

Chaos DNA Optimal Algorithm for the Parameter Estimation of Muskingum Model

Xiaohua Yang and Yuqi Li

State Key Laboratory of Water Environment Simulation,
School of Environment, Beijing Normal University,
Beijing 100875, P.R.China

Email: xiaohuayang@bnu.edu.cn, Email:angela930909@sina.com

Abstract. In order to reduce the computational amount and improve the computational accuracy for parameter optimization of Muskingum models, a new algorithm, chaos DNA optimal algorithm (CDNAOA) is proposed, in which initial population are generated by chaos mapping and searching range is automatically renewed with the excellent individuals obtained by CDNAOA. Its global convergence is analyzed. Its efficiency is verified by application of Muskingum models. Compared with standard binary-encoded genetic algorithm (SGA), differential evolution algorithm (DEA), directly search algorithm (DSA) and trial and error method (TEM), CDNAOA has higher precision and rapider convergent speed. It is good for the global optimization in the practical hydrology models.

Keywords: Muskingum model; chaos; DNA algorithm; optimization; global convergence

1. Introduction

The Muskingum model is one of the most important hydrology models. Their parameters are usually estimated from historical data using a trial and error method (TEM) [1], genetic algorithm (GA) [2], differential evolution algorithm (DEA), directly search algorithm (DSA) [3]. The parameter optimization of the complicated models is intractable mathematically with traditional optimization methods. Once an objective function has many local extreme points, the traditional optimization methods may not obtain the global optimization efficiently [2,4,5].

Recently Adleman [6] showed that DNA can be used to solve a computationally hard problem. He solved the Hamiltonian path problem for a directed graph with seven nodes and demonstrated the power of solving computational problems using molecules in solution [7]. Lipton then solved an instance of the satisfiability problem to reveal the advantage of the huge parallelism inherent in DNA computing [8].

In addition, Chaos can be considered as the movement in a limited range occurred in a deterministic nonlinear dynamic system. Such a movement is very similar to a random process. It is extremely sensitive to the initial condition, a property sometimes referred to as the instability in Liapunov's sense or the so-called butterfly effect [9]. The chaotic map has special characters such as the ergodic property, stochastic property and sensitivity dependence on initial conditions. Taking advantage of this characteristic feature of the chaotic system, one can design an efficient approach for maintaining the population diversity in the problem of interest [10-12].

In this paper, a chaos DNA optimal algorithm (CDNAOA) is presented to reduce computational amount and to improve the calculation precision in which initial population are generated by chaos mapping. It gradually directs to an optimal result with the excellent individuals obtained by CDNAOA. Its efficiency is verified by application of Muskingum model in the practical water environmental model.

2. Muskingum models

The flow conditions at two locations on a river can be related through the continuity equation.

$$\frac{dW}{dt} = I - Q \quad (1)$$

$$W = kQ' = k(xI + (1-x)Q) \quad (2)$$

In which I and Q represent the inflow and outflow discharges to the river reach at the two locations, respectively, W represents the storage volume within the reach, k and x are the Muskingum parameters, and t is time. The continuity equation can be expressed in finite difference form as:

$$\tilde{Q}(1) = Q(1) \quad (3)$$

$$\tilde{Q}(i) = c_1 I(i) + c_2 I(i-1) + c_3 \tilde{Q}(i-1), i = 2, \dots, n \quad (4)$$

$$c_1 + c_2 + c_3 = 1 \quad (5)$$

In which $\tilde{Q}(i)$ and $Q(i)$ represent the calculated outflow discharges and observed outflow discharges at time t , respectively. $I(i)$ represents the inflow discharge at time interval t_i . n is the total time number, c_1, c_2 and c_3 are the Muskingum parameters in (4).

In order to estimate the parameters of the above Muskingum models, we adopt the following objective function:

$$\begin{aligned} \min f(c_1, c_2) \\ = (\sum_{i=2}^n c_1 I(i) + c_2 I(i-1) + (1-c_1-c_2) \tilde{Q}(i-1) - Q(i)) / (n-1) \end{aligned} \quad (6)$$

3. Description of Cdnaoa

Consider the following optimization problem:

$$\begin{aligned} \min f(c_1, c_2, \dots, c_p) \\ \text{st. } a_j \leq c_j \leq b_j, \text{ for } j=1, 2, \dots, p \end{aligned} \quad (7)$$

where $c = \{c_j, j = 1, 2, \dots, p\}$, c_j is a parameter to be optimized, f is an objective function and $f \geq 0$.

The procedure of CDNAOA is shown as follows.

Step 1 (DNA encoding). Suppose DNA encoding length is m in every parameter, the j th parameter range is the interval $[a_j, b_j]$, and then each interval is divided into $2^m - 1$ sub-intervals:

$$c_j = a_j + \theta_j \cdot (2^m - 1) \cdot h_j \quad (8)$$

where the length of sub-interval of the j th parameter $h_j = \frac{b_j - a_j}{2^m - 1}$ is constant. The searching location $I_j = \theta_j \cdot (2^m - 1)$, is an integer, and $0 \leq I_j < 2^m$, θ_j is a chaos variable, $0 \leq \theta_j \leq 1$, for $j = 1, 2, \dots, p$.

The DNA code array of the j th parameter is denoted by the grid points of $\{d(j, k) \mid k = 1, 2, \dots, m\}$ for every individual:

$$I_j = \sum_{k=1}^m d(j, k) \cdot 2^{k-1} \quad (9)$$

where \oplus denotes the operator of addition modulo 2 on $\{0, 1\}$. CDNAOA's process operates on a population of individuals (also called DNA code array, strings or chromosomes). Each individual represents a potential solution to the problem. For corresponding $1 \times 1 \rightarrow T, 1 \times 0 \rightarrow A, 0 \times 1 \rightarrow G, 0 \times 0 \rightarrow C$. The first position value '1' or '0' expresses the position of DNA code and the second position value '1' or '0' expresses the true value of binary code, and the value of DNA code.

Step 2 (Creating the initial population by chaos algorithm). To cover the whole solution space and to avoid individuals entering into the same region, large uniformity random population is selected in this algorithm. The initial population of n chromosomes is generated by chaos algorithm. We give chaos variable θ_j as follows:

$$x(j, k, i) = 1 - a \cdot x(j, k, i-1)^2 + b \cdot x(j, k, i-2) \quad (10)$$

$$\theta(j, k, i) = \frac{x(j, k, i) - x_{\min}}{x_{\max} - x_{\min}}, \quad i = 1, 2, \dots, n \quad (11)$$

where $\theta(j, k, i)$ are chaos controlling variables of Henonmap system, where x_{\min} is the minimum of $x(j, k, i)$ and x_{\max} is the maximum of $x(j, k, i)$. Here we let $a = 1.41, b = 0.34$ and $\theta(j, k, -1) = 0.95$, $\theta(j, k, 0) = 0.23$, $i = 1, 2, \dots, 300$. When $a = 1.4, b = 0.3$, phase points will full in the phase space and system (10)-(11) will be chaos [9]. Once the initial father population has been generated, the decoding and fitness evaluation should be done.

Step 3 (Evaluating fitness value of each individual). The smaller the value $f(i)$ is, the higher the fitness of its corresponding i th chromosome is. So the fitness function of i th chromosome is defined as follows:

$$F(i) = \frac{1}{[f(i)]^2 + 1.0}$$

Step 4 (Selection). Select chromosome pairs randomly depending on their fitness value from the initial population. And two groups of n -chromosomes $d_1(j, k, i), d_2(j, k, i)$, are gotten ($j = 1, 2, \dots, p; k = 1, 2, \dots, m; i = 1, 2, \dots, n$).

Step 5 (Two-point crossover and two-point mutation). Perform crossover and mutation on chromosomes the same as GA in Fig. 1 and Fig. 2 as follows:

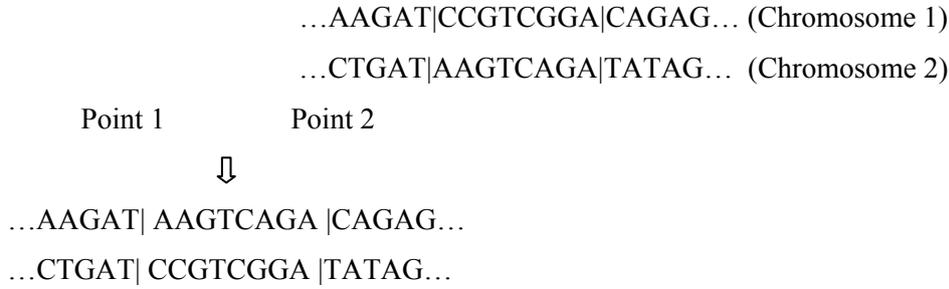


Figure 1. Flow chart describing the CDNAOA for two-point crossover

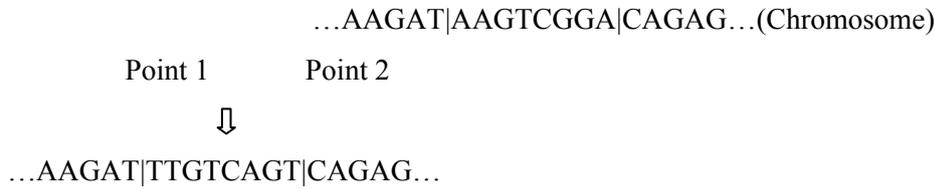


Figure 2. Flow chart describing the CDNAOA for two-point mutation

Step 6 (DNA evolution). Repeat Steps 3 ~6 until the evolution times $q = Q$ (Q is the total evolution times) or the termination condition is satisfied.

Step 7 (Optimal accelerating cycle). The parameter ranges of n_e -excellent individuals (the n_e individuals nearest to the optimum solution in the evolution) obtained by Q -times of the DNA-encoded optimal evolution alternating are regarded as the new ranges of the parameters, and then the whole process back to the DNA-encoding. The CDNAOA computation is over until the algorithm running times gets to the design T times or there exists an optimal chromosome C_{fit} whose fitness satisfies a given criterion. In the former case the C_{fit} is the fittest chromosome or the most excellent chromosome in the population. The chromosome C_{fit} represents the solution.

The CDNAOA is convergent if the shrinking interval ratio $k_j^t \leq \alpha < 1, \forall t, j = 1, 2, \dots, p$ [13]. Considering the evolution generations Q , excellent individuals n_e , optimized parameter p and accelerating evolution generations T , the probability p_e of excellent individuals surround the optimum point is $p_e = (1 - 2^{-Qn_e})^{pT}$ [13]. And the CDNAOA is global convergence with probability $p_e = 1.000\ 000$, and $n_e = 10, Q \geq 3$. From the above analysis, we can conclude that the global optimization of the CDNAOA is not only accurate but also stable.

In this paper the parameters of the CDNAOA are selected as follows: The length $e=10$, population size $n=100$, the number of excellent individuals $n_e=10$, the times of evolution alternating $Q=3$, the crossover probability $p_c=1.0$, and the mutation probability $p_m=0.5$.

4. Practical Example

Example. An automatic calibration for the Muskingum models.

The Parameter estimation of Muskingum models is studied in this paper. Time interval $\Delta t = 12h$. The observed inflow and outflow discharges for the practical example are shown in Fig. 3. The detail data can be seen in [3].

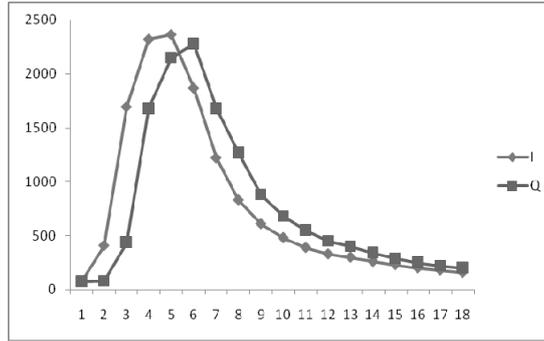


Figure 3. The figure of the observed inflow discharges I and outflow discharges Q

The parameters c_1 and c_2 are required in this model. The significance of these parameters can be seen in (4)-(6). In this work, these two parameters are estimated with respect to one criterion, namely the mean absolute error. The form of the objective function is described in (6).

The optimal parameters $c_1=0.1159$, $c_2=0.7174$, the mean absolute error f is 45.84 with CDNAOA. For CDNAOA, the evaluation number of the objective function is 900. The computational results of the above model are given in Table 1. For SGA (Standard binary-encoded genetic algorithm), the evaluation number of the objective function is 2400, and the mean absolute error f is 52.59. For DEA (Differential evolution algorithm) [3], the evaluation number of the objective function is 2400, and the mean absolute error f is 47.98. For DSA (Directly search algorithm) [3], the mean absolute error sum f is 49.55. And for TEM (Trial and error method), the mean absolute error f is 52.84.

Table 1 gives the errors comparison of several methods. From Table 1, we can see that the results achieved with our CDNAOA are satisfactory in global optimum and convergent speed. In terms of minimizing the objective function, CDNAOA has shown to be capable for Muskingum models.

TABLE I. COMPARISON WITH SEVERAL METHODS IN THE MUSKINGUM MODELS

Methods	Parameters			Mean absolute errors f
	The evaluation number	c_1	c_2	
CDNAOA	900	0.1159	0.7174	45.84
SGA	2400	0.0151	0.8699	52.59
DEA	2400	0.1269	0.7296	47.98
DSA	-	0.2310	0.5380	49.55
TEM	-	0.1175	0.7554	52.84

5. Conclusion

In this paper, the chaos DNA optimal algorithm (CDNAOA) is developed for the parameter optimization of Muskingum models. The circulating mechanism of CDNAOA has been studied. The corresponding global convergent ability is analyzed for the new CDNAOA. The global optimization of the CDNAOA is not only accurate but also stable. Because the DNA-encoding, the chaos initial population and accelerating cycle are adopted, the efficiency and accuracy of the new CDNAOA algorithm are very high compared with SGA, DEA, DSA and TEM methods. It has been applied to the Muskingum models, and the results are encouraging. This paper provides a good optimal algorithm for the parameter estimation of the practical hydrology models.

6. Acknowledgment

This work was supported by the National Basic Research Program of China (No. 2010CB951104), the Project of National Natural Science Foundation of China (No. 50939001, 51079004), the Specialized Research Fund for the Doctoral Tutor Program of Higher Education (No. 20100003110024), and the Program for Changjiang Scholars and Innovative Research Team in University (No. IRT0809).

7. References

- [1] H.J. Chu, "The Muskingum Flood Routing Model using a Neuro-Fuzzy Approach," *KSCE Journal of Civil Engineering*, vol. 13, May 2009, pp. 371-376, doi:10.1007/s12205-009-0371-6.
- [2] S. Mohan, "Parameter estimation of nonlinear Muskingum models using genetic algorithm," *Journal of Hydraulic Engineering*, ASCE, vol. 123, Feb. 1997, pp. 137-142, doi:10.1061/(ASCE)0733-9429.
- [3] X.J. Xu and X.X. Zhong, "Differential Evolution for Parameter Estimation of Muskingum Model," *Complex Systems and Complexity Science*, vol. 5, Mar. 2008, pp. 85-91.
- [4] J. Yoon and G. Padmanabhan, "Parameter estimation of linear and nonlinear Muskingum Routing Models," *J. Water Resour. Planning Mgmt*, vol. 119, May 1993, pp. 600-610, doi:10.1061/(ASCE)0733-9496.
- [5] X.H. Yang, Z.F. Yang, and Z.Y. Shen, "GHHAGA for Environmental Systems Optimization," *Journal of Environmental Informatics*, vol. 5, Jan. 2005, pp. 36-41, doi:10.3808/jei.200500044.
- [6] L. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, Nov. 1994, pp. 1021-1024, doi:10.1126/science.7973651.
- [7] C.W. Yeh, C.P. Chu, and K.R. Wu, "Molecular solutions to the binary integer programming problem based on DNA computation," *Biosystems*, vol. 83, May 2006, pp. 56-66, doi:10.1016/j.biosystems.2005.09.005.
- [8] R.J. Lipton, "DNA Solution of hard computational problems," *Science*, vol. 268, Apr. 1995, pp. 542-545, doi:10.1126/science.7725098.
- [9] E.N. Lorenz, "Deterministic non-periodic flows," *Jat Mos Sci.*, vol. 20, Feb. 1963, pp. 130-141, doi:10.1175/1520-0469.
- [10] Q.Z. Lu, G.L. Shen, and R.Q. Yu, "A chaotic approach to maintain the population diversity of genetic algorithm in network training," *Computational Biology and Chemistry*, vol. 27, Feb. 2003, pp. 363-371, doi:10.1016/S1476-9271(02)00083-X.
- [11] X.H. Yang, Z.F. Yang, X.A. Yin, and J.Q. Li, "Chaos gray-coded genetic algorithm and its application for pollution source identifications in convection-diffusion equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, Aug. 2008, pp. 1676-1688, doi:10.1016/j.cnsns.2007.03.003.
- [12] B. Liu, L. Wang, and Y.H. Jin, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, May 2005, pp. 1261-1271, doi:10.1016/j.chaos.2004.11.095.
- [13] X.H. Yang, Z.F. Yang, G.H. Lu, and J.Q. Li, "A Gray-encoded, hybrid-accelerated, genetic algorithm for global optimizations in dynamical systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, Apr. 2005, pp. 355-363, doi:10.1016/j.cnsns.2003.12.005.