^{t+1} & \cdots & c_{1,N}^{t+1} & d_1^{t+1} \\ c_{2,1}^{t+1} & c_{2,2}^{t+1} & \cdots & c_{2,N}^{t+1} & d_2^{t+1} \\ c_{3,1}^{t+1} & c_{3,2}^{t+1} & \cdots & c_{3,N}^{t+1} & d_3^{t+1} \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ c_{4,1}^{t+1} & c_{4,2}^{t+1} & \cdots & c_{4,N}^{t+1} & d_4^{t+1} \\ c_{5,1}^{t+1} & c_{5,2}^{t+1} & \cdots & c_{5,N}^{t+1} & d_5^{t+1} \end{bmatrix}_{t+1}$$

(a) Mutations incite the offspring to have properties distinct from their parents. In GA this operator is aimed at local area search and to exploit results (Faramarzi et al. 2020). Take this two-point mutation, the values are replaced through randomly selected values from the scope of the search region in everyone dimension, which retains the variability and generates new solutions.

(5) **Reproduction**: For the aim of imposing the GA to disseminate the genetic substance more greatly from the best parents, the Roulette wheel selection (RWS) (Goldberg 1989) part was used to format the copulation pairs (Kuzmanovski, Lazova, and Aleksovska 2007). Through enhanced evolution, the [GA+ACO] and [ACO+GA] subpopulations are further integrated. Units with equivalent amount from the original population are stochastic picked by the proportional RWS (Goldberg 1989) for the evolution afterward. As such, through utilizing GA and ACO approaches to conduct exploitation and exploration in the resolving region respectively. Accordingly, it is anticipated to receive the superior solution regarding their best complementary features.

(a) The (F_2, Z_2) subset is adopted in this step as a testing set in the following method. First, the predictions $\hat{Z}_{2,1}, \hat{Z}_{2,2}, \ldots, \hat{Z}_{2,T}$ of the T RBFNuNet established in the previous step and the corresponding MAE_t are calculated as follows:

$$MAE_t = T^{-1}\left(\sum_{t=1}^{T} \left|Z_2 - \hat{Z}_{2,t}\right|\right) \tag{11}$$

(a) The pair(C_t, w_t)related with the maximum error is substituted by the best RBFNuNet of the previous iteration so that the optimal solution survives in all iterations (this substitution will not occur in the initial iteration). The RBFNuNet related with the minimum error is stored for further adoption. The purpose is to offer higher survival possibility for the RBFNuNet related with smaller error values. Thus, the probability of selection p_t of each C_t is calculated through Eq. (12)

$$p_t = MAE_t^{-1} \cdot \left[\sum_{t=1}^{T} (MAE_t^{-1}) \right]^{-1} \tag{12}$$

and the cumulative probability q_t is calculated through Eq. (13).

$$q_t = \sum_{i=1}^{t} p_i \tag{13}$$

Subsequently, since the characteristic of regional search with GA approach, regardless what the values of objective function of the units within population are, them absolutely have the possibility to make advancement with few heritable operators and entry the inferior iteration of population to fulfill.

(6) **Termination**: ECs apply global search method for optimization without former heuristics mechanism for any particular domain. Additionally, ECs conform the survival of the fittest principle and convergence to a better solution at each iteration (Dey, Bhattacharyya, and Maulik 2014). Thereby, the IGACO algorithm will perform until returning to step (2) only in a definite count of iterations has been arrived.

Hence, enforcing an evolution procedure through the ACO approach would obtain an advanced population, which is superior than the original population. In addition, the superiority of the characteristic of universal search in ACO approach permits extensive exploration on dimensionality region among different examinations and the resolving interval is able to be enlarged. In addition, as the IGACO algorithm evolutions, the units of the population progress gradually. In this process, the IGACO algorithm corresponds the nature of GA approach, insures the inherited diversity in the advanced evolution, and makes enhances to receive a new improved population. Moreover, on the basis of the GA approach in the IGACO algorithm to estimate the fitness function of unit parameters set solution within the population, the dominance solutions will be achieved progressively. And then, the resolving region within population could be refined gradually and convergence toward to the global area optimal solution.

In the following experiment, the IGACO algorithm stops and the RBFNuNet corresponding to the maximum fitness value is chosen. Lastly, it is validated through adopting the (F_3, Z_3) subset, which has not been adopted throughout the whole learning procedure. Once those crucial parameter values are settled, RBFNuNet starts the training of approximation and learning via five continuous test functions.

Experimental Results

This section concentrated on learning and adjusting the relevant parameters in RBFNuNet for function approximation task. The goal is to acquire the optimal appropriate fitness values regarding the parameters solution of the RBFNuNet.

The purpose is then to decide the adequate values of the parameters set from the searching region in the examination. The proposed IGACO algorithm will progressively may tune and therefore acquire the solutions of parameters value set for RBFNuNet.

In this section, all experiments have been implemented in Java 4.7.3a and conducted adopting a standard commercial Laptop (Microsoft Windows 10 64-bit operating system, Intel Core^{TM} i7-4770 3.4 GHz CPU with 16 GB RAM).

Benchmark Problems Experiment

Experimental exam function causes wonderful approximation to reimburse RBFNuNet for the effect of nonlinear mapping correlation. This paper utilizes five continuous test functions that are always applied in the literature to be the competitive benchmark of measured algorithms.

The unimodal functions are trialed for benchmarking the exploitation of algorithms as them have only one global optimal. On the other hand, the multimodal and compound functions, and them have many local optimal which in return are adequate for benchmarking the expression of algorithms and prevent local optimal as well as exploration assessment (Saremi et al., 2017). Therefore, the examination contains the following five benchmark problems, including Griewank, Sphere, Rosenbrock (Bilal et al. 2020; Shelokar et al. 2007), Mackey-Glass time series (Liu et al. 2014; Whitehead and Choate 1996), and B2 (Shelokar et al. 2007) continuous test functions.

The first examination, Griewank function (Bilal et al. 2020; Shelokar et al. 2007) is presented as follows:

$$GR(x_j, x_{j+1}) = \sum_{j=1}^{n} \frac{x_j^2}{4000} - \prod_{j=1}^{n} \cos\left(\frac{x_{j+1}}{\sqrt{j+1}}\right) + 1 \tag{14}$$

(a) search domain: $-100 \leqq x_j \leqq 100$, $j = 1$;
(b) one global minimum: $(x_1, x_2) = (0, 0)$; $GR(x_1, x_2) = 0$.

In the second examination, Sphere function (Bilal et al. 2020; Shelokar et al. 2007) is presented as follows:

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{15}$$

(a) search domain: $-100 \leq x_i \leq 100$, $i = 1$;
(b) one global minimum: $(x_1, x_2) = (0, 0)$; $SP(x_1, x_2) = 0$.

In the third examination, Rosenbrock function (Bilal et al. 2020; Shelokar et al. 2007) is presented as follows:

$$RS(x_j, x_{j+1}) = \sum_{j=1}^{n-1} [100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2] \tag{16}$$

(a) search domain: $-30 \leqq x_j \leqq 30$, $j = 1$;
(b) one global minimum: $(x_1, x_2) = (1, 1)$; $RS(x_1, x_2) = 0$.

In the fourth examination, the Mackey-Glass time series (Liu et al. 2014; Whitehead and Choate 1996) is presented as follows:

$$\frac{dx(t)}{d(t)} = 0.1x(t) + 0.2 \cdot \frac{x(t-17)}{1 + x(t-17)^{10}} \tag{17}$$

where $x(t)$ is the value of time series at time step t. The research for the retrieval t ranges from 118 to 1118 with the Mackey-Glass time series function, from which 1000 specimens were randomly produced (Whitehead and Choate 1996). The data set is established with second-order Runge-Kutta method and with step size of 0.1 (Song et al. 2011).

In the fifth examination, B2 function (Shelokar et al. 2007) is presented as follows:

$$B2(x_j, x_{j+1}) = x_j^2 + 2x_{j+1}^2 - 0.3\cos(3\pi x_j) - 0.4\cos(4\pi x_{j+1}) + 0.7 \tag{18}$$

(a) search domain: $-100 \leqq x_j \leqq 100$, $j = 1$;
(b) one global minima: $(x_1, x_2) = (0, 0)$; $B2(x_1, x_2) = 0$.

Parameter Setup

There are several associated parameter values within RBFNuNet that must be set prior to execute training for function approximation. In addition, the IGACO algorithm is considered the better method to train RBFNuNet than the trial and error way in the literature since it has a preset range for each benchmark function associated to its own search domain. These estimated algorithms are started with the evaluation of the parameters setting for five benchmark functions listed in Table 1.

In the IGACO algorithm, four parameters (i.e., mutation rate, crossover rate, pheromone exam, and pheromone diminish), which have significant influence on estimation results are inspected. Simultaneously, this examination referred to the associated literature for the interval of the parameter values situation. Latter, the configuration of the parameters setup for the IGACO algorithm is revised by consulting to the Taguchi experimental (Taguchi et al. 2005) design with inspection mode displace for applying trial and error

Table 1. The parameters setting for several benchmark problems in the experiment.

	Continuous test function				
Description	Griewank	Sphere	Rosenbrock	Mackey-Glass time series	B2
Search domain	[−100, 100]	[−100, 100]	[−30, 30]	[0.4, 1.6]	[−100, 100]
The widths on RBFNuNet	[43000, 44200]	[0.1, 0.9]	[200, 700]	[0.1, 0.3]	[23000, 26000]

Table 2. Parameter values setting for the IGACO algorithm.

Parameter	Description	Value
E	The maximum number of iterations	1000
g_t	The number of the RBFNuNet hidden neuron centers	[1, 50]
d_i^t	The width of RBFNuNet hidden neuron	[1100, 39000]
δ	The learning rate of the RBFNuNet	0.34
T	Population size	50
K	The number of ant	100
η	The pheromone exam	0.48
∂	The pheromone impressibility	0.72
φ	The pheromone diminish rate	0.37
H	A constant	0.68
P_c	Crossover rate (2-point crossover)	[0.45, 0.65]
P_m	Mutation rate (2-point mutation)	[0.1, 0.18]

process (Yin et al. 2020). The Taguchi method (Taguchi et al. 2005) is utilized where orthogonal arrays are adopted to significantly reduce the number of experiments (Taguchi et al. 2004). Besides, Taguchi suggested that the signal-to-noise (S/N) ratio is a well selection for performance evaluation. A realistic solution for the present experimentation should be as large as possible (Kuo et al. 2015). Thus, the Taguchi trial analysis and trials were configured in a L_9 (3^4) orthogonal array (i.e., 4 factors with 3 levels, and 9 experiments) for the IGACO algorithm after the experiment was carried out 50 times. Meantime, the MINITAB 18 (statistical software) was used in the analysis of parameter design for the IGACO algorithm, where the stability of system quality in the experiment is assessed by the S/N ratio (Lin et al. 2009). After that, the maximum count of iterations is fix at 1,000 to set as termination situation in the examination. Finally, the assessment of the parameter values setting for the IGACO algorithm was executed with the detail exhibited in Table 2.

Performance Assessment and Comparison

The adjusting of all measured algorithms on few solution sets of parameters (i.e., the center within hidden neuron, width, and weight) configuration for RBFNuNet that are yielded via the population during the manipulation of the progression sequence in the examination are dissertated in this section. After that, 1000 stochastically yielded datasets are partition into three sections (i.e., 65% training dataset, 25% testing dataset, and 10% validation dataset) (Looney 1996) to train RBFNuNet. In which, we can examine the studying status and adjust the parameters' arrangement. Afterward, this

study employs these measured algorithms to resolve the optimal solution sets of parameters configuration for RBFNuNet. And then it stochastically yields non-repetitive 65% training dataset from 1000 yielded specimen and transfer the dataset into RBFNuNet for training. With the identical manner, it stochastically yields nonrepetitive 25% testing dataset to examine unit's parameters configuration solution within population and estimates the fitness function. So far, RBFNuNet has applied 90% dataset in studying phase. After one thousand iterations in the evolvement operation, the optimal solution sets of parameters configuration for RBFNuNet are acquired. Lastly, it stochastically yields nonrepetitive 10% validation dataset to certificate how the parameters configuration solution of unit approximates the five examinations and remain the root mean square error (RMSE) values to explain the studying phase of RBFNuNet. In case the data extracting stage mentioned above have fulfilled, all measured algorithms are ready to enforce. The studying and certification phases mentioned above were implemented 50 times before the average of RMSE (i.e., $RMSE_{avg}$) values were assessed. The values of the $RMSE_{avg}$ and standard deviation (SD) for all measured algorithms estimated from the examination are revealed in Table 3.

In Table 3, the presented outcomes evidence that IGACO algorithm acquires the accurate enough values with steady representation during the studying process of the examination. Consequently, RBFNuNet may achieve the single solution set of parameters configuration from the progress process within population, which has implemented the circumstance with dominant function approximation. When the training of RBFNuNet via the IGACO algorithm is fulfilled, the unit with the optimal solution sets of parameters configuration (i.e., the center within hidden neuron, width, and weight) in studying phase is the RBFNuNet setting in certain.

Furthermore, when a large amount of training specimens is adopted compared to the number of model parameters, the problem of overtraining can be considered minor (Shinozaki and Ostendorf 2008). As shown in Table 4, the values of training and validation expression are persistently small, which represents that RBFNuNet trained through the IGACO algorithm offers inevitable stability. Hence overfitting and over-training problems do not emerge in the experiment utilizing the IGACO algorithm. Such result not only satisfies for the training and validation set, a generalization could also be made regarding other unseen dataset. Additionally, since the numerical results contrast are significant in Table 4, the superiority of expression results received from the IGACO algorithm when inspected with different datasets is clearly presented. Consequently, the IGACO algorithm indicates exceptional studying through five benchmark continuous test functions and reveals superior approximation consequences.

Table 3. Result comparison among relevant algorithms employed in this experiment.

Benchmark problem	Griewank		Sphere		Rosenbrock		Mackey-Glass time series		B2	
Competitive algorithm	Training set	Validation set	Training set	Validation set	Training set	Validation set	Training set	Validation set	Training set	Validation set
RBFNuNet (Chen, Wu, and Luk 1999)	27.64 ± 2.82	28.34 ± 3.68	2746.24 ± 334.61	2432.34 ± 320.35	12058.19 ± 1360.41	12969.45 ± 2623.37	703 ± 41.6	712 ± 53.6	5803.27 ± 413.66	5935.23 ± 690.28
GA (Holland 1992)	19.76 ± 16.31	54.43 ± 17.55	1272.21 ± 109.30	1507.23 ± 373.41	1674.13 ± 376.29	2591.39 ± 252.34	2.75E-3 ± 1.52E-4	2.53E-3 ± 1.95E-4	1131.06 ± 394.38	3874.13 ± 539.20
GA-based (Sarimveis et al. 2004)	17.23 ± 16.31	52.65 ± 16.23	1137.54 ± 104.71	1463.28 ± 314.86	1552.56 ± 335.47	2362.67 ± 236.33	2.67E-3 ± 1.45E-4	2.51E-3 ± 1.36E-4	1088.39 ± 324.72	3529.44 ± 505.26
ACO (Dorigo, Maniezzo, and Colorni 1996)	7.57 ± 13.49	51.34 ± 10.25	958.14 ± 11.39	1204.71 ± 221.25	1233.21 ± 146.43	1984.16 ± 360.30	2.70E-3 ± 1.59E-4	2.46E-3 ± 1.92E-4	1014.39 ± 134.27	2782.31 ± 451.63
ACO-based (Tabakhi et al., 2014)	7.26 ± 11.09	48.51 ± 10.24	913.86 ± 10.73	1239.09 ± 207.35	1181.23 ± 117.81	1897.11 ± 315.05	2.88E-3 ± 1.56E-4	2.41E-3 ± 1.79E-4	2147.82 ± 119.14	2502.03 ± 430.63
GA-ACO (Luan et al. 2019)	6.69 ± 3.53	15.28 ± 4.37	413.51 ± 40.67	361.02 ± 25.39	912.31 ± 20.27	626.54 ± 28.23	2.07E-3 ± 1.81E-4	1.95E-3 ± 1.46E-4	298.16 ± 72.81	356.77 ± 16.95
IGACO	**5.15 ± 2.61**	**13.30 ± 2.43**	**362.04 ± 28.29**	**297.24 ± 19.72**	**705.14 ± 12.13**	**587.17 ± 21.36**	**1.69E-3 ± 1.41E-4**	**1.52E-3 ± 1.28E-4**	**251.48 ± 50.24**	**317.39 ± 15.52**

Table 4. Contrast of the best learning expression among relevant algorithms in the experiment.

Benchmark problem Competitive algorithm	Griewank	Sphere	Rosenbrock	Mackey-Glass time series	B2
			RMSE		
RBFNuNet (Chen, Cowan, and Grant 1991)	3.421	7.288	5.392	4.173	7.193
GA (Holland 1992)	***5.175E-1***	42.831E-2	***47.816E-5***	35.794E-2	***25.269E-2***
GA-based (Sarimveis et al. 2004)	5.142E-1	39.377E-2	47.211E-5	33.528E-2	24.843E-2
ACO (Dorigo, Maniezzo, and Colorni 1996)	5.092E-1	***61.395E-2***	43.843E-5	***38.271E-2***	22.075E-2
ACO-based (Tabakhi et al., 2014)	5.027E-1	53.471E-2	40.349E-5	31.604E-2	19.845E-2
GA-ACO (Luan et al. 2019)	4.828E-1	3.205E-2	3.914E-5	4.383E-2	5.660E-2
IGACO	**4.179E-1**	**2.078E-2**	**3.242E-5**	**2.737E-2**	**4.735E-2**

Table 5. Comparison of the time consumed (in seconds) among relevant algorithms arriving at the preset RMSE threshold.

Experiment (Benchmark problem)	Griewank	Sphere	Rosenbrock	Mackey-Glass time series	B2
RMSE threshold	***5.175E-1***	***61.395E-2***	***47.81E-5***	***38.271E-2***	***25.26E-2***
Competitive algorithm					
OLS (Chen, Cowan, and Grant 1991)	–	–	–	–	–
GA (Holland 1992)	15.265	4.231	3.793	5.098	663.513
GA-based (Sarimveis et al. 2004)	11.389	6.379	5.285	4.872	349.228
ACO (Dorigo, Maniezzo, and Colorni 1996)	13.749	7.263	3.346	8.271	526.381
ACO-based (Tabakhi et al., 2014)	10.598	11.450	5.813	6.095	276.313
GA-ACO (Luan et al. 2019)	2.489	2.817	1.205	2.337	2.601
IGACO	**1.871**	**2.279**	**0.748**	**1.593**	**1.926**

Note. "–" means that it failed to achieve the preset RMSE threshold

On the other hand, the comparison of the best learning expression during the training is addressed in Table 4. In which, it can be concluded that the IGACO algorithm has the smallest gains in RMSE value among relevant algorithms. It produced the lowest RMSE values and the optimal tuning of parameter setting in RBFNuNet and thus the IGACO algorithm was able to reach the best expression. Besides, this paper utilizes the proper RMSE value for five benchmark problems from Table 4 and yields as the threshold. The time consumed (in seconds) for all algorithms were listed in Table 5.

The results given in Table 5 represent that the proposed IGACO algorithm spends the least time among relevant algorithms to achieve the current RMSE threshold value for five benchmark problems. Consequently, experimental results in Tables 4 and 5 indicate that the IGACO algorithm surpasses other algorithms in terms of fitting preciseness and execution time.

Table 6. The data period of the spot gold price forecast exercise.

Spot gold price forecast	The observations: month/day/year (number of sampling)	
	Learning set (90%)	Forecasting set (10%)
Data Period	Feb 1st 2008~ Dec-23 2008 (228)	Dec-29 2008~ Feb 2nd 2009 (24)

Practical Exercise for the Spot Gold Price Forecast

It has been indicated that RBFNuNet is able to reach precise approximation on five benchmark problems through the proposed IGACO algorithm. The results are compared with other algorithms in literatures with indication of the preciseness of the IGACO algorithm.

This assessment attempts to study the preciseness of forecast on spot gold price records of the London Afternoon (PM) Gold Price from Feb 1st, 2008 to Feb 2nd, 2009 (252 records in total) which is utilized as observations in this study. The data period of this exercise is presented in Table 6.

Additionally, the forecasting and verification of spot gold price is directly priced in US dollar (US$). Besides, the analysis has supposed that the effect of exogenous interfere variables did not emerge and the spot gold price data was not disturbed via any external events.

Build Box-Jenkins Models

Time series data are often assessed in expectation of discovering a historical pattern that can be utilized in the forecasting. Box and Jenkins (1976) developed the ARIMA methodology to forecast time series events. In this section, in order to assure the predictions of Box-Jenkins models could be fulfilled, the case study for the spot gold price forecast was utilized to inspect the models. In addition, EViews™ 11.0 and SPSS™ 16.0 (statistical software) were adopted for the decomposition of Box-Jenkins models to estimate the numerical results. If the data are stationary, model estimation can be implemented directly; otherwise, differencing must be executed to make it stationary.

Further, the study implemented spot gold price forecast based on Box-Jenkins models. The ARIMA (p, d, q) modeling procedure has three steps: (a) to identify the model order (i.e., p, d, and q); (b) to estimate the model coefficients; and (c) then to forecast the data (Babu and Reddy 2014). Next, this study executes the data identification of ARIMA models via augmented Dickey-Fuller (Dickey & Fuller, 1981) (ADF) testing. Thus, it can adopt ARIMA (p, d, q) models to proceed measure and forecast of the spot gold price data. Moreover, the optimal model (Engle, Robert, and Yoo 1987) was filtered out by applying the criteria of Akaike information criterion (AIC) (i.e., AIC value = 8.551) (Akaike 1974). Based on the results, it concludes that the AIC value of ARIMA (2, 1, 2) model is the smallest (i.e., adjusted R-square = 0.0043) among every candidate ARIMA models, showing that it

is the optimal model and thus the most adequate one for the spot gold price data. The results of model diagnosis indicate that the values of Q-statistic (i.e., Ljung-Box statistic) (Kmenta 1986) are greater than 0.05 in result of ARIMA models, in which are serial noncorrelation (i.e., white noise) and had been adequate fitted. This study adopts the fittest ARIMA (2, 1, 2) model, which has verified model estimation and diagnosis to proceed the spot gold price forecast.

Parameters Setup for the Spot Gold Price Forecast Exercise

There are some values of parameters within RBFNuNet that must be set up prior to executing training for the exercise of forecasting analysis. Thus, the parameters' setting for the IGACO algorithm is received according to relevant literatures and Taguchi method. Moreover, the MINITAB 18 (statistical software) was applied in the analysis of parameter design. The Taguchi trials were configured in a L_9 (3^4) orthogonal array for the IGACO algorithm after the experiment was executed for 40 times. Finally, the IGACO algorithm was conducted with parameters setting listed in Table 7.

Error Estimate for Spot Gold Price Forecast

Looney (1996) suggests taking 65% of the parent database for training, 25% for testing, and 10% for validation respectively. On the other hand, most studies in the literatures have applied convenient ratio of splitting for in- and out-of-samples such as 70:30%, 80:20%, or 90:10% (Zou et al. 2007). Thus, this study uses the ratio of 90% (228 observations):10% (24 observations) as the basis of division. The spot gold price records are retrieved from Feb 1st, 2008 to Feb 2nd, 2009 (252 observations). The application example with the spot gold price forecast is based on time series data period and utilized for forecast analysis.

Table 7. Parameters setup for the IGACO algorithm in the spot gold price forecast exercise.

Parameter	Description	Value
E	The maximum number of iterations	1000
g_t	The number of the RBFNuNet hidden neuron centers	[1, 50]
d_i^t	The width of RBFNuNet hidden neuron	[1000, 38000]
δ	The learning rate of the RBFNuNet	0.25
T	Population size	50
K	The number of ant	100
η	The pheromone exam	0.4
∂	The pheromone impressibility	0.9
φ	The pheromone diminish rate	0.45
H	A constant	0.75
P_c	Crossover rate (two-point crossover)	[0.6, 0.7]
P_m	Mutation rate (two-point mutation)	[0.15, 0.25]

Accordingly, the learning stage of RBFNuNet will be based on daily spot gold price data; it includes training and testing sets (i.e., 65%+25%). The training began with entering in turn four observations retrieved from 65% training set to RBFNuNet. In this process, the unit parameters solution within the population inspects along with the whole evolution procedure, thus the fitness values of all units within the population could be estimated with the 25% testing set. At this point, 90% of the spot gold price data had been adopted to the learning stage of RBFNuNet, which actually generated a unit parameters solution with the most accurate forecasting. Consequently, it was necessary for the approximation expression of the RBFNuNet prediction to be assessed with the 10% validation set. Besides, the following predicted values were produced in turn from the moving window procedure. The first 90% of the observations were adopted for model estimation while the residual 10% were adopted for validation and gradually to move toward prediction. As summary, this section addresses how data is input to RBFNuNet for forecasting through several algorithms, and how the result is compared with Box-Jenkins models (i.e., ARIMA (2, 1, 2) model).

Moreover, the RMSE, mean absolute error (MAE), and mean absolute percentage error (MAPE) are the most common error estimates applied in business, and thus were utilized to assess the forecast models (Co and Boosarawongse 2007). Further, in Chen et al. (2020), the RMSE denoted the sample SD of the variances between observed and predicted values. As one of the commonly adopted error measure pointers in statistics, the RMSE was expounded as:

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \tag{19}$$

The MAE was the average of the absolute error between y_i and $\hat{y}_i$. It is expounded as follows:

$$MAE=\frac{1}{N}\sum_{i=1}^{N}\left|y_i-\hat{y}_i\right| \tag{20}$$

The MAPE was a statistical estimate of forecast for the preciseness of a forecast way. It normally represents the percentage of the output error and is expounded as below (Chen et al. 2020):

$$MAPE=\frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i-\hat{y}_i}{y_i}\right|\times 100\% \tag{21}$$

Table 8. The forecasting errors comparison for relevant algorithms used in the spot gold price forecast exercise.

	Error		
Competitive algorithm	RMSE	MAE	MAPE (%)
RBFNuNet (Chen, Wu, and Luk 1999)	8.217	5.729	2.134
GA (Holland 1992)	66.406E-4	68.053E-4	1.957
GA-based (Sarimveis et al. 2004)	53.037E-4	47.738E-4	1.423
ACO (Dorigo, Maniezzo, and Colorni 1996)	41.481E-4	22.391E-4	1.215
ACO-based (Tabakhi et al., 2014)	29.120E-4	18.925E-4	1.064
GA-ACO (Luan et al. 2019)	7.471E-4	11.301E-4	14.186E-2
IGACO	**6.236E-4**	**8.572E-4**	**11.874E-2**
ARIMA (2, 1, 2) model	16.172	11.173	1.328

Table 9. The statistical results for T-test among relevant algorithms.

	Paired differences			
Competitive algorithm	Mean	Standard deviation	*T*	Significant (two-tailed)
RBFNuNet (Chen, Wu, and Luk 1999)	−625.1048	641.0314	−3.163	0.017*
GA (Holland 1992)	−306.3764	622.7529	−1.381	0.143*
GA-based (Sarimveis et al. 2004)	−284.0823	551.5489	−1.026	0.172*
ACO (Dorigo, Maniezzo, and Colorni 1996)	−517.2013	633.3081	−2.736	0.034*
ACO-based (Tabakhi et al., 2014)	−369.8834	548.7602	−1.475	0.027*
GA-ACO (Luan et al. 2019)	−204.0923	526.3407	−0.928	0.019*
IGACO	**−125.3682**	**434.1382**	**−0.647**	**0.773**
ARIMA (2, 1, 2) model	−226.1536	584.4939	−1.349	0.235

*: 5% significance level
Note: Mean is equal to arithmetic average.

The algorithms of forecasting expressions mentioned earlier with the exercise data are shown in Table 8. The numerical results derived from RMSE, MAE, and MAPE (%) of the proposed IGACO algorithm were the smallest ones among relevant algorithms.

As for the verification of statistically significant difference, we received the results significantly while conducting the matched paired sample tests of T-test (5% significance level) with the absolute error from the estimated dataset of the source data in all algorithms. As result, the forecasting verification and the *T*-test results among relevant algorithms are presented in Table 9, which shows that the IGACO algorithm and ARIMA (2, 1, 2) model are not statistically significant (i.e., p value larger than 0.05 and it does not appear significant deviation between the predicted and actual values) and therefore provide more accurate forecasting than other algorithms.

Also, the statistical results reveal that the IGACO algorithm has the best expression for most accurate forecasting among relevant algorithms. Accordingly, the proposed IGACO algorithm can significantly provide the best results while the comparison results for the spot gold price (US$) forecast exercise is presented in Figure 5.

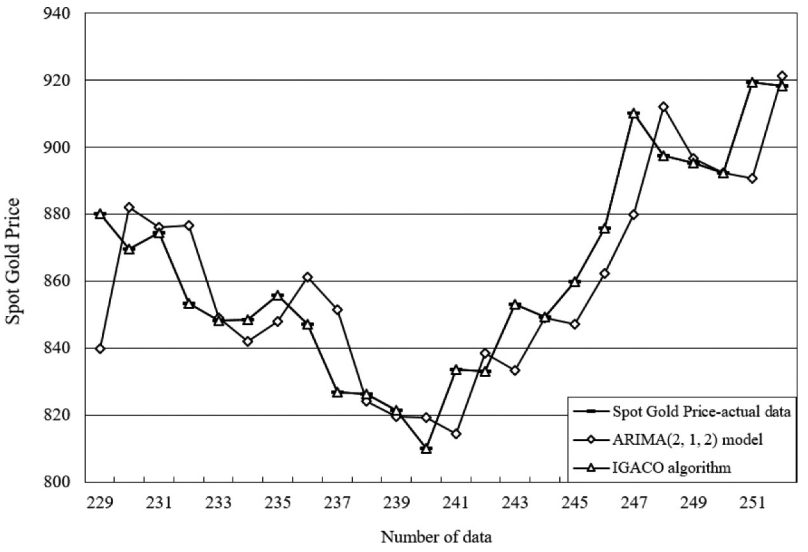

Figure 5. The forecasting results comparison of the proposed IGACO algorithm and Box-Jenkins model for the spot gold price forecast exercise.

Conclusions

This study proposed the IGACO algorithm by integrating GA and ACO approaches, which provides the solution of RBFNuNet parameter values. In addition, the spot gold price forecast exercise and the tuning values of parameters with RBFNuNet adopting the trained algorithm have been addressed. The empirical results indicated that GA and ACO approaches can be collaborated intelligently and exploit into an integrated algorithm which is achieving the optimal training representation among relevant algorithms in this paper. Furthermore, algorithm evaluation results for five benchmark continuous test functions and the spot gold price forecast exercise exhibits that the proposed IGACO algorithm surpassed relevant algorithms and the traditional ARIMA models in terms of forecasting preciseness and execution time. This analytical implication will be favorable in practice to allow lower financial risk and could be practical to determine advisable marketing strategy.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

ORCID

Zhen-Yao Chen http://orcid.org/0000-0002-2349-5486

References

Abualigah, L., and E. Hanandeh. 2015. Applying genetic algorithms to information retrieval using vector space model. *International Journal of Computer Science, Engineering and Applications* 5 (1):19–28. doi:10.5121/ijcsea.2015.5102.

Abualigah, L. M., A. T. Khader, and E. S. Hanandeh. 2018. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational Science* 25:456–66. doi:10.1016/j.jocs.2017.07.018.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19 (6):716–23. doi:10.1109/TAC.1974.1100705.

Amar, M. N., N. Zeraibi, and K. Redouane. 2018. Optimization of WAG process using dynamic proxy, genetic algorithm and ant colony optimization. *Arabian Journal for Science and Engineering* 43 (11):6399–412. doi:10.1007/s13369-018-3173-7.

Ansari, M., F. Othman, and A. El-Shafie. 2020. Optimized fuzzy inference system to enhance prediction accuracy for influent characteristics of a sewage treatment plant. *Science of the Total Environment* 722:137878–90. doi:10.1016/j.scitotenv.2020.137878.

Ayala, H. V. H., and L. D. S. Coelho. 2016. Cascaded evolutionary algorithm for nonlinear system identification based on correlation functions and radial basis functions neural networks. *Mechanical Systems and Signal Processing* 68-69:378–93. doi:10.1016/j.ymssp.2015.05.022.

Babu, C. N., and B. E. Reddy. 2014. A moving-average-filter-based hybrid ARIMA-ANN model for forecasting time series data. *Applied Soft Computing* 23 (10):27–38. doi:10.1016/j.asoc.2014.05.028.

Baeck, T., D. B. Fogel, and Z. Michalewicz. 2018. *Evolutionary Computation 1 Ba*sic Algorithms and Operators. New York, US: CRC press, Taylor & Francis Group.

Barra, T. V., G. B. Bezerra, and L. N. de Castro. 2006. An immunological density-preserving approach to the synthesis of RBF neural networks for classification. *The 2006 IEEE International Joint Conference on Neural Network Proceedings. Vancouver, BC, Canada*: 929–35.

Bilal, M. P., A. Abraham, A. Abraham, A. Abraham, and A. Abraham. 2020. Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence* 90:103479–502. doi:10.1016/j.engappai.2020.103479.

Box, G. E. P., and G. M. Jenkins. 1976. *Time series analysis, forecasting and control. San Francisco*. CA, USA: Holden-Day.

Cadenas, E., W. Rivera, R. Campos-Amezcua, and C. Heard. 2016. Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. *Energies* 9 (2):1–15. doi:10.3390/en9020109.

Chen, S., C. F. N. Cowan, and P. M. Grant. 1991. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 2 (2):302–09. doi:10.1109/72.80341.

Chen, S., Y. Wu, and B. L. Luk. 1999. Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks* 10 (5):1239–43. doi:10.1109/72.788663.

Chen, X., H. Tianfield, C. Mei, W. Du, and G. Liu. 2017. Biogeography-based learning particle swarm optimization. *Soft Computing* 21 (24):7519–41. doi:10.1007/s00500-016-2307-7.

Chen, X., H. Tianfield, and K. Li. 2019. Self-adaptive differential artificial bee colony algorithm for global optimization problems. *Swarm and Evolutionary Computation* 45:70–91. doi:10.1016/j.swevo.2019.01.003.

Chen, X., X. Cai, J. Liang, and Q. Liu. 2018. Ensemble learning multiple LSSVR with improved harmony search algorithm for short-term traffic flow forecasting. *IEEE Access* 6:9347–57. doi:10.1109/ACCESS.2018.2805299.

Chen, Y., J. Fan, Z. Deng, B. Du, X. Huang, and Q. Gui. 2020. PR-KELM: Icing level prediction for transmission lines in smart grid. *Future Generation Computer Systems* 102:75–83. doi:10.1016/j.future.2019.08.002.

Co, H. C., and R. Boosarawongse. 2007. Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. *Computers and Industrial Engineering* 53 (4):610–27. doi:10.1016/j.cie.2007.06.005.

Comuzzi, M. 2019. Optimal directed hypergraph traversal with ant-colony optimization. *Information Sciences* 471:132–48. doi:10.1016/j.ins.2018.08.058.

Cui, L., G. Li, X. Wang, Q. Lin, J. Chen, N. Lu, and J. Lu. 2017. A ranking-based adaptive artificial bee colony algorithm for global numerical optimization. *Information Sciences* 417:169–85. doi:10.1016/j.ins.2017.07.011.

Day, P., S. Iannucci, and I. Banicescu. 2020. Autonomic feature selection using computational intelligence. *Future Generation Computer Systems* 111:68–81. doi:10.1016/j.future.2020.04.015.

Del Ser, J., E. Osaba, D. Molina, X. S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, and F. Herrera. 2019. Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation* 48:220–50.

Deniz, A., H. E. Kiziloz, T. Dokeroglu, and A. Cosar. 2017. Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques. *Neurocomputing* 241:128–46. doi:10.1016/j.neucom.2017.02.033.

Denker, J. S. 1986. Neural network models of learning and adaptation. *Physica D* 22:216–32. doi:10.1016/0167-2789(86)90242-3.

Dey, S., S. Bhattacharyya, and U. Maulik. 2014. Quantum inspired genetic algorithm and particle swarm optimization using chaotic map model based interference for gray level image thresholding. *Swarm and Evolutionary Computation* 15:38–57. doi:10.1016/j.swevo.2013.11.002.

Dickey, D. A., and W. A. Fuller. 1981. Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica* 49(4): 1057–72.

Dorigo, M., V. Maniezzo, and A. Colorni. 1996. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26 (1):29–41. doi:10.1109/3477.484436.

Dowlatshahi, M., V. Derhami, and H. Nezamabadi-pour. 2017. Ensemble of filter-based rankers to guide an epsilon-greedy swarm optimizer for high-dimensional feature subset selection. *Information* 8 (4):152. doi:10.3390/info8040152.

Dowlatshahi, M. B., and V. Derhami. 2017. Winner determination in combinatorial auctions using hybrid ant colony optimization and multi-neighborhood local search. *The Journal of Artificial Intelligence and Data Mining* 5:169–81.

Du, K. L., and M. Swamy. 2016. *Ant colony optimization, in: Search and optimization by metaheuristics.*Midtown Manhattan, New York City, US: Springer International Publishing: 191-99.

Dzalbs, I., and T. Kalganova. 2020. Accelerating supply chains with ant colony optimization across a range of hardware solutions. *Computers and Industrial Engineering* 147:106610–23. doi:10.1016/j.cie.2020.106610.

Engle, R. F., F. Robert, and B. S. Yoo. 1987. Forecasting and Testing in Cointegrated Systems. *Journal of Econometrics* 35:588–89. doi:10.1016/0304-4076(87)90085-6.

Erdem, E., and J. Shi. 2011. ARMA based approaches for forecasting the tuple of wind speed and direction. *Applied Energy* 88 (4):1405–14. doi:10.1016/j.apenergy.2010.10.031.

Faramarzi, A., M. Heidarinejad, B. Stephens, and S. Mirjalili. 2020. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems* 191:105190–210. doi:10.1016/j.knosys.2019.105190.

George Lindfield, J. P. 2019. *Optimization methods*. Cambridge, Massachusetts, US: Academic Press: 433-83.

Ghafil, H. N., and K. Jarmai. 2020. Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications. *Applied Soft Computing* 93:106392–410. doi:10.1016/j.asoc.2020.106392.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Hajirahimi, Z., and M. Khashei. 2019. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence* 86:83–106. doi:10.1016/j.engappai.2019.08.018.

Hamida, Z., F. Azizi, and G. Saad. 2017. An efficient geometry-based optimization approach for well placement in oil fields. *Journal of Petroleum Science and Engineering* 149:383–92. doi:10.1016/j.petrol.2016.10.055.

Holland, J. H. 1992. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, MA, US: MIT Press, Google books.

Holland, O. 2008. Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. *Signal Processing* 88:1858–67. doi:10.1016/j.sigpro.2008.01.026.

Huang, Y., and Z. He. 2020. Carbon price forecasting with optimization prediction method based on unstructured combination. *Science of the Total Environment* 725:138350–63. doi:10.1016/j.scitotenv.2020.138350.

Huseyin, A., and F. Tansu. 2019. Wind speed forecasting by subspace and nuclear norm optimization based algorithms. *Sustainable Energy Technologies and Assessments* 35:139–47. doi:10.1016/j.seta.2019.07.003.

Islam, J., P. M. Vasant, B. M. Negash, M. B. Laruccia, M. Myint, and J. Watada. 2020. A holistic review on artificial intelligence techniques for well placement optimization problem. *Advances in Engineering Software* 141:102767–86. doi:10.1016/j.advengsoft.2019.102767.

Jakobsson, S., B. Andersson, and F. Edelvik. 2009. Rational radial basis function interpolation with applications to antenna design. *Journal of Computational and Applied Mathematics* 233 (4):889–904. doi:10.1016/j.cam.2009.08.058.

Jose-Garcia, A., and W. Gomez-Flores. 2016. Automatic clustering using nature-inspired metaheuristics: A survey. *Applied Soft Computing* 41:192–213. doi:10.1016/j.asoc.2015.12.001.

Kaur, S., L. K. Awasthi, A. Sangal, and G. Dhiman. 2020. Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence* 90:103541–70. doi:10.1016/j.engappai.2020.103541.

Khashei, M., M. Bijari, G. Ali, and R. Ardali. 2009. Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). *Neurocomputing* 72 (4–6):956–67. doi:10.1016/j.neucom.2008.04.017.

Khashei, M., and Z. Hajirahimi. 2018. A comparative study of series arima/mlp hybrid models for stock price forecasting. *Communication in Statistics-Simulation and Computation* 47:1–16.

Kmenta, J. 1986. *Elements of econometrics (2nd ed.)*. New York: Macmillan Publishing Co.

Kouziokas, G. N. 2020. A new W-SVM kernel combining PSO-neural network transformed vector and Bayesian optimized SVM in GDP forecasting. *Engineering Applications of Artificial Intelligence* 92:103650–60. doi:10.1016/j.engappai.2020.103650.

Kozak, J., and U. Boryczka. 2015. Multiple Boosting in the Ant Colony Decision Forest meta-classifier. *Knowledge-Based Systems* 75:141–51. doi:10.1016/j.knosys.2014.11.027.

Kristjanpoller, W., and E. Hernandez. 2017. Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications* 84:290–300. doi:10.1016/j.eswa.2017.05.024.

Kuo, R. J., Y. H. Lee, F. E. Zulvia, and F. C. Tien. 2015. Solving bi-level linear programming problem through hybrid of immune genetic algorithm and particle swarm optimization algorithm. *Applied Mathematics and Computation* 266:1013–26. doi:10.1016/j.amc.2015.06.025.

Kuzmanovski, I., S. D. Lazova, and S. Aleksovska. 2007. Classification of perovskites with supervised self-organizing maps. *Analytica Chimica Acta* 595 (1–2):182–89. doi:10.1016/j.aca.2007.04.062.

Li, M., S. Lian, F. Wang, Y. Zhou, B. Chen, L. Guan, and Y. Wu. 2020. A decision support system using hybrid AI based on multi-image quality model and its application in color design. *Future Generation Computer Systems* 113:70–77. doi:10.1016/j.future.2020.06.034.

Lin, C. F., C. C. Wu, P. H. Yang, and T. Y. Kuo. 2009. Application of Taguchi method in lightemitting diode backlight design for wide color gamut displays. *Journal of Display Technology* 5 (8):323–30. doi:10.1109/JDT.2009.2023606.

Liu, B., H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell. 2014. An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques. *IEEE Transactions on Antennas and Propagation* 62 (1):7–18. doi:10.1109/TAP.2013.2283605.

Liu, H., S. Shi, P. Yang, and J. Yang. 2018. An improved genetic algorithm approach on mechanism kinematic structure enumeration with intelligent manufacturing. *Journal of Intelligent and Robotic Systems* 89 (3–4):343–50. doi:10.1007/s10846-017-0564-z.

Looney, C. G. 1996. Advances in feedforward neural networks: Demystifying knowledge acquiring black boxes. *IEEE Transactions on Knowledge and Data Engineering* 8 (2):211–26. doi:10.1109/69.494162.

Luan, J., Z. Yao, F. Zhao, and X. Song. 2019. A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization. *Mathematics and Computers in Simulation* 156:294–309. doi:10.1016/j.matcom.2018.08.011.

Ma, H., S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou. 2019. Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey. *Swarm and Evolutionary Computation* 44:365–87. doi:10.1016/j.swevo.2018.04.011.

Moayedi, H., A. Moatamediyan, H. Nguyen, X. N. Bui, D. T. Bui, and A. S. A. Rashid. 2019. Prediction of ultimate bearing capacity through various novel evolutionary and neural network models. *Engineering with Computers* 36 (2):671–87. doi:10.1007/s00366-019-00723-2.

Mortazavi, A., V. Toğan, and M. Moloodpoor. 2019. Solution of structural and mathematical optimization problems using a new hybrid swarm intelligence optimization algorithm. *Advances in Engineering Software* 127:106–23. doi:10.1016/j.advengsoft.2018.11.004.

Mustaffa, Z., Y. Yusof, and S. S. Kamaruddin. 2014. Enhanced artificial bee colony for training least squares support vector machines in commodity price forecasting. *Journal of Computational Science* 5 (2):196–205. doi:10.1016/j.jocs.2013.11.004.

Nagra, A. A., F. Han, Q. H. Ling, and S. Mehta. 2019. An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization. *IEEE Access* 7:50388–99. doi:10.1109/ACCESS.2019.2903137.

Nanda, S. J., and G. Panda. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation* 16:1–18. doi:10.1016/j.swevo.2013.11.003.

Nasir, A. N. K., and M. O. Tokhi. 2015. Novel metaheuristic hybrid spiral-dynamic bacteria-chemotaxis algorithms for global optimization. *Applied Soft Computing* 27:357–75. doi:10.1016/j.asoc.2014.11.030.

Oprea, M. 2020. A general framework and guidelines for benchmarking computational intelligence algorithms applied to forecasting problems derived from an application domain-oriented survey. *Applied Soft Computing* 89:106103–27. doi:10.1016/j.asoc.2020.106103.

Pendharkar, P. C. 2015. An ant colony optimization heuristic for constrained task allocation problem. *Journal of Computational Science* 7:37–47. doi:10.1016/j.jocs.2015.01.001.

Qasem, S. N., S. M. Shamsuddin, and A. M. Zain. 2012. Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. *Knowledge-Based Systems* 27:475–97. doi:10.1016/j.knosys.2011.10.001.

Qiu, X., and H. Y. K. Lau. 2014. An AIS-based hybrid algorithm for static job shop scheduling problem. *Journal of Intelligent Manufacturing* 25:489–503. doi:10.1007/s10845-012-0701-2.

Rais, H. M., and T. Mehmood. 2018. Dynamic ant colony system with three level update feature selection for intrusion detection. *International Journal of Network Security* 20 (1):184–92.

Rani, R. H. J., and T. A. A. Victoire. 2018. Training radial basis function networks for wind speed prediction using pso enhanced differential search optimizer. *PLoS One* 13 (5):1–35. doi:10.1371/journal.pone.0196871.

Saremi, S., S. Mirjalili, and A. Lewis. 2017. Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software* 105: 30–47.

Sarimveis, H., A. Alexandridis, S. Mazarakis, and G. Bafas. 2004. A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms. *Computers and Chemical Engineering* 28:209–17. doi:10.1016/S0098-1354(03)00169-8.

Savsani, P., R. L. Jhala, and V. Savsani. 2014. Effect of hybridizing Biogeography-Based Optimization (BBO) technique with Artificial Immune Algorithm (AIA) and Ant Colony Optimization (ACO). *Applied Soft Computing* 21:542–53. doi:10.1016/j.asoc.2014.03.011.

Shelokar, P. S., P. Siarry, V. K. Jayaraman, and B. D. Kulkarni. 2007. Particle swarm and colony algorithms hybridized for improved continuous optimization. *Applied Mathematics and Computation* 188:129–42. doi:10.1016/j.amc.2006.09.098.

Shinozaki, T., and M. Ostendorf. 2008. Cross-validation and aggregated EM training for robust parameter estimation. *Computer Speech and Language* 22 (2):185–95. doi:10.1016/j.csl.2007.07.005.

Song, H. J., C. Y. Miao, R. Wuyts, Z. Q. Shen, M. D'Hondt, and F. Catthoor. 2011. An extension to fuzzy cognitive maps for classification and prediction. *IEEE Transactions On Fuzzy Systems* 19 (1):116–35. doi:10.1109/TFUZZ.2010.2087383.

Song, W., W. Ma, and Y. Qiao. 2017. Particle swarm optimization algorithm with environmental factors for clustering analysis. *Soft Computing* 21 (2):283–93. doi:10.1007/s00500-014-1458-7.

Su, S.-F., -C.-C. Chuang, C. W. Tao, J.-T. Jeng, and -C.-C. Hsiao. 2012. Radial basis function networks with linear interval regression weights for symbolic interval data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42 (1):69–80. doi:10.1109/TSMCB.2011.2161468.

Sulaiman, M. H., Z. Mustaffa, M. M. Saari, and H. Daniyal. 2020. Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence* 87:103330–42. doi:10.1016/j.engappai.2019.103330.

Tabakhi, S., P. Moradi, and F. A. Tab. 2014. An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence* 32: 112–23.

Taguchi, G., S. Chowdhury, and Y. Wu. 2005. *Taguchi's quality engineering Handbook., Hoboken*. NJ, USA: Wiley.

Taguchi, G., R. Jugulum, and S. Taguchi. 2004. *Computer-based robust engineering: Essentials for DFSS*. Milwaukee, WI, US: ASQ Quality Press.

Talbi, E. G. 2009. *Metaheuristics: From design to implementation., Hoboken, NJ*. Wiley: USA.

Tian, Z. 2020. Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM. *Engineering Applications of Artificial Intelligence* 91:103573–96. doi:10.1016/j.engappai.2020.103573.

Truong, V.-H., and S.-E. Kim. 2018. Reliability-based design optimization of nonlinear inelastic trusses using improved differential evolution algorithm. *Advances in Engineering Software* 121:59–74. doi:10.1016/j.advengsoft.2018.03.006.

Wang, B., M. Yu, X. Zhu, L. Zhu, and Z. Jiang. 2019. A robust decoupling control method based on artificial bee colony-multiple least squares support vector machine inversion for marine alkaline protease MP fermentation process. *IEEE Access* 7:32206–16. doi:10.1109/ACCESS.2019.2903542.

Wang, L., T. Wang, J. Wu, and G. Chen. 2017a. Multi-objective differential evolution optimization based on uniform decomposition for wind turbine blade design. *Energy* 120:346–61. doi:10.1016/j.energy.2016.11.087.

Wang, S., C. Yu, D. Shi, and X. Sun. 2018. Research on speed optimization strategy of hybrid electric vehicle queue based on particle swarm optimization. *Mathematical Problems in Engineering 2018*: 1–14.

Wang, W., S. Yuan, J. Pei, and J. Zhang. 2017b. Optimization of the diffuser in a centrifugal pump by combining response surface method with multi-island genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 231 (2):191–201. doi:10.1177/0954408915586310.

Wen, F., X. Yang, X. Gong, and K. K. Lai. 2017. Multi-scale volatility feature analysis and prediction of gold price. *International Journal of Information Technology and Decision Making* 16 (1):205–23. doi:10.1142/S0219622016500504.

Whitehead, B. A., and T. D. Choate. 1996. Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks* 7 (4):869–80. doi:10.1109/72.508930.

Xiao, Y., S. Wang, M. Xiao, J. Xiao, and Y. Hu. 2017. The analysis for the cargo volume with hybrid discrete wavelet modeling. *International Journal of Information Technology and Decision Making* 16 (3):851–63. doi:10.1142/S0219622015500285.

Xiaowei, H., Z. Xiaobo, Z. Jiewen, S. Jiyong, Z. Xiaolei, and M. Holmes. 2014. Measurement of total anthocyanins content in flowering tea using near infrared spectroscopy combined with ant colony optimization models. *Food Chemistry* 164:536–43. doi:10.1016/j.foodchem.2014.05.072.

Xu, B., X. Chen, and L. Tao. 2018. Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization. *Information Sciences* 435:240–62. doi:10.1016/j.ins.2018.01.014.

Yan, L., H. Wang, X. Zhang, M.-Y. Li, J. He, and S. B. Jadhao. 2017. Impact of meteorological factors on the incidence of bacillary dysentery in Beijing, China: A time series analysis (1970-2012). *PLoS One* 12 (8):1–13. doi:10.1371/journal.pone.0182937.

Yan, X., P. Li, K. Tang, L. Gao, and L. Wang. 2020. Clonal selection based intelligent parameter inversion algorithm for prestack seismic data. *Information Sciences* 517:86–99. doi:10.1016/j.ins.2019.12.083

Yang, Z., K. Li, Y. Guo, H. Ma, and M. Zheng. 2018. Compact real-valued teaching-learning based optimization with the applications to neural network training. *Knowledge-Based Systems* 159:51–62. doi:10.1016/j.knosys.2018.06.004.

Yang, Z., M. Mourshed, K. Liu, X. Xu, and S. Feng. 2020. A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting. *Neurocomputing* 397:415–21. doi:10.1016/j.neucom.2019.09.110.

Yin, X., Z. Niu, Z. He, Z. S. Li, and D. Lee. 2020. An integrated computational intelligence technique based operating parameters optimization scheme for quality improvement oriented process-manufacturing system. *Computers and Industrial Engineering* 140:106284–98. doi:10.1016/j.cie.2020.106284.

Zhang, F., and Z. Liao. 2013. Gold price forecasting based on RBF neural network and hybrid fuzzy clustering algorithm. *Proceedings of the Seventh International Conference on Management Science and Engineering Management*. Philadelphia, US. Springer, 73–84.

Zhang, H., Q. Zhang, L. Ma, Z. Zhang, and Y. Liu. 2019. A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Information Sciences* 490:166–90. doi:10.1016/j.ins.2019.03.070.

Zhang, M., N. Tian, V. Palade, Z. Ji, and Y. Wang. 2018. Cellular artificial bee colony algorithm with gaussian distribution. *Information Sciences* 462:374–401. doi:10.1016/j.ins.2018.06.032.

Zhao, H., C. Zhang, and B. Zhang. 2020. A decomposition-based many-objective ant colony optimization algorithm with adaptive reference points. *Information Sciences* 540:435–48. doi:10.1016/j.ins.2020.06.028.

Zhao, W., L. Yan, and Y. Zhang. 2018a. Geometric-constrained multi-view image matching method based on semi-global optimization. *Geospatial Information Science* 21:115–26.

Zhao, Y., R. Liu, X. Zhang, and A. Whiteing. 2018b. A chance-constrained stochastic approach to intermodal container routing problems. PLoS One 13 (2):1–22

Zhou, H., H. Zhang, and C. Yang. 2020. Hybrid-model-based intelligent optimization of ironmaking process. *IEEE Transactions on Industrial Electronics* 67 (3): 2469–79.

Zhou, Q., Y. Rong, X. Y. Shao, P. Jiang, Z. M. Gao, and L. C. Cao. 2018. Optimization of laser brazing onto galvanized steel based on ensemble of metamodels. *Journal of Intelligent Manufacturing* 29 (7):1417–31. doi:10.1007/s10845-015-1187-5.

Zhu, Z., L. Chen, C. Yuan, and C. Xia. 2018. Global replacement-based differential evolution with neighbor-based memory for dynamic optimization. *Applied Intelligence* 48 (10):1–15. doi:10.1007/s10489-018-1147-9.

Zou, H. F., G. P. Xia, F. T. Yang, and H. Y. Wang. 2007. An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting. *Neurocomputing* 70 (16–18):2913–23. doi:10.1016/j.neucom.2007.01.009.