Engineering Applications of Artificial Intelligence 46 (2015) 258–268

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model

X.Y. Chen, K.W. Chau *, A.O. Busari

Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 10 February 2015
Received in revised form
18 September 2015
Accepted 18 September 2015
Available online 23 October 2015

Keywords:
Differential evolution
Artificial bee colony
Ant colony optimization
Downstream river flow forecasting
Hybrid neural network

ABSTRACT

Population-based optimization algorithms have been successfully applied to hydrological forecasting recently owing to their powerful ability of global optimization. This paper investigates three algorithms, i.e. differential evolution (DE), artificial bee colony (ABC) and ant colony optimization (ACO), to determine the optimal one for forecasting downstream river flow. A hybrid neural network (HNN) model, which incorporates fuzzy pattern-recognition and a continuity equation into the artificial neural network, is proposed to forecast downstream river flow based on upstream river flows and areal precipitation. The optimization algorithm is employed to determine the premise parameters of the HNN model. Daily data from the Altamaha River basin of Georgia is applied in the forecasting analysis. Discussions on the forecasting performances, convergence speed and stability of various algorithms are presented. For completeness' sake, particle swarm optimization (PSO) is included as a benchmark case for the comparison of forecasting performances. Results show that the DE algorithm attains the best performance in generalization and forecasting. The forecasting accuracy of the DE algorithm is comparable to that of the PSO, and yet presents weak superiority over the ABC and ACO. The Diebold–Mariano (DM) test indicates that each pair of algorithms has no difference under the null hypothesis of equal forecasting accuracy. The DE and ACO algorithms are both favorable for searching parameters of the HNN model, including the recession coefficient and initial storage. Further analysis reveals the drawback of slow convergence and time-consumption of the ABC algorithm. The three algorithms present stability and reliability with respect to their control parameters on the whole. It can be concluded that the DE and ACO algorithms are considerably more adaptive in optimizing the forecasting problem for the HNN model.

1. Introduction

River flow forecasting is a prerequisite for many water resource applications such as flood warning and reservoir design. The hydrological process of river flows is so complex that simple data-driven models are not able to describe its behavior. It is therefore worth investigating suitable models for the highly-nonlinear and seasonal river flows. The reasoning ability of fuzzy-based neural networks has led to an increasing interest within the hydrology community. So far a number of neuro-fuzzy computing techniques and fuzzy set theories have already been applied in hydrological modeling (Li and Chen, 2010; Tabari et al., 2012; Rath et al., 2013). Incorporating a fuzzy concept activation function into an artificial neural network (ANN) model is a good alternative to the latter since the former can be flexibly implemented. Qiu et al. (1998) introduced a fuzzy pattern-recognition activation function (from the input layer to the hidden layer) into an ANN model to forecast annual runoff. This function classified runoff into a number of categories during wet and dry seasons, and therefore reflected nonlinear and seasonal characteristics of the river system. Zhao and Chen (2008) developed a mixed forecasting model combining neural networks with fuzzy pattern-recognition, considering the fuzziness in the concept of similar basins. This practice, in which activation functions in the ANN structure are modified with characterization, is more easily executed and also addresses the fuzzy behavior of river flows. In this paper, this type of fuzzy pattern-recognition activation function is introduced into a hybrid neural network (HNN) model, representing uncertainties of the river flow problem.

* Corresponding author.
E-mail address: cekwchau@polyu.edu.hk (K.W. Chau).

http://dx.doi.org/10.1016/j.engappai.2015.09.010
0952-1976/

As an important issue in hydrological modeling, conceptual models have been developed for river flow forecasting over the past few decades. Rainfall-runoff models have extensively emerged in river flow forecasting by considering simplified forms of physical laws (Moore, 2007; Bhadra et al., 2010; Rezaeianzadeh et al., 2013). Another practical approach is to directly integrate physical equations into the ANN models. Yang et al. (1998) developed a hydrological modeling network (HYMN) model using the continuity equation and considering nodes in the hidden or output layers as storage reservoirs. This method has been successful applied to compute the monthly river flow at the Salford University station in the Irwell River basin using daily evaporation data and precipitation records from six upstream stations. Li and Gu (2003) further expanded the HYMN model to sediment yield forecasting. These investigations are rather limited, whereas they attain the possibility of directly integrating fundamental physical principles into an ANN structure. This paper contributes to assembling the fuzzy pattern-recognition concept and continuity equation into the ANN framework, with an explicit emphasis on the seasonal and non-stationary features of the river flows.

The proposed forecasting model is highly nonlinear and varying with time. Traditional optimization algorithms, e.g. the Gauss–Newton algorithm and gradient descent method are not applicable for this non-differentiable and multi-dimensional problem. Recently, population-based optimization algorithms have attracted the interest of many researchers (Blum and Roli, 2003; Chiong et al., 2012). They are characterized by a population consisting of possible solutions to the problem, which are modified by applying different types of operators and thus moving towards a near-optimal solution area. These algorithms are very powerful in finding a global optimum since they simultaneously search in many directions by using a population of possible solutions. Generally, there are two categories of population-based optimization algorithms: evolutionary algorithms and swarm intelligence algorithms (Blum et al., 2012). Typical methods of evolutionary algorithms are the genetic algorithm (GA) and differential evolution (DE), which attempt to simulate natural evolution. The DE algorithm was proposed by Storn and Price (1995) and has been applied extensively in hydrological modeling (e.g., Babu and Angira, 2003; Vasan and Simonovic, 2010; Liu and Pender, 2013). It shows better performances than the GA in terms of convergence characteristics and computational efficiency (Wang et al., 2008; Li et al., 2013; Song et al., 2014). In view of its ability to handle optimization problems that are non-differentiable, nonlinear, non-continuous and varying with time (Rocca et al., 2011), it is adopted in this paper for the comparative study.

The second category, swarm intelligence-based algorithms are inspired by the collective behavior of animal societies, including particle swarm optimization (PSO), artificial bee colony (ABC) and ant colony optimization (ACO). This paper focuses on ABC and ACO as stochastic global optimization algorithms. The ABC algorithm was introduced and popularized by Karaboga (2005) to solve numerical optimization problems. It has predictive capability comparable to the GA, PSO and DE algorithms on numerical test functions (Karaboga and Akay, 2009). Hybrid models that combine ABC algorithms with ANNs have been developed recently (Karaboga et al., 2007; Kisi et al., 2012). The ABC algorithm has been found to apply in many fields, e.g. HVAC systems (Zhang et al., 2013), reservoir optimum problems (Hossain and El-shafie, 2013) and protein structure optimization (Li et al., 2015). Another swarm-based optimization method ACO was derived from the food searching behavior of ants (Dorigo et al., 1996). Similar to the ABC algorithm, it is a meta-heuristic technique available to solve non-linear optimization problems with high dimensionality and inequality constraints. Coupling ACO algorithms with feed-forward neural network training has proven to be successful (Li and Chung, 2005; Shelokar et al., 2007; Socha and Blum, 2007). The potential to apply ACO to the field of river flow forecasting is clear, e.g., see its application in water resource problems (Maier et al., 2003; Jalali et al., 2006; Kumar and Reddy, 2006).

Optimization of neural networks has always been an open research. It is imperative to solve the disadvantages of traditional learning algorithms, such as poor generalization, slow convergence speed and easily plunging into local optima. The main objective of this paper is therefore to incorporate population-based optimization algorithms (i.e. DE, ABC and ACO) into a HNN model and compare their optimization ability, stability and reliability, and thereby determine the most adaptive optimization algorithm for the river flow forecasting problem.

The rest of this paper is structured in the following manner. A description of the HNN model for downstream river flow forecasting is presented in Section 2. In Section 3, a brief review of the three population-based optimization algorithms is provided. Section 4 introduces the case study site, while Section 5 summaries the computational results and comparisons of different optimization algorithms. Finally, the conclusion is reported in Section 6.

2. Hybrid neural network (HNN) model

The ANN is an efficient data-driven model for real-time forecasting. For a typical three-layer feed-forward ANN, the nodes in the input layer (input data introduced to the network) are linked with a predetermined activation function to those in the hidden layer, and then to the nodes in the output layer with a similar operation. An objective function with premise parameters (to be adjusted) is defined by comparing the difference between computed and target outputs. The fitness value of the objective function as well as associated parameters has to be adapted in the calibration process using optimization techniques. The optimal parameters are, therefore, determined corresponding to the most approximate computed outputs. Usually tan-sigmoid and linear functions are adopted as the activation functions to capture the relation of nodes between two layers. Nonetheless, they have no physical meanings and render the ANN a real 'black-box' model, and therefore, are unsuitable for the forecasting of river flows, due to the nonlinearity, non-stationary and seasonal behavior of river flows. It is essential to describe the hydrological processes through an adequate nonlinear and fuzzy model.

In the HNN model, the framework of the traditional ANN with three layers is maintained and, yet, activation functions with special significance are introduced. A conceptual function with fuzzy pattern-recognition idea from the input layer to the hidden layer is defined as follows

$$Q_i = \frac{1}{\sum_{l=1}^{C} \frac{\sum_{j=1}^{k} \left[w_{ji} \left(Q_j^{in} - M_i \right) \right]^2}{\sum_{j=1}^{k} \left[w_{ji} \left(Q_j^{in} - M_l \right) \right]^2}} \tag{1}$$

wherein Q_i ($i=1, 2, \ldots, s$) are nodes in the hidden layer and Q_j^{in} ($j=1, 2, \ldots, k$) denote nodes in the input layer. The parameter w_{ji} stands for weight parameter from the input layer to the hidden layer. A model vector is defined as $M=[M_i]=[M_l]$ in the hidden layer. The corresponding parameter C refers to the number of elements in the model vector. Note that C is the number of nodes in the hidden layer as well (i.e. $C=s$). Based on this activation function, the nodes in the hidden layers are elaborated by classifying the inputs into a number of categories. A higher value of C implies that there will be more categories in the hidden layer, which therefore represents a higher degree of nonlinearity of the model. In this paper, $C=11$ and $M=[1.0, 0.9, \ldots, 0.1, 0]$ are adopted

to obtain fuzzified nodes in the hidden layer. The largest membership value 1.0 represents the "wet" model in wet season and the smallest value 0 represents the "dry" model in dry season, respectively. Thus seasonal effects and fuzzy features are accentuated using the membership function.

The ensuing task is to compute the nodes in the output layer which are represented by storage reservoirs, that receive flows from upstream river reaches. The process is akin to that the nodes in the output layer acquire their values from nodes in the hidden layer. By using the continuity equation, the downstream river flows is computed as follows

$$\frac{\partial S_h}{\partial T} = \sum_{i=1}^{s} \overline{w}_{ih} Q_i - Q_h \tag{2}$$

where S corresponds to water storage, Q is water discharge and T is time. Meanwhile, i (1, 2, …, s) refer to the reservoirs in the hidden layer and h (1, 2, …, r) refer to the reservoirs in the output layer. The fraction of water from a reservoir in the hidden layer entering into a reservoir in the output layer is denoted by $\overline{w}_{ih}$. The continuity equation is employed to model the process of upstream water flowing towards downstream sections.

By discretizing Eq. (2), the water storage S in the output layer at time $T+\Delta T$ is expressed in the following equation:

$$S_{h(T+\Delta T)} = S_{h(T)} + \left(\sum_{i=1}^{s} \overline{w}_{ih} Q_{i(T)} - Q_{h(T)} \right) \times \Delta T \tag{3}$$

The parameter ΔT is the time step, representing the time of flows from the hidden layer to the output layer. After having introduced the following two parameters

$$P_{h(T)} = \sum_{i=1}^{s} \overline{w}_{ih} Q_{i(T)} \times \Delta T \tag{4}$$

$$\lambda_{h(T)} = 1 - \frac{Q_{h(T)} \times \Delta T}{S_{h(T)} + P_{h(T)}} \tag{5}$$

Eq. (3) is simplified as follows

$$S_{h(T+\Delta T)} = \lambda_{h(T)} \times (S_{h(T)} + P_{h(T)}) \tag{6}$$

where λ_h is regarded as a recession coefficient and assumed to be independent of time (Yang et al., 1998). The computation of water storage is facilitated by a recession coefficient, which represents the storage capability of the reservoir. The initial water storage $S_{h\ (T=0)}$ of each reservoir is prescribed before the computation, and the storage at any given time can be computed from the initial one in every time step. Thus, the storage becomes viable once the initial storage and time step are determined. Considering the nonlinear relation between the discharge and the storage of the reservoirs, the discharge in the output layer is evaluated by the following equation:

$$Q^{out}_{h(T+\Delta T)} = \frac{1}{1+\exp\left[-(S_{h(T+\Delta T)} + P_{h(T+\Delta T)})\right]} \tag{7}$$

This nonlinear function is represented by an empirical expression (Yang et al., 1998), revealing the nonlinear feature of the reservoir. It is noted that the water storage is a time-varying parameter, indicating that the current discharge would be affected by the previous storage. This incorporates the physical phenomenon of flow in a river basin in which larger storage in the wet season will result in higher discharge. Assuming flows from the hidden layer reach reservoirs in the output layer with time step ΔT, the storage of the reservoirs is computed from Eq. (6) and the eventual discharge becomes the output.

The implementation of the HNN model is accomplished by integrating the fuzzy pattern-recognition and continuity equation into the neural network. The superiority to traditional ANNs is attributed to the consideration of the fuzzy behavior of river flows as well as the storage capability of observed river stations. To conduct the forecasting, three kinds of parameters are necessarily determined: weight parameters w_{ji} and $\overline{w}_{ih}$; recession coefficients λ_h and initial storages $S_{h(T=0)}$ for the reservoirs in the output layer. In this study, there is only one desired output, that is, the reservoir in the output layer ($h=1$). The corresponding set of optimization parameters is $\{w_{11}, \ldots, w_{ks}, \overline{w}_{11}, \ldots, \overline{w}_{s1}, \lambda, S_{(T=0)}\}$. The training of the HNN model is characterized as a process of searching the optimal combination of parameters in their solution spaces.

3. Population-based optimization algorithms

3.1. Differential evolution algorithm

The DE algorithm conducts mutation, crossover and selection operations based on the differences of randomly sampled pairs of solutions in the population. Mutation operation acts as a search mechanism while the crossover operator recombines the parent vector with the mutated one. Afterwards, all solutions have an equal chance of being selected as parent. The strategies are described as follows.

Firstly, define a population of D dimensional parameter vectors, with NP population size. Each individual (target vector) for generation G is then represented as $X_{i,G}$ ($i=1, 2, \ldots, NP$) and $X_{i,G} = \{x_{i1,G}, x_{i2,G}, \ldots, x_{iD,G}\}$ constitutes all parameters to be optimized. Donor vector in the next generation $V_{i,G+1}$ is generated from three randomly selected vectors $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ as follows

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \tag{8}$$

in which F acts as a mutation factor which is a random number uniformly distributed within the range [0, 2]. The randomly selected indexes $r1$, $r2$, $r3 \in \{1, 2, \ldots, NP\}$ must be different from one another and from the running index i as well.

The crossover operation defines a trial vector $U_{i,G+1} = \{u_{i1,\ G+1}, u_{i2,G+1}, \ldots, u_{iD,G+1}\}$ as follows

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{if } rand_{ij} \le CR \text{ or } j = I_{rand} \\ x_{ij,G} & \text{if } rand_{ij} > CR \text{ and } j \ne I_{rand} \end{cases} \tag{9}$$

where $j=1, 2, \ldots, D$; CR is a crossover constant in the range of [0, 1]; $rand_{ij}$ is a random number within 0 and 1; and I_{rand} is a random index from [1, 2, …, D], ensuring that at least one element in the trial vector is obtained from $V_{i,G}$. The change of the diversity of the population is controlled by the CR value.

The selection operation chooses the vector in the next generation by the following equation for a minimizing problem

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \le f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{10}$$

in which the objective function associated with $X_{i,G}$ is denoted as $f(X_{i,G})$. In other words, when comparing the target vector with the trial vector, a lower function yielding value is admitted to the succeeding generation. In general, DE algorithm is a comparatively simple algorithm because only three control parameters (i.e. NP, F and CR) are required to be prescribed.

3.2. Artificial bee colony

The ABC algorithm draws its inspiration from the foraging behavior of honey bee swarms. A possible solution of the optimization problem is viewed as a food source for the artificial bees. In the foraging process, bees interact with each other aiming to maximize the nectar amount of the food source. The distribution

and duties of grouped employed, onlooker and scout bees have been described in detail (Karaboga and Akay, 2009).

In this scheme, a population of food source position $\{X_1, X_2, \ldots, X_{NP}\}$ is initialized randomly. NP denotes the number of population, and refers to the number of food sources as well. Each food source $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ is a D-dimensional vector, containing D variables for the optimal problem. During the employed bees' phase, a neighbor solution v_{ij} is generated from an original one x_{ij} by the following equation

$$v_{ij} = x_{ij} + rand(-1, 1) \times (x_{ij} - x_{kj}) \tag{11}$$

where the subscript $j \in [1, D]$ are randomly chosen indexes and $k \in [1, NP]$ is a random neighbor index which should be different from i. After evaluating the new neighbor and original solution by the fitness value of the optimal problem, the better solution will be kept in the population. Employed bees will then transmit the information concerned with the source to the onlooker bees. The onlooker bees employ a roulette-wheel like selection depending on a probability value as follows

$$p_i = \frac{f(X_i)}{\sum_{k=1}^{NP} f(X_k)} \tag{12}$$

where $f(X_i)$ is the fitness value of the solution X_i. The probability value p_i is proportional to the fitness value for a selected solution. If p_i is larger than a random number drawn for each solution within the range [0, 1], the onlooker bees will make a local search as in Eq. (11) to find a neighbor solution. During the employed and onlooker bees' phases, when the nectar of a food source being exploited is exhausted, the source will be abandoned. That is, if the original solution is kept and not replaced by a new neighbor one for exceeding a prescribed number of cycles, namely *Limit*, it is postulated to be an exhausted one and has to be replaced by a new random solution. The scout bees determine a new food source by the following expression:

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

in which, $x_j^{\min}$ and $x_j^{\max}$ are the minimum and maximum value of the corresponding solution, respectively. Similarly, the new solution will be evaluated and compared with the existing one. The solution with a "rich source" (minimum fitness value) will be selected as the optimal one. There are two control parameters in an ABC algorithm: *NP* and *Limit*.

3.3. Ant colony optimization

The ACO algorithm simulates ants' behavior in which their fundamental objectives are to find the shortest path between food source and their nest. There is a hypothetical chemical substance named pheromone laid by other ants in their trails, which works as a communication mechanism. The ants will choose paths independently according to the pheromone intensity. For a specific solution component, if its pheromone value is higher, it is more likely to be selected. The pheromone value will be reinforced if the corresponding solution component belongs to the best solution. The following discussion outlines the basic concept of the ACO algorithm.

Table 1
Pheromone table for each parameter in ACO algorithm.

$x_j (j=1, 2, \ldots, D)$				
Tag	1	2	…	N
Solution component	a_{j1}	a_{j2}	…	a_{jN}
Pheromone intensity	τ_{j1}	τ_{j2}	…	τ_{jN}

Given that there are D parameters to be optimized in the search space. Each parameter's definition space is split into a set of discrete points. As shown in Table 1, for parameter x_j ($j=1, 2, \ldots, D$) within a range of $[a_{j1}, a_{jN}]$, there are totally N discrete points when the feasible range is uniformly divided into $N-1$ shares. Each point corresponds to a candidate value of the parameter, namely solution component a_{jk} ($k=1, 2, \ldots, N$). Suppose an ant i ($i=1, 2, \ldots, NP$) is on its path to search D parameters, it can only choose a value for each parameter among the candidate points and record the corresponding tag. In the meantime, the pheromone intensity is needed for each candidate point, represented by τ_{jk} for tag k. When an ant reaches the parameter x_j, each solution component has the following probability to be selected

$$p_{jk} = \frac{\tau_{jk}}{\sum_{1 \le m \le N} \tau_{jm}} \tag{14}$$

With probability value larger than a random value within the range [0, 1], the corresponding point a_{jk} will be selected for this parameter. After the ant finishes its tour and all parameters are selected as $X_i = \{x_1, x_2, \ldots, x_D\}$, it will return to its nest and update the pheromone intensity according to the following equations:

$$\tau_{jk}(i+1) = \rho \tau_{jk}(i) + \Delta \tau_{jk} \tag{15}$$

$$\Delta \tau_{jk} = \begin{cases} Q/f(X_i) & \text{if } a_{jk} \text{ is selected as } x_j \text{ and belongs to } X_i \\ 0 & \text{else} \end{cases} \tag{16}$$

The first term in the right hand side of Eq. (15) reflects an evaporation process in which $\rho \in [0, 1]$ is the coefficient of pheromone duration. The second term indicates the reinforcement in which $\Delta \tau_{jk}$ is the amount of pheromone retained in the solution component being a part of X_i. The reinforcement value $\Delta \tau_{jk} = 0$ if component a_{jk} is not within the best solution. In particular, the pheromone constant Q in Eq. (16) is an user-defined parameter which is the same for all ants, and $f(X_i)$ is the fitness value associated with X_i. The parameters of the ACO algorithm to be prescribed are respectively *NP*, *N*, ρ and *Q*.

This paper employs population-based algorithm to deal with the non-differentiable, combinatorial and multi-dimensional optimization problem for the HNN model. Given that the number of parameters is D ($D = k \times s + s \times 1 + 1 + 1$), i.e. the sum of the number of w_{ji}, $\overline{w}_{ih}$, λ and $S_{(T=0)}$, this set of parameters is represented by a vector X_i. As outlined in Fig. 1, the steps for optimizing parameters, when taking the DE algorithm as an example, are as follows: (1) randomly generate a population $\{X_1, X_2, \ldots, X_{NP}\}$ within their interval $[x_j^{\min}, x_j^{\max}]$; (2) define an objective function of the HNN model and set the running generation $G=1$; (3) train the HNN model with current updated parameters and obtain the corresponding fitness value of the objective function; (4) apply the mutation and crossover processes to construct donor and trial vectors, and finally select one vector with a minimum fitness value from the trial and target vectors; (5) reset $G=G+1$; (6) if $G > MI$ (max iteration), the stopping criterion is satisfied and output the optimal set of parameters; or go to step (3) and (4) for the next generation. In particular, for step (4), the configuration of the ABC and ACO algorithms is demonstrated in Fig. 1 as well. The ABC algorithm undergoes the employed, onlooker and scout bees' phases orderly. As for the case of the ACO algorithm, firstly set $i=1$ and initialize equal pheromone intensity; release ant X_i, and let it select a candidate value for all D parameters; update the pheromone intensity and reset $i=i+1$; if $i <= NP$, release the succeeding ant and begin its tour, otherwise all ants have finished the search. All these three techniques show an easy realization in the process of optimizing parameters of the HNN model.

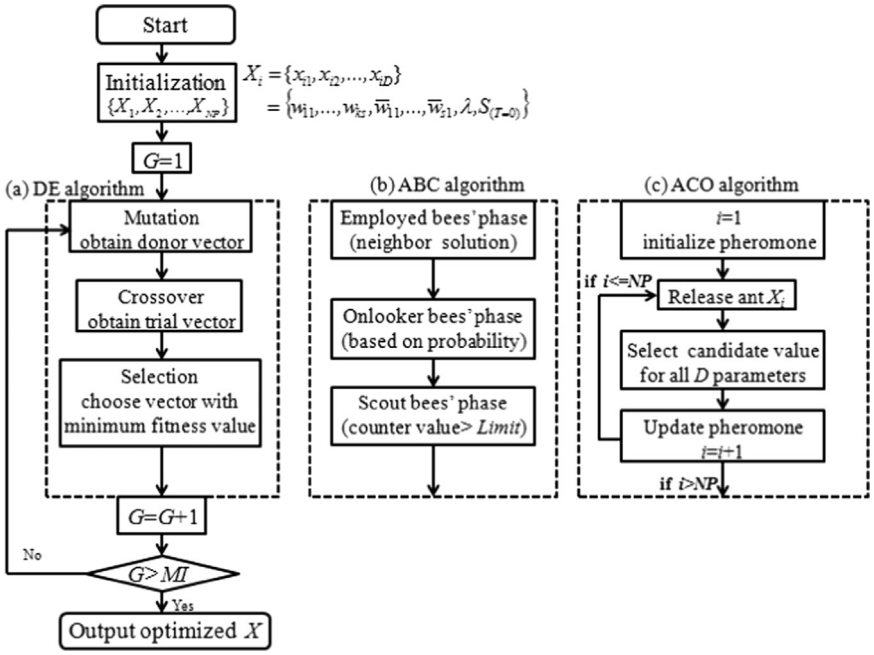

Fig. 1. Framework for optimizing parameters by (a) DE (b) ABC (c) ACO algorithm.

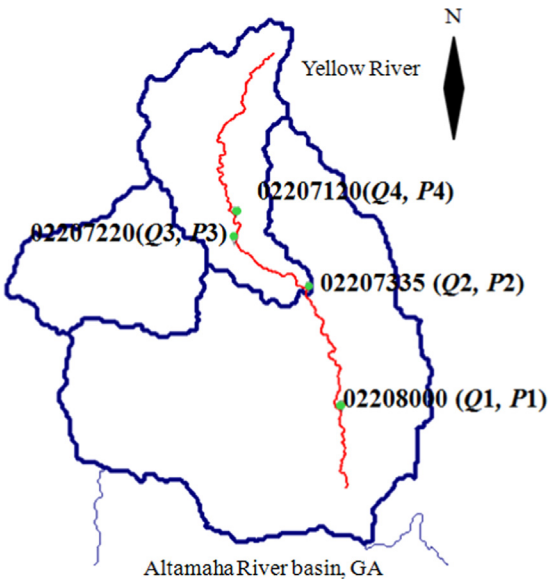

Fig. 2. Locations of four observed river stations along Yellow River in Altamaha River basin.

4. Study case and data preparation

The study area for the present study is the Altamaha River basin, situated on the Atlantic coast in the U.S. state of Georgia. The Yellow River flows southward in the basin, as a tributary of the Ocmulgee River. The locations of four river stations along the Yellow River in the watershed are shown in Fig. 2. Daily river discharge and precipitation data is taken from these four hydrological stations. The simplified letters Q and P refer to discharge and precipitation, respectively. In addition, Table 2 provides the identity (ID), name, location and elevation of each station.

The input variable selection is critical to the performance of neural networks (Hu et al., 2015a, 2015b; Xiong et al., 2015). For a river system, the simulation of target outflows with appropriate upstream inputs is desirable to avoid flooding on the downstream site during the rainy season. Apart from upstream river flows, areal precipitation over a river basin has been pronounced as a powerful influence on river flows as well (Jiang et al., 2007; Jena et al., 2014; Rezaeianzadeh et al., 2014). The objective of this study is to forecast daily discharge in station 02208000, i.e. $Q1$, based on discharges from three upstream stations ($Q2$, $Q3$ and $Q4$) and the areal precipitation AP. The time step ΔT is recognized as 1 day since the travel time of flow from station 02207335 to 02208000 is roughly 16 h. Therefore, the HNN model provides a one-day-ahead forecast for the flow in station 02208000. It is a one-step-head forecasting for this river basin. Since multi-step-ahead forecasting is important as well for this comparative study (Taieb et al., 2012; Xiong et al., 2013; Bao et al., 2014b, 2014a), we will compare the optimization algorithms from the perspective of multi-step-ahead forecasting in another study case in the future.

The areal precipitation over the river basin is computed by the Thiessen polygon method, details of which can be found in Sen (1998). In this study case, the weight factors taken up by each polygon of the basin are 0.218, 0.296, 0.141 and 0.345, respectively, for the four observed stations from 02207120 to 02208000. Thus areal precipitation is computed as a weighted precipitation of these four stations. As a smoothly averaged precipitation over the river basin, areal precipitation is adopted as an effective input variable to compute downstream river flow.

Daily dataset of four years spanned from 2010 to 2013 is collected as an example (from the USGS website http://waterdata.usgs.gov/ga/nwis/rt). It is important to note that the discharge and precipitation time series are characterized by high nonlinearity, non-stationary and seasonal behavior. In order to examine the generalization and forecasting ability of the model, the data are separated into training, validation and testing sets. The data for the training period are from 1st January 2010 to 31st December 2011, taking roughly 50% of all data. Data in the year 2012 are set aside for validation to ensure that the network does not over-fit the training data. It is carried out by strictly terminating the training at the point where the error in the validation set begins to rise. The remaining data are utilized to gauge the forecasting performance in the testing period.

Table 2
Stations' ID, name, location and elevation along Yellow River.

Station ID	Station name	Latitude	Longitude	Elevation (above NGVD29)
02207120	Yellow River at GA 124, near Lithonia	33°46′22″N	84°03′30″W	222.504 m
02207220	Yellow River at Pleasant Hill road, near Lithonia	33°44′01″N	84°03′43″W	213.735 m
02207335	Yellow River at Gees Mill road	33°40′01″N	83°56′17″W	192.368 m
02208000	Yellow River at Rocky Plains road	33°29′59.5″N	83°53′03″W	164.592 m

Table 3
Training and testing performances of various algorithms in terms of RMSE, NSCE and ACC.

Algorithm	Training			Testing		
	RMSE(m^3/s)	NSEC	ACC	RMSE(m^3/s)	NSEC	ACC
DE	**2.5534**	**0.9812**	0.7627	**8.1500**	**0.8294**	0.7937
ABC	3.0827	0.9727	0.7279	9.0319	0.7905	0.7905
ACO	2.8174	0.9772	**0.7658**	8.8573	0.7985	0.7853
PSO	2.8306	0.9769	0.7139	8.1502	**0.8294**	**0.8150**

5. Results and discussion

A complete assessment of model performances is performed employing three evaluation measures. These include: the root mean square error (RMSE), the Nash–Sutcliffe efficiency coefficient (NSEC) and the accuracy (ACC). They are formulated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Q_i - \hat{Q}_i)^2} \tag{17}$$

$$NSEC = 1 - \sum_{i=1}^{N}(Q_i - \hat{Q}_i)^2 / \sum_{i=1}^{N}(Q_i - \overline{Q})^2 \tag{18}$$

$$\mathrm{ACC} = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{\left|Q_i - \hat{Q}_i\right|}{Q_i} \tag{19}$$

in which Q_i and $\hat{Q}_i$ are respectively observed and computed discharges, $\overline{Q}$ is the averaged observed data, and N is the number of observations. The RMSE is an absolute error measure and a lower value indicates better performance while NSEC and ACC values of 1 suggest perfect fits.

In the following, the generalization and forecasting performances of the HNN model are investigated by contrasting different optimization algorithms. Section 5.2 then gives a detailed analysis on the convergence speed and the ability of parameter optimization. Finally, the stability of each algorithm is discussed since the control parameters are of great concern. All three algorithms are implemented in Matlab on an x64-based PC processor.

5.1. Comparison of forecasting performances

For completeness' sake, the PSO algorithm is included as a benchmark case for the comparison of forecasting performances. Table 3 provides the training and testing performances of various algorithms in terms of RMSE, NSEC and ACC. The DE algorithm outperforms its counterparts in both training and testing period with respect to RMSE and NSEC values. The high value of NSEC (0.8294) in the testing period suggests that DE algorithm can be well combined with the HNN model for the forecasting. Although obtaining a lower ACC value than the PSO in the testing stage, the DE algorithm displays excellent performances of generalization and forecasting. It is apparent that the HNN model is fully trained to provide sufficiently accurate forecasting by the DE algorithm. The advantage of the DE over the ABC algorithm is relatively evident, as there is a 9.76% reduction in RMSE value and 4.92% increase in NSEC value in the testing period. When compared with the PSO algorithm, ABC does not show superiority in this study case. Comparison of performances between DE and ACO implies that the ACO algorithm can be recommended as an effective method as well. It has comparable performances with the PSO algorithm, particularly for the RMSE and ACC values in the training period. For the perspective of various algorithms, the HNN model superbly captures the input–output mapping when optimized with population-based optimization algorithms. To obtain the above results, parameters used are: $MI=500$, $NP=550$, $F=0.4$ and $CR=1.0$ in the DE algorithm; $MI=2000$, $NP=440$ and $Limit=214$ in the ABC algorithm; $MI=500$, $NP=780$, $N=45$, $\rho=0.3$ and $Q=1.0$ in the ACO algorithm.

The scatter plots of observed and computed discharges by these four algorithms are presented in Fig. 3. The HNN model exhibits good matches between the observed and computed data series in the testing period. The plots of intensively distributed dots along the ideal line within 0–50 m^3/s suggest that the low river flows are mostly well forecasted. This is because the frequent occurrences of low values allow a better generalization of the trained model. The performance of the ABC algorithm is not as good comparing to the DE, ACO and PSO with the evidence of five apparent dots over-forecasting with a range of 113–226 m^3/s. The result is consistent with the high RMSE value of 9.0319 m^3/s in Table 3. The improvement of performances for the DE over ACO algorithm is not obvious from the scatter plots.

The capability of capturing extreme values is critical to evaluate the performance of the HNN model. Fig. 4 presents the time series of observed and computed discharges by the DE algorithm and marks five observed extreme points. The quantitative values are provided in Table 4, along with other algorithms. Whilst the peak value (extreme point 2) is over-forecasted by all four algorithms, the DE algorithm produces relatively closer result. It performs well at extreme point 3 although the forecasted one is slightly underestimated. The ACO algorithm gives the most approximate values for point 4 and 5. The relative mean errors between the computed and observed extreme values by all four algorithms do not vary significantly. Accordingly, all four algorithms show equal performance on capturing extreme values. They have operators providing variable step size and diversity, thus can perform the perturbation of the proposed HNN model.

The mean, variance and mean with 95% confident interval by these four algorithms are provided in Table 5. This statistical analysis is necessary to check the significance of the differences. The mean value attained by the PSO algorithm is closer to the observed one as compared with the others. These four algorithms obtain higher variance values than the observations, which indicate that the computed results are widely distributed. The most approximated value to the observations is obtained by the DE algorithm (16.3556 m^3/s) for the mean with 95% confident interval, however, slight discrepancy is observed when compared with other algorithms. Thus the forecasting performances by these four optimization algorithms are comparable.

The Diebold–Mariano (DM) test is employed to assess the statistical significance of the comparison of the results. It is used to

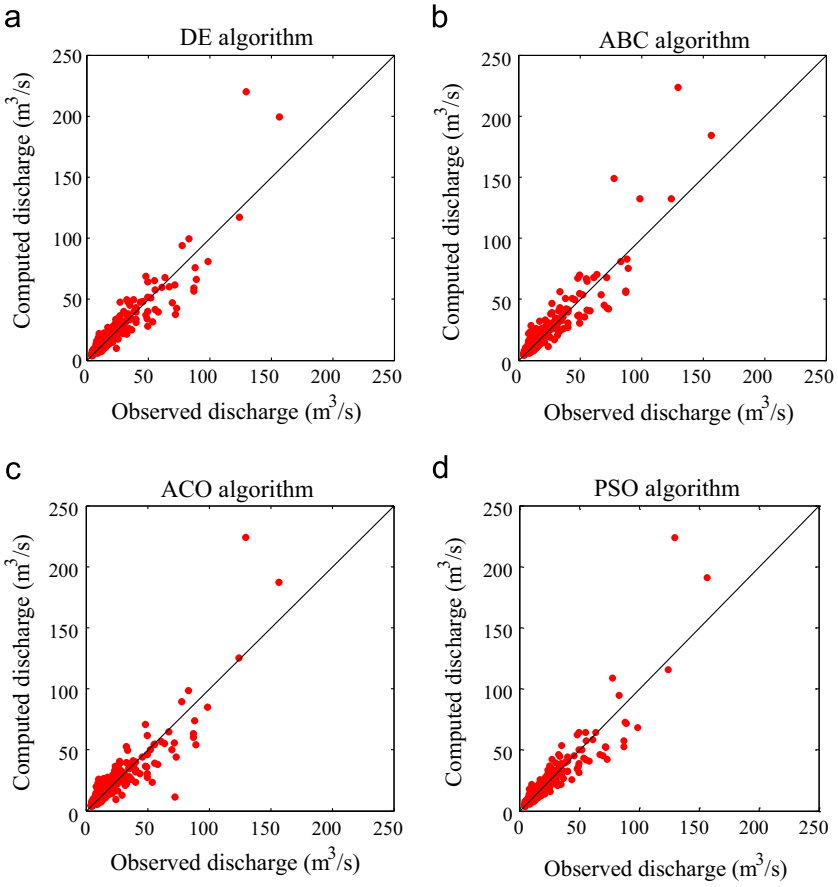

Fig. 3. Observed and computed discharges of various algorithms in the testing period.

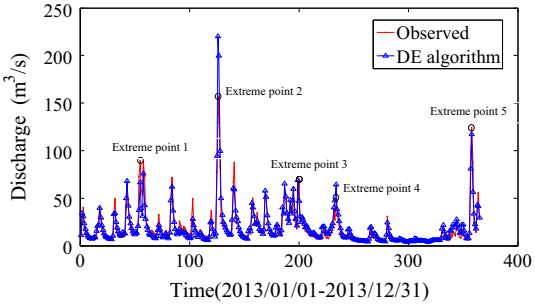

Fig. 4. Observed and computed discharges by DE algorithm and marked extreme points for observed discharges.

Table 4
Observed and computed extreme values of various algorithms.

Algorithm	Extreme point 1 (m^3/s)	Extreme point 2 (m^3/s)	Extreme point 3 (m^3/s)	Extreme point 4 (m^3/s)	Extreme point 5 (m^3/s)	Relative mean error
Observed	89.4811	156.8751	69.6593	49.8376	124.0276	
DE	66.4992	**219.9800**	**68.8210**	64.1180	117.3200	**0.2024**
ABC	**75.2180**	223.3100	67.5240	69.5910	132.1700	0.2151
ACO	54.3910	224.1600	71.0100	**61.9400**	**125.1600**	0.2184
PSO	71.8230	223.7800	62.4530	64.5090	115.9200	0.2174

compare the accuracy of two forecasts, by computing the DM statistics on the base of the loss differential defined as the difference of the squared forecast errors (Diebold and Mariano, 1995; Harvey et al., 1997; Rech, 2002). Under the null hypothesis of equal forecasting accuracy, the asymptotical distribution of DM is

Table 5
Statistical analysis of various algorithms in the testing period.

Algorithm	Mean (m^3/s)	Variance (m^6/s^2)	Mean of 95% confident interval (m^3/s)
Observed	18.6845	389.4509	18.5668
DE	19.2846	453.6750	**16.3556**
ABC	19.6943	512.7502	16.1543
ACO	18.8421	433.9878	16.0703
PSO	**18.5927**	441.1913	15.8193

Table 6
DM test for the forecasting accuracy between each pair of algorithms.

Algorithms	DE	ABC	ACO	PSO
DE		−1.0389	−1.4719	−0.0005
ABC	1.0389		0.1905	1.2904
ACO	1.4719	−0.1905		1.1300
PSO	0.0005	−1.2904	−1.1300	

standard normal. Suppose that the significance level of the test is 0.05, the null hypothesis of no difference will be rejected if the computed DM statistic falls outside the range of −1.96 to 1.96. Table 6 provides the DM statistics between each pair of algorithms. Observation revealed that all the DM statistics are within the range of [−1.96, 1.96], which give all the algorithm equal chance of forecast. This supports the reliability of the comparative study of optimization algorithms. The DM statistics between DE and the other three algorithms are −1.0389, −1.4719 and −0.0005 respectively, and these negative values reveal a preference with regard to the DE algorithm. Inversely, the positive DM statistics between

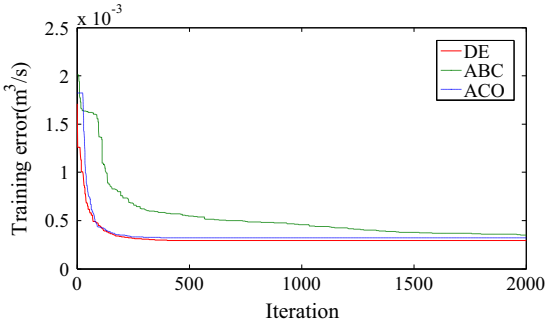

Fig. 5. Training errors (RMSE) of various algorithms versus running iteration.

Table 7
Running time and values of recession coefficient and initial storage of various algorithms.

Algorithm	Max iteration (*MI*)	Running time (h)	Parameters	
			Recession coefficient λ	Initial storage $S_{(T=0)}$
DE	500	3.1	0.3609	0.1327
ABC	2000	29.4	0.3032	0.0532
ACO	500	4.4	0.3864	0.1365

ABC and the other three algorithms indicate lower accuracy. The PSO and DE algorithms show almost equal accuracy, as indicated by the low DM statistic (0.0005). These results are consistent with the above discussions on forecasting performances.

5.2. Convergence and determination of optimal parameters

Here, the convergence speed of population-based optimization algorithms is investigated. The training error as a function of running iteration is illustrated in Fig. 5. The training error decreases rapidly when the number of iterations is less than 200 and remains approximately constant when the number of iterations approaches 500 times for DE and ACO algorithms. That is, these two algorithms can achieve convergence with 500 time runs. In contrast, the ABC algorithm reaches the minimum training error after running 2000 iterations. The minimum training errors produced by these three algorithms are sufficiently small, however, indicating that the precision is satisfactory. These three algorithms have a powerful global search ability to prevent the optimization from the premature convergence. As shown in Table 7, the running times for each algorithm are 3.1, 29.4 and 4.4 h, respectively. The ABC algorithm displays slow convergence speed, and thereby, is time consuming. This is mainly due to the localization of the method itself, in which two phases with equally large population of bees are assigned in the searching process.

As mentioned above, there are D parameters to be identified, including weight parameters, the recession coefficient and the initial storage. Fig. 6 plots the weight parameters from the input to the hidden layer and vice versa by these three algorithms. The x axis in Fig. 6(a) represents a set of weight parameters $\{w_{11}, w_{21}, w_{31}, w_{41}, w_{12}, .., w_{4,11}\}$ consisting of 44 parameters while that in Fig. 6(b) represents a set of weight parameters $\{\overline{w}_{11}, \overline{w}_{21}, \overline{w}_{31}, \ldots, \overline{w}_{11,1}\}$ consisting of 11 parameters. The y axis in Fig. 6 represents the value with respect to each weight parameter. Although without much regulation, some weight parameters produced by different algorithms are quite close to one another in Fig. 6(a). The ABC algorithm displays a worse performance than the other two because quite a number of parameters are not well searched and distributed around the maximum and minimum attainable values. This may be explained by the fact that

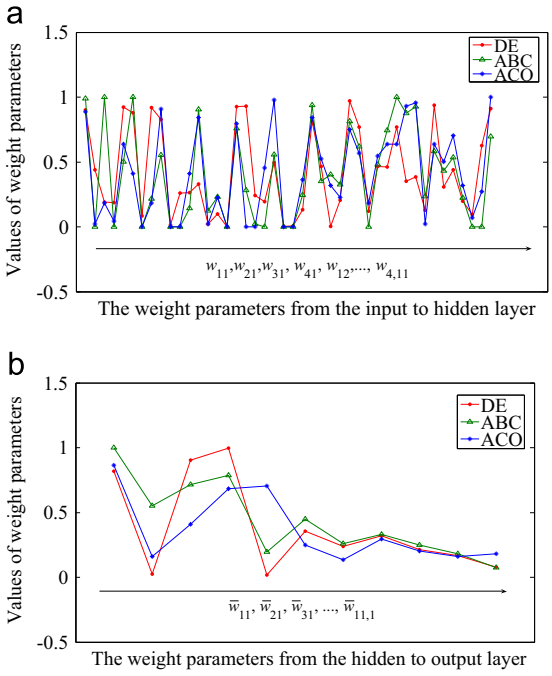

Fig. 6. The weight parameters of various algorithms (a) from the input to hidden layer (b) from the hidden to output layer.

the best solution discovered by the ABC algorithm is not always held in the population and it might be replaced with a random solution by scout bees. Therefore, the contribution of the best solution may not represent the production of trial solutions as with the DE and ACO algorithms. In view of those from the hidden to the output layer in Fig. 6(b), the DE algorithm does not show superiority. In particular, for the last five parameters, these three algorithms present competitive capability of optimizing parameters. In addition, as indicated in Table 7, the ABC algorithm obtains quite different values for the recession coefficient and initial storage while the DE and ACO algorithms tend to attain reliable values. Thereby, the DE and ACO algorithms have the feature of rapid convergence on global optima and high precision in searching parameters of the HNN model. They are consistently better than the ABC algorithm in time and derivation efficiency.

5.3. Stability of optimization algorithms

A comprehensive comparative study on the stability of these three algorithms is further carried out. The parameter of population size *NP* poses a great impact on the performance of population-based optimization algorithms. A large *NP* can endow the algorithms with powerful ability of exploring more possible solutions, although computationally intensive. Fig. 7 presents the verification of stability of *NP* in terms of RMSE of various algorithms. The DE algorithm has an adaptive relation between *NP* and RMSE, where the RMSE value gradually decreases with increasing *NP*. The choice of *NP* assures the stability of the DE algorithm. This is most likely due to donor and trial vectors which provide diversity of population and thus weaken the dependency of the performance on population size. The ABC and ACO algorithms could not reveal such consistency, as *NP* significantly affects the RMSE value. Nevertheless, the variation of RMSE with *NP* is within a reasonable range.

Mutation factor and crossover constant are two other specific parameters of the DE algorithm, and their influences on the RMSE value are demonstrated in Fig. 8. The results of Fig. 8(a) are obtained with fixed $CR=1.0$ while those for Fig. 8(b) are obtained

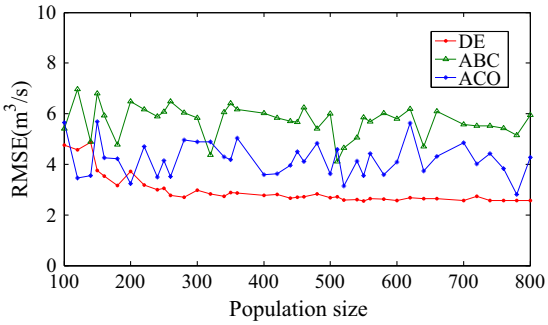

Fig. 7. The influence of population size on forecasting performances by various algorithms.

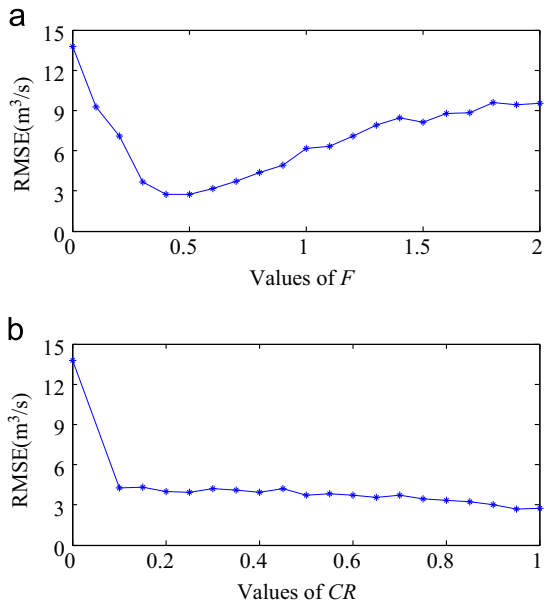

Fig. 8. The influences of (a) *F* (b) *CR* on forecasting performances by DE algorithm.

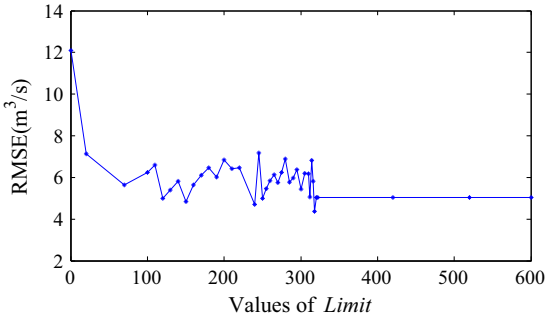

Fig. 9. The influence of *Limit* on forecasting performances by ABC algorithm.

with fixed $F=0.4$. The best RMSE value is attained when *F* and *CR* are recognized as 0.4 and 1.0, respectively. However, the plots of RMSE versus *F* and *CR* are quite different. As *F* increases from 0 to 2, RMSE first drops to a minimum value and then begins to increase. This implies that RMSE is sensitive to *F* for the DE algorithm. As a basic requirement, the chosen *F* should be able to provide a diversity of mutated vectors. With respect to *CR*, RMSE decreases sharply from 0 to 0.1, and fluctuates within a narrow range when *CR* increases. The DE algorithm presents a certain degree of stability and reliability in terms of *CR* within the interval between [0.1, 1.0].

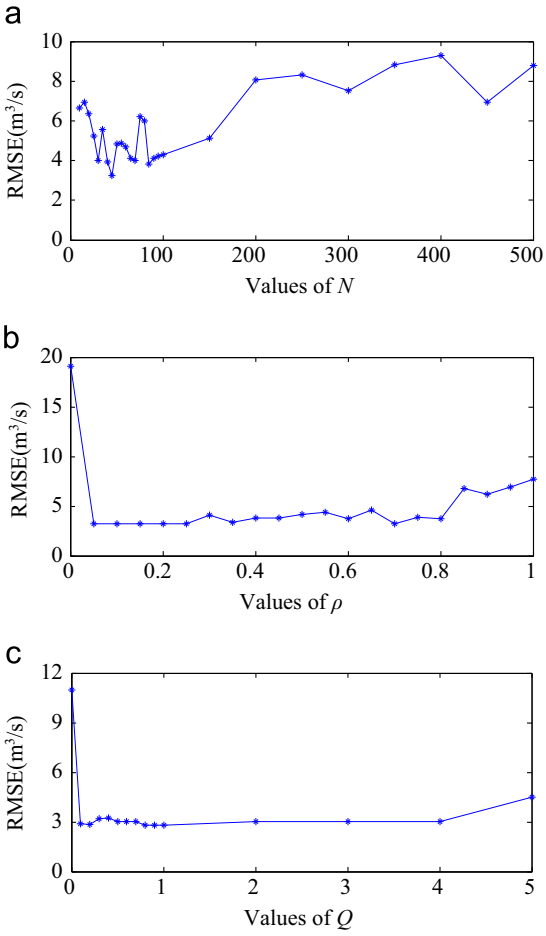

Fig. 10. The influences of (a) *N* (b) ρ (c) *Q* on forecasting performances by ACO algorithm.

Limit is an inclusive control parameter for the ABC algorithm, which governs the number of scout bees' exploitation. As illustrated in Fig. 9, the value of *Limit* is recommended to be smaller than the number of food source *NP* since a too high value of *Limit* will lead to an ineffective scout bees' phase. A too small *Limit* value which indicates that employed and onlooker bees do not work is not appreciated as well. In an ABC searching process, both exploration by scout bees and exploitation by employed and onlooker bees must be efficiently carried out (Karaboga, 2005). In this simulation, the ABC algorithm presents stability in terms of *Limit* in the interval of [80, 320]. A slight advantage of the ABC over the other two algorithms is that there is only one control parameter in addition to *NP*.

Fig. 10 presents the sensitivities of RMSE value to the number of solution components *N*, pheromone duration coefficient ρ and pheromone constant *Q* respectively for the ACO algorithm. The precision of finding the optimal solution is low and the best solution may not be obtained if *N* is small and the amount of available solutions is insufficient. On the contrary, a larger value of *N* does not definitely ensure better searching results even though there are more solution components to be selected. As displayed in Fig. 10(a) when $\rho=0.3$ and $Q=1.0$, the RMSE value strongly depends on *N* with irregular fluctuations. Generally *N* can be enacted to be less than 100 when the parameters are in the interval between [0, 1]. The fixed parameters given in Fig. 10(b) are $N=45$ and $Q=1.0$ whilst those in Fig. 10(c) are $N=45$ and $\rho=0.3$. The ACO algorithm exhibits stability regarding ρ and *Q*, apart from infeasible cases both on the boundary value 0. An appreciable value of ρ enables the ant to effectively forget the fallacious

solution component and explore a new one. The value of Q is associated with the fitness value, and a range of [0.1, 1] is suggested to assure the stability of the ACO algorithm in this study. However, these values for the control parameters vary with the dimensions of the problem or with other specific characteristics. The recommended values of these parameters may not be applicable in other studies.

6. Conclusions

In this work, the performances of three population-based optimization algorithms (i.e. DE, ABC and ACO algorithms) for evolving a HNN model are compared. The purpose is to determine a relatively reliable and stable optimization algorithm for downstream river forecasting. For completeness' sake, the PSO algorithm is included as a benchmark case for the comparison of forecasting performances. The HNN model is triggered for the forecasting, by integrating fuzzy pattern-recognition and a continuity equation into the neural network. Upstream river flows and areal precipitation over the entire river basin are adopted as input variables in the Altamaha river basin in Georgia. The capability of capturing extreme values as well as three statistical indicators (RMSE, NSEC and ACC) are employed to evaluate the forecasting performance. The DM test is examined to compare the forecasting accuracy between each pair of algorithms. The ability of parameter optimization and stability of each algorithm in terms of its control parameters are investigated. Results indicate that the DE algorithm presents the best performance on the generalization and forecasting for the HNN model, which is quite comparable to the PSO algorithm. The ACO is a feasible algorithm yielding satisfactory forecasting results as well. The ABC algorithm does not exhibit a comparable efficiency, with slow convergence and time consumption. The ACO is found to give a consistent parameter optimization of the HNN model just as the DE algorithm, particularly in terms of the recession coefficient and initial storage. The DE algorithm performs reliably well when the population size *NP* varies while the performances of the ABC and ACO are sensitive to *NP*. Regarding swarm-based heuristic algorithms, the stability of the ABC is similar to the ACO with the advantage of employing fewer control parameters. In summary, the DE and ACO algorithms are well adapted to the optimization problem for the HNN model. They can be efficiently used for solving nonlinear and non-differentiable optimization problems in multi-dimensional space.

Acknowledgments

This research was supported by Central Research Grant of Hong Kong Polytechnic University (4-ZZAD).

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.engappai.2015.09.010.

References

Babu, B.V., Angira, R., 2003. Optimization of water pumping system using differential evolution strategies. In: Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems. Citeseer.

Bao, Y., Xiong, T., Hu, Z., 2014a. Multi-step-ahead time series prediction using multiple-output support vector regression. Neurocomputing 129, 482–493.

Bao, Y., Xiong, T., Hu, Z., 2014b. PSO-MISMO modeling strategy for multistep-ahead time series prediction. Cybernetics, IEEE Transactions on 44 (5), 655–668.

Bhadra, A., Bandyopadhyay, A., Singh, R., Raghuwanshi, N.S., 2010. Rainfall-runoff modeling: comparison of two approaches with different data requirements. Water Resources Management 24 (1), 37–62.

Blum, C., Chiong, R., Clerc, M., De Jong, K., Michalewicz, Z., Neri, F., Weise, T., 2012. Evolutionary optimization. In: Chiong, R., Weise, T., Michalewicz, Z. (Eds.), Variants of Evolutionary Algorithms for Real-world Applications, 2012. Springer, Berlin, pp. 1–29.

Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Computing Surveys 35 (3), 268–308.

Chiong, R., Weise, T., Michalewicz, Z., 2012. Variants of evolutionary algorithms for real-world applications, 2012. Springer, Berlin.

Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. Journal of Business & Economic Statistics 13 (3), 253–263.

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

Harvey, D., Leybourne, S., Newbold, P., 1997. Testing the equality of prediction mean squared errors. International Journal of Forecasting 13 (2), 281–291.

Hossain, M.S., El-shafie, A., 2013. Application of artificial bee colony (ABC) algorithm in search of optimal release of Aswan High Dam. Journal of Physics: Conference Series 423, 01 2001, IOP Publishing.

Hu, Z., Bao, Y., Chiong, R., Xiong, T., 2015a. Mid-term interval load forecasting using multi-output support vector regression with a memetic algorithm for feature selection. Energy . http://dx.doi.org/10.1016/j.energy.2015.03.054, in press.

Hu, Z., Bao, Y., Xiong, T., Chiong, R., 2015b. Hybrid filter-wrapper feature selection for short-term load forecasting. Engineering Applications of Artificial Intelligence 40, 17–27.

Jalali, M.R., Afshar, A., Marino, M.A., 2006. Reservoir operation by ant colony optimization algorithms. Iranian Journal of Science and Technology, Transaction B: Engineering 30 (B1), 107–117.

Jena, P.P., Chatterjee, C., Pradhan, G., Mishra, A., 2014. Are recent frequent high floods in Mahanadi basin in eastern India due to increase in extreme rainfalls? Journal of Hydrology 517, 847–862.

Jiang, T., Su, B., Hartmann, H., 2007. Temporal and spatial trends of precipitation and river flow in the Yangtze River Basin, 1961–2000. Geomorphology 85 (3), 143–154.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Erciyes University, Engineering Faculty, Computer Engineering Department, p. 200, Technical Report-tr06.

Karaboga, D., Akay, B., 2009. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214 (1), 108–132.

Karaboga, D., Akay, B., Ozturk, C., 2007. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, Modeling Decisions for Artificial Intelligence. Springer, Berlin Heidelberg, pp. 318–329.

Kisi, Ö., Ozkan, C., Akay, B., 2012. Modeling discharge-sediment relationship using neural networks with artificial bee colony algorithm. Journal of Hydrology 428, 94–103.

Kumar, D.N., Reddy, M.J., 2006. Ant colony optimization for multi-purpose reservoir operation. Water Resources Management 20 (6), 879–898.

Li, B., Chiong, R., Lin, M., 2015. A balance-evolution artificial bee colony algorithm for protein structure optimization based on a three-dimensional AB off-lattice model. Computational Biology and Chemistry 54, 1–12.

J.B. Li Y.K. Chung A novel back-propagation neural network training algorithm designed by an ant colony optimization. In: Proceedings of Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES. IEEE pp. 1–5.

Li, M., Chen, S.Y., 2010. Fuzzy variable classified method and its application in basin floods. Fuzzy Information and Engineering, 2010. Springer, Berlin Heidelberg, pp. 627–636.

Li, X.T., Liu, H.W., Yin, M.H., 2013. Differential evolution for prediction of longitudinal dispersion coefficients in natural streams. Water Resources Management 27 (15), 5245–5260.

Li, Y.T., Gu, R.R., 2003. Modeling flow and sediment transport in a river system using an artificial neural network. Environmental Management 31 (1), 122–134.

Liu, Y., Pender, G., 2013. Automatic calibration of a rapid flood spreading model using multiobjective optimisations. Soft Computing 17 (4), 713–724.

Maier, H.R., Simpson, A.R., Zecchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y., Tan, C. L., 2003. Ant colony optimization for design of water distribution systems. Journal of Water Resources Planning and Management 129 (3), 200–209.

Moore, R.J., 2007. The PDM rainfall-runoff model. Hydrology and Earth System Sciences 11 (1), 483–499.

Qiu, L., Chen, S.Y., Nie, X.T., 1998. A forecast model of fuzzy recognition neural networks and its application. Advances in Water Science 9 (3), 258–264.

Rath, S., Nayak, P.C., Chatterjee, C., 2013. Hierarchical neurofuzzy model for real-time flood forecasting. International Journal of River Basin Management 11 (3), 253–268.

Rech, G., 2002. Forecasting with artificial network models: SSE/EFI Working Paper Series in Economics and Finance 491.

Rezaeianzadeh, M., Stein, A., Tabari, H., Abghari, H., Jalalkamali, N., Hosseinipour, E. Z., Singh, V.P., 2013. Assessment of a conceptual hydrological model and artificial neural networks for daily outflows forecasting. International Journal of Environmental Science and Technology 10 (6), 1181–1192.

Rezaeianzadeh, M., Tabari, H., Yazdi, A.A., Isik, S., Kalin, L., 2014. Flood flow forecasting using ANN, ANFIS and regression models. Neural Computing and Applications 25 (1), 25–37.
Rocca, P., Oliveri, G., Massa, A., 2011. Differential evolution as applied to electromagnetics. Antennas and Propagation Magazine, IEEE 53 (1), 38–49.
Sen, Z., 1998. Average areal precipitation by percentage weighted polygon method. Journal of Hydrologic Engineering 3 (1), 69–72.
Shelokar, P.S., Siarry, P., Jayaraman, V.K., Kulkarni, B.D., 2007. Particle swarm and ant colony algorithms hybridized for improved continuous optimization. Applied Mathematics and Computation 188 (1), 129–142.
Socha, K., Blum, C., 2007. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Computing and Applications 16 (3), 235–247.
Song, X.H., Li, L., Zhang, X.Q., Huang, J.Q., Shi, X.C., Jin, S., Bai, Y.M., 2014. Differential evolution algorithm for nonlinear inversion of high-frequency Rayleigh wave dispersion curves. Journal of Applied Geophysics 109, 47–61.
Storn, R., Price, K., 1995. Differential Evolution-a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. International Computer Science Institute, Berkeley.
Tabari, H., Kisi, Ö., Ezani, A., Talaee, P.H., 2012. SVM, ANFIS, regression and climate based models for reference evapotranspiration modeling using limited climatic data in a semi-arid highland environment. Journal of Hydrology 444, 78–89.
Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Systems with Applications 39 (8), 7067–7083.
Vasan, A., Simonovic, S.P., 2010. Optimization of water distribution network design using differential evolution. Journal of Water Resources Planning and Management 136 (2), 279–287.
K. Wang X.D. Wang J.S. Wang M.L. Jiang G.Y. Lv G.L. Feng X.L. Xu Solving parameter identification problem of nonlinear systems using differential evolution algorithm. In: Proceedings of the Second International Symposium on IEEE Intelligent Information Technology Application, 2008, IITA'08. pp. 687–691.
Xiong, T., Bao, Y., Hu, Z., 2013. Beyond one-step-ahead forecasting: evaluation of alternative multi-step-ahead forecasting models for crude oil prices. Energy Economics 40, 405–415.
Xiong, T., Bao, Y., Hu, Z., Chiong, R., 2015. Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms. Information Sciences 305, 77–92.
Yang, R.F., Ding, J., Liu, G.D., 1998. Preliminary study of hydrology-based artificial neural network. Journal of Hydraulics 8, 23–27.
Zhang, X., Fong, K.F., Yuen, S.Y., 2013. A novel artificial bee colony algorithm for HVAC optimization problems. HVAC&R Research 19 (6), 715–731.
Zhao, R.H., Chen, S.Y., 2008. Hydrological sciences for managing water resources in the Asian developing world. IAHS-AISH publication, Wallingford, pp. 39–48.