Research Article

A Class of Parameter Estimation Methods for Nonlinear Muskingum Model Using Hybrid Invasive Weed Optimization Algorithm

Aijia Ouyang,^{1,2} Li-Bin Liu,³ Zhou Sheng,⁴ and Fan Wu¹

¹College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China

²School of Information Science and Engineering, Hunan City University, Yiyang, Hunan 413000, China

³Department of Mathematics and Computer Science, Chizhou College, Chizhou, Anhui 247000, China

⁴College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi 530004, China

Correspondence should be addressed to Fan Wu; wufan.whu@163.com

Received 27 January 2015; Revised 7 May 2015; Accepted 18 May 2015

Academic Editor: Gisele Mophou

Copyright © 2015 Aijia Ouyang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nonlinear Muskingum models are important tools in hydrological forecasting. In this paper, we have come up with a class of new discretization schemes including a parameter θ to approximate the nonlinear Muskingum model based on general trapezoid formulas. The accuracy of these schemes is second order, if $\theta \neq 1/3$, but interestingly when $\theta = 1/3$, the accuracy of the presented scheme gets improved to third order. Then, the present schemes are transformed into an unconstrained optimization problem which can be solved by a hybrid invasive weed optimization (HIWO) algorithm. Finally, a numerical example is provided to illustrate the effectiveness of the present methods. The numerical results substantiate the fact that the presented methods have better precision in estimating the parameters of nonlinear Muskingum models.

1. Introduction

Flood routing plays an essential role in calculating the shape of flood waves along open river channels. In general, there are two basic methods to route floods: hydrologic routing and hydraulic routing. Hydrologic routing method is based on the storage-continuity equation whereas the hydraulic routing method is based on continuity and momentum equations. The Muskingum model, as one hydrologic routing approach, is a popular model for flood routing, whose storage depends upon the water inflow and outflow. Analysis of model features shows that classical Muskingum models can be classified into two types in form: one is linear model and the other is nonlinear model.

In the hydrological routing approach, it is very important to estimate the parameters K and x by using historical inflow and outflow records for a specific length of a reach. Generally speaking, the parameters K and x in (2) are graphically

estimated by a trial-and-error procedure. If x is obtained, the values of $[xI(t) + (1-x)Q(t)]$ are calculated by using observed data in both upstream and downstream of a river and further plotted against S . The particular value which generates the loop is accepted as the best estimate of x . The slope of the straight line fitted through the loop derives K .

In order to estimate the parameters K and x , Yoon and Padmanabhan [1] suggested and discussed three methods. Chen and Yang [2] brought forward a gray-encoded accelerating genetic algorithm. Ouyang et al. [3] developed an adaptive PSO algorithm.

In the last two decades, some nonlinear optimization algorithms have been adopted to estimate the parameters K , x , and m of the aforementioned nonlinear Muskingum model. Commonly used algorithms include genetic algorithm [4], BFGS technique [5], particle swarm optimization algorithm [6], immune clonal selection algorithm [7], parameter-setting-free harmony search algorithm [8],

Nelder-Mead simplex algorithm [9], differential evolution algorithm [10], hybrid harmony search algorithm [11], and modified honey bee mating optimization algorithm [12].

IWO algorithm [13] is a new bionic intelligent algorithm which simulates the spatial weed diffusion, growth, reproduction, and competitive survival of the invasive weeds [14]. IWO algorithm has been widely used in a variety of optimization problems and practical engineering problems: such as multiobjective optimization problem [15], parameter estimation of chaotic systems [16], model order reduction problem [17], global numerical optimization [18], antenna arrays problem [19, 20], unit commitment problem [21], optimal power flow problem [22], flow shop scheduling problem [23], traveling salesman problem [24], and economic dispatch [25].

Furthermore, we have applied the hybrid invasive weed optimization (HIWO) algorithm to the parameter estimation problems of nonlinear Muskingum model for the first time.

The remainder of the paper is organized as follows. In Section 2, we give the preliminary knowledge and the procedure of parameter derived on Muskingum models. Section 3 presents a class of new routing procedure of nonlinear Muskingum model. The basic principle of the IWO, Nelder-Mead simplex method (NMSM), and HIWO algorithms is detailed in Section 4. Section 5 shows numerical examples and analysis of results before Section 6 concludes the paper.

2. Muskingum Model

McCarthy in [26] analyzed the data from the Muskingum River in Ohio and identified the following relationship:

$$\frac{dS(t)}{dt} = I(t) - Q(t), \quad (1)$$

$$S(t) = K [xI(t) + (1-x)Q(t)], \quad (2)$$

where $S(t)$, $I(t)$, and $Q(t)$ denote channel storage, inflow rate, and outflow rate at time t , respectively; K and x denote storage time constant and weighting factor for the river.

Although the trial-and-error procedure has been used for many years, it is time consuming and entails much subjective interpretation. Therefore, in order to avoid subjective interpretations of observed data in estimating K and x , a finite difference scheme is used to resulting ordinary differential equation (1), yielding

$$\frac{S(i) - S(i-1)}{\Delta t} = \left[\frac{I(i) + I(i-1)}{2} \right] - \left[\frac{Q(i) + Q(i-1)}{2} \right], \quad (3)$$

where

$$S(i) = K [xI(i) + (1-x)Q(i)],$$

$$S(i-1) = K [xI(i-1) + (1-x)Q(i-1)], \quad (4)$$

$$i = 2, \dots, n,$$

and $Q(i)$ and $I(i)$ represent observed outflow discharges and inflow discharge, respectively, at time t_i , Δt is the time step, and n is the total time number.

As far as we know, linear Muskingum model (2) is not a well-fitting technique to observe data. Thus, some studies [4, 27, 28] proposed the following two nonlinear Muskingum models:

$$S(t) = K [xI(t) + (1-x)Q(t)]^m, \quad (5)$$

$$S(t) = K [xI^m(t) + (1-x)Q^m(t)], \quad (6)$$

where m is an additional parameter. Obviously, the parameter K in these nonlinear Muskingum models is no longer an approximation of the traveling time of the flood wave. Because the nonlinear Muskingum model in (6) is not as popular as the one in (5), many authors focus on the model in (5) for comparison purpose.

A review of the mentioned literature in Section 1 shows that the following routing procedure has been frequently used to obtain the calculated outflow discharges $\tilde{Q}(i)$ at time $t = t_i$ (see [5, 9]).

Step 1. Assume values for the three parameters, K , x , and m .

Step 2. Calculate $S(i)$ using (5), where the initial outflow is the same as the initial inflow.

Step 3. Calculate the time rate of changing storage volume $\Delta S(i)$ using the following equation:

$$\frac{\Delta S(i)}{\Delta t} = -\left(\frac{1}{1-x}\right)\left(\frac{S(i)}{K}\right)^{1/m} + \left(\frac{1}{1-x}\right)I(i). \quad (7)$$

Step 4. Use the following equation to estimate the next accumulated storage:

$$S(i+1) = S(i) + \Delta S(i). \quad (8)$$

Step 5. Calculate the next outflow as follows:

$$\tilde{Q}(i+1) = \left(\frac{1}{1-x}\right)\left(\frac{S(i+1)}{K}\right)^{1/m} - \left(\frac{x}{1-x}\right)\tilde{I}(i+1), \quad (9)$$

where $\tilde{Q}(i)$ represents the calculated outflow discharges at time t_i and $\tilde{I}(i+1) = (I(i) + I(i+1))/2$.

Step 6. Repeat Steps 2–5.

In a word, almost all of the above-mentioned optimization approaches used to calculate the parameters K , x , and m of the nonlinear Muskingum models are based on the following approximate formula:

$$\begin{aligned} \frac{\Delta S(t_i)}{\Delta t} &\approx \frac{dS(t_i)}{dt} \\ &= -\left(\frac{1}{1-x}\right)\left(\frac{S(t_i)}{K}\right)^{1/m} + \left(\frac{1}{1-x}\right)I(t_i). \end{aligned} \quad (10)$$

Obviously, the truncation error of this approximate formula (10) is only $O(\Delta t)$. Thus, it is necessary to construct a scheme of higher accuracy to approximate differential equation (1). Based on the generalized trapezoid formula [29], this paper first develops a class of new difference schemes which contain a parameter θ to approximate differential equation (1). In doing so a class of new unconstrained nonlinear optimization problems which contain a parameter θ are obtained. It should be noted that the accuracy of the presented difference schemes to approximate differential equation (1) can be improved to third order, when $\theta = 1/3$. In other words, we can get a parameter estimation model with better accuracy, if $\theta = 1/3$.

3. A Class of New Routing Procedure of Nonlinear Muskingum Model

Considering that (5) is the most popular nonlinear Muskingum model, this paper follows the trend and focuses on the same model although other equations can be used. Firstly, we rewrite (1) as

$$\frac{dS(t)}{dt} = S(t) + p(t), \quad (11)$$

where $p(t) = I(t) - Q(t) - S(t)$.

Secondly, applying the generalized trapezoid formula (see [29]) to (1), we can have

$$\begin{aligned} \tilde{S}(i) &= S(i+1) - \Delta t \frac{dS(i+1)}{dt}, \\ S(i+1) &= S(i) + \frac{\Delta t}{2} \left[(1-\theta) \frac{dS(i)}{dt} + \theta \frac{d\tilde{S}(i)}{dt} + \frac{dS(i+1)}{dt} \right]. \end{aligned} \quad (12)$$

min $f(K, x, m)$

$$= \sum_{i=1}^{n-1} \left| \left(1 - \frac{\Delta t}{2} \theta \right) K [xI(i+1) + (1-x)Q(i+1)]^m - \left(1 - \frac{\Delta t}{2} \theta \right) K [xI(i) + (1-x)Q(i)]^m - \frac{\Delta t}{2} [I(i) - Q(i)] + \frac{\Delta t}{2} (1 - \Delta t \theta) [I(i+1) - Q(i+1)] \right|. \quad (17)$$

Remark 1. By using Taylor expansion, it is easy to see that the order of above difference schemes (15) is $O((\Delta t)^2)$ if $\theta \neq 1/3$. Specially, for $\theta = 1/3$, above difference scheme (15) is of third-order accuracy (see [30]).

4. Hybrid Invasive Weed Optimization (HIWO) Algorithm

4.1. Invasive Weed Optimization (IWO) Algorithm. IWO algorithm is an optimization one based on swarm intelligence [13], which can steer the search process through cooperation and competition between individuals within groups. Compared with other evolutionary algorithms, IWO algorithm maintains a population-based global search strategy. Weeds

Combining (12) and (11), we have

$$\tilde{S}(i) = (1 - \Delta t) S(i+1) - \Delta t p(i+1), \quad (13)$$

$$\begin{aligned} S(i+1) &= S(i) + \frac{\Delta t}{2} \left[p(i+1) + p(i) + (1-\theta) S(i) \right. \\ &\quad \left. + \theta \tilde{S}(i) + S(i+1) \right]. \end{aligned} \quad (14)$$

Substituting (13) into (14), we finally obtain

$$\begin{aligned} &\left(1 - \frac{\Delta t}{2} \theta \right) S(i+1) \\ &= \left(1 - \frac{\Delta t}{2} \theta \right) S(i) + \frac{\Delta t}{2} [I(t_j) - Q(i)] \\ &\quad + \frac{\Delta t}{2} (1 - \Delta t \theta) [I(i+1) - Q(i+1)]. \end{aligned} \quad (15)$$

Furthermore, by substituting (5) into (15), we have

$$\begin{aligned} &\left(1 - \frac{\Delta t}{2} \theta \right) K [xI(i+1) + (1-x)Q(i+1)]^m \\ &= \left(1 - \frac{\Delta t}{2} \theta \right) K [xI(i) + (1-x)Q(i)]^m \\ &\quad + \frac{\Delta t}{2} [I(i) - Q(i)] \\ &\quad + \frac{\Delta t}{2} (1 - \Delta t \theta) [I(i+1) - Q(i+1)]. \end{aligned} \quad (16)$$

With the aim of estimating the parameters K , x , and m , we can construct the following unconstrained nonlinear optimization problem:

refer to those strong and aggressively growing plants, which pose a serious threat to cultivated crops [31].

IWO algorithm contains four core operation steps: initialization of the population, growth and reproduction, diffusion, and competitive exclusion, which will be briefly introduced in the following sections.

4.1.1. Population Initialization. IWO regards for elements of algorithm design inspired by nature, which is a numerical optimization calculation method based on population. Prior to the optimization iteration, we need to set some parameters in the IWO algorithm: the initial weed population number $P_{\min}$, the maximum weed population number $P_{\max}$, the maximum seed number $s_{\max}$, the minimum seed number $s_{\min}$, the maximum generation times G , the problem dimension d , the

nonlinear exponent n , the interval step initial value σ_{init} , and the end value σ_{final} , then randomly generate P_{min} numbers of W , and ultimately calculate the adapted degree f of W .

Consider the following minimized (or maximized) function optimization problems:

$$\min(\max) \quad f(x), \quad x = [x_1, x_2, \dots, x_d], \quad (18)$$

where $x_{\min,i} \leq x_i \leq x_{\max,i}$, $i = 1, 2, \dots, d$, $x_{\min,i}$ and $x_{\max,i}$ represent the upper bound and lower bound of the i_{th} dimensional variable, respectively, and d represents the dimensions of the problem.

Each weed in the IWO algorithm represents a solution; that is, $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$, where x_i denotes the i_{th} individual in the population. Once the IWO is initialized, the algorithm will generate randomly P_{min} numbers of individuals to form the initial population, as shown in the following formula:

$$x_{i,k}^0 = x_{\min,k} + r \cdot (x_{\max,k} - x_{\min,k}), \quad (19)$$

where, $k = 1, 2, \dots, d$, $i = 1, 2, \dots, P_{\text{min}}$, r represents any uniformly distributed random number between 0 and 1.

4.1.2. Growth and Reproduction. In the standard IWO algorithm, each weed breeds seeds according to its fitness (survival ability). The better fitness the individual has, the more seeds it can produce. This is in line with the natural growth law; that is, there is a linear relationship between the number of seeds produced by parent generation and the adaptable degree of matrix. The worse adaptable to the environment the weeds are, the more seeds they will produce as the current research focuses on the minimum value optimization problem. The number of seeds produced by each strain of weed is decided by the following equation [14]:

$$N_s = \text{floor} \left(\frac{f_{\max} - f}{f_{\max} - f_{\min}} (s_{\max} - s_{\min}) + s_{\min} \right), \quad (20)$$

where N_s denotes the number of seeds produced per strain of weed; f denotes the fitness value of the weed; $f_{\min}$ denotes the minimum fitness value of the weed; $f_{\max}$ denotes the maximum fitness value of the weed; $s_{\max}$ denotes the maximum number of seeds produced by each strain of weed; $s_{\min}$ denotes the minimum number of seeds produced per strain of weed; $\text{floor}(x)$ means the number of taking downward to obtain an integer to the nearest one.

4.1.3. The Spatial Diffusion. During the optimization iteration of the IWO algorithm, the way of producing seeds is as follows: taking the parent generation as the benchmark, copy as many as N_s for each solution x_i in the parent generation, which will be added with normal random parameters, r ($r \in [-\sigma, \sigma]$), whose mean value $\mu = 0$ and the standard deviation is σ , respectively, so that each strain of weed will generate N_s seeds.

Normal distribution probability is generally known as the Gaussian distribution one. As one of the best important probability distribution ways in mathematics, physics, engineering, and other fields, it has a significant effect on many aspects of the statistics. If the random variable r is amenable to such a probability distribution whose mathematical expectation is μ and variance σ^2 , denoting as $N(\mu, \sigma^2)$, its probability density function can be described as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left(-\frac{(x-\mu)^2}{2\sigma^2} \right). \quad (21)$$

The random variable is also named a normal random variable, with expectation μ determining its position and the standard deviation σ determining the amplitude of distribution. This normal distribution random parameter r is the standard normal distribution whose expectation is 0 and the variance is 1. If the specified expectation is μ and the variance is σ , then we only need to add up $r = r * \sigma + \mu$.

The method of seed production is shown as follows:

$$S_{i,j,k}^g = W_{i,k}^g + R_{i,j,k}^g, \quad (22)$$

where, $i = 1, 2, \dots, P$, g means the current generation. If $g = 1$, $P = P_{\text{min}}$; otherwise $P = P_{\text{max}}$, $j = 1, 2, \dots, N_s$, $k = 1, 2, \dots, d$. W denotes weed space, S denotes seed space, and R refers to the normal distribution random sequence.

In the process of iterative computing, the standard deviation of each generation changes in accordance with the following equation:

$$\sigma = \left(\frac{G-g}{G} \right)^n (\sigma_{\text{init}} - \sigma_{\text{final}}) + \sigma_{\text{final}}, \quad (23)$$

where σ_{init} denotes the initial value of the interval step, σ_{final} denotes the final value of the interval step, σ denotes the current value of stepsize, G denotes the maximum generation time, g denotes the current generation, and n denotes the nonlinear adjustment factor.

It can be seen from formula (23) that seeds distribute around the weeds in larger step at the preliminary stage of iteration.

4.1.4. Competitive Exclusion. After some generations of the algorithm evolution, the plant population will reach the maximum population P_{max} because of rapid propagation. However, it is hoped that the number of the plants with good survival ability (fitness) is larger than that of the plants with poor survival ability (fitness). The executive way of competitive survival rule is as follows: these newborn seeds and parent weeds will be sorted in order of the fitness value after the growth and spatial distribution of each plant. After that, we have to select P_{max} strains of plants in accordance with the fitness values from small to large (we assume that the fitness function is to compute the minimum value) and then get rid of the remaining plants.

It is quite clear that the function of the competitive exclusion operator is to ensure that the next generation of individuals is superior to the current one, as a consequence of which, the average performance of the population will improve and gradually obtain the optimal or near-optimal solutions.

4.2. Nelder-Mead Simplex Method. Nelder-Mead simplex method (NMSM) [32] is also known as a derivative-free scheme method for unconstrained optimization problems. The advantages of NMSM mainly lie in its low computation complexity, fast computation speed, and strong local search ability [33].

The NMSM [32] was proposed by scholars Nelder and Mead, who tried to obtain a minimum value for solving an objective function in a multidimensional space in 1965. The NMSM has been applied to solving all kinds of optimization problems as it can obtain an accurate value for objective functions [34]. Let $f(x)$ denote an objective function, and x_i denotes the group of points in the current simplex, $i \in \{1, \dots, N+1\}$. The NMSM method is inclusive of the following steps [34].

Step 1 (order). The system of inequalities, $f(x_1) \leq f(x_2) \leq \dots \leq f(x_i) \leq \dots \leq f(x_{N+1})$, must meet the demands of the $N+1$ vertices.

Step 2 (reflection). Firstly, compute the reflection point x_R of the simplex by $x_R = \bar{x} + \alpha(\bar{x} - x_{N+1})$, ($\alpha > 0$), where $\bar{x} = (1/N) \sum_{i=1}^N x_i$ is the centroid of the n best points. Then calculate $f_R = f(x_R)$. If $f_1 \leq f_R \leq f_N$, accept the reflected point x_R and terminate the iteration.

Step 3 (expansion). If $f_R < f_1$, calculate the expansion point x_E of the simplex by $x_E = \bar{x} + \beta(x_R - \bar{x})$, ($\beta > 1$ and $\beta > \alpha$). Compute $f_E = f(x_E)$. If $f_E < f_R$, accept x_E and terminate the iteration; otherwise, accept x_R and terminate the iteration.

Step 4 (contraction). If $f_R \geq f_N$, implement a contraction operation between $\bar{x}$ and the better point of x_R and x_{n+1} .

- (a) Outside contraction: if $f_N \leq f_R < f_{n+1}$, calculate the outside contraction point x_C by $x_C = \bar{x} + \gamma(x_R - \bar{x})$, ($0 < \gamma < 1$). Compute $f_C = f(x_C)$. If $f_C < f_R$, accept x_C and terminate the iteration; otherwise, continue with Step 5.
- (b) Inside contraction: if $f_R \geq f_{n+1}$, compute the inside contraction point x_{CC} by $x_{CC} = \bar{x} - \gamma(\bar{x} - x_{N+1})$. Calculate $f_{CC} = f(x_{CC})$. If $f_{CC} < f_R$, accept x_{CC} and end the iteration; if not, continue with Step 5.

Step 5 (shrink). Calculate the N points by $v_i = x_1 + \sigma(x_i - x_1)$ ($0 < \sigma < 1$). And calculate $f(v_i)$, $i = 2, \dots, N+1$. The unordered vertices of the simplex at the next iteration consist of $x_1, v_2, \dots, v_{N+1}$. Then, go back to Step 1.

4.3. The Basic Steps of HIWO Algorithm. The main steps of executing the HIWO algorithm are shown in Algorithm 2, where W is the weed populations and b is the best individual.

Algorithm 2 (HIWO algorithm). Require $W, G, P_{\max}, P_{\min}, d, n, \sigma_{\text{init}}, \sigma_{\text{final}}, s_{\max}, s_{\min}, x_{\min}$, and $x_{\max}$. Ensure b .

- (1) Extract the set of reliable negative and/or positive samples T_n from U_n with the help of P_n .
- (2) Initialize the weed population W_0 in accordance with formula (19), let $g = 0$.
- (3) Calculate the adaptability of weeds of W_0 .
- (4) If the termination criterion is met, stop the iteration of the algorithm; otherwise execute the next step.
- (5) Calculate the standard deviation for each generation in accordance with formula (23).
- (6) Calculate the seed number produced by each individual of parent weed W_g using formula (20); compute the fitness of seeds S_g according to formula (22) until the numbers of W_g and S_g are greater than or equal to the maximum population $P_{\max}$.
- (7) Compute the adaptability of seeds S_g .
- (8) After sorting W_g and S_g in accordance with the fitness value, choose the number $P_{\max}$ of the best results as the next generation weed W_{g+1} .
- (9) Update the best individual b_g of current generation.
- (10) Execute NMSM algorithm and implement a precise search.
- (11) Let $g = g + 1$; turn to Step 3.

5. Experimental Results and Discussion

Here, all the algorithms (BFGS, NMSM, IWO, PSO, and HIWO) are implemented by MATLAB 7.0.4 and all numerical experiments will be run on a PC with CPU Intel CORE (TM) 2 Duo T6600 2.20 GHz, with 2.00 GB of RAM and with Windows 7 as the operating system.

5.1. Experimental Results of Benchmark Functions. In order to verify the performance of the HIWO algorithm, we have selected 20 benchmark functions as the test set. For the convenience of comparison, the functions in this paper are exactly the same as the test functions in literature [14, 35]. $F1-F9$ belong to multipeak functions, and each of them owns multiple local extremum, in which $F7$ has $100! = 9.33E + 157$ local extreme points, and $F10$ is the simple unimodal function in 2-dimension and 3-dimension, but it is seen as the multimodal function in more dimensions. Its search process is easy to get into local optimum. These functions can test the multimodal search capability of the algorithm to test the performance of the algorithm well. Table 1 shows the characteristics of the test function, including the search area, the theoretical optimal value, and dimension. The 20 optimization functions used in the experiment are from $F1$ to $F20$ in literature [35].

The five algorithms BFGS, NMSM, IWO, PSO, and LEA [35] are selected to compare performance with HIWO algorithm. The experimental data is cited from literature [35].

TABLE 1: The characteristics of benchmark functions.

F	Domain	Optimum	D	F	Domain	Optimum	D
$F1$	$[-500, 500]$	-12569.5	30	$F11$	$[-100, 100]$	0	30
$F2$	$[-5.12, 5.12]$	0	30	$F12$	$[-1.28, 1.28]$	0	30
$F3$	$[-32, 32]$	0	30	$F13$	$[-10, 10]$	0	30
$F4$	$[-600, 600]$	0	30	$F14$	$[-100, 100]$	0	30
$F5$	$[-50, 50]$	0	30	$F15$	$[-100, 100]$	0	30
$F6$	$[-50, 50]$	0	30	$F16$	$[-5, 5]$	-1.031629	2
$F7$	$[0, \pi]$	-99.2784	100	$F17$	$[-5, 10] \times [0, 15]$	0.398	2
$F8$	$[-\pi, \pi]$	0	100	$F18$	$[-2, 2]$	3	2
$F9$	$[-5, 5]$	-78.33236	100	$F19$	$[-5, 5]$	0.000308	4
$F10$	$[-5, 10]$	0	100	$F20$	$[0, 1]$	-3.32	6

The number of independent running on PC for benchmark functions is 30. The function evaluations (FES) of IWO, PSO, and HIWO algorithms for functions $F1$ – $F15$ are $1E5$; the FES of IWO, PSO, and HIWO algorithms for functions $F16$ – $F20$ are $1E4$. The FES of the LEA algorithm for functions $F1$ – $F15$ and $F16$ – $F20$ are much more than $1E5$ and $1E4$, respectively.

The parameter settings of the three intelligence algorithms (PSO, IWO, and HIWO) are as follows, where P is the size of population, G is the maximum number of generations, and $MaxX$ is the maximum value of objective space. For the three algorithms, the G is the same, $G = 5000$ (from $F1$ to $F15$) or $G = 500$ (from $F16$ to $F20$); the setting of FES is the same.

- (i) The PSO algorithm: population size $P = 20$; learning factor $C1 = C2 = 1.49445$; the initial weight value $W_{start} = 0.9$; the final weight value $W_{end} = 0.4$; the maximum velocity $V_{max} = MaxX/5.0$; the minimum velocity $V_{min} = -MaxX/5.0$; the calculation formulation of weight factor for the current generation is $W = W_{start} - (W_{start} - W_{end}) * \text{pow}(\text{double}(g)/G, 2)$.
- (ii) The IWO algorithm: the minimum number of weed populations $P_{min} = 8$; the maximum number of weed populations $P_{max} = 10$; the initial step $Q_{init} = 10$; the final step $Q_{fin} = 1e-3$; the minimum number of seeds $S_{min} = 2$; the maximum number of seeds $S_{max} = 3$; nonlinear index $n = 3$.
- (iii) The HIWO algorithm: the minimum number of weed populations $P_{min} = 4$; the maximum number of weed populations $P_{max} = 5$; the settings of other parameters of HIWO are the same as IWO.

Table 2 shows the performance comparison of different algorithms for 20 functions. In Table 2, “mean” denotes the mean values of 30 independent running processes; “std” denotes the standard deviation of 30 independent running processes. It can be seen from Table 2 that BFGS algorithm is the worst in terms of calculation precision; the BFGS algorithm cannot find the optimum solution or approximate optimum solution in solving 20 functions. The calculation precision of NMSM is better than the one of BFGS, but the calculation precision of NMSM for most of the functions is not ideal, except for functions $F16$ – $F19$. The calculation

precision of IWO is better than the one of PSO; IWO has obvious advantages in terms of calculation precision for solving 11 functions. However, PSO is better than IWO in solving 7 functions. IWO and PSO are just the same in terms of calculation precision when they solve functions $F16$ and $F18$. The LEA algorithm outperforms the aforementioned four algorithms. However, the performance of LEA is worse than the one of HIWO. LEA is much greater than HIWO algorithm in the case of function evaluations; the average optimal values of the LEA algorithm are better than those of HIWO for solving $F3$ and $F7$; mean values and standard deviations of the remaining 18 functions by LEA are lagging behind those of HIWO. HIWO can find the optimum solution or approximate optimum solution in 13 functions except functions $F3$, $F5$, $F6$, $F8$, $F9$, $F12$, and $F19$.

Table 3 shows the significant level test ($\alpha = 0.05$) paired sample Wilcoxon signed rank test result; HIWO has obvious advantages when compared the other algorithms. The statistical results of IWO, PSO, and HIWO are presented by three metrics: obvious advantages (represented by B), roughly equal (represented by E), and a distinct disadvantage (indicated by W). It is shown that the HIWO algorithm has distinct advantages in terms of calculation precision.

It can be seen from Figure 1 to Figure 20 that, compared with IWO and PSO, HIWO has faster convergence speed in solving benchmark functions.

The convergence speed of HIWO is faster than those of IWO and PSO except for $F16$ and $F18$ from Figure 1 to Figure 20. It is shown that the HIWO algorithm can improve the probability of global search and keep the good balance between global search and local search. We can conclude that HIWO is characterized by fast convergence speed, high calculation precision, and strong robustness.

5.2. Experimental Results of Muskingum Model. In this section, we use actual observed data of flood runoff process between Chenggouwan reach and Linqing reach of the Nanyunhe river in the Haihe Basin, Tianjin, China (the length of the river segment is 83.8 km, where there is no tributary, but a levee control exists on both sides). Lifting irrigation can occur during the water delivery and flood water may discharge into the reach when rainfall is excessively high. But these situations have little effect on the flood, the routing time

TABLE 2: Mean value and standard deviation comparison of different algorithms for benchmark functions.

<i>F</i>	BFGS		NMSM		IWO	
	Mean	std	Mean	std	Mean	std
<i>F1</i>	-2198.02	2.26E + 03	-6521.42	3.80E + 02	-7340.4	6.33E + 02
<i>F2</i>	3.13E + 02	3.75E + 01	4.39E + 01	8.02	5.82E + 01	1.53E + 01
<i>F3</i>	1.98E + 01	5.08E - 01	1.01E + 01	1.03	3.80E - 03	2.37E - 04
<i>F4</i>	4.61E + 02	5.52E + 01	3.34E + 01	1.23E + 01	8.62E - 03	7.21E - 03
<i>F5</i>	3.52E + 08	9.55E + 07	8.93	5.39	3.46E - 03	1.86E - 02
<i>F6</i>	6.27E + 08	2.00E + 08	2.58E + 03	3.70E + 03	3.70E - 04	1.97E - 03
<i>F7</i>	-31.84	1.86E + 01	-50.03	4.38	-45.91	2.83
<i>F8</i>	4.15E + 07	1.50E + 07	1.03E + 07	2.34E + 06	9.13E + 05	3.53E + 05
<i>F9</i>	-33.97	2.71	-47.24	9.07	-67.53	1.46
<i>F10</i>	6.43E + 06	7.64E + 05	1.50E + 05	4.35E + 04	1.40E + 02	6.14E + 01
<i>F11</i>	4.33E + 04	1.32E + 04	2.36E + 03	7.10E + 02	2.68E - 05	2.94E - 06
<i>F12</i>	4.61	1.58	1.64E - 01	8.19E - 02	1.55E - 02	6.64E - 03
<i>F13</i>	1.74E + 11	3.49E + 11	9.06	2.8	2.29E - 02	1.53E - 03
<i>F14</i>	1.55E + 05	1.14E + 05	2.05E + 04	1.35E + 04	4.02E - 04	1.77E - 04
<i>F15</i>	8.32E + 01	1.09E + 01	1.84E + 01	2.64	2.28E - 03	1.80E - 04
<i>F16</i>	27.55	4.24E + 01	-1.03163	7.65E - 18	-1.03163	2.49E - 09
<i>F17</i>	8.96	7.04	0.397888	2.01E - 06	0.397913	3.85E - 12
<i>F18</i>	1246.7	1.82E + 03	3	2.82E - 17	3	9.89E - 08
<i>F19</i>	1.09E - 01	6.03E - 02	6.31E - 04	4.99E - 04	1.12E - 03	2.16E - 04
<i>F20</i>	-1.1488	4.12E - 01	-3.2375	9.02E - 02	-3.2031	2.15E - 06
<i>F</i>	PSO		LEA		HIWO	
	Mean	std	Mean	std	Mean	std
<i>F1</i>	-7307.54	6.00E + 02	-12569.45	4.83E - 04	-12569.49	1.68E - 13
<i>F2</i>	4.29E + 01	1.20E + 01	2.10E - 18	3.36E - 18	0	0
<i>F3</i>	1.14E - 01	3.47E - 01	3.27E - 16	3.00E - 17	2.37E - 15	1.67E - 16
<i>F4</i>	1.43E - 02	1.68E - 02	6.10E - 16	2.51E - 17	0	0
<i>F5</i>	6.91E - 03	2.59E - 02	2.48E - 06	2.28E - 06	8.03E - 17	1.16E - 27
<i>F6</i>	5.49E - 03	1.13E - 02	1.73E - 04	1.21E - 04	6.33E - 19	4.48E - 27
<i>F7</i>	-69.32	4.61	-93.01	2.31E - 02	-90.38	1.86E - 02
<i>F8</i>	6.23E + 06	3.41E + 06	1.63E - 06	6.53E - 07	3.28E - 13	3.56E - 13
<i>F9</i>	-66.07	1.34	-78.31	6.13E - 03	-78.3323	8.01E - 18
<i>F10</i>	5.07E + 04	5.04E + 04	5.61E - 01	1.08E - 01	1.87E - 35	2.64E - 35
<i>F11</i>	7.70E - 12	3.97E - 11	4.73E - 16	6.22E - 17	0	0
<i>F12</i>	2.23E - 03	7.70E - 04	5.14E - 03	4.43E - 04	3.29E - 04	2.36E - 04
<i>F13</i>	2.38E - 05	5.97E - 05	4.25E - 19	4.24E - 19	0	0
<i>F14</i>	1.68E + 02	8.98E + 02	6.78E - 18	5.43E - 18	2.23E - 283	3.16E - 283
<i>F15</i>	5.29E - 01	2.22E - 01	2.68E - 16	6.26E - 17	1.50E - 160	2.13E - 160
<i>F16</i>	-1.03163	3.42E - 18	-1.03108	3.36E - 07	-1.031628	1.01E - 18
<i>F17</i>	0.397889	2.68E - 06	0.398	2.65E - 08	0.397887	0
<i>F18</i>	3	1.05E - 17	3.00003	6.25E - 05	3	2.51E - 18
<i>F19</i>	3.27E - 03	6.72E - 03	3.51E - 04	7.36E - 05	3.12E - 04	5.93E - 06
<i>F20</i>	-3.2451	8.59E - 02	-3.301	7.83E - 03	-3.322	1.38E - 17

FIGURE 1: Convergence of $F1$. FIGURE 2: Convergence of $F2$.

duration $\Delta t = 12$ h in 1960. More detailed data can be seen in [3].

Next, we will carry out numerical experiments from the following two aspects. On the one hand, for the purpose of illustrating the advantages of using the general trapezoid formula to approximate nonlinear Muskingum model (5), we use the HIWO to solve optimization problem (17) by choosing different θ . Table 4 lists the average absolute errors (AAE) and average relative errors (ARE) of calculated outflows with those of observed outflows for different θ in 1960, where the AAE and the ARE are given as follows:

$$AAE = \frac{1}{n} \sum_{i=1}^n |\bar{Q}(t_i) - Q(t_i)|,$$

FIGURE 3: Convergence of $F3$. FIGURE 4: Convergence of $F4$.

$$ARE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\bar{Q}(t_i) - Q(t_i)}{Q(t_i)} \right|. \quad (24)$$

It can be seen from Table 4 that the experiment results are much better, when $\theta = 1/3$, confirming the error estimation in this paper.

On the other hand, in order to test the performance of HIWO, we list the numerical results of BFGS [5], PSO [6], NMSM [9], IWO, and HIWO in 1960, where the initial parameter values of the NMSM and BFGS are chosen as $[K, x, m] = [0, 1, 1]$ and $\theta = 1/3$. Once the parameters K , x , and m are calculated, we can use (16) to calculate

FIGURE 5: Convergence of $F5$. FIGURE 7: Convergence of $F7$. FIGURE 6: Convergence of $F6$. FIGURE 8: Convergence of $F8$.

TABLE 3: Wilcoxon statistic result comparison by HIWO and other algorithms.

Metric	BFGS	NMSM	IWO	PSO	LEA
B	20	19	18	18	18
E	—	1	2	2	—
W	—	—	—	—	2

the next calculated outflow $\tilde{Q}(t_i)$, where $\tilde{Q}(t_1) = Q(t_1)$. Obviously, given that (16) is a nonlinear equation about $Q(t_i)$, the classical Newton iteration can be applied for solving it. The numerical results are displayed in Table 5. Furthermore,

TABLE 4: Numerical results with different θ for flood routing in 1960.

θ	AAE	ARE (%)	θ	AAE	ARE (%)
$\theta = 0$	6.844292	1.964226	$\theta = 1/7$	6.813278	2.010455
$\theta = 1/6$	19.629630	5.454065	$\theta = 2/3$	6.838390	2.007756
$\theta = 1/5$	7.037453	2.039629	$\theta = 3/4$	6.837713	2.006820
$\theta = 1/4$	7.108511	2.090138	$\theta = 5/6$	6.843642	2.006165
$\theta = 1/3$	6.829915	2.010007	$\theta = 1$	6.833480	2.004863

we have used the same parameters presented in Table 5 to validate the effectiveness of flood routing in 1961 and 1964, with the AAE and the ARE, respectively, listed in Table 6.

FIGURE 9: Convergence of F_9 . FIGURE 11: Convergence of F_{11} . FIGURE 10: Convergence of F_{10} . FIGURE 12: Convergence of F_{12} .

TABLE 5: Result comparison of different methods in 1960.

Algorithm	K	x	m	AAE	ARE (%)
BFGS [5]	10.815576	0.982641	1.021860	7.866438	2.345940
PSO [6]	13.095689	0.800769	0.995361	6.831708	2.010701
NMSM [9]	0.001628	0.758777	2.337256	9.871414	3.308699
IWO	4.998601	0.881355	1.136204	7.241756	2.148295
HIWO	13.281324	0.800085	0.993263	6.829915	2.010007

As the numerical results in Tables 5 and 6 show, HIWO performs overall the best among the five algorithms. Meanwhile, the parameters obtained by the HIWO algorithm in

TABLE 6: Result comparison of different methods in 1961 and 1964.

Algorithm	1961		1964	
	AAE	ARE (%)	AAE	ARE (%)
BFGS [5]	4.634922	1.156255	8.745983	2.185487
PSO [6]	3.980577	0.991698	8.710431	2.179195
NMSM [9]	10.990222	2.808921	5.334347	1.709903
IWO	4.154933	1.064907	8.475477	2.136220
HIWO	3.971981	0.990858	8.716951	2.180527

1960 can be effectively applied to the changes in river outflows of flood forecast in the future.

FIGURE 13: Convergence of $F13$. FIGURE 15: Convergence of $F15$. FIGURE 14: Convergence of $F14$. FIGURE 16: Convergence of $F16$.

Finally, the observed and calculated flows in 1960, 1961, and 1964 are given in Figures 21, 22, and 23, respectively. Figures 21–23 show that we can calculate approximations of flood outflows, by using the HIWO algorithm to estimate the parameters of unconstrained optimization problem (17), which could be highly close to observed outflows. To sum up, the numerical results indicate that the proposed method is efficient for estimating the parameters of nonlinear Muskingum model.

6. Conclusions

In this paper, a class of approaches is developed for the parameter optimization of nonlinear Muskingum models.

These methods are based on the general trapezoid formulas and a hybrid invasive weed optimization algorithm. In order to improve the precision of parameter estimation of the nonlinear Muskingum model, we first use the general trapezoid formulas to approximate the nonlinear Muskingum model and then obtain a class of high accuracy difference schemes. Last but not least, a hybrid invasive weed optimization algorithm is developed to better estimate the parameters K , α , and m of the nonlinear Muskingum model. By virtue of the adopted Nelder-Mead simplex algorithm, the efficiency and accuracy of the new algorithm are much better compared to those of existing algorithms. The proposed algorithm has

been applied to the nonlinear Muskingum model and offers similarly encouraging results.

Appendix

Consider

$$F1 = \sum_{i=1}^d -x_i \sin \left(\sqrt{|x_i|} \right),$$

$$F2 = \sum_{i=1}^d \left(x_i^2 - 10 \cos (2\pi x_i) + 10 \right),$$

$$F3 = -20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos (2\pi x_i) \right) + 20 + \exp (1),$$

$$F4 = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1,$$

$$F5 = \frac{\pi}{d} \left\{ 10 \sin^2 (\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 \cdot [1 + 10 \sin^2 (\pi y_{i+1})] + (y_d - 1)^2 \right\} + \sum_{i=1}^d u(x_i, 10, 100, 4), \quad (\text{A.1})$$

$$\text{where } y_i = 1 + \frac{1}{4} (x_i + 1), \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a, \end{cases}$$

$$F6 = \frac{1}{10} \left\{ \sin^2 (3\pi x_1) + \sum_{i=1}^{d-1} (x_i - 1)^2 \cdot [1 + \sin^2 (3\pi x_{i+1})] + (x_d - 1)^2 [1 + \sin^2 (2\pi x_d)] \right\} + \sum_{i=1}^d u(x_i, 5, 100, 4),$$

$$F7 = - \sum_{i=1}^d \sin(x_i) \sin^{20} \left(\frac{i \times x_i^2}{\pi} \right),$$

$$F8 = \sum_{i=1}^d \left[\sum_{j=1}^d (\chi_{ij} \sin \omega_j + \psi_{ij} \cos \omega_j) - \sum_{j=1}^d (\chi_{ij} \sin x_j + \psi_{ij} \cos x_j) \right]^2,$$

where χ_{ij} and ψ_{ij} are random integers in $[-100, 100]$ and ω_j is a random number in $[-\pi, \pi]$:

$$F9 = \frac{1}{d} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i),$$

$$F10 = \sum_{i=1}^{d-1} [100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2],$$

$$F11 = \sum_{i=1}^d x_i^2,$$

$$F12 = \sum_{i=1}^d x_i^4 + \text{rand} [0, 1),$$

$$F13 = \sum_{i=1}^d |x_i| + \prod_{i=1}^d |x_i|,$$

$$F14 = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2,$$

$$F15 = \max \{ |x_i|, i = 1, 2, \dots, d \},$$

$$F16 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

$$F17 = \left(x_2 - \frac{5.1}{4\pi} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cdot \cos x_1 + 10,$$

FIGURE 17: Convergence of $F17$. FIGURE 19: Convergence of $F19$. FIGURE 18: Convergence of $F18$. FIGURE 20: Convergence of $F20$.

$$\begin{aligned}
 F18 = & \left[1 + (x_1 + x_2 + 1)^2 \right. \\
 & \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \left. \right] \left[30 \right. \\
 & + (2x_1 + 3x_2)^2 \\
 & \cdot (18 - 32x_1 + 12x_1^2 - 48x_2 + 36x_1x_2 + 27x_2^2) \left. \right], \\
 F19 = & \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_ix_2)}{b_i^2 + b_ix_3 + x_4} \right]^2,
 \end{aligned} \tag{A.2}$$

where $a_1, \dots, a_{11} = [0.1957, 0.1947, 0.1735, 0.1600, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235, 0.0246]$ and $[b_1, \dots, b_{11}] = [4, 2, 1, 1/2, 1/4, 1/6, 1/8, 1/10, 1/12, 1/14, 1/16]$:

$$F20 = - \sum_{i=1}^4 c_i \exp \left[- \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right], \tag{A.3}$$

FIGURE 21: Comparison between calculated values and observed values in 1960.

FIGURE 23: Comparison between calculated values and observed values in 1964.

FIGURE 22: Comparison between calculated values and observed values in 1961.

where

$$[c_1, c_2, c_3, c_4] = [1, 1.2, 3, 3.2],$$

$$[a_{ij}]_{4 \times 6} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix},$$

$$\begin{aligned} & [p_{ij}]_{4 \times 6} \\ &= \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1415 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}. \end{aligned} \quad (\text{A.4})$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China (11301044), the key projects of excellent young talents fund in universities of Anhui province (2013SQRL095ZD), the project supported by the Scientific Research Fund of Hunan Provincial Education Department (Grant no. 13C333), and the project supported by the Science and Technology Research Foundation of Hunan Province (Grant no. 2014GK3043).

References

- [1] J. Yoon and G. Padmanabhan, "Parameter estimation of linear and nonlinear Muskingum models," *Journal of Water Resources Planning and Management*, vol. 119, no. 5, pp. 600–610, 1993.
- [2] J. Chen and X. Yang, "Optimal parameter estimation for Muskingum model based on Gray-encoded accelerating genetic algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 12, no. 5, pp. 849–858, 2007.

- [3] A. Ouyang, Z. Tang, K. Li, A. Sallam, and E. Sha, "Estimating parameters of muskingum model using an adaptive hybrid PSO algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 1, Article ID 1459003, 29 pages, 2014.
- [4] S. Mohan, "Parameter estimation of nonlinear muskingum models using genetic algorithm," *Journal of Hydraulic Engineering*, vol. 123, no. 2, pp. 137–142, 1997.
- [5] Z. W. Geem, "Parameter estimation for the nonlinear Muskingum model using the BFGS technique," *Journal of Irrigation and Drainage Engineering*, vol. 132, no. 5, pp. 474–478, 2006.
- [6] H.-J. Chu and L.-C. Chang, "Applying particle swarm optimization to parameter estimation of the nonlinear muskingum model," *Journal of Hydrologic Engineering*, vol. 14, no. 9, pp. 1024–1027, 2009.
- [7] J. Luo and J. Xie, "Parameter estimation for nonlinear muskingum model based on immune clonal selection algorithm," *Journal of Hydrologic Engineering*, vol. 15, no. 10, Article ID 006010QHE, pp. 844–851, 2010.
- [8] Z. W. Geem, "Parameter estimation of the nonlinear muskingum model using parameter-setting-free harmony search," *Journal of Hydrologic Engineering*, vol. 16, no. 8, pp. 684–688, 2011.
- [9] R. Barati, "Parameter estimation of nonlinear muskingum models using nelder-mead simplex algorithm," *Journal of Hydrologic Engineering*, vol. 16, no. 11, pp. 946–954, 2011.
- [10] D.-M. Xu, L. Qiu, and S.-Y. Chen, "Estimation of nonlinear muskingum model parameter using differential evolution," *Journal of Hydrologic Engineering*, vol. 17, no. 2, pp. 348–353, 2012.
- [11] H. Karahan, G. Gurarslan, and Z. Geem, "Parameter estimation of the nonlinear muskingum flood-routing model using a hybrid harmony search algorithm," *Journal of Hydrologic Engineering*, vol. 18, no. 3, pp. 352–360, 2013.
- [12] M. Niazkar and S. H. Afzali, "Assessment of modified honey bee mating optimization for parameter estimation of nonlinear muskingum models," *Journal of Hydrologic Engineering*, vol. 20, no. 4, Article ID 04014055, 2015.
- [13] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, 2006.
- [14] S. Peng, A. Ouyang, and J. J. Zhang, "An adaptive invasive weed optimization algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 2, Article ID 1559004, pp. 1–20, 2015.
- [15] D. Kundu, K. Suresh, S. Ghosh, S. Das, and B. K. Panigrahi, "Multi-objective optimization with artificial weed colonies," *Information Sciences*, vol. 181, no. 12, pp. 2441–2454, 2011.
- [16] M. Ahmadi and H. Mojallali, "Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems," *Chaos, Solitons & Fractals*, vol. 45, no. 9-10, pp. 1108–1120, 2012.
- [17] D. I. Abu-Al-Nadi, O. M. Alsmadi, Z. S. Abo-Hammour, M. F. Hawa, and J. S. Rahhal, "Invasive weed optimization for model order reduction of linear MIMO systems," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 4570–4577, 2013.
- [18] A. Basak, D. Maity, and S. Das, "A differential invasive weed optimization algorithm for improved global numerical optimization," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6645–6668, 2013.
- [19] Z. D. Zaharis, C. Skeberis, T. D. Xenos, P. I. Lazaridis, and J. Cosmas, "Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data," *IEEE Transactions on Broadcasting*, pp. 455–460, 2013.
- [20] Z. D. Zaharis, P. I. Lazaridis, J. Cosmas, C. Skeberis, and T. D. Xenos, "Synthesis of a near-optimal high-gain antenna array with main lobe tilting and null filling using taguchi initialized invasive weed optimization," *IEEE Transactions on Broadcasting*, vol. 60, no. 1, pp. 120–127, 2014.
- [21] B. Saravanan, E. R. Vasudevan, and D. P. Kothari, "Unit commitment problem solution using invasive weed optimization algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 21–28, 2014.
- [22] M. Ghasemi, S. Ghavidel, E. Akbari, and A. A. Vahed, "Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos," *Energy*, vol. 73, pp. 340–353, 2014.
- [23] Y. Zhou, H. Chen, and G. Zhou, "Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem," *Neurocomputing*, vol. 137, pp. 285–292, 2014.
- [24] Y. Zhou, Q. Luo, H. Chen, A. He, and J. Wu, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, no. 3, pp. 1227–1236, 2015.
- [25] A. K. Barisal and R. C. Prusty, "Large scale economic dispatch of power systems using oppositional invasiveweed optimization," *Applied Soft Computing*, vol. 29, pp. 122–137, 2015.
- [26] G. T. McCarthy, "The unit hydrograph and flood routing," in *Proceedings of the Conference of North Atlantic Division*, U.S. Army Corps of Engineers, Providence, RI, USA, 1938.
- [27] M. A. Gill, "Flood routing by the Muskingum method," *Journal of Hydrology*, vol. 36, no. 3-4, pp. 353–363, 1978.
- [28] V. P. Singh and P. D. Scarlatos, "Analysis of nonlinear Muskingum flood routing," *Journal of Hydraulic Engineering*, vol. 113, no. 1, pp. 61–79, 1987.
- [29] M. M. Chawla, M. A. Al-Zanaidi, and D. J. Evans, "Generalized trapezoidal formulas for parabolic equations," *International Journal of Computer Mathematics*, vol. 70, no. 3, pp. 429–443, 1999.
- [30] Z. Sheng, A. Ouyang, L.-B. Liu, and G. Yuan, "A novel parameter estimation method for muskingum model using new newton-type trust region algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 634852, 7 pages, 2014.
- [31] G. Pan, K. Li, A. Ouyang, X. Zhou, and Y. Xu, "A hybrid clustering algorithm combining cloud model IWO and k-means," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 6, Article ID 1450015, 2014.
- [32] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [33] X. Ren, R. Hao, Z. Sun, and B. Shi, "Quantum behaved particle swarm optimization algorithm based on simplex method," *Microelectronics & Computer*, vol. 27, no. 1, pp. 154–157, 2010.
- [34] A. Ouyang, K. Li, T. K. Truong, A. Sallam, and E. H. Sha, "Hybrid particle swarm optimization for parameter estimation of Muskingum model," *Neural Computing & Applications*, vol. 25, no. 7-8, pp. 1785–1799, 2014.
- [35] Y. Wang and C. Dang, "An evolutionary algorithm for global optimization based on level-set evolution and latin squares," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 579–595, 2007.