Linear genetic programming for time-series modelling of daily flow rate

Aytac Guven

Civil Engineering Department, Gaziantep University, 27310 Gaziantep, Turkey.
e-mail: aguven@gantep.edu.tr

In this study linear genetic programming (LGP), which is a variant of Genetic Programming, and two versions of Neural Networks (NNs) are used in predicting time-series of daily flow rates at a station on Schuylkill River at Berne, PA, USA. Daily flow rate at present is being predicted based on different time-series scenarios. For this purpose, various LGP and NN models are calibrated with training sets and validated by testing sets. Additionally, the robustness of the proposed LGP and NN models are evaluated by application data, which are used neither in training nor at testing stage. The results showed that both techniques predicted the flow rate data in quite good agreement with the observed ones, and the predictions of LGP and NN are challenging. The performance of LGP, which was moderately better than NN, is very promising and hence supports the use of LGP in predicting of river flow data.

1. Introduction

The prediction of discharge in rivers plays an important role in river hydrology and water resources management. There are several crucial points for understanding the key statistical characteristics of available discharge data, some of which are: hydro-geological risk prevention, efficient operations of storage reservoirs for hydro-electric or other purposes, and design of many hydro-geological works (Bordignon and Lisi 2000). It is generally accepted that river flow processes, especially daily discharges, are seasonal and nonlinear, since the processes generally pronounce seasonal means, variances, and the underlying mechanisms of streamflow generation are likely to be quite different during low, medium, and high flow periods, especially when extreme events occur (Wang *et al* 2006).

Several linear and nonlinear methods have been applied in the prediction of discharge in rivers and successful results have been reported. These studies have focused on the prediction of discharge based on *stage-discharge*, *rainfall-discharge* or *time-series of discharge* relationships, using either conventional methods (Wang *et al* 2004; Habib and Meselhe 2006; Baiamonte and Ferro 2007; Jain 2008) or new so-called 'soft computing techniques' such as Neural Networks (NNs), Genetic Algorithms (GAs), Genetic Programming (GP), Fuzzy Logic (FL), and Machine Learning (MA) (Maier and Dandy 2000; Kisi 2004; Lopes and Weinert 2004; Cigizoglu and Kisi 2005; Kisi and Cigizoglu 2007; Aytek and Alp 2008; Guven and Gunal 2008; Guven *et al* 2008; Tayfur and Moramarco 2008).

For the last ten years, GAs and GP have been pronounced as alternative and robust methods in the prediction of water engineering data (Drecourt 1999; Whigham and Crapper 2001; Muttil and Liong 2004; Kalra and Deo 2007; Charhate *et al* 2007, 2008; Singh *et al* 2007; Aytek *et al* 2008; Deo *et al* 2008; Gaur and Deo 2008; Guven *et al* 2008; Jain and Deo 2008; Kalra *et al* 2008; Ustoorikar and Deo 2008). Linear genetic programmingc (LGP), which is an extension of GP is under consideration in this study. Although a few studies on GP application on hydrologic data exist in the

Keywords. Genetic programming; neural networks; daily flows; flow forecasting.

literature, no studies have been found about the application of LGP in hydrologic data. In this sense, the present study can be considered as a pioneering study presenting the usage of LGP in the prediction of hydrologic data.

Referring to the embedding theorem of Takens (1981), the time-series modelling of daily discharge can be formulated as:

$$Q(t) = f\{Q(t-n),\ Q(t-2\cdot n),\ldots,\ Q(t-(d-1)\cdot n)\}, \tag{1}$$

where n is the constant time delay between samples, d is the embedded dimension and f is a function. The objective is to find an optimal analytical model that can explain the behaviour of dynamical river flow systems. There are several conventional statistical methods to cope with this type of problem, most of them being based on ARMA-derived (Auto-Regressive Moving Average) methods. Alternatively, several heuristic models have been proposed (Bordignon and Lisi 2000; Cigizoglu and Kisi 2000; Kisi 2004; Lopes and Weinert 2004; Tayfur *et al* 2007; Firat 2008; Tayfur and Moramarco 2008).

Evolutionary computational methods have been successfully applied in time-series modelling of flow discharge problems, with limited number of examples. Namely, Lopes and Weinert (2004) used Gene-Expression Programming in modelling monthly time series of unregulated Rio Grande river flow at Furnas Dam in Brazil; Tayfur and Moramarco (2008) predicted hourly-based flow discharge hydrographs from level data by using GAs; Preis and Otsfeld (2008) presented a coupled model tree-genetic algorithm scheme for predicting flow and water quality constituents in watersheds.

This study proposes to employ an emerging strong evolutionary computational technique, LGP in predicting daily time series of river flow data. Also, multilayer perceptron NNs empowered by GA, and generalized regression NNs were proposed for time-series modelling of the same discharge data. Different heuristic scenarios were developed and accordingly, LGP and NN models were developed based on these scenarios. The performance of each model was compared based on the well-known statistical performance measures. The results were tabulated and illustrated in scatter and time-series diagrams.

2. Linear genetic programming (LGP)

GP technique is an automatic, computerized creation of computer programs in order to solve a selected problem using Darwinian natural selection. LGP, a linear variant of GP, uses a specific linear representation of computer programs. The name 'linear' refers to the structure of the (imperative) program representation, and does not stand for functional genetic programs that are restricted to a linear list of nodes only. On the contrary, genetic programs normally represent highly nonlinear solutions in this meaning (Brameier 2004). The main characteristic of LGP in comparison to conventional tree-based GP is that the expressions of a functional programming language (like LISP) are substituted by programs of an imperative language (like C or C++).

The main characteristics of LGP are the graph-based data flow that results from a multiple usage of indexed variables (registers, $r[i]$) and evolving programs in a low-level language, in which the solutions are directly manipulated as binary machine codes and executed without using an interpreter (Banzhaf *et al* 1998; Bramier 2004). In this way the computer program can be evolved very quickly (Bhattacharya *et al* 2001; Brameier and Banzhaf 2001; Foster 2001).

Each individual program in LGP is represented by a variable-length sequence of simple C language instructions. These instructions operate on one or more registers ($r[i]$) or constants (c) from predefined sets (Bramier 2003; Oltean and Groşan 2003). An example of LGP program can be:

```
Void LGP
double v[3];
{
r[0]+ = v[0];
r[1]= r[0] - v[2];
r[0]/ = v[1];
r[2]= - v[3];
r[0]* = 2.53;
r[0]/ = r[0]* r[2];
}
```

where $v[i]$ represents the input and output variables used in LGP modelling.

The *function set* of the system can be composed of arithmetic operations $(+, -, /, *)$, conditional branches (if $v[i] \le v[k]$), and function calls ($f \in \{e^x,\ x$, sin, cos, tan, log, sqrt, ln, power$\}$. Each function implicitly includes an assignment to a variable $v[i]$, which facilitates the use of multiple program outputs in LGP, whereas in tree-based GP, the side effects need to be incorporated explicitly (Brameier and Banzhaf 2001). After several trials, the functional set and operational parameters given in table 1 have been used in LGP modelling during this study.

Table 1. *Parameters of the LGP model.*

Parameter	Description of parameter	Setting of parameter
p_1	Function set	+, −, ∗, /, $\surd$, power
p_2	Population size	250
p_3	Mutation frequency %	95
p_4	Cross-over frequency %	50
p_5	Number of replication	10
p_6	Block mutation rate %	30
p_7	Instruction mutation rate %	30
p_8	Instruction data mutation rate %	40
p_9	Homologous cross-over %	95
p_{10}	Program size	Initial 80, maximum 256

LGP utilizes two-point string cross-over. A segment of random position and random length is selected in both parents and exchanged between them. If one of the resulting children would exceed the maximum length, cross-over is abandoned and restarted by exchanging equalized segments (Brameier and Banzhaf 2001).

An operand or operator of an instruction is changed by mutation into another symbol over the same set. LGP also employs a special kind of mutation (called *macro mutation*) which deletes or inserts an entire instruction.

The fitness of an LGP individual may be computed by using the equation:

$$f = \sum_{j=1}^{N} |O_j - E_j|, \qquad (2)$$

where O_j is the value returned by a chromosome for the fitness case j, and E_j is the expected value for the fitness case j.

In LGP, the maximum size of the program is usually restricted to avoid over-growing programs without bound (Brameier and Banzhaf 2001). In this study, the maximum size of each program has been set to 256, starting with 80 instructions per program. This configuration has been tested for each LGP model and has been experienced to be sufficient.

The best individual (program) of a trained LGP can be converted into a functional representation by successive replacements of $v[i]$ starting with the last effective instruction (Oltean and Groşan 2003). Further details on LGP can be found in Brameier and Banzhaf (2001) and Bramier (2003).

3. Neural networks

In this study, the Multilayer Perceptron Neural Networks (MLPNNs) (Rumelhart 1986) with one single hidden layer, and Generalized Regression Neural Networks (GRNNs) (Specht 1991) were employed. In the architecture, logistic transfer function $(y = 1/(1 + e^{-x}))$ is utilized. The Levenberg–Marquardt back-propagation algorithm was employed to optimize the weights in neural networks. The interested reader should refer to Cigizoglu and Alp (2006) and Tayfur (2002) for further details on NN modelling.

One of the main issues in MLPNNs modelling is to obtain the optimal network architecture (number of input–number of hidden neurons–number of output) (Guven *et al* 2006; Guven and Gunal 2008). Most of the studies in the literature searched for optimal MLPNNs architecture by increasing the number of neurons in each trial and monitoring the performance of the model based on an error criterion. However, this method usually leads to local optimum and the resultant NN model may suffer from over- or under-predictions in validation datasets. However, GRNNs do not require an iterative training procedure as in MLPNN algorithm. GRNN approximates any arbitrary function between input and output vectors, drawing the function estimate directly from the training data (Cigizoglu and Alp 2006).

To overcome the above-mentioned issue, a genetic algorithm was used to obtain the global optimal architecture of the proposed MLPNN models. The algorithm of genetic search for an optimal NNs architecture applies a systematic evolutionary search to determine the best number of hidden neurons to get the best generalization capacity. A combination of genetic operators of selection, cross-over and mutation is used. The neural network, produced from each generation, is trained to optimize the number of hidden neurons to minimize the fitness function (error) between the observed and predicted output. The architecture is ranked and the best architecture giving the minimal error is chosen (Guven and Gunal 2008).

4. Auto-regressive model

Auto-regressive (AR) model is the most widely used traditional time-series analysis. The model is usually referred to as the AR(p) model where p is the order model. The AR(p) model for time-series of daily flow rate can be represented as:

$$Q(t) = c + \sum_{i=1}^{p} \alpha_i Q(t-i) + \varepsilon(t), \tag{3}$$

where α_i is the model parameter, c is a constant, and $\varepsilon(t)$ is random error.

In this study, the AR(p) models are developed using an Excel add-in software. The model parameters are selected using Akaike's Information Criterion (AIC), given in equation (12).

5. Study area and data used

The dataset used in this study was obtained from the U.S. Geological Survey (USGS). The time series of daily discharge data from station 01470500 (lat. 40′31′21″, long. 75′59′55″) Schuylkill River at Berne, PA, USA are used. The drainage area at this site is 919.45 km^2. Information on the daily time series for the station can be acquired from the USGS web server (http://www.usgs.gov). The data from October 1, 2002 to September 30, 2006 were chosen for training of the proposed LGP and NN models, and data of October 1, 2006 to September 30, 2007 were chosen for testing the models. Additionally, a dataset from October 1, 2007 to September 30, 2008 was reserved for application. During the analysis of data, a scattered relationship between time-lagged discharge values was observed. The discharge values ranges between 2.123 and 971.03 m^3/s.

6. Development and evaluation of the models

As explained in the preceding section, the daily discharge data between 10.01.2002 and 09.30.2006 were used as a training set (1824 sets) and the data between 10.01.2006 and 09.30.2007 as a testing set (730 sets) for calibration and validation of both LGP and NN models. Referring to Tayfur and Guldal (2006), firstly the cross-correlation between the discharge at present time ($Q(t)$) and time-lagged discharges ($Q(t-1), \ldots, Q(t-d)$) was evaluated. Figure 1 shows the cross-correlation values between $Q(t)$ and each time-lagged Q values. It is clearly seen from this figure that, R values are higher than 0.3 for four time lags ($t-1$, $t-2$, $t-3$ and $t-4$), and equal or lower than 0.2 after a time lag of 5 days. Similar findings were observed in LGP and NN modelling, namely, adding $Q(t-5)$ and the further time-series to the input set did not make any change in the training performance of both LGP and NN models. Therefore, it was decided to use the time series from $t-1$ to $t-4$ as input set in present modelling applications.

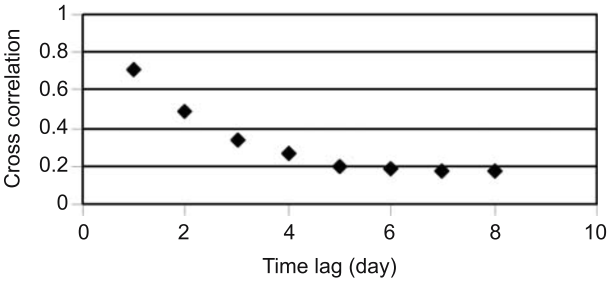

Figure 1. Cross correlations between time-lagged Q (m^3/s) values.

The relationship in equation (1) was used while developing different time-series scenarios for model development. The first scenario was $Q(t) = f(Q(t-1))$ and for each next scenario, $Q(t-2)$ to $Q(t-d)$ were added to the input set, one after another:

$$Q(t) = f(Q(t-1)), \tag{4}$$

$$Q(t) = f(Q(t-1), Q(t-2)), \tag{5}$$

$$Q(t) = f(Q(t-1), Q(t-2), Q(t-3)), \tag{6}$$

$$Q(t) = f(Q(t-1), Q(t-2), Q(t-3), Q(t-4)). \tag{7}$$

The testing results of LGP and NN models, in terms of coefficient of determination R^2, mean square error (MSE) and mean absolute error (MAE) (see Tayfur and Singh 2006 for definitions), are given in table 2. The common statistical characteristics (maximum, minimum, average and standard deviation) of the observed and predicted $Q(t)$ values for testing set are given in table 3. Also the predictions of the proposed LGP and NN models to the observed discharges ($Q(t)$) for testing set are illustrated in figures 2 and 3, respectively.

Referring to table 2, it can be stated that the LGP2 model outperformed the other LGP models with the highest R^2 (0.691) and the lowest MSE (107.914 m^6/s^2) and MAE (0.072) values. Another important finding from table 2 is that, LGP and NN models gave the best testing results for input set $\{Q(t-1), Q(t-2)\}$, among corresponding four scenarios given in equations (4)–(7). No significant

Table 2. *Testing results of LGP and NN models.*

Model	MSE (m^6/s^2)	MAE	R^2	AIC	Optimal architecture
LGP1	127.214	4.441	0.616	1792.74	–
LGP2	**107.914**	**3.960**	**0.691**	**1728.69**	–
LGP3	126.374	5.170	0.615	1782.33	–
LGP4	140.065	6.706	0.584	1790.33	–
MLPNN1	272.35	5.168	0.615	2077.13	1–9–1
MLPNN2	**222.11**	**4.509**	**0.677**	**2054.16**	2–12–1
MLPNN3	295.809	7.128	0.591	2062.03	3–7–1
MLPNN4	1970.50	8.321	0.246	3012.91	4–23–1
GRNN1	403.095	13.206	0.419	2217.70	1–5–1
GRNN2	**262.48**	**6.565**	**0.645**	**2136.46**	2–9–1
GRNN3	335.067	8.235	0.517	2238.23	3–13–1
GRNN4	309.045	6.700	0.539	2296.72	4–17–1
AR(3)	278.75	5.413	0.630	2067.96	–

discrepancy was observed among the testing results of MLPNN and GRNN models.

Table 3 shows that all of the models failed to predict $Q_{\max}$ in the testing period ($305.75\,m^3/s$), and prediction of LGP2 model is the closest one ($141.81\,m^3/s$). Also, moderate differences were observed among statistical measures of the observed testing set and those of the predicted ones. The underlying reason for this can be explained by the considerable discrepancy observed between statistical characteristics of the training and testing periods.

Referring to the overall performance of all models given in tables 2 and 3, LGP2 can be said to outperform all other models. The simplified analytic form of LGP2 model is given in equation (8), for the purpose of re-evaluation in further studies:

$$Q(t) = \sqrt{\mathbf{F}_1} \times Q(t-1) \cdot 2^{\mathbf{F}_2} - 3\mathbf{F}_2, \tag{8}$$

where

$$\mathbf{F}_1 = (2(\mathbf{F}_3^2 - 1)^2 - \mathbf{F}_2)\frac{Q(t-1)}{Q(t-2)} - \frac{Q(t-1)}{Q(t-2)}, \tag{9}$$

$$\mathbf{F}_2 = \frac{Q(t-1)Q(t-2)}{-2Q^2(t-2) - 4.01Q(t-2) + 2.394}, \tag{10}$$

$$\mathbf{F}_3 = \left(\left(2\left(-\frac{Q(t-1)}{F_2}\right)^{0.5} - 5.56\right) \times \frac{\mathbf{F}_2}{Q(t-2)}\right) - 0.587. \tag{11}$$

In order to get more reliable evaluation and comparison, AR(p) model is also used as a traditional method for time-series modelling of daily flow rate. Therefore, the same training and testing datasets utilized in LGP and NN modelling are also employed in developing different AR(p) models. The AIC results of AR(p) models, p ranging from 1 to 10, for the testing set are illustrated in figure 4. The aim is to select the optimal model, which gives the minimum AIC. It is clearly seen that AR(3) model, containing $Q(t-1)$, $Q(t-2)$ and $Q(t-3)$ as inputs and $Q(t)$ as output, can be chosen as the optimal AR model with minimum AIC of 2067.96. The testing results of AR(3) are given in tables 2 and 3. The results clarify that, the testing results of AR(3) model can said to be competitive with those of NN models.

7. Further application

7.1 *Estimation of flood and minimum discharges*

During the testing period, a number of floods occurred, the maximum of which was recorded on November 17, 2006 with a magnitude of $305.75\,m^3/s$. The LGP2 model predicted it as $147.81\,m^3/s$, which is the closest value among those of the proposed models. MLPNN2 model predicted the flood as $143.33\,m^3/s$ with a second best performance. Table 4 shows the observed and estimated flood and minimum flow values that occurred at testing period. It is clearly seen that, LGP models generally predicted the maximum and minimum discharge values better than the NN models. AR(3) model predictions are observed to be almost as

Table 3. *Statistical performance of the proposed models for testing set.*

	$Q_{\max}$ (m^3/s)	$Q_{\min}$ (m^3/s)	Q_{mean} (m^3/s)	S_x (m^3/s)	C_v
Observed $Q(t)$ (m^3/s)	305.75	3.20	21.67	25.69	1.19
LGP1	83.21	3.54	20.35	21.19	1.04
LGP2	141.81	4.93	21.32	20.03	0.94
LGP3	78.92	4.23	21.79	20.45	0.94
LGP4	83.92	6.79	23.38	21.37	0.91
MLPNN1	82.08	3.61	19.86	18.07	0.91
MLPNN2	123.32	2.99	24.41	24.99	1.02
MLPNN3	82.50	4.88	21.64	18.95	0.88
MLPNN4	83.87	3.88	24.24	51.12	2.11
GRNN1	68.14	8.28	24.19	18.71	0.69
GRNN2	84.76	5.84	22.50	22.57	1.00
GRNN3	69.34	9.32	25.81	18.33	0.71
GRNN4	69.84	4.36	22.02	21.12	0.96
AR(3)	112.71	6.83	22.58	30.31	1.34

Note: Q_{mean} – mean observed discharge, S_x – standard deviation, $Q_{\min}$ – minimum observed discharge, $Q_{\max}$ – maximum observed discharge, C_v – coefficient of variation.

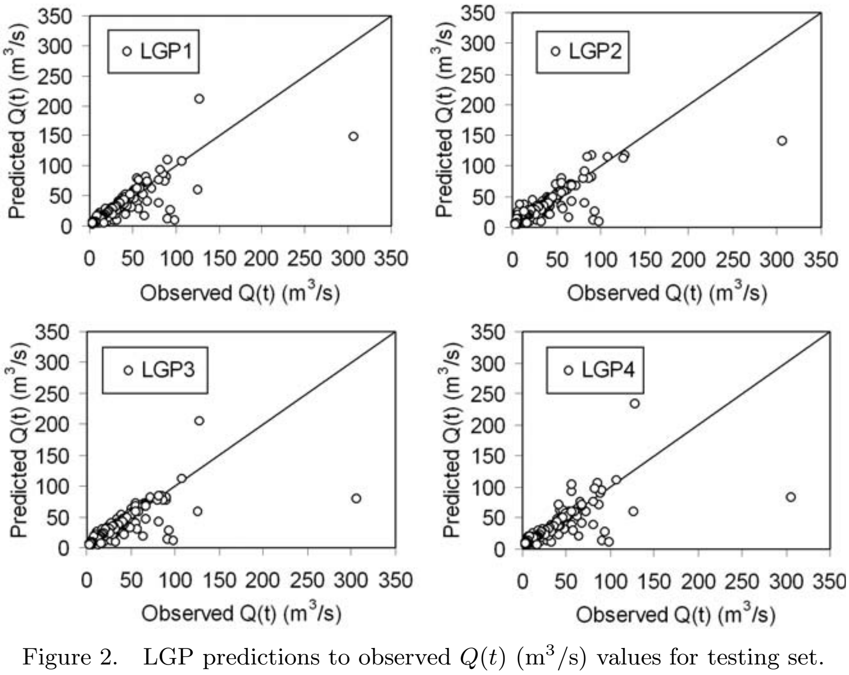

Figure 2. LGP predictions to observed $Q(t)$ (m^3/s) values for testing set.

good as LGP2 model in the prediction of maximum $Q(t)$ values, especially better than those for the last five peak discharges, but AR(3) predictions are much worse than those of LGP2 for minimum $Q(t)$ values (see table 4). LGP models' predictions are closer to observed minimum discharges, compared to other models. As a general view from table 4, the predictions are higher than almost all the observed minimum $Q(t)$ values, whereas all models underpredicted the observed peak $Q(t)$.

7.2 *Generalization of LGP and NN models*

The optimal NN architecture, which is related to the number of neurons in the hidden layer, is one of the most important tasks in MLPNN studies. Generally, the trial and error approach is used. In this study, the best architecture of the network was obtained by a genetic algorithm, which avoids local optimum problems associated with multilayer perceptron modelling, and grant for a global optimal

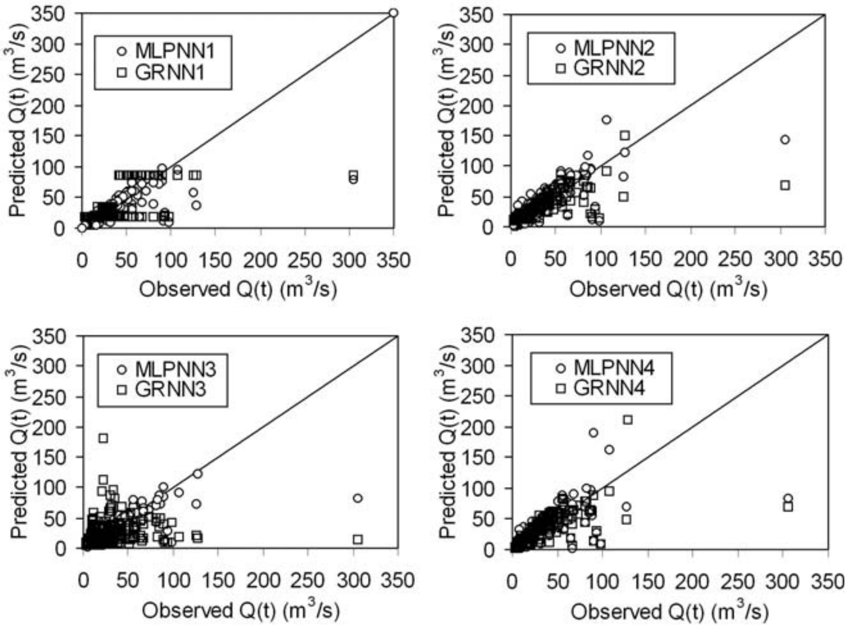

Figure 3. MLPNN and GRNN predictions to observed $Q(t)$ (m^3/s) values for testing set.

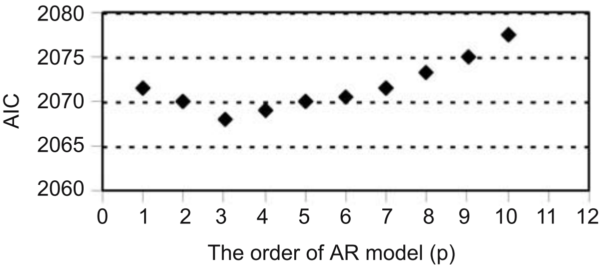

Figure 4. AIC values for AR(p) models for different orders.

solution. There are several performance criteria for measuring the generalization capacity of soft computing techniques. In this study, Akaike Information Criterion (AIC) defined by Akaike (1974) is utilized to evaluate the robustness of the proposed LGP and NN models.

$$\mathrm{AIC} = N \ln(\mathrm{MSE}) + 2k, \tag{12}$$

where N is the number of samples in the testing set, and k is the number of network weights.

AIC is used to measure the exchange between testing performance and network size. The goal is to minimize AIC to obtain a network with the best generalization. The AIC values for the predictions of proposed models for testing set is given in table 2. Table 2 also shows the optimal architecture (number of inputs–number of hidden neurons–number of output) of each NN models.

Equation (14) implies that AIC increases with increasing number network weights (i.e., number of hidden neurons in NN models), however, table 2 shows that MLPNN2 has the lowest AIC value as 2054.16 among MLPNN models despite its higher number of neurons (12) than MLPNN1 (7 neurons) and MLPNN3 (9 neurons). This may seem as a contradiction but actually, the lowest MSE (222.11 $\mathrm{m}^6/\mathrm{s}^2$) of MLPNN2, among other NN models, revealed the lowest AIC (= 2054.16) despite the higher number of neurons it employed. Table 5 gives the number of network weights of NN models, which support the above findings.

As seen in table 2, AIC value of LGP2 model is the lowest one among the others, and MLPNN2 model with 2–4–1 architecture, has the lowest AIC value (2054.16) among the NN models.

7.3 *Application*

In this section, the robustness of the proposed LGP and NN models are evaluated based on the application period. In other words, the models proposed in this study are run for another dataset. Time-series of discharge for October 1, 2007 to September 30, 2008 has been taken into consideration for application period. It is emphasized that the data of this period was neither used in the training nor testing period. The statistical results of $Q(t)$ predictions to application data are given in table 5. LGP2 model has the highest R^2 as 0.59, and the lowest MSE as 214.87 $\mathrm{m}^6/\mathrm{s}^2$. Among the NN models, MLPNN2 can be said to have performed better than the others with AIC = 2064.63 and $R^2 = 0.55$ (see table 5).

The observed cumulative hydrograph during the application period and the predictions of

Table 4. *Observed and estimated peak discharge values in testing set.*

Date	$Q(t)$ (m^3/s)	LGP2	MLPNN2	GRNN2	AR(3)
Maximum discharge values (m^3)					
11.17.2006	305.75	147.81	143.33	68.14	112.71
11.18.2006	127.68	117.46	111.06	149.75	105.70
04.16.2007	125.98	113.29	82.13	49.63	109.46
04.17.2007	107.30	115.61	176.25	91.53	110.73
03.02.2007	98.24	10.33	9.23	14.52	76.36
11.16.2006	93.14	25.06	31.85	29.23	83.55
10.28.2006	90.88	12.65	12.24	16.77	96.46
11.19.2006	89.74	82.34	93.67	22.01	95.57
10.29.2006	89.18	117.58	97.08	62.71	96.15
03.24.2007	86.63	80.12	116.67	65.06	75.65
Minimum discharge values (m^3)					
08.04.2007	3.20	4.93	5.21	10.28	6.83
08.03.2007	3.23	4.61	5.40	10.49	6.99
08.05.2007	3.31	5.00	4.91	10.24	6.76
09.25.2007	3.31	5.02	5.07	10.37	7.26
09.24.2007	3.34	5.29	6.39	10.48	5.53
08.02.2007	3.37	4.66	1.65	11.27	7.45
08.19.2007	3.43	5.11	6.15	10.46	4.91
08.18.2007	3.45	5.53	6.68	10.56	7.57
09.23.2007	3.51	5.37	6.49	10.51	7.37
08.15.2007	3.54	5.25	6.30	10.56	7.81

Table 5. *Statistical results of model predicitons in application data.*

Model	MSE (m^6/s^2)	R^2	AIC	No. of fitting parameters (k)
LGP1	258.55	0.50	2051.61	12
LGP2	214.87	0.59	1980.06	10
LGP3	250.21	0.55	2039.64	12
LGP4	273.86	0.50	2076.61	14
MLPNN1	319.76	0.49	2149.17	22
MLPNN2	228.58	0.55	2064.63	41
MLPNN3	253.72	0.53	2088.72	34
MLPNN4	479.91	0.41	2497.36	122
GRNN1	472.70	0.40	2275.84	14
GRNN2	323.23	0.50	2173.11	32
GRNN3	413.46	0.26	2314.97	58
GRNN4	312.55	0.48	2300.83	102
AR(3)	310.07	0.55	2101.93	4

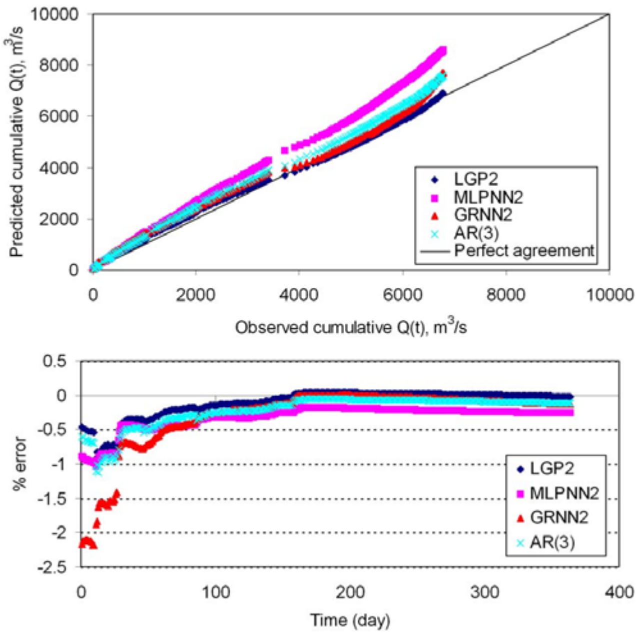

Figure 5. Observed and predicted cumulative hydrograph for application period.

LGP2, MLPNN2, GRNN2 and AR(3) models are illustrated in figure 5. The figure also shows the percentage of error for the corresponding prediction during the application period. LGP2 model is observed to capture the cumulative hydrograph with almost perfect agreement, with the lowest percentage of error during the whole period. It is clearly seen from the figure that all the models underpredicted the cumulative discharges. LGP2's predictions are clearly seen to be closer to the observed ones, compared to other models.

The last column in table 5 shows the number of fitting parameters (k) included in the corresponding model. This column shows that the LGP models have significantly less number of model parameters than the NN models. The proposed LGP models can be attributed as more practical and robust than the NN models, considering the network size and predicting performance together.

8. Conclusions

The main goal of this study is to propose LGP as an alternative method of predicting time series of river discharge data, and to evaluate the performance of the proposed LGP models with well-validated MLPNN and GRNN techniques. For this purpose, different LGP and NN models have been developed, and the statistical performance of each model was evaluated based on well-known performance measures. The results showed that both LGP and NN techniques predicted the daily time series of discharge with quite good agreement with the observed data. As a conventional method, autoregressive (AR) models were also developed for time-series modelling of the same datasets. As a general view, LGP models performed moderately better than NNs and AR(3) models. Especially, in predicting the peak (maximum and minimum) discharge values, LGP2 model was clearly superior to those of NNs and AR(3). Consequently, the results of this study are very promising and support the use of LGP in predicting the nonlinear and dynamic river flow parameters.

Acknowledgements

The data used in this study were downloaded from the web server of the USGS. The author wishes to thank the staff of the USGS who are associated with data observation, processing, and management of USGS websites. The financial support of Scientific Research Projects Governing Unit of Gaziantep University during this study is gratefully acknowledged.

References

Akaike H 1974 A new look at the statistical model identification; *IEEE Transactions on Automatic Control* **19(6)** 716–723.

Aytek A, Guven A, Yuce M İ and Aksoy H 2008 An explicit neural network formulation for evapotranspiration; *Hydrol. Sci.* **53(4)** 893–904.

Aytek A and Alp M 2008 An application of artificial intelligence for rainfall-runoff modelling; *J. Earth Syst. Sci.* **117(2)** 145–155.

Banzhaf W, Nordin P, Keller R and Francone F 1998 Genetic Programming; Morgan Kauffman, San Francisco, CA.

Bhattacharya M, Abraham A and Nath B 2001 A linear genetic programming approach for modelling electricity demand prediction in Victoria; In: *Proceedings of the hybrid information systems, first international workshop on hybrid intelligent systems*, Adelaide, Australia, 379–393.

Brameier M and Banzhaf W 2001 A comparison of linear genetic programming and neural networks in medical data mining; *IEEE Transactions on Evolutionary Computation* **5** 17–26.

Brameier M 2004 On linear genetic programming; Ph.D. thesis. University of Dortmund.

Baiamonte G and Ferro V 2007 Simple flume for flow measurement in sloping channel; *J. Irrig. Drain. Eng.* **133(1)** 71–78.

Bordignon S and Lisi F 2000 Nonlinear analysis and prediction of river flow time series; *Environmetrics* **11** 463–477.

Charhate S B, Deo M C and Sanil Kumar V 2007 Soft and hard computing approaches for real time prediction of currents in a tide dominated area; *J. Engineering for the Maritime Environment* **221** 147–163.

Charhate S B, Deo M C and Londhe S N 2008 Inverse modelling to derive wind parameters from wave measurements; *Applied Ocean Research* **30(2)** 120–129.

Cigizoglu H K and Kisi O 2000 Flow prediction by two back propagation techniques using k-fold partitioning of neural network training data; *Nordic Hydrology* **36(1)** 1–16.

Cigizoglu H K and Alp M 2006 Generalized regression neural network in modelling river sediment yield; *Advanced Engineering Software* **37** 63–68.

Deo O, Jothiprakash V and Deo M C 2008 Genetic Programming to predict spillway scour; *Int. J. Tomography and Statistics* **8(8)** 32–45.

Drecourt J P 1999 Application of Neural Networks and Genetic Programming to Rainfall Runoff Modeling; Danish Hydraulic Institute (Hydro-Informatics Techonologies HIT), D2K-0699-1.

Firat A 2008 Comparison of artificial intelligence techniques for river flow forecasting; *Hydrol. Earth Syst. Sci.* **12** 123–139.

Foster J A 2001 Discipulus: A commercial genetic programming system; *Genetic Programming and Evolvable Machines* **2** 201–203.

Gaur S and Deo M C 2008 Real time wave forecasting using genetic programming; *Ocean Engineering* **35** 1166–1172.

Guven A, Gunal M and Cevik A 2006 Prediction of pressure fluctuations on stilling basins; *Can. J. Civ. Eng.* **33(11)** 1379–1388.

Guven A and Gunal M 2008 A genetic programming approach for prediction of local scour downstream hydraulic structures; *J. Irrig. Drain. Eng.* **132(4)** 241–249.

Guven A, Aytek A, Yuce M İ and Aksoy H 2008 Genetic programming-based empirical model for daily reference evapotranspiration estimation; *CLEAN-Soil, Air, Water* **36(10–11)** 905–912.

Habib E H and Meselhe E A 2006 Stage-discharge relations for low-gradient tidal streams using data driven models; *J. Hydraul. Eng.* **132(5)** 482–492.

Jain K S 2008 Development of integrated discharge and sediment rating relation using a compound neural network; *J. Hydrol. Eng.* **13(3)** 124–131.

Jain P and Deo M C 2008 Artificial intelligence tools to forecast ocean waves in real time; *The Open Ocean Engineering Journal* **1** 13–21.

Kalra R and Deo M C 2007 Genetic programming to retrieve missing information in wave records along the west coast of India; *Applied Ocean Research* **29(3)** 99–111.

Kalra R, Deo M C, Kumar R and Agarwal V K 2008 Genetic programming to estimate coastal waves from deep water measurements; *Int. J. Ecology and Development* **10(8)** 67–76.

Kisi O 2004 River flow modelling using artificial neural networks; *J. Hydrol. Eng.* **9(1)** 60–63.

Kisi O and Cigizoglu H K 2007 Comparison of different ANN techniques in river flow prediction; *Civil Engineering and Environmental Systems* **24(3)** 211–231.

Lopesi H S and Weinert W R 2004 A gene-expression programming system for time series modelling; In: *Proc. XXV Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE)*, Recife, Brazil, CD-ROM Version.

Maier H R and Dandy G C 2000 Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications; *Environ. Modelling Software* **15** 101–123.

Muttil N and Liong S Y 2004 A superior exploration–exploitation balance in shuffled complex evolution; *J. Hydraul. Eng.* **130(2)** 211–220.

Oltean M and Groşan C 2003 A comparison of several linear genetic programming techniques; *Complex Systems* **14(1)** 1–29.

Preis A and Otsfeld A 2008 A coupled model tree-genetic algorithm scheme of flow and water quality predictions in watersheds; *J. Hydrol.* **349** 364–375.

Rumelhart D E, Hinton G E and Williams R J 1986 Learning internal representations by error propagation; In: *Paralled Distributed Processing. Explorations in the Microstructure of Cognition* (eds) Rumelhart D E, McClelland J L, and the PDP Research Group, Volume 1: Foundations (Cambridge: MIT Press), pp 318–362.

Singh A K, Deo M C and Sanil Kumar V 2007 Neural network – genetic programming for sediment transport; *ICE J. Maritime Engineering* **160(MA3)** 113–119.

Specht D F 1991 A general regression neural network; *IEEE Trans. Neural Netw.* **2(6)** 568–576.

Tayfur G 2002 Artificial neural networks for sheet sediment transport; *Hydrol. Sci.* **47(6)** 879–892.

Tayfur G and Singh V P 2006 ANN and fuzzly logic models for simulating event-based rainfall-ruoff; *J. Hydraul. Eng.* **132(12)** 1321–1330.

Tayfur G and Guldal V 2006 Artificial neural networks for estimating daily total suspended sediment in natural streams; *Nordic Hydrology* **37** 69–79.

Tayfur G, Moramarco T and Singh V P 2007 Predicting and forecasting flow discharge at sites receiving significant lateral inflow; *Hydrological Processes* **21** 1848–1859.

Tayfur G and Moramarco T 2008 Predicting hourly-based flow discharge hydrographs from level data using genetic algorithms; *J. Hydrol.* **352** 77–93.

Takens F 1981 Detecting strange attractors in turbulence in turbulence; In: *Dynamical systems and turbulence* (eds) Hand D and Young L S, Springer-Verlag, Berlin, pp 366.

United States Geographical Survey (USGS) web server: http://www.usgs.gov

Ustoorikar K and Deo M C 2008 Filling up gaps in wave data with genetic programming; *Marine Structures* **21** 177–195.

Wang W, Van Gelder P H A J M and Vrijling J K 2004 Periodic autoregressive models applied to daily streamflow; *Proceedings of the 6th International Conference on Hydroinformatics*, World Scientific, Singapore, 1334–1341.

Wang W, Van Gelder P H A J M, Vrijling J K and Ma J 2006 Forecasting daily streamflow using hybrid ANN models; *J. Hydrol.* **324** 383–399.

Whigham P A and Crapper P F 2001 Modeling rainfall-runoff using Genetic Programming; *Mathematical and Computer Modelling* **33** 707–721.

MS received 16 October 2008; revised 4 February 2009; accepted 5 February 2009