



遺傳規劃法應用於水庫入流量預測之研究

A Study on Reservoir Inflow Forecasting Using Genetic Programming

陳永祥

YUNG-HSIANG CHEN

經濟部水利署科長
國立臺灣大學生物環境系統工程學研究所
博士

張斐章*

FI-JOHN CHANG

國立臺灣大學生物環境系統工程學研究所
教授

摘 要

流量預測對於水資源系統規劃及操作甚為重要，尤以中、長期流量預測對於乾旱時期供水系統為甚。而在各種供水系統中，水庫向來被視為最重要且有效之蓄水設施，其入流量如能於事前精確地預測，將有助於水庫操作與管理。惟因流量之形成具高度非線性及時變性，且隨空間分布而異，故甚難預測。近來有甚多研究發展出不同之流量預測模式以求更精確之結果，而以達爾文進化論為理論基礎之遺傳規劃法(或稱基因規劃法)為其中較為嶄新之一種，由於該法具探索及學習隱藏於資料關係之能力，且能自動化地以數學運算元求解，並將模式以方程式展現，故已應用於多種領域。本研究之目的係將遺傳規劃法應用於週期性之流量預測，分別以石門水庫之旬入流量及集水區平均雨量資料進行不同之輸入組合，再藉由根均方差最小之目標函數，建構合適之入流量預測模式。研究結果顯示，遺傳規劃法所建構之預測模式表現較時間序列模式及線性迴歸模式為佳，且輸入資料經標準化後亦能增加模式之精確度。

關鍵詞：流量預測，旬流量，降雨-逕流模式，遺傳規劃法，石門水庫入流量。

ABSTRACT

Streamflow forecasting is of significant importance for planning and operation of water resource systems. Mid-term streamflow forecasting is especially important for the operation of water supply systems over drought seasons. Among the water supply systems, reservoirs should be regarded as the most important and effective water storage facilities. If the seasonal inflows of reservoir can be forecasted precisely beforehand, it may benefit the reservoir operation and management. However, streamflow formation is a highly non-linear, time varying, spatially distributed process and difficult to be forecasted. A large variety of models have been proposed to get more accurate and reliable performance for streamflow forecasting. Among the rainfall-runoff models, genetic programming (GP) has the advantages of its ability to learn relationships hidden in data and express them automatically in a mathematical manner. This study applied GP to establish the ten-daily inflow forecasting models for Shihmen Reservoir catchment. The forecasting of 10-day reservoir inflows reveals the excellent effectiveness of GP, and standardization is beneficial to modeling for seasonal time series.

Keywords: Streamflow forecasting, Ten-daily streamflow, Rainfall-runoff model, Genetic programming, Shihmen reservoir inflow.

* 通訊作者，國立臺灣大學生物環境系統工程系教授，10617台北市大安區羅斯福路4段1號，changfj@ntu.edu.tw

I. INTRODUCTION

Streamflow forecasting is of significant importance for planning and operation of water resource systems, such as water supply, flood mitigation, irrigation, drainage, water quality, power generation, recreation, and fish and wildlife propagation. For hydrologic component, there is a need for short-term (hourly or daily), mid-term (weekly, ten-daily or monthly), and long-term (yearly) forecasts of streamflow events in order to optimize the system or to plan for future expansion or reduction. Mid-term streamflow forecasting is especially important for the operation of water supply systems over low-flow or drought seasons. Among the water supply systems, reservoirs should be regarded as the most important and effective water storage facilities, which have the functions of modifying uneven distribution of rainfalls and allocating water resources. Thus if the reservoir inflows can be forecasted precisely beforehand, it may benefit the reservoir operation and management.

Streamflow formation is a highly non-linear, time varying, and spatially distributed process and difficult to be forecasted. A large variety of models have been proposed to obtain more accurate and reliable performance for streamflow forecasting (Chang and Chen, 2001; Chen *et al.*, 2002; Chang *et al.*, 2003; Chen and Chang, 2009). Among them the rainfall-runoff models are the most common for hydrologists to forecast streamflow. Nevertheless, the rainfall-runoff relationships in the models are very complex and complicated to comprehend due to the tremendous spatial and temporal variability of watershed characteristics, heterogeneity in precipitation, as well as a number of variables involved in modeling the physical processes (Moradkhani *et al.*, 2004). Besides, the quality of the forecasts generally depends on the quality of the simulation models, the accuracy of the observed precipitation, and the efficiency of the data

assimilation procedure. Without a large number of parameter calibration data that are difficult to collect all of them, the data-driven models, built upon only the observations of the input and output, are regarded as simpler rainfall-runoff models. Among the data-driven models, genetic programming (GP) has the advantages of its ability to learn relationships hidden in data and express them automatically in a mathematical manner and then is applied in this study.

A great number of previous studies applied genetic programming to their fields. However, it seems that not many efforts have been made to the applications to rainfall-runoff or streamflow modeling. Babovic (1996) introduced the GP paradigm in the area of water engineering first soon after Koza, who first proposed GP in 1992. Drecourt (1999) applied genetic programming and artificial neural networks (ANNs) to forecast runoff with hourly and daily precipitation and runoff data and comparing the efficiency of the two tools with a classic lumped model and naive prediction. Madsen *et al.* (2000) used GP, ANN, and autoregressive (AR) models to analyze the data assimilation in rainfall-runoff forecasting. Babovic and Bojkov (2001) modeled the runoff of a semi-arid catchment with GP and ANN with hourly and daily precipitation and runoff data. They recommended for further study the application of both data mining techniques for more comprehensive, and longer-term runoff prediction. Drecourt (2001) applied GP, ANN, and AR models to analyze the data assimilation in rainfall-runoff forecasting. The work was an extension of Madsen *et al.* (2000). Muttil and Liong (2001) used the Group Method of Data Handling (GMDH) technique for selecting the significant variables to be used as input to GP, which leads to improve runoff forecasting. Chen *et al.* (2002) combined grammatical evolution and genetic algorithm, two branches of the evolutionary algorithm, to establish the ten-daily streamflow forecasting models for Dechi Reservoir in central Taiwan. They showed

that the forecast performance of their ten-daily models is better than that of multiple regressions. Chen *et al.* (2005) applied GP to river routing model for Chilung river catchment. They found that the unrestricted syntax tree mode is superior to the restricted syntax tree mode in an hourly river routing.

II. Heredity Plan Law

Genetic programming (GP) is a relatively new automated, domain-independent technique that was first proposed by John Koza in 1989. Inspired by the theory of evolution and current understanding of biology and nature evolution, GP mimics in a very simplified way the evolutionary processes occurring in nature. Via evolving computer programs for solving or approximately solving problems, the aim of GP is to get the best solution by selecting among a population of potential solutions. As members of a species are selected in nature, the selection is according to their adaptation to the environment.

The evolutionary algorithms (EAs) have become very popular recently because of the success in searching complex non-linear spaces and their robustness in practical applications (Babovic and Bojkov, 2001). GP is one of the EAs and normally regarded as a variant or specialization of genetic algorithms (GAs), invented by John Holland in 1975. GP is a much more powerful technique than GAs in the sense that encoding is much more expressive and it is much more likely to find solutions that were not anticipated by the designer. In a word, the output of the GA is a quantity, while the output of the GP is another computer program.

GP starts initially with a number of randomly created computer programs, which are possible solutions to a given optimization problem. Each program is regarded as an individual and all individuals form a population. Each individual in the population is generally represented as a binary tree (GP tree) composed of functions and data/

terminals appropriate to the problem domain. The function set may contain standard arithmetic operators, mathematical functions, logical operators, and domain-specific functions, whereas the terminal set usually consists of feature variables and constants. Besides, each individual in the population is assigned a fitness value, evaluating how well it performs in the problem environment. The fitness value is computed by a problem-dependent fitness function. Common fitness functions include mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), percentage absolute error (PAE), coefficient of determination (CoD), coefficient of correlation (CC), the goodness-of fit with respect to the benchmark (G_{bench}), or the length of the equations (Babovic and Bojkov, 2001; Chen *et al.*, 2002; Muni and Pal, 2004; Chang *et al.*, 2005). Then, the population is progressively evolved over a series of generations. The evolutionary search for every generation uses biologically inspired operations like reproduction, crossover, and mutation. The search will be terminated when the smallest fitness value of any individual in a generation is less than a given criterion or the preset generation has been reached.

The use of GP includes the following main steps and the schematic flowchart is shown in Fig. 1:

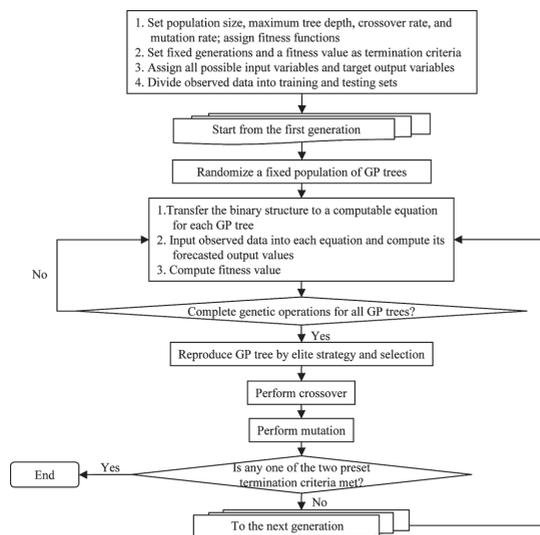


Fig. 1. Flowchart of genetic programming.

1. Settings:

Set population size, maximum tree depth, crossover rate, and mutation rate; assign fitness functions (e.g., RMSE and CC); set fixed generations and fitness values as termination criteria; assign all possible input variables and target output variables; divide the observed data into training and testing sets.

2. Initialization:

Generate an initial population of binary trees formed by random compositions of the non-terminal nodes (standard arithmetic operators, mathematical functions, and/or logical operators) and terminal nodes (variables and constants). Fig. 2 shows an arbitrary GP tree.

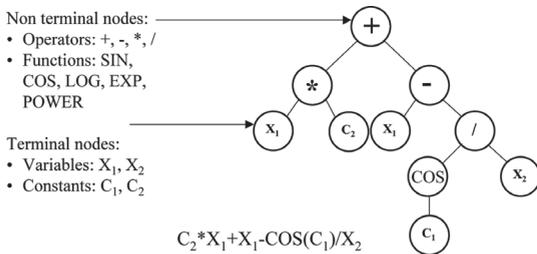


Fig. 2. Binary tree structure of GP.

3. Decoding and computation of fitness:

- (1) Transfer the binary structure to a computable equation for each GP tree.
- (2) Input training and testing data into each equation and compute its forecasted output values for training and testing parts, respectively.
- (3) Compute the fitness value for each GP tree by inputting the target and forecasted output values into the preset fitness functions for training and testing data, respectively; record the best.

4. Genetic operation:

If any one of the termination criteria has not been met in the current generation, create a new population of binary trees by the following operations:

(1) Reproduction

Elite and selection of GP trees are two strategies of reproduction. One or a few with

better fitness from parent GP trees in the current generation, called elite GP trees, are selected and kept unchanged in the next generations. Then two that are not elite GP trees are randomly selected by specified selection method (e.g., tournament selection) and one with better fitness is put into a match pool, where consists of all GP trees except elite ones in the current generation.

(2) Crossover

Crossover of two GP trees, selected in the match pool, is performed by the following steps:

- A. Randomly select a node in one tree and separate the tree into main-tree part (without the selected node) and sub-tree part (with the selected node).
- B. Randomly select a node in another tree and separate into main-tree and sub-tree parts in the same way.
- C. Interchange the two sub-trees, i.e., combine the main-tree part of the first tree with the sub-tree part of the second tree to form a new GP tree; combine the main-tree part of the second tree with the sub-tree part of the first tree to form another new GP tree. Fig. 3 illustrates a schematic crossover for two GP trees.

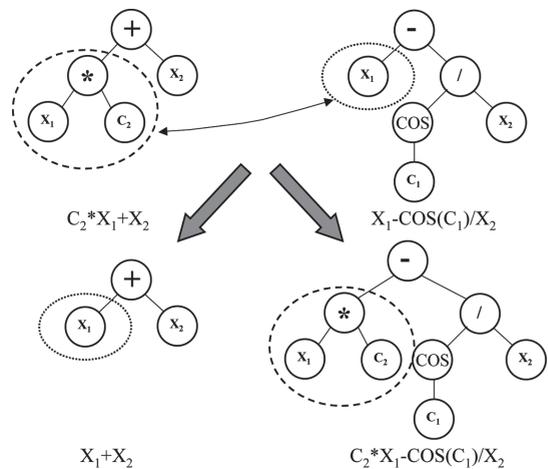


Fig. 3. Crossover operation of GP trees.

(3) Mutation

Mutation of GP trees is performed by the following steps:

- A. Randomly select a node in a tree as a mutation

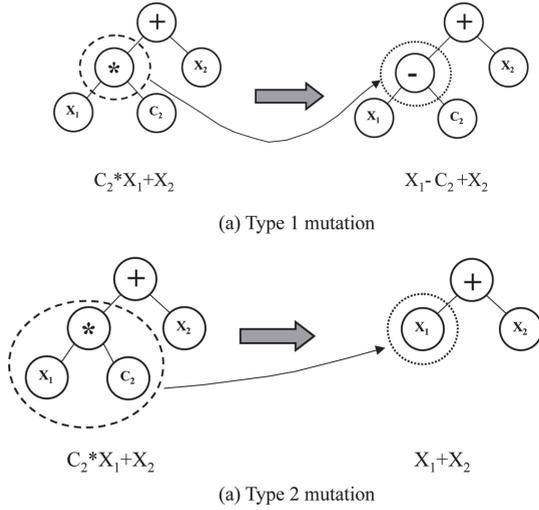


Fig. 4. Mutation operation of GP tree.

point.

- B. Generate a new node or a new sub-tree.
 - C. Replace the selected node with new generated node (type 1 mutation); or replace the sub-tree rooted at the selected node with new generated sub-tree (type 2 mutation). Fig. 4 shows two types of schematic GP mutation. It should be noted that the maximum depth of the mutated tree must be less than or equal to the preset maximum tree depth.
5. Repeat steps 3 and 4 for offspring GP trees in the next generations until one of the termination criteria have been met.

III. Research Case

The objective of this study is to establish the ten-daily inflow forecasting models for Shihmen Reservoir catchment. The performances of the GP models are compared with AR(1), ARMAX, and linear regression models.

The Shihmen Reservoir is located in the upper reaches of the Tahan River, a branch of Tamshui River in northern Taiwan. The catchment area of the reservoir is 763.4 km² and the effective water storage is 251.88 million cubic meters. The reservoir serves a number of purposes, including

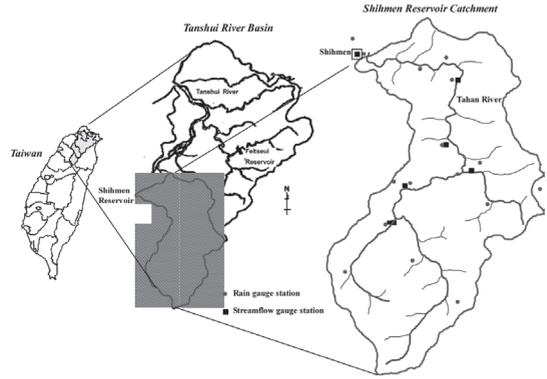


Fig. 5. Location of study area.

irrigation, hydroelectric power, fresh water supply, flood prevention and sightseeing. Supplying water to 28 districts and homing to 3.4 million people, the reservoir is very important water source for the livelihoods of the people living in northern Taiwan.

Locations of the study area and gauge stations are shown in Fig. 5. The weighted average rainfalls over the catchment and reservoir inflow measurements with time step of ten days are available from the gauging stations. The data were collected from the gauging stations during the period from 1965 to 2003 and divided into a training set, 30 years, and testing set, 9 years. The total numbers of data sets are 1,080 ten-days for training and 324 ten-days for testing, respectively.

Table 1 shows the different cases of selected input data to GP in this study. The six cases can be divided into two groups: the inputs with observed data (cases 1, 2, and 3) and those with standardized data (cases 3, 4, and 5). Since standardization of seasonal time series data is usually used to eliminate the effects of periodical trends, the purpose of the second group is to investigate whether standardization could improve the performance of GP.

The standardization of rainfall and inflow data can be attained by Eqs. (1) and (2), respectively.

$$P_{ij}^s(t) = \frac{P_{ij}(t) - \overline{P_j(t)}}{\sigma_{P_j}(t)} \quad (1)$$

Table 1. The cases of different input data to GP in this study

	Inputs	Forecasting
Case 1	Random selection of P(t), P(t-1), P(t-2)	Q(t+1)
Case 2	Random selection of Q(t), Q(t-1), Q(t-2)	Q(t+1)
Case 3	Random selection of P(t), P(t-1), P(t-2), Q(t), Q(t-1), Q(t-2)	Q(t+1)
Case 4	Random selection of PS(t), P ^S (t-1), P ^S (t-2)	Q(t+1)
Case 5	Random selection of Q ^S (t), Q ^S (t-1), Q ^S (t-2)	Q(t+1)
Case 6	Random selection of PS(t), P ^S (t-1), P ^S (t-2), Q ^S (t), Q ^S (t-1), Q ^S (t-2)	Q(t+1)

Note:

1. P and Q mean observed rainfall and inflow, respectively. The superscript S means standardization, i.e., P^S and Q^S mean standardized rainfall and inflow, respectively.
2. Time step is defined as ten days.

$$Q_{ij}^S(t) = \frac{Q_{ij}(t) - \overline{Q_j(t)}}{\sigma_{Q_j}(t)} \quad (2)$$

where $P_{ij}^S(t)$ is the standardized rainfall of the j th 10-days in i th year; $P_{ij}(t)$, the observed rainfall of the j th 10-days in i th year; $P_j(t)$, the average of $P_{ij}(t)$; $\sigma_{P_j}(t)$, the standard deviation of $P_{ij}(t)$; $Q_{ij}^S(t)$, the standardized inflow of the j th 10-days; $Q_{ij}(t)$, the observed inflow of the j th 10-days in i th year; $Q_j(t)$, the average of $Q_{ij}(t)$; and $\sigma_{Q_j}(t)$ is the standard deviation of $Q_{ij}(t)$.

The GP modeling is implemented in MATLAB, a widely used programming environment available for a large number of computer platforms. GPLAB is a genetic programming toolbox of MATLAB, which was developed by Silva (2004) at Evolutionary and Complex Systems Group of University of Coimbra, Portugal. The architecture of GPLAB follows a highly modular and parameterized structure, which different users may use at various levels of depth and insight. There are a number of parameters in GPLAB users could set. Table 2 shows the parameters that are set by the authors or selected from the default values in GPLAB.

Root mean square error (RMSE) and

Table 2. Parameters of GPLAB in this study

Population size	60
Generation	120
Functions	+, -, *, /, SIN, COS
Terminals	1, random number from 0 to 1, rainfall variables, inflow variables
Operators	Reproduction, crossover and mutation
Maximum tree depth	6
Other parameters	Default values in GPLAB

coefficient of correlation (CC) are selected as the fitness functions in this study and expressed as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^N (Q_f(t) - Q_o(t))^2}{N}} \quad (3)$$

$$CC = \frac{\sum_{t=1}^N [Q_f(t+1) - \overline{Q_f(t+1)}][Q_o(t+1) - \overline{Q_o(t+1)}]}{\sqrt{\sum_{t=1}^N [Q_f(t+1) - \overline{Q_f(t+1)}]^2} \sqrt{\sum_{t=1}^N [Q_o(t+1) - \overline{Q_o(t+1)}]^2}} \quad (4)$$

where $Q_f(t+1)$ is the forecasted inflow at time $t+1$; $Q_f(t+1)$, the average of $Q_f(t+1)$ over N ; $Q_o(t+1)$, the observed inflow at time $t+1$; and $Q_o(t+1)$ is the average of $Q_o(t+1)$ over N ; N is the number of data.

IV. Result and Discussion

1. Table 3 shows the best optimal results of the GP modeling for the six cases in Table 1. For example, for the model of case 4 (GP4) the input is only the standardized rainfall at time t ; the optimal RMSE on the training and testing data sets are 44.01 and 45.37, respectively; the optimal coefficients of correlation on the training and testing data sets are 0.56 and 0.58, respectively. To further investigate if only the standardized rainfall at time t (the variable optimized in case 4) and a random number from 0 to 1 as inputs for GP modeling could have better performance than GP4, the other runs are performed and the optimal result of the model (GP4a) is listed in case 4a of

Table 3. The best optimal results of the GP modeling in six cases

GP Eq.		Case 1	Case 2	Case 3	Case 4	Case 4a	Case 5	Case 6
		$0.368P(t) + 13.578$	$0.635Q(t) + 10.32$	Eq. 5	$0.462P^S(t) - 0.048$	$0.46 P^S(t) - 0.25$	$0.499Q^S(t)$	Eq. 6
RMSE	Training	46.6	48.6	46.4	44.0	35.5	46.0	47.4
	Testing	56.7	59.6	57.0	45.4	37.3	51.9	47.8
CC	Training	0.53	0.48	0.48	0.56	0.55	0.52	0.57
	Testing	0.56	0.51	0.51	0.58	0.58	0.54	0.58

Note:

- Eq. 5: $0.407 \left[P(t) + \frac{P(t-1)}{P(t)+0.302} + 0.423Q(t-1) + 1.963 \right]$
- Eq. 6: $\{P^S(t) + \sin[\sin(Q^S(t-1))]\} \cdot \sin\{\cos[\cos(Q^S(t))]\} \cdot \cos\{\cos[\sin(P^S(t) - Q^S(t-1))]\}$

Table 3. It can be found that the performance of GP4a is better than GP4 model after GP tuning. Besides, from Table 3 it can be also found that the RMSE values in cases 4, 5, and 6 are much smaller than those in cases 1, 2, and 3. The result reveals that standardization of input data can improve the performance of the proposed GP. This could be because the standardized rainfall or streamflow used as inputs provides the flow anomaly (e.g., below or above average flow) and supports information relevant to the dry or wet in climate scenario; as a result, they give better 10-day forecasts than that of non-standardized flow as inputs.

- The results in Table 3 also show that GP4 or GP4a models (only including the rainfall data) have better performance than the models of cases 5 and 6 (only including the inflow data) in both training and testing phases. Since the quick response from heavy precipitation can be in a matter of few hours for the catchment of size 763.4 km², so the rainfall information is important in forecasting the 10-day ahead reservoir inflow.
- Figs. 6(a) and (b) display the comparison between forecasted and observed inflow time series for GP4a model in Table 3. Since the performance of forecasting is our focus, we pay more attention to the results of testing. From Fig. 6 (b) it can be seen GP performs well on the testing data set in the forecasting of low inflows. However, most of

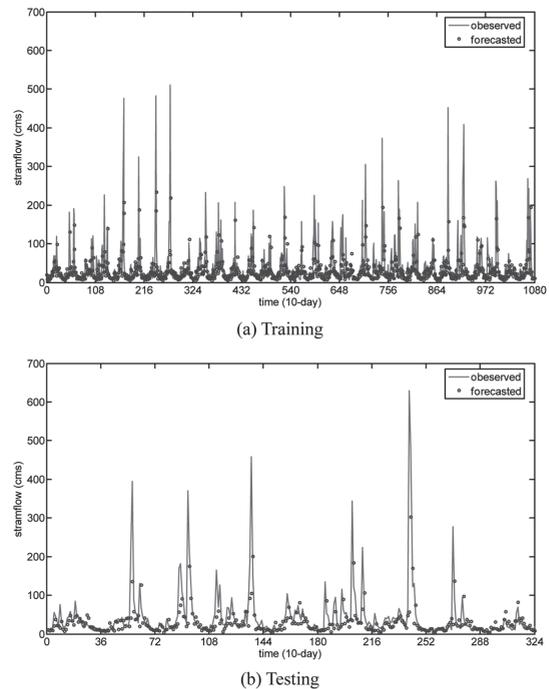


Fig. 6. Comparison between forecasted and observed inflow time series for case 4a.

high inflow forecasting, especially the streamflow more than 150 m³/s, shows underestimation. This is mainly because there are only a few typhoon events with limited observed high flow data for modeling.

- Since GP4a model, the best model in Table 3, is linear, four linear models, ARMAX(1, 2, 1), AR(1), linear regression 1 (LR1), and linear regression 2 (LR2) models, are used to compare with it. The performances of the four models with

Table 4. Performance of AR(1), ARMAX, and two linear regression models

Models	Forecasting	RMSE		CC	
		Training	Testing	Training	Testing
AR(1)	Q(t+1)	47.2	58.9	0.42	0.32
ARMAX(1, 2, 1)	Q(t+1)	46.8	59.2	0.55	0.55
LR1	Q(t+1)	54.5	64.5	0.50	0.53
LR2	Q(t+1)	44.1	41.1	0.48	0.44

Note:

1. AR(1) model: $Q^S(t+1) = 0.6186 \cdot Q^S(t+1) + e(t+1)$, where $e(t+1)$ is an error term at time $t+1$.
2. ARMAX(1, 2, 1) model: $Q^S(t+1) = 0.7257 \cdot Q^S(t+1) + 0.1866 \cdot P^S(t) - 0.06515 \cdot P^S(t-1) + e(t+1) - 0.2239 \cdot e(t)$.
3. Linear regression 1 (LR1): $Q^S(t+1) = 0.745 Q^S(t) + 0.036$
4. Linear regression 2 (LR2): $Q^S(t+1) = 0.099 P^S(t) + 0.008$
5. The time series for input were standardized before modeling for AR(1), ARMAX and linear regression models.
6. A time step is defined as ten days, so “t+1” means ten days ahead.

Table 5. Percentage of improvement for the RMSE values on training and testing data of GP4a model, compared with ARMAX(1, 2, 1), AR(1) and linear regression 2 (LR2) models

Models	ARMAX (1, 2, 1)	GP4a	AR(1)	GP4a	LR2	GP4a
RMSE on training data	46.8	35.5	47.2	35.5	44.1	35.5
Percentage of improvement	-	24.1%	-	24.8%	-	19.5%
RMSE on testing data	59.2	37.3	58.9	37.3	41.1	37.3
Percentage of improvement	-	37.0%	-	36.7%	-	9.2%

Note: GP4a means the model of case 4a in Table 3; LR2 means linear regression 2 model in Table 4.

standardized input data are tabulated in Table 4. The comparison between GP4a and the three models with better performance is shown in Table 5. Again, since the performance of forecasting is our focus, we pay more attention to the results of testing. Compared with the ARMAX(1, 2, 1) model, the percentage of improvement in terms of the RMSE values on testing data for GP4a model is 37.0%. Compared with the AR(1) model, the percentage of improvement in terms of the RMSE values on testing data for GP4a model is 36.7%. Compared with the LR2 model, the percentage of improvement in terms of the RMSE values on testing data for GP4a model is 9.2%. In addition to the admirable effectiveness of GP, it can also be found that the coefficient of correlation on testing data for GP4a model is superior to that for other models.

5. Two of the GP parameters, maximum tree depth and input variables, are determined after a number of trials in this study. The settings of the parameters,

usually relying on the features of data, could be regarded as a tradeoff in terms of the number or amount of the parameters. In terms of maximum tree depth, too small value of tree depth could construct a simpler architecture of GP tree and might not meet the demand for accuracy, whereas too large tree depth could generate a complicated GP tree and might reduce the generalization ability due to over-fitting. The selection of inputs, taken as another example, also has influence on optimal results of times series forecasting. If too many possible time-dependent variables are input to GP model, the optimal results could be hard to converge since the best combination of inputs would be rarely achieved.

V. Conclusion

1. GP has the advantages of its ability to learn relationships hidden in data and express them automatically in a mathematical manner. That is,

GP finds solutions to an optimization problem without continuous interruption of the designer. However, unless the problem and its solutions can be clearly verified, it is still hard to confirm if the best-so-far solution is a global optimum.

2. The present study shows a successful application of GP on the seasonal streamflow forecasting. The forecasting of 10-day reservoir inflows reveals the excellent effectiveness of GP, and standardization is beneficial to modeling for seasonal time series.

Reference

1. Babovic, V., Emergence, Evolution, Intelligence. Hydroinformatics, Balkema, Rotterdam, 1996.
2. Babovic, V. and Bojkov, V. H., "Runoff Modeling with Genetic Programming and Artificial Neural Networks", D2K Technical Report D2K TR 0401-1, DHI Water & Environment, 2001.
3. Babovic, V., "Data Mining in Hydrology". Hydrological Processes, 19, pp. 1511-1515, 2005.
4. Chang, F. J., Chen, Y. C., A Counterpropagation Fuzzy-Neural Network Modeling Approach to Real-time Streamflow Prediction. Journal of Hydrology 245, pp153-164, 2001.
5. Chang, L. C., Chang, F. J., Chiang, Y. M., A Two-step Ahead Recurrent Neural Network for Streamflow Forecasting. Hydrological Processes 18, 81-92, 2003.
6. Chen, C. S., Chung, Y. D., and Fang, W. C., "River Routing Using Genetic Programming", Journal of Taiwan Water Conservancy, 53 (4), 2005. (in Chinese)
7. Chen, L., Jian, B. K., and Tsai, J. S., "Study on Streamflow Forecasting Using Evolutional Algorithm", 2002 Conference on Water Resources Management, Taiwan, 2002. (in Chinese)
8. Chen Y. H., Chang, F. J., Evolutionary Artificial Neural Networks for Hydrological Systems Forecasting, Journal of Hydrology, 367 (1-2), 125-137, 2009.
9. Drecourt, J.-P., "Application of Neural Networks and Genetic Programming to Rainfall Runoff Modeling", D2K TR 0699-1, DHI Water & Environment, 1999.
10. Drecourt, J.-P. and Madsen, H., "Role of Domain Knowledge in Data-driven Modeling", 4th DHI Software Conference, Scanticon Conference Centre, Denmark, 2001.
11. Koza, J. R., Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992.
12. Holland, J. H., Adaptation in Natural and Artificial Systems, second ed., Massachusetts Institute of Technology, Cambridge, 1975.
13. Madar, J., Abonyi, J., Szeifert, F., "Genetic Programming for the Identification of Nonlinear Input-output Models", <http://www.fmt.vein.hu/softcomp/gp/ie049626e.pdf>, 2005.
14. Madsen, H., Butts, M. B., Khu, S. T., and Liang, S. Y., "Data Assimilation in Rainfall-runoff Forecasting", Hydroinformatics 2000, 4th International Conference on Hydroinformatics, Cedar Rapids, Iowa, USA, 2000.
15. Moradkhani, H. S., Sorooshian, S., Gupta, H., and Houser, P., "Dual State-Parameter estimation of hydrological models using ensemble Kalman Filter", Advance Water Resources, 28, pp. 135-147, 2004.
16. Muni, D. P. and Pal, N. R., "A Novel Approach to Design Classifiers Using Genetic Programming", IEEE Transaction on Evolutionary Computation, 8 (2), pp. 183-196, 2004.
17. Muttil, N. and Liang, S. Y., "Improving Runoff Forecasting by Input Variable Selection in Genetic Programming", World Water Congress 2001, No. 111, 2001.
18. Silva, S., "GPLAB - A Genetic Programming Toolbox for MATLAB, <http://gplab.sourceforge.net>, 2004.
19. The GP Tutorial. <http://www.geneticprogramming.com/Tutorial/index.html>.

收稿日期：民國 98 年 02 月 17 日

修正日期：民國 98 年 04 月 29 日

接受日期：民國 98 年 05 月 08 日