Australian Journal of Basic and Applied Sciences, 4(11): 5663-5667, 2010
ISSN 1991-8178

Review of Genetic Programing in Water Resource Engineering

[1]Omolbani Mohamad Rezapour and [2]Lee Teang Shui, [3]Amir Ahmad Dehghani

[1,2]Faculty of Engineering, University Putra Malaysia
[3]Grogan University of Agriculture Science and Natural Resources

Abstract: Correct estimation of sediment volume carried by a river is very important for many water resources projects. Empirical relations such as sediment rating curves are often applied to determine the average relationship between discharge and suspended sediment load. This type of models generally underestimates or overestimates the amount of sediment. During recent decades, some black box models based on artificial neural networks have been developed to overcome this problem.
GP has been applied to a wide range of problems in artificial intelligence, engineering and science applications, industrial, and mechanical models. The main purpose of this paper is literature review of Genetic Programming for suspended sediment estimation.

Key words: Genetic programming, Suspended Sediment, Model estimation.

INTRODUCTION

Right estimation of sediment volume carried by a river is very important for many water resources projects. The prediction of river sediment load also constitutes an important matter in hydraulic and healthful engineering. It is well known fact that all reservoirs are designed to a volume known as''the dead storage'' to fit the sediment income that will collect over a specified period called the economic life. The underestimation of sediment yield results in insufficient reservoir capacities while the overestimation will lead to over-capacity reservoirs.

Only the appropriate reservoir design and operation is sufficient to justify every effort to determine sediment yield accurately, but in sanitary engineering the prediction of river sediment load has an additional significance, especially if the particles also transport pollutants. On the other hand, the sediment can aggrades channel beds with excess sand and gravel for tens to hundreds of kilometers downstream. Such aggradations promote lateral migration of channels and may cause serious flooding during rainstorms, due to loss of channel capacity necessary to convey floodwaters. The assessment of the volume of sediment carries significance also for the flooding problem.

A number of attempts have been made to relate the amount of sediment transported by a river to flow conditions such as discharge, velocity and shear stress. However, none of the quations derived have received universal acceptance. Usually, either the weight or the concentration of sediment is related to the discharge. These two forms are often used interchangeably.

Empirical relations such as sediment rating curves are often applied to determine the average relationship between discharge and suspended sediment load. This type of models generally underestimates or overestimates the amount of sediment. During recent decades, some black box models based on artificial neural networks have been developed to overcome this problem. But these type of models are implicit that can not be simply used by other investigators. Therefore it is still necessary to develop an expressed model for the discharge-sediment relationship.

Overview of Genetic Programming:

In this section, a brief overview of the GP and Gene Expression Programming (GEP) is given for motivation. GP is first proposed by Koza (1992). It is a generalization of genetic algorithms (GAs) (Goldberg, 1989). GP starts with an initial population of randomly generated computer programs composed of functions and terminals appropriate to the problem domain. The functions may be standard arithmetic operations, standard programming operations, standard mathematical functions, logical functions, or domain-specific functions.

Corresponding Author: Omolbani Mohamad Rezapour, University Putra Malaysia
E-mail; nrezaii2000@yahoo.com
00601-73601749

Depending on the particular problem, the computer program may be boolean-valued, integer-valued, real-valued, complex-valued, vector-valued, symbolic-valued, or multiple-valued. GEP is, like GAs and GP, a genetic algorithm as it uses populations of individuals, selects them according to fitness, and introduces genetic variation using one or more genetic operators (Ferreira, 2006). GEP is an extension to GP that evolves computer programs of different sizes and shapes encoded in linear chromosomes of fixed length. One strength of the GEP approach is that the creation of genetic diversity is extremely simplified as genetic operators work at the chromosome level. Another strength of GEP consists of its unique, multigenic nature which allows the evolution of more complex programs composed of several subprograms. As a result GEP surpasses the old GP system in 100-10,000 times (Ferreira, 2001a; Ferreira, 2001b).The fundamental difference between GAs, GP and GEP is due to the nature of the individuals: in GAs the individuals are linear strings of fixed length (chromosomes); in GP the individuals are nonlinear entities of different sizes and shapes (parse trees); and in GEP the individuals are encoded as linear strings of fixed length (the genome or chromosomes) which are afterwards expressed as nonlinear entities of different sizes and shapes (i.e., simple diagram representations or expression trees). There are five major preliminary steps for solving a problem by using GP; (i) ''the set of terminals''; the independent variables of the problem, the state variables of the system and the functions with no arguments (ii) ''the set of functions''; arithmetic operations, testing functions (such as IF and CASE statements) and boolean functions (iii) ''the fitness measure'' which identifies the way of evaluating how good a given program solves a particular problem (iv) ''control parameters''; the values of the numerical parameters and qualitative variables for controlling the run, and (v) ''stop condition''; the criterion for designating a result and terminating a run.. The process begins with the random generation of the chromosomes of a certain number of individuals (the initial population). Then these chromosomes are expressed and the fitness of each individual is evaluated against a set of fitness cases (also called selection environment which, in fact, is the input to a problem). The individuals are then selected according to their fitness (their performance in that particular environment) to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment, selection, and reproduction with modification. The process is repeated for a certain number of generations or until a good solution has been found (Ferreira, 2006).

GP, a branch of the genetic algorithm (GA) Holland(1975), is a method for learning the most ”fit” computer programs by means of artificial evolution. In other words, its behavior forms a metaphor of the processes of evolution in nature. GP, similar to GA, initializes a population that compounds the random members known as chromosomes (individual). Afterward, fitness of each chromosome is evaluated with respect to a target value. The principle of Darwinian natural selection is used to select and reproduce “fitter” programs. The main difference between GP and GA is the representation of the chromosomes and final solution. A genetic algorithm creates equal length strings of numbers (chromosomes) in the form of binary or real which represent the solution. However, GP creates equal or unequal length computer programs [a symbolic expression that consists of variables (terminal) and several mathematical operators (function)] in the LISP language or other computer languages as the solution.

Therefore, unlike GA, in GP there is no need to define the form of the objective function a priori. In fact, it is the GP that determines not only the coefficients and parameters of the objective function, but also and more importantly, the form of the objective function itself. This is one of the advantages of GP as compared to GA. Although research on GP techniques dates back to the 1960s and 1970s, GP emerged as a distinct discipline presented by Koza (1990). In brief, five stages are employed in GP to solve a problem.

1. Initialize a population of programs. Create a population of randomly generated programs in LISP language.
2. Selection. The randomly generated programs with the higher fitness will “win” and must be copied to the next generation. There are several different types of selection used in GP such as roulette-wheel selection, tournament, and ranking.
3. Transform the winner programs. The two winner programs (GP solution) are then copied and transformed probabilistically by: exchanging parts of the winner programs with each other to create two new programs (crossover) and randomly changing each of the winner programs to create new program. A function can only replace a function, a terminal can only replace a terminal, and an entire subtree can replace another subtree (mutation).
4. Replace the “loser” programs. Replace the “loser” programs in the population with the transformed “winner” programs. The winners of the selection remain in the population unchanged.
5. Iterate until convergence. Repeat steps 2-4 until a program is developed that predicts the behavior properly.

McBean and Al-Nassri (1988) examined this issue and concluded that the practice of using sediment load vs. discharge is misleading because the goodness of fit implied by this relationship is spurious. Instead they recommended that the regression link be established.

Over the years, researchers have proposed several rating curves to determine the average relationship between discharge and suspended sediment load (Thomas, 1985; Asselman, 2000; Picoet *et al*., 2001; Overleir, 2004; Crowder *et al*., 2007).

The physically based models are based on the simplified partial differential equations of flow and sediment flux as well as on some unrealistic simplifying assumptions for flow and empirical relationships for erosive effects of rainfall and flow. Examples of such models are presented by Wicks and Bathurst (1996), Kothyari *et al*. (1997), Refsgaard (1997), and others. They are highly sophisticated and complex models that have the advantages of having components that correspond to physical processes and of being theoretically capable of taking into account the spatial variation of catchment properties as well as uneven distribution of precipitation and evapotranspiration. The sophistication and complexity of the model should, however, be keyed to utilizable information about the catchment characteristics and density and frequency of the available input data. Especially because the real spatial distribution of precipitation is not presently measurable for much of the world, process-oriented distributed models offer no practically significant advantage over lumped models and have many practical disadvantages (Guldal and Muftuoglu, 2001).

GP has been applied to a wide range of problems in artificial intelligence, engineering and science applications, industrial, and mechanical models. GP can be successfully applied to areas, where (i) the interrelationships among the relevant variables are poorly understood (or where it is suspected that the current understanding may well be wrong), (ii) finding the size and shape of the ultimate solution is hard and a major part of the problem, (iii) conventional mathematical analysis does not, or cannot, provide analytical solutions, (iv) an approximate solution is acceptable (or is the only result that is ever likely to be obtained), (v) small improvements in performance are routinely measured (or easily measurable) and highly prized, (vi) there is a large amount of data, in computer readable form, that requires examination, classification, and integration (such as molecular biology for protein and DNA sequences, astronomical data, satellite observation data, financial data, marketing transaction data, or data on the World Wide Web) (Banzhaf *et al*., 1998).

It was observed that only a few studies existed in the literature related to the use of GP in the field of water resources engineering. Cousin and Savic (1997), Savic *et al*. (1999), Drecourt (1999), Whigham and Crapper (1999, 2001), Babovic and Keijzer (2002) applied GP to rainfall-runoff modeling. Babovic *et al*. (2001) applied GP to sedimentary particle settling velocity equations. Harris *et al*. (2003) studied on velocity predictions in compound channels with vegetated floodplains using GP. Dorado *et al*. (2003) studied on prediction and modeling of the rainfall-runoff transformation of a typical urban basin using artificial neural networks (ANNs) and GP. Giustolisi (2004) determined Chezy resistance coefficient in corrugated channels by using GP. Rabunal *et al*. (2007) determined the unit hydrograph of a typical urban basin using GP. Only two studies were observed for sediment modeling using GP approach; Babovic (2000) used experimental flume data utilized by Zyserman and Fredsoe (1994) and expressed a new formulation for bed concentration of suspended sediment. Kizhisseri *et al*. (2005) used GP methodology to explore a better correlation between the temporal pattern of fluid field and sediment transport by utilizing two datasets; one from numerical model results and other from Sandy Duck field data.

Aytek ali (2007) their study proposed genetic programming (GP) as a new approach for the expressed formulation of daily suspended sediment-discharge relationship. They compared expressed models obtained using the GP with rating curves and multi-linear regression techniques in suspended sediment load estimation. They were used the daily streamflow and suspended sediment data from two stations on Tongue River in Montana as case studies. Their results indicate that the proposed GP formulation performs quite well compared to sediment rating curves and multi-linear regression models and is quite practical for use.

Johari, (2006). They developed genetic model for soil analysing. They use terminal set that consists of initial void ratio, initial gravimetric water content, logarithm of suction normalized with respect to atmospheric air pressure, clay content, and silt content the input of GP model. The output terminal set consisted of the gravimetric water content corresponding to the assigned input suction. The function set includes operators such as plus, minus, product, division, and power. Their results from pressure plate tests carried out on clay, silty clay, sandy loam, and loam compiled in the SoilVision software were adopted as a database for developing and validating the genetic model. For this purpose, they employed after data digitization, GP software (GPLAB) provided by MATLAB for the analysis. Furthermore, GP simulations were compared with the experimental results as well as the models proposed by other investigators. Their comparison indicated superior performance of the proposed model for predicting the SWCC.

Guven (2008) their study was that genetic programming _GP_ as a new tool for prediction of local scour downstream of grade-control structures. Their objective of their study were to provide an alternative formulation to conventional regression based equations and verify the superiority of GP over regression analysis. The training and testing patterns of the proposed GP formulation are based on well established and widely dispersed experimental results from the literature. Linear and nonlinear regression-based equations were derived throughout regression analysis on dimensionless parameters obtained from dimensional analysis. The GP-based formulation results were compared with experimental results and other equations and found to be more accurate.)

Soon Thiam Khu(2001) they employed, GP functions as an error updating scheme to complement a rainfall-runoff model, MIKE 11/NAM. Hourly runoff forecasts of different updating intervals are performed for forecast horizons of up to nine hours. Their results showed that the proposed updating scheme is able to predict the runoff quite accurately for all updating intervals considered and particularly for updating intervals not exceeding the time of concentration of the catchment. Their results were also compared with those of an earlier study, by the World Meteorological Organization, in which autoregression and Kalman filter were used as the updating methods. Comparisons showed that GP is a better updating tool for real-time flow forecasting. Another important finding from their study was that nondimensionalizing the variables enhances the symbolic regression process significantly.

(journal of hydrology-1)their study emphasized the inclusion of sea surface temperature (SST) in addition to the spatio-temporal rainfall distribution via the Next Generation Radar (NEXRAD), meteorological data via local weather stations, and historical stream data via USGS gage stations to collectively forecast discharges in a semi-arid watershed in south Texas. Two types of artificial intelligence models, including genetic programming (GP) and neural network (NN) models, were employed comparatively. Four numerical evaluators were used to evaluate the validity of a suite of forecasting models. Research findings indicate that GP-derived streamflow forecasting models were generally favored in the assessment in which both SST and meteorological data significantly improve the accuracy of forecasting. Among several scenarios, NEXRAD rainfall data were proven its most effectiveness for a 3-day forecast, and SST Gulf-to-Atlantic index shows larger impacts than the SST Gulf-to-Pacific index on the streamflow forecasts. The most forward looking GP-derived models can even perform a 30-day streamflow forecast ahead of time with an r-square of 0.84 and RMS error 5.4 in their study.

REFERENCES

Asselman, N.E.M., 2000. Fitting and interpretation of sediment rating curves. J. Hydrol., 234: 228-248.

Aytek, A., O¨zgu¨r Kis_i, 2008. A genetic programming approach to suspended sediment modelling. Journal of Hydrology, 351: 288- 298.

Babovic, V., 2000. Data mining and knowledge discovery in sediment transport. Comput.-Aided Civ. Infrastruct. Eng., 15(5): 383-389.

Babovic, V., M. Keijzer, 2002. Declarative and preferential bias in GP-based scientific discovery. Genet. Program. Evol. Mach. 3(1).

Babovic, V., M. Keijzer, D.R. Aguilera, J. Harrington, 2001. Automatic Discovery of Settling Velocity Equations. D2K Technical Report, D2K-0201-1.

Banzhaf, W., P. Nordin, R.E. Keller, F.D. Francone, 1998. Genetic Programming. Morgan Kaufmann, San Francisco, CA.

Cousin, N., D.A. Savic, 1997. A rainfall-runoff model using genetic programming. Centre for Systems and Control Engineering, Report No. 97/03, School of Engineering, University of Exeter, Exeter, United Kingdom, pp: 70.

Dorado, J., J.R. Rabunal, A. Pazos, D. Rivero, A. Santos, J. Puertas, 2003. Prediction and modelling of the rainfall-runoff transformation of a typical urban basin using ANN and GP. Appl. Artif. Intell., 17: 329-343.

Drecourt, J.P., 1999. Application of neural networks and genetic programming to rainfall-runoff modeling. D2K Technical Report 0699-1-1, Danish Hydraulic Institute, Denmark.

Ferguson, R.I., 1986. River loads underestimated by rating curves. Water Resour. Res., 22(1): 74-76.

Ferreira, C., 2001a. Gene expression programming in problem solving. In: 6th Online World Conference on Soft Computing in Industrial Applications (invited tutorial).

Ferreira, C., 2001b. Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst., 13(2): 87-129.

Ferreira, C., 2006. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. Springer, Berling Heidelberg New York, pp: 478.

Giustolisi, O., 2004. Using genetic programming to determine Chezy resistance coefficient in corrugated channels. J. Hydroinform, 157-173.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass.

Guven, A., M. ASCE, M. Guna, 2008. Genetic Programming Approach for Prediction of Local Scour Downstream of Hydraulic Structures. Journal of Irrigation and Drainage engineering. ASCE., 134(2): 241.

Harris, E.L., V. Babovic, R.A. Falconer, 2003. Velocity predictions in compound channels with vegetated floodplains using genetic programming. Int. J. River Basin Manage, 1(2): 117-123.

Holland, J.H., 1975. Adaptation in natural and artificial system, University of Michigan Press, Ann Arbor, Mich.

Johari, A., G. Habibagahi, A. Ghahramani, 2006. Prediction of Soil-Water Characteristic Curve Using Genetic Programming.J. Geotechnical and Geoenvironmental engineering. ASCE., 132(5): 661.

Kizhisseri, A.S., D. Simmonds, Y. Rafiq, M. Borthwick, 2005. An Evolutionary computation approach to sediment transport modeling. In: Fifth International Conference on Coastal Dynamics, Barcelona, Spain.

Kothyari, U.C., A.K. Tiwari, R. Singh, 1997. Estimation of temporal variation of sediment yield from small catchments through the kinematic method. J. Hydrol., Amsterdam, 203: 39-57.

Koza, J.R., 1990. A paradigm for genetically breeding populations of computer programs to solve problems, Computer Science Dept., Stanford Univ., Margaret Jacks Hall, Stanford, Calif.

Koza, J.R., 1992. Genetic Programming: On the Programming of Computers by means of Natural Selection. The MIT Press, Cambridge, MA.

McBean, E.A., S. Al-Nassri, 1988. Uncertainty in suspended sediment transport curves. J. Hydr. Eng. ASCE, 114(1): 63-74.

Makkeasorn, M., N.B. Chang, X. Zhou, 2008. Short-term streamflow forecasting with global climate change implications - A comparative study between genetic programming and neural network models. Journal of Hydrology, 352: 336-354.

Overleir, A.P., 2004. Accounting for heteroscedasticity in rating curve estimates. J. Hydrol., 292: 173-181.

Picoet, C., B. Hingray, J.C. Olivry, 2001. Empirical and conceptual modeling of the suspended sediment dynamics in large tropical African River: The Upper Niger River Basin. J. Hydrol., 250: 19-39.

Rabunal, J.R., J. Puertas, J. Suarez, D. Rivero, 2007. Determination of the unit hydrograph of a typical urban basin using genetic programming and artificial neural networks. Hydrol. Process, 21: 476-485.

Refsgaard, J.C., 1997. Parameterization, calibration and validation of distributed hydrological models. J. Hydrol., Amsterdam, 198: 69-97.

Thiam Khu, S., S. Liong, W. Babovic, H. Madsen and N. Muttil, 2001. Genetic programming and its application in real-time runoff forecasting. Journal of American water resources association, 37(2): 439-451.

Thomas, R.B., 1985. Estimating total suspended sediment yield with probability sampling. Water Resour. Res., 21: 1381-1388.

Wicks, J.M., J.C. Bathurst, 1996. SHESED: a physically based, distributed erosion and sediment yield component for the SHE hydrological modeling system. J. Hydrol., 175: 213-238.

Whigham, P.A., P.F. Crapper, 1999. Time series modeling using genetic programming: an application to rainfall-runoff models. In: Spector, L. *et al*. (Eds.), Advances in Genetic Programming. The MIT Press, Cambridge, MA, pp: 89-104.

Whigham, P.A., P.F. Crapper, 2001. Modeling rainfall-runoff using genetic programming. Math. Comput. Model., 33: 707-721.

Zyserman, J.A., J. Fredsoe, 1994. Data analysis of bed concentration of suspended sediment. J. Hydr. Eng., ASCE, 120(9): 1021-1042.