Irrig Sci (2011) 29:135–145
DOI 10.1007/s00271-010-0225-5

ORIGINAL PAPER

Daily pan evaporation modeling using linear genetic programming technique

Aytac Guven · Özgür Kişi

Received: 2 February 2010 / Accepted: 7 July 2010 / Published online: 22 July 2010

Abstract This paper investigates the ability of linear genetic programming (LGP), which is an extension to genetic programming (GP) technique, in daily pan evaporation modeling. The daily climatic data, air temperature, solar radiation, wind speed, pressure and humidity of three automated weather stations, Fresno, Los Angeles and San Diego in California, are used as inputs to the LGP to estimate pan evaporation. The LGP estimates are compared with those of the Gene-expression programming (GEP), which is another branch of GP, multilayer perceptrons (MLP), radial basis neural networks (RBNN), generalized regression neural networks (GRNN) and Stephens–Stewart (SS) models. The performances of the models are evaluated using root mean square errors (RMSE), mean absolute error (MAE) and determination coefficient (R^2) statistics. Based on the comparisons, it was found that the LGP technique could be employed successfully in modeling evaporation process from the available climatic data.

Introduction

Estimating evaporation loss is an important issue for monitoring, survey and management of water resources. The estimation of this loss is imperative in the planning and management of irrigation practices in many areas where water resources are scarce. Temperate areas of rainy seasons are mainly concerned with the shortage of water in the dry seasons for the different uses. Despite this importance, evaporation is one of the less understood components of the hydrologic cycle (Brutsaert 1982).

A number of researchers have attempted to predict pan evaporation values from climatic variables (Stephens and Stewart 1963; Rahimikhoob 2009), and most of these methods require data not easily available. Simple methods that are reported (e.g., Stephens and Stewart 1963) fit a linear relationship between the variables. However, the evaporation process is highly non-linear, as it is evidenced by many of the estimation procedures. Many researchers have emphasized the need for correct estimations of pan evaporation in hydrologic modeling studies. This requirement could be addressed through better models that will address the inherent non-linearities in the pan evaporation process.

According to the recent experiments, artificial neural networks (ANN) may offer a promising alternative in evapotranspiration estimation (Al-Ghobari 2000; Kumar et al. 2008; Gundekar et al. 2008) and pan evaporation modeling (Keskin and Terzi 2006; Kisi 2009a, b; Tabari et al. 2010). Keskin and Terzi (2006) developed MLP models for modeling daily pan evaporation and found that the ANN model shows a considerably better performance over the conventional method. Kisi (2009a) used three different ANN techniques, namely the MLP, RBNN and GRNN, in daily pan evaporation modeling and found that the MLP and RBNN performs much better than the GRNN. Kisi (2009b) investigated the accuracy of MLP and RBNN methods in monthly evaporation estimation and found that the two ANN methods could be successfully used in monthly evaporation modeling. Tabari et al. (2010) used an

Communicated by S. Azam-Ali.

A. Guven (✉)
Civil Engineering Department, Hydraulics Division,
Gaziantep University, 27310 Gaziantep, Turkey
e-mail: aguven@gantep.edu.tr

Ö. Kişi
Civil Engineering Department, Hydraulics Division,
Erciyes University, 38039 Kayseri, Turkey

ANN and multivariate regression models to estimate daily pan evaporation. The ANN has the ability to learn complex and non-linear relationships that are difficult to model with conventional techniques. This is the main reason for ANN becoming so popular. However, there are some disadvantages of the ANN. The ANN is a black-box model, and network structure is hard to determine. It is usually determined using a trial-and-error approach, i.e., sensitivity analysis (ASCE Task Committee 2000). The ANN training algorithm has the danger of getting stuck into local minima, etc. However, there are also a number of studies that present the proposed ANN models in terms of explicit formulations, which were developed using the weights and bias of the well-trained ANN models and the transfer functions utilized (Guven et al. 2006; Guven and Gunal 2008a).

The focus of this paper is to model empirically the daily pan evaporation using the linear genetic programming (LGP), which is a new branch of genetic programming technique. GP (Koza 1992) has been applied to a wide range of problems in artificial intelligence, engineering and science applications, industrial and mechanical models, but LGP has been rarely applied in engineering and science area (Guven and Kişi 2009).

GP technique has been successfully applied in the field of water resources engineering, with only a number of studies of Whigham and Crapper (2001), Babovic and Keijzer (2002), Dorado et al. (2003), Giustolisi (2004), Rabunal et al. (2007) Guven et al. (2007), Guven and Gunal (2008b), Aytek and Kisi (2008). However, the use of LGP has been recorded by only a few studies. More recently, Guven (2009) modeled the time series of daily flow rate in rivers using LGP, and Guven et al. (2009) presented LGP as an alternative tool in the prediction of scour depth around a circular pile due to waves in medium dense silt and sand bed.

This study is concerned with the application of LGP for modeling daily pan evaporation $(E)_0$ based on climatic variables. The ability of LGP is evaluated based on field data and compared to GEP developed by authors and MLP, RBNN, GRNN and SS methods employed in the previous work of Kisi (2009a). This is the first study that investigates the accuracy of LGP in modeling evaporation in the hydrological context.

Methodology

Linear genetic programming

Linear genetic programming (LGP) is characterized with a linear representation of individuals. The main characteristic of LGP in comparison with tree-based GEP is that expressions of a functional programming language (like LISP) are substituted by programs of an imperative language (like C; Brameier and Banzhaf 2001). Nordin's idea of using machine code for evolution was the most radical "down-to-bones" approach (Nordin 1994; Brameier and Banzhaf 2001) in this context. The methodology was subsequently expanded (Nordin and Banzhaf 1995) and led to the automatic induction of machine code by genetic programming (AIMGP) system (Nordin 1997; Banzhaf et al. 1998; Brameier and Banzhaf 2001). In AIMGP, individuals are manipulated directly as binary machine code in memory and are executed directly without passing an interpreter during fitness calculation. This results in a significant speedup compared with interpreting systems like GEP. Because of their dependence on specific processor architectures, however, AIMGP systems are restricted in portability (Brameier and Banzhaf 2001; Brameier 2004; Banzhaf et al. 1998). Present LGP technique utilizes individual programs, which are represented as variable-length strings composed of simple C instructions. An extract from a linear genetic program is illustrated as follows:

```
Void LGP
double v[3];
{
r[0]+ = −0.997;
r[0]* = v[0]
r[1]+ = r[0];
r[0]− = v[2];
r[0]/ = v[1];
r[2]− = r[0];
r[0]/ = r[2];
}
```

where v[*i*] represents the input and output variables used in LGP modeling, r[*i*] are the temporary computation variables in the programs LGP creates. LGP uses these temporary computation variables to store values while performing calculations. The output of these programs is the value remaining in r[0] after the program executes (Guven 2009; Guven and Kişi 2009). After LGP model is evolved and finished, the value then in v[*i*] is also treated by LGP as the output of the program for testing fitness. The number of computation variables is defined by the user. It was observed that using two computation variables in present modeling gave the optimal training results.

The *function* (*instruction*) *set* of LGP system is composed of arithmetic operations (+, −, /, *), conditional branches (*if*, *then*) and function calls (*power*, *trigonometric*, *logarithmic*, *etc*.). The function set used in present LGP modeling is listed in Table 1. Each element of instructon

Table 1 Parameters of the LGP1 and LGP2 model

Parameter	Description of parameter	Setting of parameter
p_1	Function set	+, −, *, /, $\surd$, power
p_2	Population size	500 (LGP1), 250 (LGP2)
p_3	Mutation frequency %	95
p_4	Crossover 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 crossover %	95
p_{10}	Program size	Initial 80, maximum 512 (LGP1), initial 64, maximum 256 (LGP2)

set implicitly includes an assignment to a variable (*destination variable*). This facilitates the use of multiple program outputs in LGP, whereas in tree-based GEP those side effects need to be incorporated explicitly (Brameier and Banzhaf 2001). LGP uses variable-length chromosome, which performs better than GEP that uses fixed-length chromosome.

Instructions operate either on two variables (*operand variables*) or on one variable and one integer constant. Variables and constants form the “terminal set” of LGP. Each instruction is encoded into a four-dimensional vector that holds the instruction identifier, indexes of all participating variables and a constant value (optionally). Because each vector component uses one byte of memory only, the maximum number of variables is restricted to 256 and constants range from 0 to 255 at maximum. This representation allows an efficient recombination of the programs as well as an efficient interpretation (Brameier and Banzhaf 2001).

The evolutionary algorithm (EA) of present LGP applies *tournament selection* and puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. The loser of each tournament is replaced by a copy of the winner. In such a *steady-state* EA, the population size is always constant and determines the number of individuals created in one *generation*. A segment of random position and random length is selected in each of the two parents and exchanged. If one of the resulting children would exceed the maximum length, crossover is aborted and restarted with exchanging equally sized segments. The crossover points only occur *between* instructions. *Inside* instructions, the *mutation* operation randomly replaces the instruction identifier, a variable or the constant (if existent) by equivalents from valid ranges. Constants are modified through a certain standard deviation (*mutation step size*) from the current value. Exchanging a variable, however, can have an enormous effect on the program flow that might be the reason why in LGP, high mutation rates have been experienced to produce better results (Brameier and Banzhaf 2001).

In GP, the maximum size of the program is usually restricted to prevent programs from growing without bound (Brameier and Banzhaf 2001). In present LGP, the maximum number of instructions allowed per program has been set to 256 for two-input models (assigned as LGP2) and 512 for five-input models (LGP1), starting with 64 80 instructions per program. For all tested problems, this configuration has been experienced to be a sufficient maximum length.

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 Grosan 2003). The further details on LGP can be found in Brameier and Banzhaf (2001) and Brameier (2004).

Gene-expression programming

The Gene-expression programming (GEP) technique, which is tree-based variant of the GP technique, involves computer programs (mathematical expressions, decision trees, polynomial constructs, logical expressions), which are then expressed or translated into expression trees (ETs); Guven and Gunal 2008b). ETs are sophisticated computer programs that are usually evolved to solve a particular problem and are selected according to their fitness at solving that problem (Ferreira 2001a, b).

The brief algorithm of GEP starts with randomly generating the chromosomes of each individual of the population. GEP chromose is formed by genes linked by an arithmetic operators (+, −, *, /). In present study, the GEP genes are linked with addition. In fact, this is one of the drawbacks of GEP, that it is not a good idea to assume that the genes may be linked by either multipication or addition. But using all combinations of operators as linking function would increase the completiy of the problem (Oltean and

Grosan 2003). GEP chromosomes are expressed, and each individual is evaluated based on a fitness function and selected to reproduce with modification, leaving progeny with new traits. The individuals of the new generation are, in their turn, subjected to some developmental process such as expression of the genomes, confrontation of the selection environment and reproduction with modification. The process is repeated either for predefined number of generations or until a solution is achieved (Ferreira 2001a, b). The further details of GEP technique can be found in Ferreira (2001a, b), and GEP applications in water resources engineering are given in Guven and Gunal (2008b), and Guven and Aytek (2009).

Case study

The daily climatic data of three automated weather stations, Fresno Station (Latitude 36°47′N, Longitude 119°43′W), Los Angeles Station (Latitude 33°56′N, Longitude 118°24′W) and San Diego Station (Latitude 32°44′N, Longitude 117°10′W) operated by the US

Table 2 Error statistics for each model in test period

Models	Input set	RMSE	MAE	R^2	k	AIC	MDL
Fresno station							
MLP1[a]	T, SR, W, P and H	0.20	0.14	0.998	36	−1,102.9	−1,068.7
RBNN1[a]	T, SR, W, P and H	0.20	0.15	0.998	49	−1,076.9	−1,030.3
GRNN1[a]	T, SR, W, P and H	0.40	0.29	0.993	40	−588.9	−550.9
LGP1	T, SR, W, P and H	0.77	0.20	0.972	5	−184.8	−180.1
GEP1	T, SR, W, P and H	0.63	0.16	0.983	6	−325.6	−319.9
MLP2[a]	T and SR	0.74	0.53	0.974	13	−193.8	−181.5
RBNN2[a]	T and SR	0.73	0.52	0.974	40	−149.7	−111.7
GRNN2[a]	T and SR	0.78	0.56	0.971	10	−161.4	−151.9
LGP2	T and SR	0.74	0.15	0.974	12	−190.3	−184.9
GEP2	T and SR	0.79	0.17	0.970	6	−163.2	−157.5
SS[a]	T and SR	0.81	0.58	0.969	2	−149.8	−147.9
Los Angeles station							
MLP1[a]	T, SR, W, P and H	0.20	0.16	0.988	50	−1,074.9	−1,027.4
RBNN1[a]	T, SR, W, P and H	0.20	0.16	0.988	55	−1,064.9	−1,012.6
GRNN1[a]	T, SR, W, P and H	0.46	0.32	0.952	40	−486.9	−448.9
LGP1	T, SR, W, P and H	0.19	0.05	0.989	26	−1,165.3	−1,140.6
GEP1	T, SR, W, P and H	0.26	0.05	0.982	6	−968.3	−962.6
MLP2[a]	T and SR	1.02	0.76	0.716	17	48.5	64.6
RBNN2[a]	T and SR	1.01	0.75	0.722	19	45.3	63.3
GRNN2[a]	T and SR	1.05	0.77	0.710	10	55.6	65.1
LGP2	T and SR	1.01	0.20	0.722	18	41.5	58.6
GEP2	T and SR	1.04	0.20	0.705	9	47.9	56.4
SS[a]	T and SR	1.18	0.81	0.657	2	124.8	126.7
San Diego station							
MLP1[a]	T, SR, W, P and H	0.20	0.16	0.987	50	−1,074.9	−1,027.4
RBNN1[a]	T, SR, W, P and H	0.20	0.15	0.988	49	−1,076.9	−1,030.3
GRNN1[a]	T, SR, W, P and H	0.37	0.29	0.963	40	−645.8	−607.8
LGP1	T, SR, W, P and H	0.19	0.04	0.988	21	−1,175.8	−1,155.9
GEP1	T, SR, W, P and H	0.30	0.05	0.973	9	−868.7	−860.1
MLP2[a]	T and SR	0.93	0.65	0.737	17	−19.0	−2.8
RBNN2[a]	T and SR	0.92	0.65	0.736	40	19.1	57.1
GRNN2[a]	T and SR	0.94	0.66	0.724	10	−25.2	−15.7
LGP2	T and SR	0.90	0.14	0.728	5	−70.6	−65.9
GEP2	T and SR	0.96	0.14	0.719	6	−16.8	−11.1
SS[a]	T and SR	1.04	0.72	0.703	2	32.6	34.5

[a] The test results of the model were obtained from the previous study, Kisi (2009a)

Environmental Protection Agency (US EPA), are used in the current study. The elevations are 102, 30 and 4 m for the Fresno, Los Angeles and San Diego stations, respectively. The measured daily climatic data for these stations were downloaded from the US EPA web server (http://www.epa.gov/ceampubl/tools/metdata/us_met.htm).

The data sample consisted of 4 years (1987–1990) of daily records of air temperature (*T*), solar radiation (SR), wind speed (*W*), pressure (*P*), humidity (*H*) and pan evaporation (*E*). For each station, the first 3-year (1,461 daily values, 75% of the whole data 1,461) data were used to train the models and the remaining data (365 daily values, 25% of the whole data 365) were used for testing. The detailed information about the climatic data of Fresno, Los Angeles and San Diego stations can be obtained from the study of Kisi (2009a).

Application and results

Kisi (2009a) tried several input combinations using MLP for the estimation of pan evaporation in his study. Then, he compared the MLP estimates with the four- and two-parameter RBNN, GRNN and SS models. He used root mean square error (RMSE), mean absolute error (MAE) and determination coefficient (R^2) statistics for the

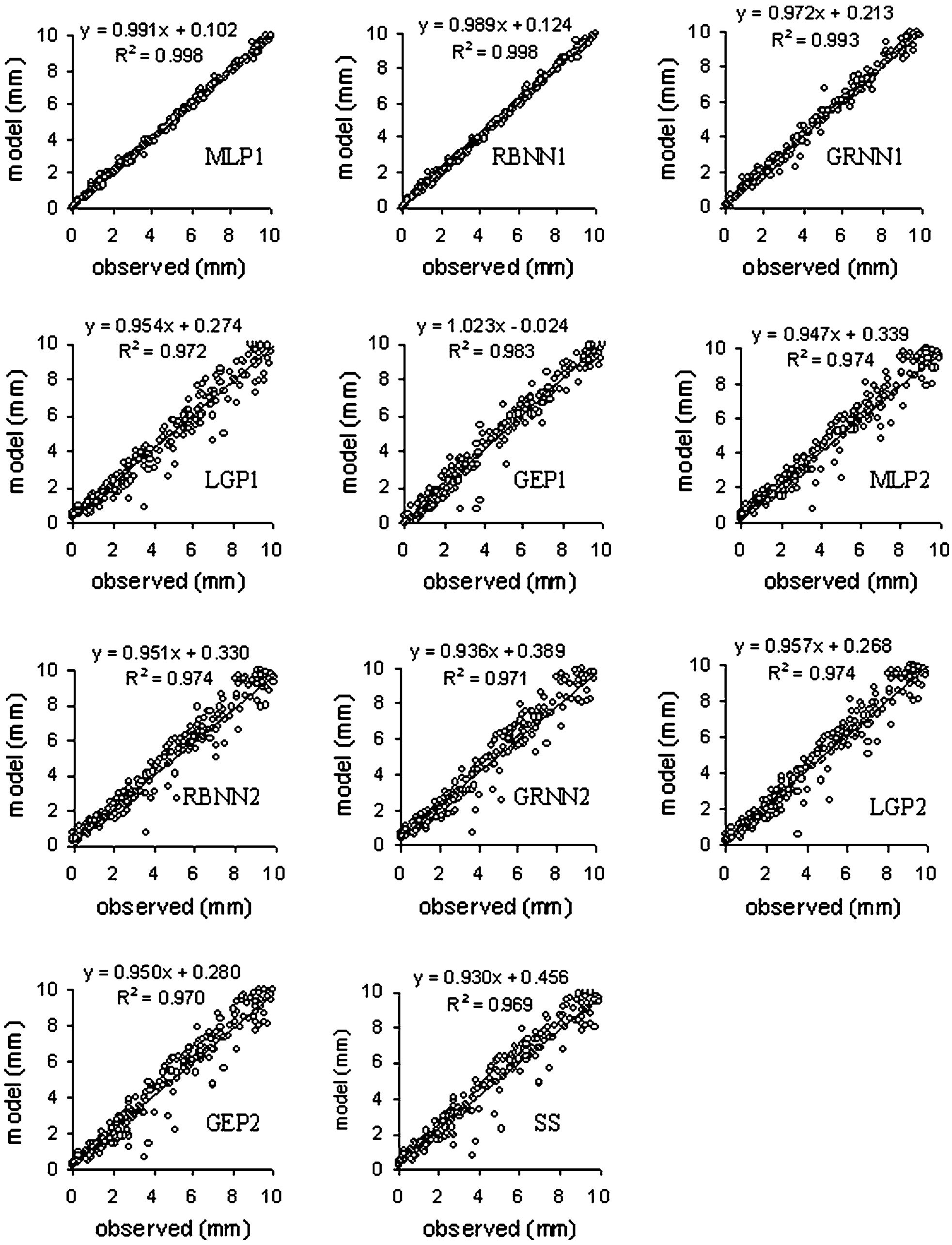

Fig. 1 The observed and estimated pan evaporation of the Fresno station in test period

Table 3 Total pan evaporation estimates in test period—Fresno station

Fresno station	Total pan evaporation (mm)	Relative error (%)
Observed	2,288	–
MLP1[a]	2,305	0.8
RBNN1[a]	2,307	0.8
GRNN1[a]	2,300	0.6
LGP1	2,284	−0.2
GEP1	2,332	1.9
MLP2[a]	2,289	0.1
RBNN2[a]	2,295	0.3
GRNN2[a]	2,283	−0.2
LGP2	2,289	0.04
GEP2	2,278	−0.4
SS[a]	2,294	0.3

[a] The test results of the model were obtained from the previous study, Kisi (2009a)

evaluation of ANN and SS models in test period, and he found that the MLP and RBNN models showed almost same accuracy and these two performed better than the GRNN and SS methods. In this study, four- and two-parameter LGP and GEP models are developed using the same data, and the results are compared with those of the MLP, RBNN, GRNN and SS models employed in Kisi (2009a). The generalization capacity of each model is also compared with each other. One of the most important issues in soft computing methods is the generalization capacity, which means that the proposed model should give the best performance in testing data with the optimal size (i.e., number of network weights employed). Particularly, ANN modeling suffers from obtaining the optimal architecture which is related to the number of neurons in hidden layer. Generally, trial-and-error approach is used in order to obtain the optimal ANN architecture (Guven et al. 2006; Guven 2009).

There are several performance criteria for measuring the generalization capacity of soft computing techniques. In the current study, Akaike information criterion (AIC) defined by Akaike (1973) and minimum description length (MDL) introduced by Rissanen (1978) are utilized to evaluate the robustness of the LGP, GEP and ANN models.

$$\mathrm{AIC} = N \times \mathrm{Ln}(\mathrm{MSE}) + 2k \tag{1}$$

$$\mathrm{MDL} = N \times \mathrm{Ln}(\mathrm{MSE}) + 0.5k \times \mathrm{Ln}N \tag{2}$$

where N is the number of samples in the testing set, MSE is mean square error, and k is the number of network weights. AIC is used to measure the exchange between testing performance and network size, while MDL criterion tries to combine the model's error with the number of degrees of freedom to determine the level of generalization. The goal is to minimize both AIC and MDL to obtain a network with the best generalization. Equations 1 and 2 imply that the value of AIC and MDL increases with increasing number of network weights (k), but if the MSE of the model is much lower than that of another model, its AIC and MDL may be lower despite its relatively larger network size. This can be attributed as a two-objective optimization problem, first objective to minimize the size of model and the second one to obtain the lowest MSE of testing whatever the size of the model is.

The formulae of the four-parameter GEP1 and LGP1 and two-parameter GEP2 and LGP2 models obtained for the Fresno Station in the present study, respectively, are

$$E = \frac{T - (3.959H \times T)/H^{-1}(T \times \mathrm{SR} + T)}{241.22 - 7.229T - W \times P} \times \left(T + \frac{\mathrm{SR}(\mathrm{SR} + W + P)}{T}\right) \tag{3}$$

$$E = \left(0.863W - 1.108\frac{T \times W}{H}\right)\frac{\mathrm{SR}^2 + T \times \mathrm{SR}}{H \times W} + 0.221\mathrm{SR} \tag{4}$$

$$E = \frac{(0.628T - 0.379 - 1.341^{0.137\mathrm{SR}+0.447})\mathrm{SR}}{(-2.597\mathrm{SR}^{-2.08} + 20.01)} \tag{5}$$

$$E = 0.047\mathrm{SR}^2 + 0.106(T - (0.175F_1 \times F_2 - F_2 + F_3) \times F_4) \tag{6}$$

where $F_1 = 0.026(T - 0.356)^2 - 0.106\mathrm{SR} - 1.416$; $F_2 = 0.106(2.4 - 1.175F_1 + 2\mathrm{T})^{0.5}$; $F_3 = 1.175 - F_1 + 4.389 F_2 - F_1 \times F_2$; $F_4 = (-0.013\ (F_2 - F_3) + 0.023\mathrm{T})^{0.5}$ are the temporary computational variables.

For the Fresno Station, the GEP1, LGP1, GEP2, LGP2 (Eqs. 3–6), MLP, RBNN, GRNN and SS models are compared in Table 2. k denotes the number of parameters used in each model. The input variables used for each model are also given in this table. The GEP2, LGP2, MLP2, RBNN2, GRNN2 and SS models use two input variables. Table 2 indicates that the five-parameter ANN models show better accuracy than the LGP1 and GEP1 from the RMSE, MAE and R^2 viewpoints. From AIC and MDL criteria, it is obvious that the MLP1 and RBNN1 models are more robust and have better generalization capacity than the other models. Two-parameter MLP2, RBNN2 and LGP2 models seem to have almost equal accuracy according to the RMSE and R^2 criteria. According to the MAE criterion, however, two-parameter LGP2 and GEP2 models perform better than the MLP2 and RBNN2. The AIC and MDL values of the MLP2 and LGP2 models are almost equal to each other, and they are lower than those of the other two-parameter models. The pan evaporation estimates of each model are illustrated in

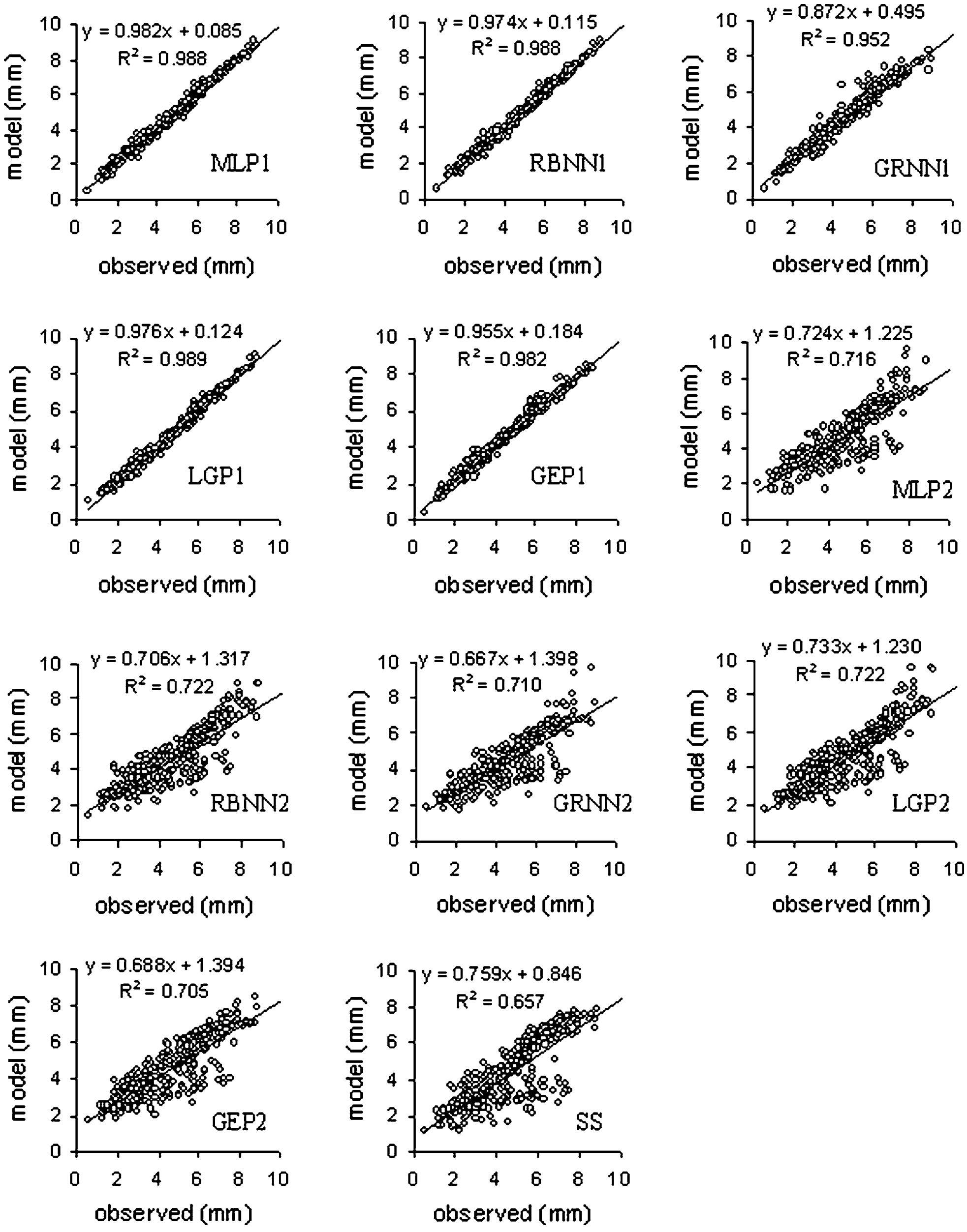

Fig. 2 The observed and estimated pan evaporation of the Los Angeles station in test period

Fig. 1 in the form of scatterplot. It is obviously seen from the scatter plots that the MLP1 and RBNN1 estimates are closer to the corresponding observed pan evaporation values comrade to those of the other models. The comparison of the models that used minimum data, i.e., MLP2, RBNN2, GRNN2, LGP2, GEP2 and the SS model, reveals that the LGP2 results are better than the other models. The MLP2, RBNN2 and LGP2 models have the same R^2 values. However, the fit line equations (assume that the equation is $y = a_0x + a_1$) in the scatterplots indicate that the a_o and a_1 coefficients for the LGP2 model are, respectively, closer to the 1 and 0 than those of the MLP2 and RBNN2 models. Total pan evaporation estimations of each model are given in Table 3. The total pan evaporation amounts were calculated by integrating the evaporation values of the test period. While the LGP1, GRNN2 and GEP2 models estimate the total pan evaporation as 2,284, 2,283 and 2,278 mm, compared to the measured 2,288 mm, with underestimations of 0.2, 0.2 and 0.4% in test period, the MLP1, RBNN1, GRNN1, GEP1, MLP2, RBNN2, LGP2 and SS models result in 2,305, 2,307, 2,300, 2,284, 2,289, 2,295, 2,289 and 2,294 mm, with overestimations of 0.8, 0.8, 0.6, 1.9, 0.1, 0.3, 0.04 and 0.3%, respectively. The LGP2 estimate is closest to the observed one. LGP1 model seems to better than the MLP1 and RBNN1 models in total evaporation estimation.

For the Los Angeles Station, the formulas of the five-parameter GEP1 and LGP1 and two-parameter GEP2 and

Table 4 Total pan evaporation estimates in test period—Los Angeles

Los Angeles station	Total pan evaporation (mm)	Relative error (%)
Observed	1,742	–
MLP1[a]	1,741	−0.1
RBNN1[a]	1,738	−0.2
GRNN1[a]	1,699	−2.4
LGP1	1,747	0.3
GEP1	1,732	−0.6
MLP2[a]	1,708	−2.0
RBNN2[a]	1,710	−1.8
GRNN2[a]	1,672	−4.0
LGP2	1,727	−0.9
GEP2	1,707	−2.0
SS[a]	1,631	−6.4

[a] The test results of the model were obtained from the previous study, Kisi (2009a)

LGP2 models obtained in the present study, respectively, are

$$E=\left(W+\frac{(\mathrm{SR}-6.567)^2}{6.029W-\mathrm{SR}}\right)^{0.5}+\left(\frac{0.35(H-\mathrm{SR})-T-W}{11.09-8.234W^{-1}}\right)+\left(\frac{\mathrm{SR}^2(W+0.18T)}{H}\right)\tag{7}$$

$$E=\left((0.211T(\mathrm{SR}+F_1)+4W+1.38)^2+(T/H)\right)^{0.5}\tag{8}$$

where $F_1 = (0.267F_2 + 1.09F_3)P^{-1}-1.505$; $F_2 = -F_4(F_5 + F_6)$; $F_3 = (-0.559(F_4 + F_2)W + \mathrm{SR} - 2.78)0.77\mathrm{T}$; $F_4 = (3F_5 + 2F_6 - P)H^{-1} + F_5 + F_6$; $F_5 = 0.288(F_6^{0.5} - T-\mathrm{SR}) - 2.451$; $F_6 = 0.288(W + H-\mathrm{SR} + P^{-1} \times H^{-1})$ are the temporary computational variables.

$$E=0.203T+0.282\mathrm{SR}-5.69T^{-1}+0.124\left(\frac{T^2-15.39}{\mathrm{SR}}\right)\tag{9}$$

$$E=0.472(\mathrm{SR}+F_1)-\frac{0.051}{F_2}\tag{10}$$

where $F_1 = 2F_3 + F_2 + F_4 + 2(F_5 \times F_3) + F_5 \times (F_3/F_6)$; $F_2 = (F_4 \times (F_1 - F_2) + 1.37)/(F_1 - F_2)$; $F_3 = 0.08(\mathrm{SR}^2/F_6) + F_5$; $F_4 = 0.63(F_6 \times (F_3 - F_6) \times F_3 + 0.92)$; $F_5 = 0.09(T - 0.49F_5^{0.5} - (F_3 - F_5) - 1.78)$; $F_6 = 0.69(0.1\mathrm{SR} + 0.1\mathrm{SR}^2)^4 + 0.996$ are the temporary computational variables.

The test results of the GEP1, LGP1, GEP2, LGP2 (Eqs. 7–10), MLP, RBNN, GRNN and SS models are compared for the Los Angeles Station in Table 2. It can be seen from the Table 2 that the LGP1 model has the lowest RMSE (0.19 mm), MAE (0.05) and the highest R^2 (0.989) statistics. The lowest AIC and MDL values of the LGP1 show its best robustness and the generalization capacity. Among the two-parameter models, the LGP2 model performs better than the others according to the various performance criteria. Figure 2 demonstrates the pan evaporation estimates of each model. The difference between the LGP1 estimates and those of the MLP1 and RBNN1 models cannot be seen from the scatterplots. However, the superiority of the LGP2 model to the other two-parameter models is obviously seen from the fit line equations and R^2 values. The overestimations and underestimations are clearly seen for the two-parameter MLP2, RBNN2, GRNN2, LGP2, GEP2 and SS models. Total pan evaporation estimations of each model in test period are compared for the Los Angeles Station in Table 4. The MLP1, RBNN1, GRNN1, GEP1, MLP2, RBNN2, GRNN2, LGP2, GEP2 and SS models, respectively, estimate the total pan evaporation as 1,741, 1,738, 1,699, 1,732, 1,708, 1,710, 1,672, 1,727, 1,707 and 1,631 mm compared to the measured 1,742 mm, with underestimations of 0.1, 0.2, 2.4, 0.6, 2, 1.8, 4, 0.9, 2 and 6.4% in test period, while the LGP1 results in 1,747 mm, with an overestimation of 0.3%. The MLP1 gives the best estimate. RBNN1 is ranked as the second best. The LGP1 estimate is close to those of the MLP1 and RBNN1 models. Unlike the Fresno Station, the models using two inputs give the worse total pan evaporation estimates than the others. Out of the two-parameter models, the LGP2 estimate is closest to the observed one.

The formulas of the five-parameter GEP1 and LGP1 and two-parameter GEP2 and LGP2 models obtained for the San Diego Station in the present study, respectively, are

$$E=\frac{(2\mathrm{SR}+(W/P)-H+W)^{0.5}P}{W+0.47P\times H+T}\left(T+\frac{0.176\mathrm{SR}}{P}+\mathrm{SR}\right)\tag{11}$$

$$E=0.139\left(0.106\left(F_1^2-F_2-\mathrm{SR}\right)+3P-1.714\right)+0.494\tag{12}$$

$$E=(0.19-0.19T)(1.46\mathrm{SR}+0.19)(0.433-0.15/T)\tag{13}$$

$$E=\left(0.006T\left(\left(\frac{(T-2\mathrm{SR})^2+\mathrm{SR}^2}{5.56\times10^4}+\mathrm{SR}\right)\mathrm{SR}-T\right)+2\mathrm{SR}\right)^{0.5}\tag{14}$$

where $F_1 = 0.118(W - H+10.65\mathrm{SR} + P) - 0.38$; $F_2 = F_3 - 2F_4 + F_5$; $F_3 = 5\mathrm{T} - 2(\mathrm{SR} + H)$; $F_4 = 0.23(F_2 \times P-\mathrm{SR}) + P$; $F_5 = 0.05(1.189(F_6^{0.5} \times \mathrm{SR} \times P^{0.25}) - 2F_7 - T)$; $F_6 = F_8 + F_3 - 2F_4 + 2\mathrm{T} - H$; $F_7 = 2(T - H) + W-\mathrm{SR}$; $F_8 = (1.4 - 0.09(P^2 + P))\mathrm{SR} + 0.67\mathrm{SR})$ are the temporary computational variables.

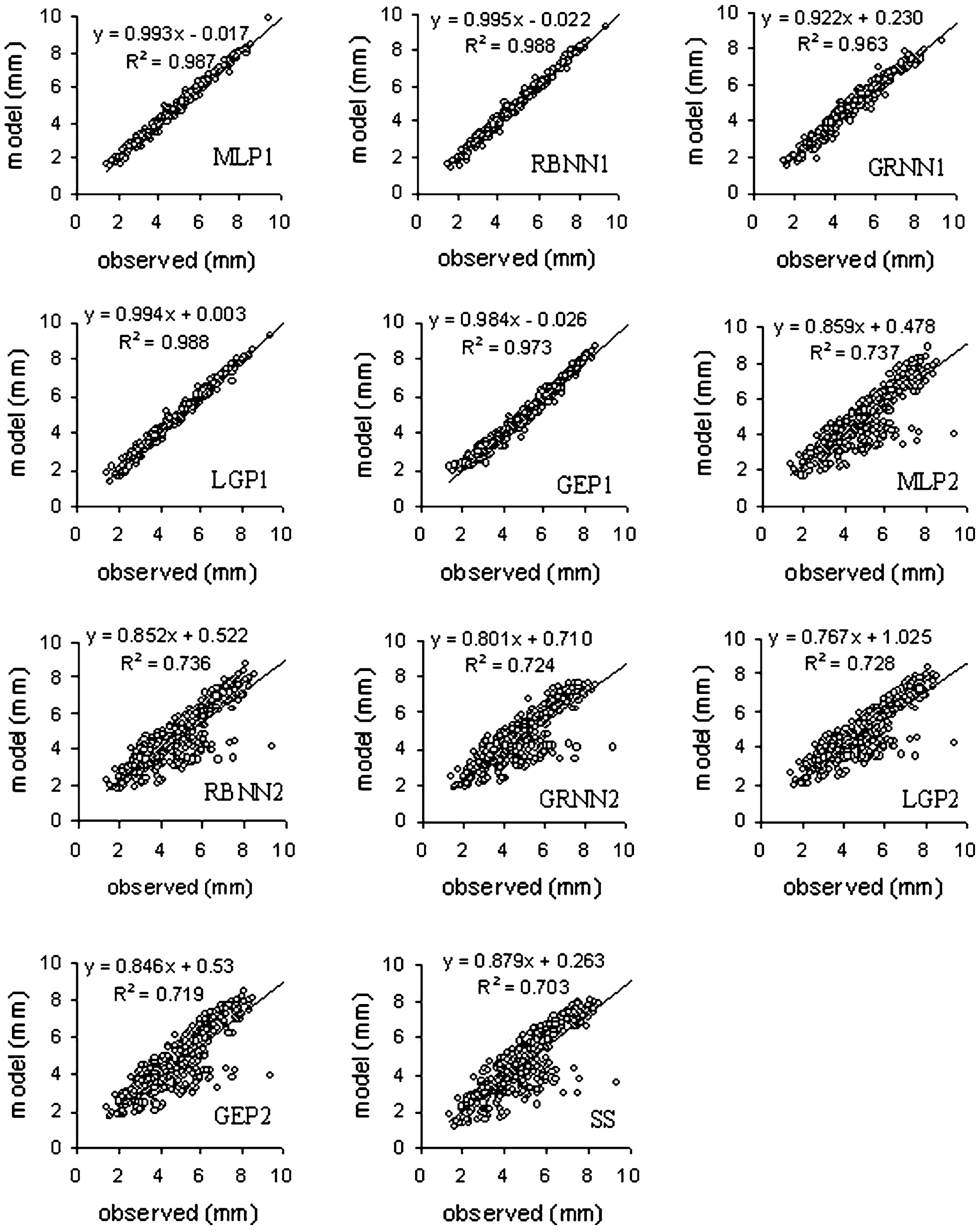

Fig. 3 The observed and estimated pan evaporation of the San Diego station in test period

For the San Diego Station, the test results of the GEP1, LGP1, GEP2, LGP2 (Eqs. 11–14), MLP, RBNN, GRNN and SS models are given in Table 2. As found in the previous application, here also the LGP1 outperforms all the other models from the RMSE, MAE and R^2 viewpoints. The generalization capacity of the LGP1 model is higher than those of the ANN models (see the AIC and MDL values). The comparison of the two-parameter MLP2, RBNN2, GRNN2, LGP2, GEP2 and SS models reveals that the LGP2 results are better than the other models from the various performance criteria (see Table 2). The observed and estimated pan evaporations of the San Diego Station in test period are shown in Fig. 3. It can be obviously seen from the fit line equations and R^2 statistics that the LGP1 is superior to the other models. The MLP1 and RBNN1 models are slightly worse than the LGP1. Table 5 compares the total pan evaporation estimations of each model in test period. The total pan evaporation estimates of the MLP1, RBNN1, GRNN1, LGP1, GEP1, MLP2, RBNN2, GRNN2, LGP2, GEP2 and SS models are 1,782, 1,785, 1,745, 1,793, 1,764, 1,722, 1,725, 1,702, 1,758, 1,719 and 1,680 mm, with underestimation errors of 1.1, 0.9, 3.1, 0.5, 2.1, 4.4, 4.3, 5.6, 2.4, 4.6 and 6.8% in test period, respectively. The LGP1 seems to have the best estimate. Here, also the RBNN1 is ranked as the second best. Among the two-parameter models, the LGP2 estimate is closest to the observed one.

Table 5 Total pan evaporation estimates in test period—San Diego

San Diego station	Total pan evaporation (mm)	Relative error (%)
Observed	1,802	–
MLP1[a]	1,782	−1.1
RBNN1[a]	1,785	−0.9
GRNN1[a]	1,745	−3.1
LGP1	1,793	−0.5
GEP1	1,764	−2.1
MLP2[a]	1,722	−4.4
RBNN2[a]	1,725	−4.3
GRNN2[a]	1,702	−5.6
LGP2	1,758	−2.4
GEP2	1,719	−4.6
SS[a]	1,680	−6.8

[a] The test results of the model were obtained from the previous study, Kisi (2009a)

Conclusions

This study investigated the ability of LGP in modeling daily pan evaporations. The accuracy of LGP has been compared to those of the GEP developed by authors and MLP, RBNN, GRNN, SS methods obtained from the previous study of Kisi (2009a). The daily climatic data of three automated weather stations, Fresno, Los Angeles and San Diego in California, are used for the model simulations. The comparison results indicated that, in general, the LGP1 model whose inputs are the *T*, SR, *W*, *P* and *H* are found to perform better than the four-parameter GEP1, MLP1, RBNN1, GRNN1 models in the estimation of daily pan evaporations. Out of the two-parameter models, the LGP2 models were generally found to be better than the GEP2, MLP2, RBNN2, GRNN2 and SS models. The LGP2 models can be successfully used in estimation of daily pan evaporations where there exist only the *T* and SR data. The total evaporation estimates of the LGP models were compared with those of the ANN and SS models. The comparison results revealed that the LGP models performed better than the other models in the estimation of total evaporation.

The LGP model presented in this study is a simple explicit mathematical formulation. However, the ANNs are black-box models that are the inputs, and outputs are known but the box is close and the model formulation is implicit. Input data are usually introduced into the black-box, and the output is obtained without understanding what happens inside the box. The LGP, which is relatively much simpler than the ANN, can be successfully used in modeling daily pan evaporations.

References

Akaike H (1973) Information theory and an extension of the maximum likelihood principle. In: Petrov BN, Csaki F (eds) Proc., 2nd Int. Symp. on information Theory. Academiai Kiado, Budapest, Hungary, pp 267–281

Al-Ghobari HM (2000) Estimation of reference evapotranspiration for southern region of Saudi Arabia. Irrig Sci 19(2):81–86

Aytek A, Kisi O (2008) A genetic programming approach to suspended sediment modeling, *J*. Hydrology 351(3–4):288–298

Babovic V, Keijzer M (2002) Declarative and preferential bias in GEP-based scientific discovery. Genet Program Evolvable Mach 3(1):41–79

Banzhaf W, Nordin P, Keller RE, Francone FD (1998) Genetic programming: an introduction. Morgan Kaufmann, San Francisco

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

Brameier M, Banzhaf W (2001) A comparison of linear genetic programming and neural networks in medical data mining. IEEE Trans Evol Comput 5:17–26

Brutsaert WH (1982) Evaporation into the atmosphere. D. Reidel Publishing Company, Dordrecht

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

Ferreira C (2001a) Gene expression programming in problem solving. In: Proceedings of the 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

Giustolisi O (2004) Using genetic programming to determine Chezy resistance coefficient in corrugated channels. J Hydroinformatics 6(3):157–173

Gundekar HG, Khodke UM, Sarkar S, Rai RK (2008) Evaluation of pan coefficient for reference crop evapotranspiration for semi-arid region. Irrig Sci 26(2):169–175

Guven A (2009) Linear genetic programming for time-series modeling pf daily flow rate. J Earth Syst Sci 118(2):137–146

Guven A, Aytek A (2009) New approach for stage-discharge relationship: gene-expression programming. J Hydrol Eng 14(8):812–820

Guven A, Gunal M (2008a) Prediction of scour downstream of grade-control structures using neural networks. J Hydraulic Eng 134(11):1656–1660

Guven A, Gunal M (2008b) Genetic programming for prediction of local scour downstream of grade-control structures. J Irrig Drainage Eng 134(2):241–249

Guven A, Kişi Ö (2009) Suspended sediment modeling using linear genetic modeling. Hydrol Sci J (under review)

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

Guven A, Aytek A, Yuce MI, Aksoy H (2007) Genetic programming-based empirical model for daily reference evapotranspiration estimation. Clean-Soil Air Water 36(10–11):905–912

Guven A, Azamathulla HMd, Zakaria NA (2009) Linear genetic programming for prediction of circular pile scour. J Ocean Eng 36(12–13):985–991

Keskin ME, Terzi O (2006) Artificial neural network models of daily pan evaporation. J Hydrol Eng 11(1):65–70

Kisi O (2009a) Daily pan evaporation modeling using multi-layer perceptrons and radial basis neural networks. Hydrol Process 23:213–223

Kisi O (2009b) Modeling monthly evaporation using two different neural computing techniques. Irrig Sci 27(5):417–430

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. The MIT Press, Cambridge

Kumar M, Bandyopadhyay A, Raghuwanshi NS, Singh R (2008) Comparative study of conventional and artificial neural network-based ETo estimation models. Irrig Sci 26(6):531–545

Nordin P (1994) A compiling genetic programming system that directly manipulates the machine-code. In: Kinnear KE (ed) Advances in genetic programming. MIT Press, Cambridge, pp 311–331

Nordin P (1997) Evolutionary program induction of binary machine code and its applications. Ph.D. dissertation: Dept. Comput. Sci., Univ. Dortmund

Nordin P, Banzhaf W (1995) Evolving turing-complete programs for a register machine with self-modifying code. In: Eshelman L (ed) Proceedings of the 6th international conference of genetic algorithms. Morgan Kaufmann, Pittsburgh, PA, USA, pp 318–325

Oltean M, Grosan C (2003) A comparison of several linear genetic programming techniques. Complex Syst 14(1):1–29

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

Rahimikhoob A (2009) An evaluation of common pan coefficient equations to estimate reference evapotranspiration in a subtropical climate (north of Iran). Irrig Sci 27(4):289–296

Rissanen J (1978). Modeling by the shortest data description. Automatica 14:465–471

Stephens JC, Stewart EH (1963) A comparison of procedures for computing evaporation and evapotranspiration. Publication 62, international association of scientific hydrology. International Union of Geodynamics and Geophysics, Berkeley, CA, pp 123–133

Tabari H, Marofi S, Sabziparvar A-A (2010) Estimation of daily pan evaporation using artificial neural network and multivariate nonlinear regression. Irrig Sci 28(5):399–406. doi:10.1007/s00271-009-0201-0

Whigham PA, Crapper PF (2001) Modeling rainfall-runoff using genetic programming. Math Comput Model 33:707–721