Rainfall–runoff modelling using genetic programming

K. Rodríguez-Vázquez, M. L. Arganis-Juárez, C. Cruickshank-Villanueva and R. Domínguez-Mora

ABSTRACT

This paper presents the application of genetic programming to the generation of models to assess the total runoff of a basin starting from the total rainfall in it and using data recorded in a sub-basin at the valley of Mexico (the Mixcoac sub-basin to the west of Mexico City). The modelling process is developed contrasting two types of models with different complexity degree: (1) a nonlinear model whose complexity is resolved using multi-objective optimization and (2) a nonlinear model with a given structure obtained by means of a physical interpretation of the dynamics of the direct and the base flow. Data from two storms (rainfall and runoff), one in 1997 and another in 1998, were used in testing the models. First, the storm in 1997 was used for the calibration step and that in 1998 for the validation step. Afterwards, the order was reversed. An interpretation of the results, focused on the applicability and possible improvement of the models in forecasting runoff, is made through their discussion and is summarized in the conclusions.

Key words | genetic algorithm, genetic programming, NARMAX model, rainfall-runoff modelling

K. Rodríguez-Vázquez (corresponding author)
IIMAS-UNAM,
Circuito Escolar, CU,
Coyoacan,
México City,
México
E-mail: *katya@uxdea4.iimas.unam.mx*

M. L. Arganis-Juárez
C. Cruickshank-Villanueva
R. Domínguez-Mora
Instituto de Ingeniería, UNAM,
Circuito Escolar,
CU, Coyoacan,
México City,
México

INTRODUCTION

One of the fundamental problems in hydrology is the modelling of the rainfall–runoff process, among other things, to be able to perform short-term forecasting of the flood likely to be induced by a given storm. Depending on the information available, the forecast can be developed from rainfall records only, or otherwise through the use of runoff measured prior to the event analyzed at a particular site.

Conventional models are generally based on excess rainfall and on direct runoff; nevertheless, the forecasting of excess rainfall is not an easy task and it is therefore desirable to develop models that take into account total rainfall and total runoff.

This paper presents the use of recently developed genetic techniques such as genetic programming (GP) (Koza 1992; Banzhaf *et al.* 1998) applied to the determination of the structure and the parameters of nonlinear autoregressive models (NARMAX) (Leontaritis & Billings 1985). On the other hand, genetic algorithms (GA) (Holland 1975; Goldberg 1989) are also proposed to obtain parameters from a non-conventional rainfall–runoff model, based on the physical interpretation of the dynamics of the direct and the base flow of a nonlinear model. In the two cases the models use data of total rainfall as well as total runoff.

For the calibration and the validation processes, data were recorded in the Mixcoac sub-basin to the west of Mexico City, Federal District, Mexico (Figure 1). Through the interpretation of the results, the advantages and limitations of the two models are discussed.

BACKGROUND

Different models are available to obtain equations that describe the rainfall–runoff relationship process whose applicability depends on the extension of the basin. In the case of urban basins, the method of the American Rational Formula is one of the most accepted and used (Aparicio 1989); when dealing with basins of natural river courses, several hydrologic procedures such as the method of the unit

doi: 10.2166/hydro.2011.105

Figure 1 | Location of the Mixcoac sub-basin, Mexico City, Federal District, Mexico.

hydrograph in its various versions such as, for example, the instantaneous geo-morphological unit hydrograph method (Eslava 1997) are used. The method proposed by the Soil Conservation Service (Aparicio 1989) is widely used to estimate the effective rainfall. Some of these methods require limited information for application purposes whereas some others demand a wide knowledge of the physiographic characteristics of the basin analyzed.

In the context of forecasting models, Chong (2002) indicates that distributed models can be identified as: based on physical fundamentals (Systeme Hydrologique Européen (SHE) and the TOP-MODEL), based on concentrated conceptual models (Sacramento, TANK, CLS) and as black-box models (the unit hydrograph and stage-regression techniques are included). Forecasting models are mainly used to solve problems of flash floods or floods induced by hurricane events.

The use of genetic programming (GP) to forecast real-time runoff has expanded in recent years. Madsen *et al.* (2000) and Drécourt & Madsen (2001) carried out comparisons on the use of auto-regressive models, AR(p), using genetic programming and neural networks in the correction of the residual error of a calibrated model (e.g. the MIKE 11/NAM model). In order to evaluate the effect of the simulation model's quality on forecasting, error correction techniques (or updating of output values) were applied to both the calibrated model and to a non-calibrated one.

Whigham & Crapper (2001) proposed the use of GP based on a context-free grammar. This work defined a function set composed of arithmetic and exponential functions. The terminal set included past rainfall values as well as the average rainfall for the last 5, 10, 15, 25, 30, 40, 50, 60 and 100 d. In 2002, Liong *et al.* applied GP in order to determine a model that described the relationship between rainfall–runoff using the classical encoding of GP as defined by Koza (1992). The function set consisted of arithmetic and, again, exponential functions, formulating the modelling problem as a regression. These two works produced models that did not provide a physical interpretation of the phenomenon.

Khu *et al.* (2004) used genetic programming and neural networks to generate an evolutionary-based real-time error updating scheme to supplement a real-time forecasting model known as WRIP (Weather Radar Information Processor) based on radar-recorded rainfall measurements.

Tests were carried out using total and effective rainfall, with both neural networks and genetic programming for error optimization. For calibration purposes, data recorded during the rainfall of December 1999 in the rural basin upstream from Taunton, UK were used and, for the validation process, an estimate was made using data from April 2000. It was possible to improve the runoff forecast with the WRIP model using both genetic programming and an artificial neural network by updating the real-time error between the measured runoff and the simulated value for up to five time intervals. The equation derived from the genetic programming can be interpreted as an advanced form of auto-regressive model.

Rabuñal *et al.* (2007) present a combination of artificial neural network (ANN) and GP for the prediction of runoff in an urban area. The base flow consists mainly of the discharge from home drainage of the inhabitants and has a very predictable pattern; this part is assigned to the ANN. In rain events the flow pattern is altered and the task of a rainfall–runoff model definition is left to GP. Nevertheless, the structure was assigned to the model to have a filter-like form, with a time delay and a exponential recession. This gives a transformation of the rain signal into a runoff response very similar to a unit hydrograph. When this is transported to discrete time intervals they obtain a linear autoregressive model with very good results

in forecasting; they emphasize that this is due to the restrictions imposed on the GP algorithm. They compared their results with those of a GP single-input single-output (SISO) NARMAX model (Rodríguez-Vázquez 2001) with no flow separation, whose results were not far behind theirs in forecast accuracy.

A recent work by Guven (2009) applied the well-known linear GP introduced by Banzhaf *et al.* (1998), defining again a similar function set as that in the work by Whigham & Crapper (2001) and Liong *et al.* (2002). These approaches presented a relatively good performance but it was difficult to show they could give insight about a physical interpretation from the generated models.

In some studies, the runoff at time t is estimated in terms of runoff occurring immediately before without involving rainfall terms. Otherwise, terms of evapotranspiration and rainfall and values of runoff in previous times are included. This is the case in a study made by Savic *et al.* (1999) that includes rainfall–runoff modeling and the identification of systems using genetic programming. The problem of system identification carries input values to an output value; the methodology involves the subdivision of the observed record set into a smaller one for the calibration period; only calibration data are used to evaluate the fitness of the genetic programming and therefore a training error is estimated. A different subset of data is used for validation purposes than in obtaining the testing error. In a first experiment, synthetically generated data were used based on a potential-type ($Q = ah_{,}^{b}$ where Q is discharge, h corresponds to rainfall, and a and b are parameters) rainfall–runoff model, assuming certain values of the parameters; subsequently, different models were obtained with genetic programming and various generations. A good solution could be found with 50 generations, determining a nonlinear model in which the flow rate at time t becomes a function of the rainfall at the same moment ($Q = a\ \sin(h)\ h + b$).

The model was applied to original data and an error term of 5% (white noise) was added to consider possible errors inherent to the data. Subsequently, these authors took into account real measured data such as rainfall, runoff and evapo-transpiration in a basin in Scotland; they obtained nonlinear models able to estimate the runoff at time t as a function only of the rainfall recorded at previous moments (at three previous intervals) or, otherwise, as a function of both the rainfall at the time of interest and up to three moments before, and the runoff at the previous moment.

METHODOLOGY

A simple genetic algorithm was used in this analysis for the optimization of parameters of the proposed models. An algorithm of genetic programming that makes it possible to fit a function with several variables was also used.

Simple genetic algorithm

The traditional genetic algorithm (Holland 1975; Goldberg 1989) generates an initial population of n individuals (in this case, parameters of the models); its fitness is evaluated with an objective function. A selection is made of the best fitted individuals with the universal stochastic method or with the roulette procedure (Goldberg 1989). Those individuals are subjected to the crossover and mutation operators and a new population is created, with n individuals that move to the next generation. The fitness evaluation process, the selection of the most suitable individuals, crossover and mutation operations and the creation of new populations are iterated until a number of generations previously established is achieved.

Genetic programming

The genetic programming algorithm (Koza 1992; Banzhaf *et al.* 1998) is a sub-class of the well-known genetic algorithm. Typically, it involves the random generation of an initial population of trees constituted by a set of functions and variables relevant to the problem to be solved, defining the objective function to evaluate the fitness of each individual. Then, as in the case of traditional genetic algorithms, a selection is made of individuals with the best fitness, and they are subjected to the operators of crossover, mutation and reproduction to be able to generate a new population representing the next generation.

Genetic programming for system identification

A common problem in the area of hydraulic engineering is the modeling and identification of systems for forecasting purposes; this process is defined as the construction of a mathematical model based on inputs and outputs of the system under study. This problem becomes complex when aspects such as nonlinearity are considered. The nature of this type of problem (frequently in a complex solution environment) lends itself to the application of genetic programming.

The model used in this work is based on the NARMAX model which is the nonlinear version of the ARMAX (Auto-Regressive Moving Average with eXogenous inputs) as expressed by Leontaritis & Billings (1985).

In order to get into the context of the proposed models, they are presented in detail in this section. First, the NARMAX model, which was calibrated considering a multi-objective function and then the NLAPI (NonLinear Antecedent Precipitation Index) model, which assumes a fast dynamics for the direct flow and a slow one for the base flow, are described.

ARMAX model

One of the most common structures representing linear models refers to the ARMAX model expressed by

$$y(k) = -\sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_e} c_i e(k-i) + e(k) \tag{1}$$

where $a_i \in \Re$, $b_i \in \Re$ and $c_i \in \Re$ are the model coefficients; and $y(k)$, $u(k)$ and $e(k)$ represent the vectors of length n_y, n_u and n_e, respectively, of data related to output, input and noise of the system, respectively. Based on this model, different algorithms of parametric estimation have been developed (Ljung 1987; Söderström & Stoica 1989).

NARMAX model

Leontaritis & Billings (1985) introduced the extension of the description of the ARMAX model with the purpose of representing nonlinear systems. According to this, the NARMAX (Nonlinear Auto Regressive Moving Average with eXogenous inputs) model is expressed as a nonlinear function $F^{\ell}(\bullet)$ of the sequences of output $y(k)$, input $u(k)$ and noise $e(k)$ of the system:

$$\begin{aligned} y(k) = F^{\ell}(&y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, \\ &u(k-n_u), e(k-1), \ldots, e(k-n_e)) + e(k) \end{aligned} \tag{2}$$

where n_y, n_u and n_e are the size of the previous samples considered as part of the function for output, input and noise signals, respectively, and ℓ is the degree of nonlinearity of the model. In the case where $\ell = 1$, the resulting model is a linear function.

Taking as a basis the nonlinear model of Equation (2), Chen & Billings (1989) demonstrated that the polynomial form of the NARMAX model is the most common expression that has proven to be fitted for practical applications. Therefore, Equation (2) can be expressed as a polynomial form defined as

$$\begin{aligned} y(k) = \theta_0 &+ \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \theta_{i_1} \theta_{i_2} x_{i_1}(k) x_{i_2}(k) \\ &+ \cdots + \sum_{i_1=1}^{n} \cdots \sum_{i_\ell=i_{\ell-1}}^{n} \theta_{i_1} \cdots \theta_{i_\ell} x_{i_1}(k) \cdots x_{i_\ell}(k) + e(k) \end{aligned} \tag{3}$$

where $n = n_y + n_u + n_e$, θ_i are scalar coefficients and $x_i(k)$ represent linear terms and nonlinear terms produced by the combinations in $y(k)$, $u(k)$ and $e(k)$. The polynomial model expressed in Equation (3) is of nonlinear nature in the variables of output, input and noise, but linear in what refers to term coefficients. Therefore, the coefficients associated with the model can be estimated by means of a least-squares algorithm. However, the main concern here is to determine the optimum structure of the model, i.e. the relevant terms and the nonlinear nature of the system.

Identification method based on Genetic Programming

In the case of Genetic Programming there is no further restriction with respect to the maximum order of nonlinearity or to the maximum number of terms in the model.

Table 1 | Parameters for the problem of modelling and system identification using genetic programming

Parameter	Description
Objective:	To find a mathematical model that reproduces the input–output relationship of the system under study
Terminals set:	$y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)$[a]
Functions set:	ADD, MULT
Fitness cases:	Number of input–output data points
Objective function:	Minimization of the mean quadratic error or a multi-criteria function
GP parameters:	MaxGen = 200, PopSize = 100, MaxDepthIni[b] = 4, MaxDepth[c] = 7
GP operators:	Crossover: 0.95, Mutation: 0.05
Finalization criterion:	Maximum number of generations

[a] Where $y(k-n_y)$ and $u(k-n_u)$ represent flow rate and rainfall, respectively.
[b] MaxDepthIni = Maximum depth defined for creating initial population and used also for mutation.
[c] MaxDepth = Maximum depth allowed after applying genetic operators.

Table 1 provides information about the parameters in the case of identification of nonlinear systems through the use of Genetic Programming (GP).

The function set has been limited to addition and multiplication because it is thought here that those are the only operations which can be given a physical interpretation in a rainfall–runoff relation. Also, the order of the polynomial obtained was restricted to 3 for the same reason and the number of terms to be less than 9 to respect parsimony.

GP encoding of NARMAX structures

The mapping process of NARMAX structures into a GP tree representation is detailed in what follows. The polynomial form of this structure can be straightforwardly expressed as a tree. Addition and product functions are only required and associated coefficients are estimated by means of a least-squares (LS) algorithm. In Figure 2, this process is exemplified. At the root node, the polynomial expression is defined and then a least-squares algorithm is applied based on measured data in order to get the set of coefficients. That is

$$y(k) = P_i(k)\hat{\theta} + \varepsilon(k) \tag{4}$$

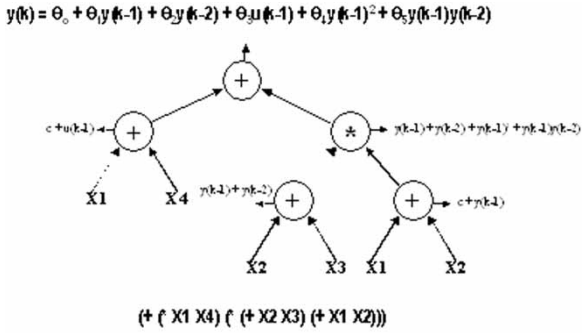

Figure 2 | NARMAX polynomial encoding.

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \quad P^T = \begin{bmatrix} p_1(k) \\ p_2(k) \\ \vdots \\ p_n(k) \end{bmatrix} = \begin{bmatrix} 1.0 \\ y(k-1) \\ y(k-2) \\ u(k-1) \\ y(k-1)^2 \\ y(k-1)y(k-2) \end{bmatrix}$$

$$\hat{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \tag{5}$$

and $\hat{\theta}$ is estimated as

$$\hat{\theta} = \left[P^T P\right]^{-1} P^T Y \tag{6}$$

It is important to point out that duplicated rows in matrix $\boldsymbol{P}$ (see Equation (15)) are deleted before the coefficient estimation and individual evaluation stage (those are eliminated during the decoding process). Thus, redundant terms in the model are removed but this fact does not

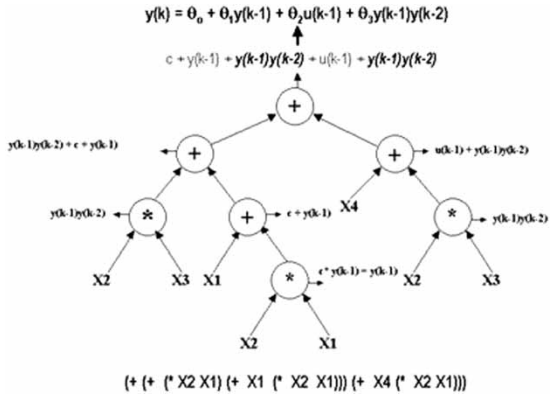

Figure 3 | NARMAX polynomial encoding (duplicated terms).

mean they are removed from GP individual trees. An example is shown in Figure 3. Decoding the GP expression, the following $\boldsymbol{P}$ matrix is obtained:

$$P^T = \begin{bmatrix} 1.0 \\ y(k-1) \\ y(k-1)y(k-2) \\ u(k-1) \\ y(k-1)y(k-2) \end{bmatrix}$$

Eliminating duplicated rows, the $\boldsymbol{P}$ matrix is reduced to

$$P'^T = \begin{bmatrix} 1.0 \\ y(k-1) \\ u(k-1) \\ y(k-1)y(k-2) \end{bmatrix}$$

Then, the least-squares algorithm is applied using $\boldsymbol{P}'$.

Genetic programming operators

Crossover, a sexual operator, works by first selecting a pair of structures from the current population. Then, a node rooted from each parent is randomly selected. These nodes become the roots for the substructures lying below the crossover point. In the next step, the substructures are exchanged between the parents, producing two new structures which are usually of different sizes to their parents. Mutation operates by randomly selecting a node, which can be either a terminal or internal point, and replacing the associated substructure with a randomly generated sub-tree up to a maximum size. A Maximum Mutation Size (MMS) parameter is introduced which is different front the maximum tree size parameter MS.

MULTI-OBJECTIVE FITNESS FUNCTION

Reformulation of the modeling and system identification processes as a multiple-objective problem is detailed in this section.

Multi-objective (or multi-criteria) optimization

Multi-objective optimization is defined as the process of finding a vector of decision variables that satisfies a set of constraints and optimizes the vector of functions whose elements represent the objective function. In general, these functions are in conflict among each other. Therefore, the term *optimize* means finding a solution that provides *acceptable* values in all objectives (Coello *et al.* 2002).

It can be mathematically expressed as:

Finding a vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ that satisfies the m inequality constraints

$$g_i(\bar{x}) \geq 0 \quad i = 1, \ldots, m \tag{7}$$

the p equality constraints

$$h_i(\bar{x}) = 0 \quad i = 1, \ldots, p \tag{8}$$

and optimizes the function vector

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})] \tag{9}$$

where $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

The set of constraints defines the feasible region $\boldsymbol{X}$; any point $\bar{x}$ in $\boldsymbol{X}$ is defined as a feasible solution. The k components of vector $\bar{f}(\bar{x})$ represent the objectives in conflict to be evaluated. Functions $g_i(\bar{x})$ and $h_i(\bar{x})$ represent the constraints imposed on the decision variables. Vector $\bar{x}^*$ denotes the optimal solution set.

From the traditional mathematical point of view, the optimization concept becomes impossible when multiple criteria are involved (Keeney & Raiffa 1976). It is therefore determined that the solution of a multi-objective problem is represented by a set of alternative solutions rather than by a single solution. This concept is known as Pareto-optimal.

Definition 1. ***Pareto optimality:*** *A solution $\bar{x}_u \in U$ corresponds to the Pareto optimum if and only if there is no $\bar{x}_v \in U$ for which $\boldsymbol{v} = f(\bar{x}_v) = (v_1, \ldots, v_n)$ dominates $\boldsymbol{u} = f(\bar{x}_u) = (u_1, \ldots, u_n)$.*

Pareto-optimal generally produces not a single solution but an array of solutions, also known as not-dominated or not-inferior solutions.

Definition 2. ***Pareto dominance:*** *A vector $\boldsymbol{u} = (u_1, \ldots, u_n)$ is said to dominate a vector $\boldsymbol{v} = (v_1, \ldots, v_n)$ if and only if $\boldsymbol{u}$ is*

partially smaller than $\boldsymbol{v}$ ($u\ p \prec v$). *That is to say*

$$\forall i \in \{1,\ldots,n\}, \quad u_i \leq v_i \quad \wedge \quad \exists i \in \{1,\ldots,n\}: u_i < v_i$$

Multi-objective genetic programming is an option in the process of modeling and system identification where aspects related to the complexity and quality of the identified model for forecasting purposes can simultaneously be evaluated. Thus, the predictive error for a storm occurred in a certain period of time (training data) as well as the predictive error of data corresponding to a storm occurred in another period of time (testing data), considering also the reduction of the complexity of the model (i.e. simple models with few, preferably linear, terms) can be simultaneously optimized producing a simple model with good performance.

The objective function (OF) proposed to assess the quality of the forecasting produced by GP is expressed in Equation (10), where Q_t is the estimated flow rate, q_t is the measured flow rate and NP the number of points:

$$OF = \min \sum_{i=1}^{NP} \frac{(Q_t - q_t)^2}{NP} \tag{10}$$

The complexity of the model is estimated by minimizing the number of terms in the model and by minimizing ℓ (the degree of nonlinearity), where the minimum value is $\ell = 1$ corresponding to a linear model.

NLAPI (nonlinear antecedent precipitation index) model

In Cruickshank (1996), an analysis of two response dynamics was presented: a fast and a slow dynamics. The author states that the total runoff generated at a basin can be determined by

$$Q_t = Q_{at} + Q_{bt} \tag{11}$$

where

$$Q_{at} = (a_0 + a_1 A_{t-ra}) hp_{t-ra} + a_2 Q_{at-1} + e_{at} \tag{12}$$

$$Q_{bt} = (b_0 + b_1 A_{t-rb}) hp_{t-rb} + b_2 Q_{bt-1} + e_{bt} \tag{13}$$

where Q_t is the total runoff at instant t [L/T^3], Q_{at} is the runoff corresponding to the quick response system (direct runoff) at instant t [L/T^3], Q_{bt} represents the runoff corresponding to the slow-response system (base) at instant t [L/T^3], hp_t is the total rainfall in the basin at instant t [L], ra corresponds to the delay in the response to the rainfall in direct runoff [T], rb is the delay in the response to the rainfall in base runoff [T], a, b are modeling parameters, A_t is the index of antecedent precipitation at instant t and e_t represents the error of the model at instant t.

The index of antecedent precipitation A_t is determined recurrently and it consists of a weighted sum of rainfall values previous to instant t:

$$A_t = (1 - \beta) A_{t-1} + \beta p_t \tag{14}$$

For hourly periods the weighting factor β is about 1/20, and for daily periods about 1/5. The parameters a_2 and b_2 of the model are the linear discharge (recession) coefficients of the rapid response and the base flow, respectively, so that their values lie between 0 and 1. Parameter a_2 has to be lower than b_2 because the direct runoff decays more rapidly than base flow runoff; a_0 and a_1 should be larger than b_0 and b_1 because, in general, the direct runoff of a hydrograph is larger than the base flow. Cruickshank (1996) applied the model successfully to monthly, daily and hourly intervals of measurement; he finds the values of the parameters assisted by the dynamic filtration technique of Kalman.

Input data

In order to estimate the model parameters and structure for a specific case, recorded data of total rainfall (at only one station) and total runoff for the storm on 13 September 1997 in the Mixcoac sub-basin in Mexico City were first used as training data. For evaluation of the behaviour of the models, data on the storm recorded in the same basin on 27 September 1998 were used as testing data. Afterwards, the order was reversed, using the last storm as the training data and the first one as the testing one. This action was taken in view of the impossibility of having more records as testing data because the gauging station was suspended after 1998. It was thought that doing this

double analysis would give a wider knowledge of the dynamics of the watershed and the applicability of the methodologies.

These were isolated storms so it was decided to subtract from the total rainfall the first five millimeters to account for interception. In all cases, the models obtained are applied using only their own calculated discharge values, that is to say, no correction is introduced from measured discharges in the forecast.

APPLICATIONS

Genetic programming: NARMAX model with 1997 storm for training

As was pointed out before, only the operations of addition and multiplication were considered as part of the set of functions. The model produced, when applied to the storm of September 1997 as training data, is that of Equation (15):

$$Q_t = a_0 + a_1 Q_{t-1} + a_2 Q_{t-2} + a_3 hp_{t-1} + a_4 hp_{t-2} + a_5 hp_{t-7} + a_6 Q_{t-1} Q_{t-7} + a_7 Q_{t-4} hp_{t-5} + a_8 hp_{t-3} hp_{t-3} + a_9 Q_{t-2} hp_{t-2} hp_{t-5} \quad (15)$$

where $a_0 = 1.108$, $a_1 = 1.049$, $a_2 = -0.298$, $a_3 = -0.082$, $a_4 = 0.910$, $a_5 = -0.254$, $a_6 = 0.0048$, $a_7 = 0.0315$, $a_8 = 0.048$, and $a_9 = 0.027$.

The mean quadratic error (OF) in the training period was equal to 0.224 $(m^3/s)^2$ with 99 observation points. Figure 4 shows recorded total rainfall *hp* (after subtracting the first 5 mm) for the storm of September 1997 together with the measured and calculated hydrographs.

The same model was tested by applying it to the storm of September 1998 with the result of a mean quadratic error of 10.9 $(m^3/s)^2$ in 99 observations. In Figure 5, the hyetograph together with the hydrograph ordinates, both observed and computed, are shown.

Genetic programming: NARMAX model with 1998 storm for training

The model produced by genetic programming when applied to the storm of September 1998 as the training data is that of

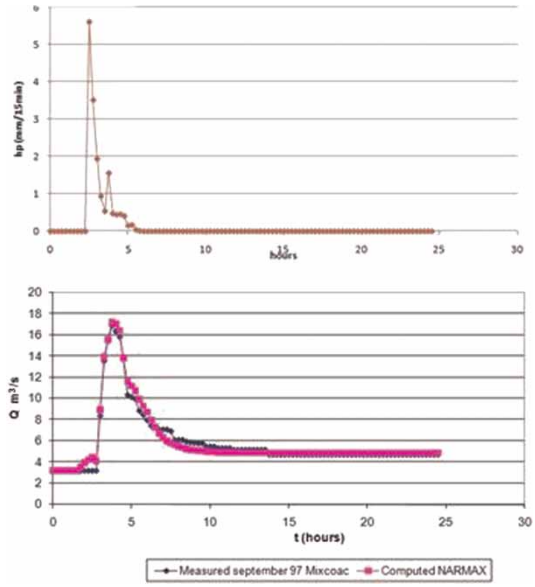

Figure 4 | Results for the storm of 1997 as the training period.

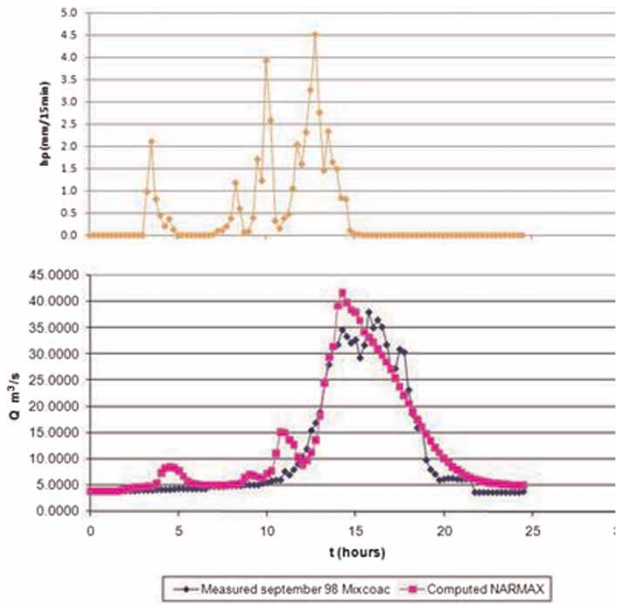

Figure 5 | Model from storm of 1997 applied to storm of 1988.

Equation (16):

$$Q_t = a_0 + a_1 Q_{t-1} + a_2 Q_{t-5} + a_3 hp_{t-3} + a_4 hp_{t-7} + a_5 Q_{t-2} Q_{t-6} + a_6 Q_{t-3} Q_{t-5} + a_7 Q_{t-7} hp_{t-1} + a_8 Q_{t-7} hp_{t-7} + a_9 hp_{t-3} hp_{t-3} \quad (16)$$

where $a_0 = 0.6980$, $a_1 = 0.9508$, $a_2 = -0.1106$, $a_3 = -0.7863$,

$a_4 = -0.3310$, $a_5 = -0.0040$, $a_6 = 0.0063$, $a_7 = 0.1205$, $a_8 = 0.0469$, and $a_9 = 0.3394$.

The mean square error in the training data given by the model was equal to 4.769 $(m^3/s)^2$, again with 99 observation points. Figure 6 presents the total rainfall *hp* that was recorded on September 1998 at the Mixcoac station together with the measured hydrograph and the one calculated with Equation (16). It may be appreciated that the estimated runoff provided a softened version of the actual record.

In order to test the model of Equation (16), it was applied to the rainfall data of the storm of September 1997. This resulted in a testing mean square error of 1.31 $(m^3/s)^2$ with 99 data points. Figure 7 shows again the hyetograph and the measured and computed hydrographs. It may be seen that the test was very successful.

Genetic algorithms

In order to adjust the NLAPI model to the measured data, a 2 interval delay (30 min) was assumed for the reaction to rainfall in direct runoff (r_a), from the observation of the storm of September 1997, which is an isolated one; a delay of seven intervals was adopted for the base runoff (r_b), taking into account some terms appearing in the

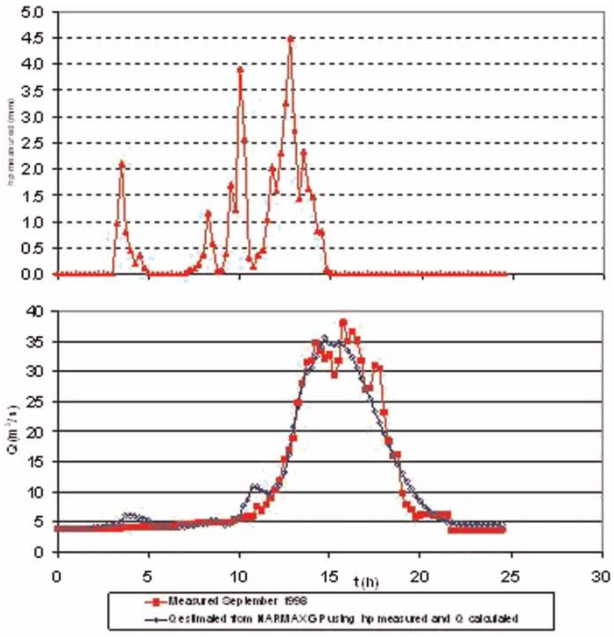

Figure 6 | Results of Equation (16): genetic programming and NARMAX model.

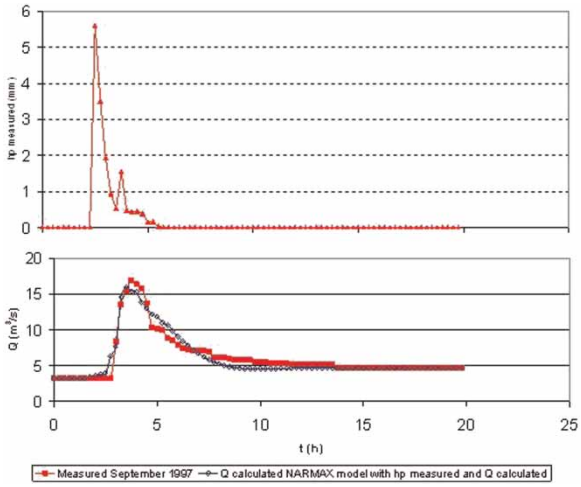

Figure 7 | Application of Equation (16), GP-based NARMAX model, to the storm of September 1997 in the Mixcoac sub-basin.

NARMAX models. Apart from that, some restrictions were imposed on the algorithm regarding the searching space and relations between the parameters, namely that a_2 and b_2 should be in the interval $0 \leq a_2, b_2 \leq 1$, that $a_2 < b_2$ and for β to be $0 \leq \beta \leq 0.1$; besides, that a_0 and a_1 be larger than b_0 and b_1, respectively, because of what was commented on before about the relative importance of direct and base runoff.

Determination of NLAPI model parameters with the storm of September 1997 as the training period

The resulting parameter values for Equations (11)–(14), obtained using genetic algorithms to optimize the adjustment, are presented in Table 2.

The training mean square error was 0.352 $(m^3/s)^2$ with 99 observation points and its graphical adjustment may be appreciated in Figure 8.

Table 2 | NLAPI model parameters

β	0.1000
a_0	0.0305
a_1	2.1058
a_2	0.7657
b_0	0.0000
b_1	0.1831
b_2	0.9978

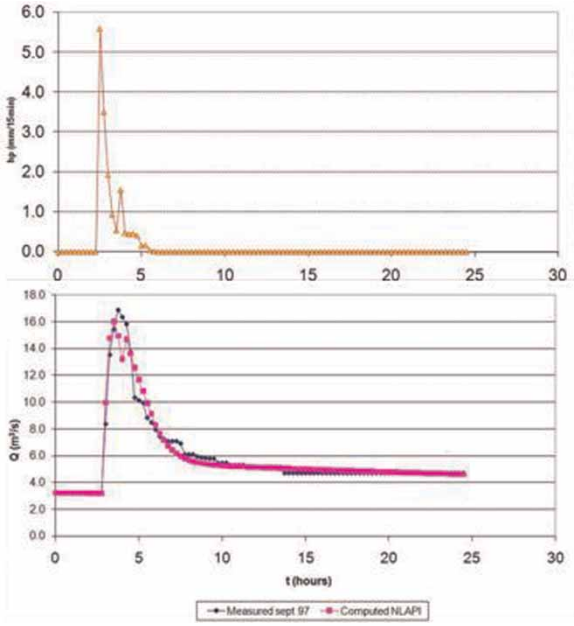

Figure 8 | NLAPI with September 1997 storm as training period.

When the model with these parameters is applied to the storm of September 1998 the result is a mean square error of 43.8 $(m^3/s)^2$, as always with 99 observation points and graphical adjustment shown in Figure 9. It may be seen that it is not a very good adjustment.

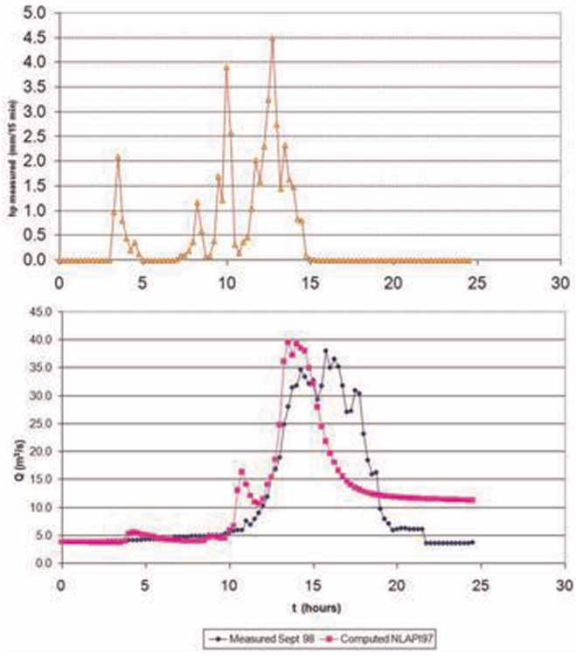

Figure 9 | September 1998 storm as testing period for NLAPI97.

Determination of NLAPI model parameters with the storm of September 1998 as the training period

For this option, the parameters obtained are shown in Table 3.

The objective function produced a training mean square error equal to 8.47 $(m^3/s)^2$ with 99 observation points and a comparison is made in Figure 10 between measured and calculated values.

When the above parameters are applied to the storm of September 1997 they have a testing mean square error of 12.5 $(m^3/s)^2$ with 99 data points and the graphical confrontation is shown in Figure 11; it is evidently not a very good

Table 3 | Results for NLAPI trained with 1998 storm

Parameter	Value
β	0.0305
a_0	0.3357
a_1	1.2513
a_2	0.8850
b_0	0.3357
b_1	0.8850
b_2	0.9461

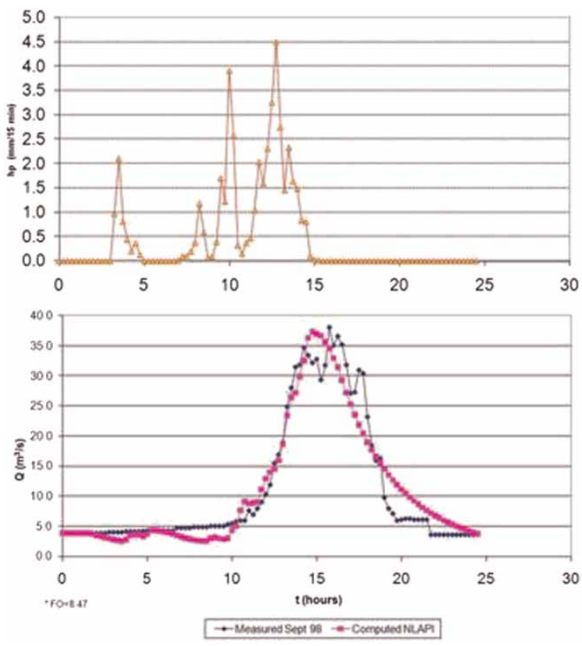

Figure 10 | Results with equations of the NLAPI model.

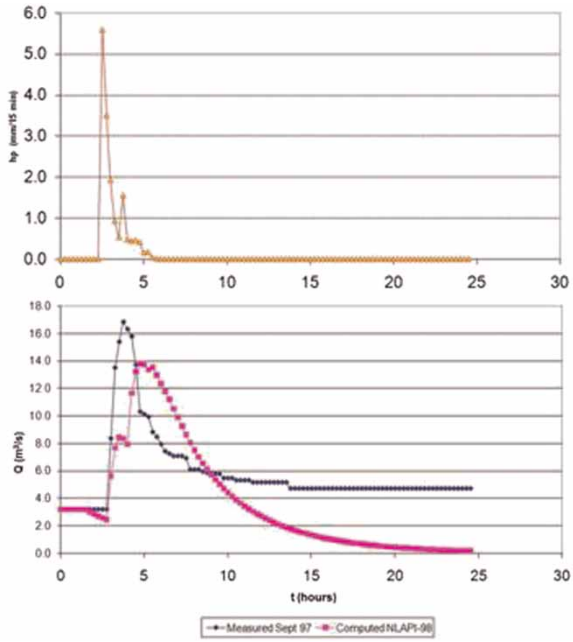

Figure 11 | NLAPI trained with 1998 storm tested against 1997 data.

adjustment. The difference is also noticeable in the obtained parameters depending on the type of storm with which the model is trained; that of 1997 is concentrated in a short time (and possibly also in space) while that of 1998 is more spread out and has several peaks. This will be discussed in more detail in the next section.

A summary of objective function values obtained by the different experiments is presented in Table 4.

DISCUSSION

It is remarkable the ability of genetic programming through the NARMAX model to detect the system's reaction to rainfall as the only input. This ability is consistent in that it shows a better model structure when it is trained with more and diverse input data. In fact, when the training is done with a short storm, as in September 1997, the model has difficulty predicting a longer storm. Nevertheless, the structure of the obtained equations has one inconvenience, which is the fact of having negative terms; they come from a mere choice of functions to minimize the error. From a physical cause and effect point of view, this has no possible explanation; this is why attempts were made, first, to eliminate those terms and, second, to interpret the remaining ones so as to give them a physical sense, more than just a mathematical formulation.

Table 4 | Summary of values of the objective function (OF) for the two stages (training and testing) and the two storms

Storm	Stage	Model NARMAX	NLAPI
Sept. 1997	Training	0.224	0.352
	Testing with 1998 model	1.31	12.5
Sept. 1998	Training	4.77	8.28
	Testing with 1997 model	10.9	43.8

The first attempt was successful by simply eliminating the negative terms and changing slightly the positive ones in Equation (15); the result was Equation (17) which, when applied to the storm of September 1997, gave an OF equal to 0.752 $(m^3/s)^2$ and may be seen in Figure 12:

$$Q_t = 1.1 + 0.76Q_{t-1} + 0.1hp_{t-1} + 1.4hp_{t-2} + 0.0315Q_{t-4}hp_{t-5} \quad (17)$$

The simplicity of Equation (15) as compared with the originals is to be noted. Nevertheless, this equation badly fits the second storm.

Going further in the analysis of the flow records of the two storms in question, it may be observed that they both begin with a constant discharge and end with another almost constant discharge. To get this result it is only necessary to have the two initial terms of the recurrent equations above: a constant term and another with the previous discharge multiplied by a constant less than 1, which reproduces the rapid flow recession. The value of the latter constant is to be chosen according to the recession slope, in the above case equal to 0.76. The first constant may be

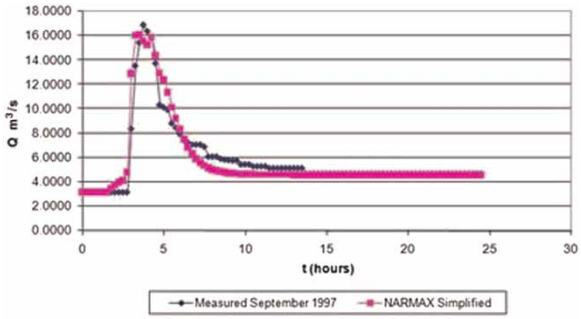

Figure 12 | Result with a simplified version of the NARMAX equation.

obtained from the value of the ending constant discharge Q_{end} as $C = (1 - 0.76)Q_{end} = 0.24 \times 4.6 = 1.1$.

So, what remains is to obtain a nonlinear function of the rain and some autoregressive discharge value as the reaction to the rain input. An equation is here presented which fits fairly well the two analysed storms (Equation (18)) with mean square errors of 2.35 and 7.46 $(m^3/s)^2$ for 1997 and 1998, respectively, and graphical representation in Figure 13:

$$Q_t = 0.99 + 0.76Q_{t-1} + 0.0035Q_{t-3}Q_{t-5} + 0.83hp_{t-2} + 0.31hp_{t-3}hp_{t-4} + 0.08Q_{t-7}hp_{t-7} \quad (18)$$

The results from the NLAPI model are useful in revealing some errors in the model application to the analyzed storms and in pointing out inconsistencies in the model itself.

The first observation has to do with the delay times adopted for the model. A delay of two 15 min intervals was chosen with basis on the observation of the lag between rain and runoff peaks of the isolated storm of September 1997; still, no attention was paid to the estimation of the watershed concentration time. From a different study (Domínguez *et al*. 2008) it was found to be 2.5 h, that is, ten 15 min intervals. For a generalized storm covering the whole watershed, this should be the direct runoff peak delay according to hydrologic analysis principles. This value is closer to the 7 interval delay adopted for the base runoff and its effect can be seen when one separates the two runoff components calculated by the model with parameters determined by GA. This is depicted in Figure 14, which is a replica of Figure 10 with base flow added; it may be seen there that a large part of the flow is taken up by the base flow while it should be much smaller than the direct flow; in some unrestricted parameter trials the base flow occupied practically the total flow. When the model with the parameters determined from the storm of September 1998 as the training period is applied to the storm of September 1997 as the testing period, the results are very bad as is shown in Figure 11. Similar, although inverse, results are obtained if the training and testing periods are interchanged.

Another weakness of the NLAPI method, as it is presented here, is that it produces unit responses which rises abruptly at the time of delay; again this goes against normal hydrograph analysis for a storm occurring uniformly in the watershed; if the concentration time of the latter is larger than the time interval, the rising limb of the unit response should last a time equal to the concentration time; for the studied watershed this is ten 15 min intervals. This seems to be in contradiction to the answer observed for the storm of September 1997 which is almost immediate; the more plausible explanation for that is that the storm was localized at the outlet of the watershed.

A further lesson which may be drawn from the genetic programming results is the existing of nonlinear effects on the runoff response to the rain that falls while runoff is

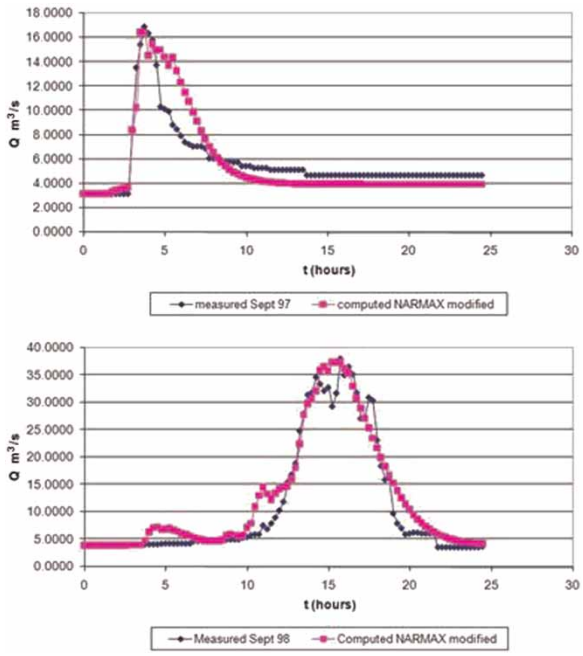

Figure 13 | Results with Equation (15) for the two storms.

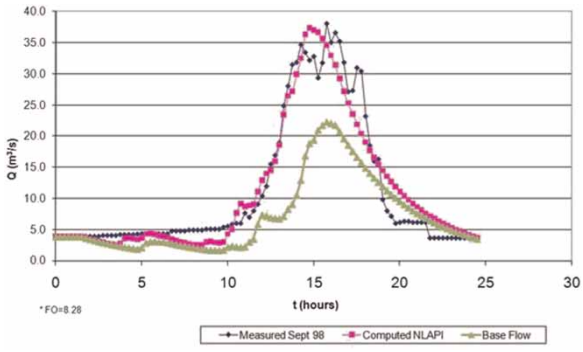

Figure 14 | Storm of September 1988 showing computed total and base flows with the NLAPI method.

occurring; the answer should be logically larger than that for the rain falling in more or less wet terrain; this is shown in many of the genetic programming terms which have products of rain and discharge. An attempt was made to include this in the NLAPI model, adding a new term for the factor multiplying rain in the direct response part of the flow which includes a previous runoff value. This is expressed in the following equation which substitutes for Equation (12):

$$Q_{at} = (a_0 + a_1 A_{t-ra} + a_h Q_{t-ra}) hp_{t-ra} + a_2 Q_{at-1} + e_a^t \qquad (19)$$

It was called NLAPIM (NLAPI modified) and applied to the storms of September 1998 and September 1997 with the following parameters which resulted in objective functions of 1.27 and 22.1 $(m^3/s)^2$, respectively (Table 5 and Figures 15 and 16).

The parameters were adjusted by trial and error so that they would fit the two storms only to show the feasibility of a better model with the new nonlinearity introduced and in spite of the described deficiencies of the model.

Table 5 | Parameters for Equations (13) and (19)

β	0.0305
a_0	0.9000
a_1	0.8000
a_2	0.8500
b_0	0.0700
b_1	0.0800
b_2	0.9970
a_h	0.0400

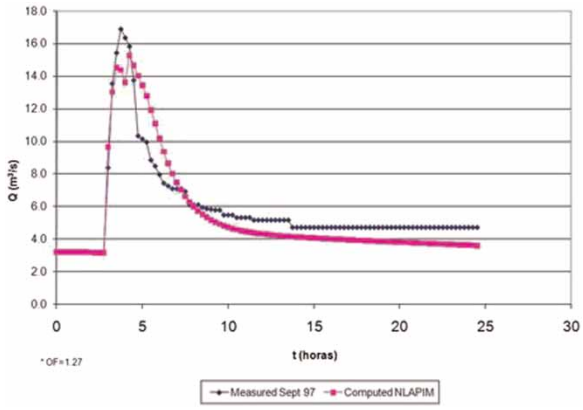

Figure 15 | Modified NLAPI applied to September 1997 storm.

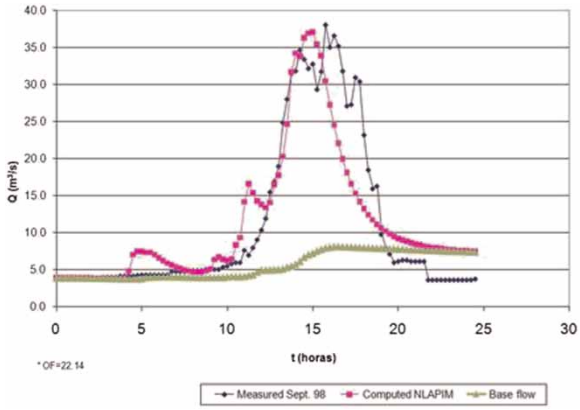

Figure 16 | Modified NLAPI applied to September 1998 storm.

CONCLUSIONS

Genetic algorithms and genetic programming were used to calculate parameters and to obtain two rainfall–runoff models considering the data of total rainfall and total runoff, providing an alternative approach to the diverse traditional models that are based on excess rainfall and on direct runoff. This was the goal of this work because it is not an easy task to obtain the value and time distribution of excess rainfall.

Two approaches were considered and compared; a fixed parameter model with two recursive equations (for direct and for base flow) which involve a antecedent precipitation index, here called NLAPI, and a nonlinear autoregressive model whose terms were found by searching multi-objective functions with genetic programming, here called NARMAX. In both cases the only external input to the system was measured rainfall and the models were left alone to produce their own outputs, be it final or internal to their autoregressive structure.

The model that best fitted the two analyzed storms in the Mixocac sub-basin, was the NARMAX model. Nonetheless, several questions are raised against both models in their plain application; in the NARMAX case, the inclusion in its structure of negative terms, difficult to interpret physically; and in the NLAPI case, the existence of inconsistencies and deficiencies. Proposals were made to surmount these

questions in the two approaches whose preliminary results point to promising investigation lines in the formulation of simple, clear and elegant black-box (input–output) models in the forecast of runoff from small watersheds.

The use of evolutionary computation techniques became useful in estimating the structure and the parameters of the rainfall–runoff models due to the complex nature involved in such processes.

In the context of multiple criteria fitness functions, it is feasible to assume the validation stage based on statistical criteria such as, for instance, the auto-correlation error, the cross-correlation error and higher-order correlation criteria that can be simultaneously evaluated during the modelling and calibration processes of parameters. It is intended to deal with this stage in forthcoming papers.

REFERENCES

Aparicio, F. J. 1989 *Fundamentos de hidrología de superficie.* Limusa, Mexico.

Banzhaf, W., Nordin, P., Keller, R. E. & Francone, F. D. 1998 *Genetic Programming: An Introduction*. Morgan Kaufmann, Sacramento, California.

Chen, S. & Billings, S. A. 1989 Representation of non-linear systems: the NARMAX model. *Int. J. Control* **49** (3), 1013–1032.

Chong, S. F. 2002 Hydrological models of precipitation. In *Proceedings of the FIFTH WMO International Workshop on Tropical Cyclones, 3–12 December 2002, Cairns, Australia*. World Meteorological Organization, United Nations, New York (CD-ROM).

Coello, C. A., Van Veldhuizen, D. A. & Lamont, G. B. 2002 *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer, Dordrecht.

Cruickshank, C. 1996 Hacia un modelo generalizado lluvia-escurrimiento. In: *Memorias del XVII Congreso Latinoamericano de Hidráulica, October, Guayaquil, Ecuador*. IAHR Internacional Association for Hydro-Enviroment Engineering and Research, Madrid, España, pp. 21–25.

Domínguez, M. R., Esquivel, G. G., Baldemar, M. A., Mendoza, R. A., Afganis, M. L. & Carrizosa, E. E. 2008 *Manual del Modelo para Pronóstico de Escurrimientos, Serie Manuales, Instituto de Ingeniería.* UNAM, Mexico.

Drécourt, J. P. & Madsen, H. 2001 Role of domain knowledge in data-driven modeling. In *Proceedings 4th DHI Software Conference & DHI Software Courses.* 6–8 June 2001, DHI Helsingør, Denmark (CD-ROM).

Eslava, H. 1997 *Programación y aplicación del hidrograma unitario instantáneo geomorfológico*. Tesis de Maestría, UNAM, Mexico.

Goldberg, D. E. 1989 *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.

Guven, A. 2009 Linear genetic programming for time-series modelling of daily flow rate. *J. Earth Sci*. **118** (2), 137–146.

Holland, J. H. 1975 *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Cambridge, Massachusetts, USA.

Keeney, R. L. & Raiffa, H. 1976 *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. John Wiley & Sons Inc., New York.

Khu, S. T., Keedwell, E. C. & Pollard, O. 2004 An evolutionary-based real-time updating technique for an operational rainfall-runoff forecasting model, In: *Complexity and Integrated Resources Management, Trans.* (C. Pahl-Wostl, S. Schmidt, A. E. Rizzoli & A. J. Jakeman, eds). In *Proceedings of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs, Manno, Switzerland*, Vol. 1, pp. 141–146.

Koza, J. R. 1992 *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, Massachusetts.

Leontaritis, I. J. & Billings, S. A. 1985 Input-output parametric models for non-linear systems. Part I and Part II. *Int. J. Control* **41** (2), 304–344.

Liong, S.-Y., Gautam, T. R., Khu, S. T., Babovic, V., Keijzer, M. & Muttil, N. 2002 Genetic programming: a new paradigm in rainfall–runoff modelling. *J. AWRA* **38** (3), 705–718.

Ljung, L. S. 1987 *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, New Jersey.

Madsen, H., Butts, M. B., Khu, S. T. & Liong, S. W. 2000 Data assimilation in rainfall-runoff forecasting. In *Hydroinformatics 2000, 4th International Conference of Hydroinformatics, 23–27 July 2000, Cedar Rapids*. Iowa Institute of Hydraulic Research, Iowa, USA, pp. 1–6 (CD-ROM).

Rabuñal, J. R., Puertas, J., Suárez, J. & Rivero, D. 2007 Determination of the unit hydrograph of a typical urban basin using genetic programming and artificial neural networks. *Hydrol. Process.* **21** (4), 476–485.

Rodríguez-Vázquez, K. 2001 Genetic programming in time series modelling: an application to meteorological data. In *Proceedings 2001 Congress on Evolutionary Computation CEC2001, 27–30 May 2001, Seoul, Korea*. IEEE Press, New York, pp. 261–266.

Savic, D. A., Walters, G. A. & Davidson, J. 1999 A genetic programming approach to rainfall–runoff modelling. *Wat. Res. Mngmnt.* **13**, 219–231.

Söderström, T. & Stoica, P. 1989 *System Identificacion*. Prentice-Hall, Englewood Cliffs, New Jersey.

Whigham, P. A. & Crapper, P. F. 2001 Modelling rainfall-runoff using genetic programming. *Math. Comput. Modell.* **33**, 707–721.

First received 27 November 2009; accepted in revised form 24 May 2010. Available online 12 April 2011