

# Evolutionary Neural Network Modeling for Describing Rainfall-Runoff Process

Alireza Nazemi<sup>1</sup>

Graduate Student, Department of Civil Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Hossein Poorkhadem,-N.

Graduate Student, Department of Mechanical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Mohammad –R., Akbarzadeh –T.

Assistant Professor, Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Seyed Mahmood Hosseini

Assistant Professor, Department of Civil Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

**Abstract.** Since the last decade, several studies have shown the ability of Artificial Neural Networks (ANNs) in modeling of rainfall-runoff process. From methodological viewpoint, ANN belongs to more general paradigm, i.e., soft computing or computational intelligence, in which independent methodologies, mostly Fuzzy Logic (FL), ANN, and Genetic Algorithms (GAs), are combined together in order to provide an intelligent behavior in computational frameworks. Consequently, in the context of rainfall-runoff modeling, this question arises whether hybridization of ANNs with other soft computing-related methodologies improves the overall performance of modeling or not. In this study, based on the idea of structure and/or parameter identification of ANNs with GAs, the evolutionary neural networks modeling paradigm is examined for describing the rainfall-runoff process. One of the benchmark data set of current literature, i.e., Leaf River basin (near Collins, Mississippi) data set, is used for simulation. The results show that on the one hand, the overall accuracy is improved; but on the other hand, in evolutionary neural modeling, the computational time is increased significantly. Hence the modeler may be faced with a trade-off problem between accuracy and computational difficulties which may have different importance in a particular rainfall-runoff problem.

## 1. Introduction

Rainfall-runoff process is accepted as one of the most complex and nonlinear real-world phenomena in the field of water engineering [Hsu, *et. al.*, 1995]. The process consists of the movement of rainfall through different media and its transformation to the runoff in channels either natural or man-made. Although the detailed investigation of the process may be more scientific-oriented than engineering-oriented, many engineering designs and applications need to have an estimation of the quantity and feature of runoff resulted from particular rainfall. Furthermore, in many other applications, different scenarios of future rainfall should be simulated in order to identify the behavior of an area. Therefore, many mathematical approaches can be found in the literature which are proposed for the purpose that an accurate estimation of runoff can be made by knowing the quantity and quality of rainfall for an assumed area.

By surveying the rainfall-runoff modeling literature, it can be realized that a diverse set of methods and models has been introduced in this regard since 1800s. From classification point of view, they can be categorized in three classes. First is the experimental models, which almost of all are primitive methods, such as rational

---

<sup>1</sup> Corresponding author, TEL: (+98)-21-8083760, Email: [ali\\_r\\_nazemi@yahoo.com](mailto:ali_r_nazemi@yahoo.com)

method. Second is the conceptual models, which are designed in order to reproduce the real physical process of rainfall-runoff transformation, occurred in nature, such as SAC-SMA model [Sorooshian, et. al., 1993]. And third is the system theoretical models, which create a black box mapping from rainfall and/or previous runoff sequences to current runoff such as ARMAX. In spite of the fact that many of these models are very detailed and well organized and the computational strength, achieved recently, a lot of difficulties remain in using them. Many studies reported that calibration of models have many computational difficulties [Sorooshian, et. al., 1993]. Because of the natural orientation of the process a great deal of uncertainty is entered to the modeling procedure. In addition, many models are based on some assumptions, which do not hold in the real process [Woolhiser, 1996]. Thus, the modeling of the rainfall-runoff process is still a challenging problem.

Nowadays, several methods, as alternatives of probability theory, have been proposed to handle uncertainties in engineering systems and models. One of the most successful and brilliant classes of methods is soft computing or computational intelligence that recently have come into the limelight [Jang, et. al., 1997]. It is originally consisted of different independent paradigms, mostly, Fuzzy Logic (FL), Artificial Neural Networks (ANNs) and Genetic Algorithms (GAs). All of these paradigms mimic human intelligence in someway. FL is a multi-variable logic inspired from human thinking process; ANNs are massive parallel computational nodes based on human neurons; and GA is the stochastic optimization, trying to simulate natural evolution based on the survival of the fittest. Although these paradigms can stand alone, the focus soft computing methods is more on the combination among them to achieve accuracy, tractability and robustness in computations and modeling [Gorzalczany, 2002]. The wide range of applications are reported in different engineering field such as control, signal processing, and real-world events modeling. In the last decade, many application examples of aforementioned individual paradigms are introduced to the literature of rainfall-runoff modeling such as [Zhu & Fujita, 1994] and [Ndiritu & Daniell, 2001]. However, these applications used only one of these paradigms in their modeling procedures.

In this study the application of a rigid framework in soft computing methodologies, i.e., evolutionary neural modeling, is examined for modeling the rainfall-runoff process. In Section 2, a brief review of rainfall-runoff modeling is presented. In Section 3, soft computing as the different view point in modeling approaches is discussed. Section 4, introduces case study data and proposed model, based on structure and parameter optimization of feedforward ANNs with GA. In section 5, results are integrated and further directions of this study are introduced; and in Section 6, conclusion is driven.

## 2. Rainfall-Runoff Modeling

Rainfall-runoff modeling is assumed to find a description for transformation of the total rainfall volume to the corresponding reduced runoff volume in an area. The total volume of the rainfall during one rainy event  $W_{rf}$  is determined based on the data collected on the watershed area  $A$ , by the following equation:  $W_{rf} = A * H$ , where  $H$  represents the average height of rainfall. The other convenient way to estimate the total rainfall volume is given as follows:  $W_{rf} = A \int_{T_0}^{T_r} i * dt$ , where  $i$  is the rainfall intensity in meter per second,  $T_0$  and  $T_r$  are starting and ending time of rainfall event. The runoff volume  $W_q$ , is estimated on the basis of hydrograph data in

the following way:  $W_Q = A \int_{T_0}^{T_r} Q * dt = A \int_{T_0}^{T_r} i_e * dt$ , where  $Q$  represents hydrograph runoff ordinate on the outlet of the watershed,  $i_e$  is net or effective intensity of the rainfall. The runoff volume or effective rain is defined by the following equation:  $H_e = W_Q / A$ . The difference is defined by the following way:  $W_{rf} - W_Q$  is the retained water volume (water volume deficit) that varies with time, i.e., similar rainfall volumes may result in significantly different runoff volumes on the same watershed profile, proving the non-stationary of the rainfall-runoff processes. The strongly emphasized non-linearity of the rainfall-runoff process is the other manifestation of the complexity of the internal watershed structure. This internal structure is a consequence of the composition of a large number of relatively permanent and changeable essential features. It is considered that more than ten factors invariably participate in the rainfall-runoff process, but this number is large for the certain physical, geographical and climatic conditions. There is no universal model designed for the rainfall-runoff process, since the model developed for the certain watershed may appear to be quite unusable for the other area. Therefore, the selection of the proper modeling approach, considering available relevant data is the first and crucial phase of the model designing process, and its success depends on whether we have good understanding or adequate initial knowledge about the process to be modeled in a certain watershed profile. The next stage assumes determination of the structure of the model and estimation of its parameters. The structure should be the one, which will perform satisfactorily the tasks stated by the modeler. The last stage, a validation of the identified model, is a test of the validity of previous performed steps of the modeling process, done by the new relevant validation set [Furundzic, 1998].

Modern literature of rainfall-runoff modeling contains two main approaches: the conceptual (physical modeling) and system theoretic modeling. The conceptual rainfall-runoff models attempts to provide reliable approximation of physical mechanisms, which determine the hydrologic cycle. The conceptual models are convenient for understanding of the hydrologic process, but they are not efficient in stream flow forecasting (prediction at specified watershed location). In such situations, the system theoretic approach is a more convenient tool. The system theoretic approach to rainfall-runoff is generally based on the differential equation models designed for direct mapping between the inputs and outputs, not taking care to the exact physical process. The ARMAX (auto-regressive moving average with exogenous inputs) linear models for time series analysis, developed by Box and Jenkins, have been frequently used [Hsu, et. al., 1995]. These models are easy to develop and practical for use, but they are not appropriate tool for modeling of nonlinear dynamic processes such as the rainfall-runoff, and may show unsatisfactory performance [Furundzic, 1998].

From the last decade a new modeling approach is added to the literature of rainfall-runoff modeling. ANNs are mathematical paradigm, which tries to represent *low-level* intelligence in natural organisms [Gorzalczany, 2002]. Although in the literature of rainfall-runoff modeling this paradigm is categorized in system theoretical approaches [Hsu, et. al., 1995], there is significant difference between their basis and inspiration. As a result, ANNs should be classified in a new category with some others having similar foundations. Usually ANNs show greater performances in comparison to conventional regression approaches based on statistical analysis. The characteristics of ANN can be approximately defined as follows [Furundzic, 1998]:

- High degree of fault tolerance and robustness for imprecision and uncertainty in unconstrained information environments.
- Distributed processing and low-level representation of information.
- Generalization ability.
- Massive parallelism.
- Learning and adaptation.

Many researchers in different areas of science and technology have used ANNs to solve problems in control, function approximation, and pattern classification. Among them, the multi-layer feedforward networks (MLP) is the mostly used network structure [Jang, et. al., 1997]. The functionality of artificial neurons is the same as natural neurons. All or part of artificial neurons are adaptive, which means that the output of these nodes depend on modifiable parameters pertaining of these nodes. The learning rule specifies how these parameters should be updated to minimize a prescribed error measure, a mathematical expression that measures the discrepancy between the network's actual output and desired output [Demuth & Beale, 2000]. In other words, an adaptive network is used for system identification, and our task is to find an appropriate network architecture and a set of parameters which can best model an unknown target system which is described by a set of input-output data pairs.

While the most common neural architecture in the field of rainfall-runoff modeling is MLP [Hsu, et. al., 1995], [Minns & Hall, 1996], several studies, particularly recent publications examined other architectures [Anamala, et. al., 2000], [Hsu, et. al., 2002]. These studies have concluded that ANNs can be assumed as a powerful alternative for describing the relationship between rainfall and runoff sequences.

From the methodological viewpoint, ANNs are classified with some other paradigms, all of them mimic the characteristics of natural life in some way. These paradigms have been proved as flexible tools especially in dynamic and uncertain information environments. The context of computational intelligence or soft computing contains all of these paradigms and their hybridization with each other, which presently is the basis of several novel investigations [Jang, et. al., 1997], [Gorzalczany, 2002].

### **3. Soft Computing: The Innovative Approach in Modeling**

As described, soft computing is a contribution of different methodologies, i.e., fuzzy systems, neural networks and evolutionary algorithms especially genetic algorithms in order to deal with uncertainty, partial truth and ambiguity in modeling of a process and/or environment. FL represents a broader class of theoretical tools, which is termed as methods of granular information processing and granular knowledge representation. The introduction of this theory allows creating formal mathematical models for imprecise and ambiguous defined terms, relations and mechanisms of approximate inference, typical of human reasoning; and for the perception of the environment by a human being. Fuzzy systems are the most used application of fuzzy sets in the field of engineering and can find many different applications in engineering problems [Bardossy and Duckstein, 1995]. The small computational time requirement, the numerical robustness, transparent programming of rule base systems, and the ability to handle non-probabilistic uncertainties are the most important properties of fuzzy systems. The second basic theoretical tools used in the construction of soft computing-based systems are ANNs. As mentioned in Section (2), they are systems of information processing which take their roots in the

biology and neuro-physiology including the knowledge of the operation of nervous systems in living organisms. The ANNs are treated as formal computation systems, which have specific properties and can be useful in the solution of some problems in the field of information processing and classification. The third essential component, beside fuzzy sets and artificial neural networks, used in the construction of soft computing-based systems, are method of evolutionary computations, which are the information processing methods based on principles imitating mechanisms of evolution, heredity and natural selection that occur in living populations. GAs are the most widely known category of evolutionary computation systems. They are method of global searching in the solution domain of the considered problem, mainly, in complex optimization tasks. Searching procedures are based on the mechanisms of natural selection of heredity, including the evolutionary principles of survival and leaving behind offspring by individuals, the best adapted to the environment, and extinction of individuals, which have the worse adaptation to the environment [Coley, 1999].

The main features of soft computing-based systems in addition to artificial intelligence systems are (1) computational adaptivity; (2) computational fault tolerance; (3) speed approaching human-like turnaround, and (4) error rates that approximate human performance [Gorzalczany, 2002]. From a superior viewpoint it can be seen that soft computing components are inspired from diverse domains of logic, biology and physiology in natural intelligent systems. In regard to specific application, different hybridization procedures can be done

The particularly complementary nature of fuzzy systems and artificial neural networks is worthwhile. On the one hand fuzzy systems are high-level information processing systems, which have the ability of computing with linguistic variables. On the other hand, as described before, neural networks are low-level computational framework. Both paradigms have universal approximation capability [Jang, et. al., 1997]. Thus, their hybridization provides a very powerful information system. The learning abilities of the hybrid system can be additionally strengthened by applying, evolutionary computations in a supportive role, which enable us to adapt both the parameters and the structure of the system. No other combination of soft computing paradigms provides a comparably high degree of mutual benefit.

Since the early of 1990s applications of soft computing paradigms have been reported in the context of rainfall-runoff modeling; however these examples almost focused on one paradigm rather than their hybridization. Section (2) referenced several applications of ANNs in the field. In all of these examples, neural networks provide a mapping between sequence of rainfall and/or other assumed inputs to sequence of runoff. Fuzzy systems also can be used as mapping paradigm; for instance, in [Bardossy and Disse, 1993] a fuzzy rule based system was proposed for describing infiltration process; and, Han et al., in 2000 used a fuzzy logic approach to river flow modeling. In addition, because of the nature of fuzzy methods, they can enter in different steps and/or frames of modeling. As an illustration, Yu and Yang in 2000 used fuzzy multi-objective function for calibration a conceptual rainfall-runoff model. In another application, [Xiong et al., 2001] used first order Takagi-Sugeno-Kang (TSK) fuzzy system for obtaining a non-linear combination of the forecasts of different rainfall-runoff models. In [Ozelkan and Duckstein, 2001] fuzzy conceptual rainfall-runoff models were introduced. With the support of these applications, it seems that FL methods may have other potentials in this context. GAs almost have been used for parameter optimization of rainfall-runoff models. The first attempt in using genetic algorithms (GA) in the field of rainfall-runoff modeling calibration was reported by Wang in 1991. Franchini in 1996, modified the Wang procedure in order to improve the efficiency. GAs are flexible and they can modified in order to increase

performance; for instance, Kuczera in 1997, reported a study to reduce the computational effort by confining the search to a subspace within which the global optimum is likely to be found. Furthermore, [Ndiritu and Daniel, 2001] reported an improved genetic algorithm for rainfall-runoff model calibration and function approximations that converge to better results than can be achieved by SCE-UA. Other evolutionary methods have also shown to be effective, as an example, Khu et. al. in 2001, used Genetic Programming (GP) in real-time runoff forecasting based on filtering the output of a certain conceptual rainfall-runoff model.

Recently, some hybrid applications have been reported as [Liong et al., 2001] used a combination of ANNs and GA for finding pareto optimal solution of a conceptual rainfall-runoff model. As an another example, Chang and Chen in 2001 used counter propagation fuzzy-neural networks modeling approach to real time stream flow prediction. These applications show that hybrid applications of soft computing in unique rainfall-runoff modeling problem may have more advantages than single paradigm approach.

#### 4. Case Study and Proposed Model

Leaf River basin rainfall-runoff data set is one of the frequently used case studies in current rainfall-runoff investigations [Sorooshian, et. al., 1993], [Hsu, et. al., 1995], [Ndiritu and Daniel, 2001] [Hsu, et. al., 2002], [Hsu, et. al., 2002]. The basin is located north of Collins, Mississippi, with an area of approximately  $1950 \text{ km}^2$ . A reliable data set is available that represents a variety of hydrologic conditions and phenomena. The data set consists of forty (1948-1987) years mean daily streamflow ( $\text{m}^3 / \text{s}$ ), daily potential evapotranspiration estimates (mm/day), and 6-hourly mean areal precipitation totals (mm per 6 hours) [Sorooshian, et. al., 1993].

The first step in modeling with ANNs is the selection of appropriate set of input variables. In many studies this step is performed by trial and error; or presumption of input variables. Furthermore, in most of the previous studies, only rainfall and runoff sequences were used for modeling. This is probably caused by the accepted habit from the past when the reliable technologies conditioned constraint dimensions of the model, or may be the insufficiency of other relevant data. Some parameters such as evapotranspiration may be useful, since they can be assumed that they have considerably influence on the rainfall-runoff process, both directly resulting in evaporation and indirectly as one of the main global determinants of the season. This appeared quite reasonable since the existing level of the hardware and software technologies is convenient for such an undertaking, so this may result in model performance improvement [Furundzic, 1998]. In this study, based on correlation analysis, proper input variables are selected from a set of potential inputs. It is considered that runoff volume in time  $t$  is a function of several potential variables as follows:

$$Q(t) = f(Evpt(t), \dots, Evpt(t-6), R_1(t), \dots, R_1(t-6), R_2(t), \dots, R_2(t-6), R_3(t), \dots, R_3(t-6), R_4(t), \dots, R_4(t-6), R(t), \dots, R(t-6), Q(t-1), \dots, Q(t-6)) \quad (1)$$

In Equation (1),  $t$  is day index,  $Q$  is runoff volume,  $Evpt$  is potential evapotranspiration,  $R_1, R_2, R_3, R_4$  are total rainfalls in first, second, third, and forth 6 hours, respectively; and  $R$  is total daily rainfall. In other word, it is assumed that  $Q(t)$  is dependent variable of 48 above independent variables, and our aim is finding the most effective independent variables based on correlation analysis. Based on the correlation analysis performed for three years 1948-1950, 12 variables, which have

the most correlation among above 48 inputs, are selected as inputs for ANN modeling:

$$Q(t) = f(Q(t-1), Q(t-2), Q(t-3), R(t-1), R(t-2), R(t-3), R(t-4), Evpt(t-3), R_1(t-3), R_2(t-3), R_3(t-3), R_4(t-3)) \quad (2)$$

The second part of ANN modeling is finding a proper structure of network, which best describes the input-output relation. Figure (1), shows a typical structure for MLP networks. In previous studies this step was performed by trial an error, which is time consuming. In addition, there is no guarantee that the selected structure found by trial and error is the best.

Finding the best structure of MLP is an optimization problem, which can be handle by GAs. For this purpose first some assumption should be considered.

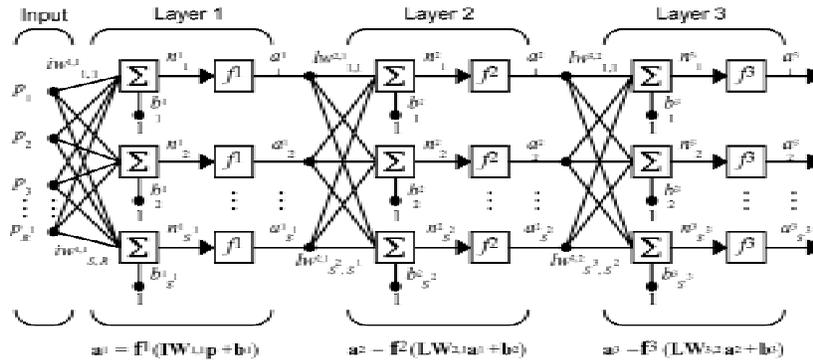


Figure (1): A typical feedforward MLP

There is a mathematical proof, describing that three layer MLP can approximate any continuous input-output mapping with any desired degree of accuracy. In this study it is assumed that MLPs have one or two hidden layer with at most 15 neurons in each layer. Activation functions are assumed as tangent sigmoid activation functions as below:

$$f(u) = \frac{1}{1 + \exp(-u)} \quad (3)$$

Consequently, GA should find optimal neurons in first and second hidden layers. Before applying evolutionary structure identification data are normalized. Data collected between 1951-1955 are used for structure identification. A simple GA procedure with binary coding is performed as below:

- *Step 1:* Initialize a population with randomly generated individuals; these individuals are MLPs genotype and evaluate the fitness value of each individual by converting them to corresponding phenotypes, i.e., MLPs. The fitness is the error criterion after particular epoch training with backpropagation.
- *Step 2:*
  - (a) Apply elitism, if required. Elitism is a policy of always keeping a certain number of best members when each new population is generated

- (b) Select two members from the population of genotypes with probability proportional to their fitness values.
  - (c) Apply a crossover operator with a probability equal to the crossover rate.
  - (d) Apply mutation with a probability equal to the mutation rate.
  - (e) Repeat (a) to (d) until enough members are generated to form the next generation.
- *Step 3:* Repeat step 2 and 3 until stopping criterion is met.

After meeting the termination criterion, a particular MLP structure is produced. This structure is the best individual in the last generation.

The third part of modeling with ANNs is finding optimize set of parameters for an assumed structure, which best describe the relation between a set of input-output pairs. In this section, hybridization between real coding GA and backpropagation is used. The real coding GA is performed for finding a set of proper initial weights for starting backpropagation procedure. The process of real code GA is the same as described above, but instead of using binary representation for genotypes, real number representation is used. Data between 1956 and 1965 is used for this part.

## 5. Results, Discussion, and Further Research

The above mentioned procedure is programmed with MATLAB® release 6.5. A Pentium IV, 1.8 GHz CPU with 256 MB RAM is used for simulation. Three different scenarios for input variables are performed. First the 12 variables assigned by correlation analysis; Second, considering the most six important variables among them, i.e.,  $Q(t-1), Q(t-2), Q(t-3), R(t-1), R(t-2), R(t-3)$ ; and third is choosing the most important input variables with evolutionary process. In later scenario, selection of proper input variables among the 12 potential variables is added to GA procedure as another decision variable. These scenarios are run in two modes, either considering elitism or not. Table (1) shows the internal parameters used in GA process.

Table (1): Properties of GAs performed in structure and parameters identification

<i>Section</i>	<i>Probability of crossover</i>	<i>Probability of mutation</i>	<i>Number of individuals</i>	<i>Number of generation</i>
Structure identification	0.8	0.01	50	50
Parameter identification	0.8	0.015	100	50
<i>Section</i>	<i>Number of elitism if used</i>	<i>Number of training epoch</i>	<i>Selection method</i>	<i>Genotypes representation</i>
Structure identification	3	50	Roulette wheel	Binary coding
Parameter identification	5	200	Roulette wheel	Real coding

Another matter to be addressed to is the unusual method of modeling. In previous studies, the data set is divided into two distinct subsets for training and validation phases. In spite of the fact that input identification, structure identification and parameter identification are related to training phase, in this study, they are done within 3 different data sequences. This unusual method has a natural inspiration. In natural systems, in which the behavior of system progresses with events, these steps

take place within different phases of system formation. In confronting with external events, a natural organism first identify the event and its characteristics, then tries to find the suitable approach in confronting and finally, by considering the new events, adopts the approach to face with it efficiently.

In addition to the fact that the above mentioned method has a natural evidence, it can represent some other features of intelligent-based systems. One of the main features of intelligence in natural organisms is the ability to memorize the indirect experiences. In psychology this is referred to unconscious memory. As it was discussed before, in training the model, the data of input identification and the structure identification is used implicitly within the model. The former is used for identification of inputs applied for structure identification and training the model and the later finds the optimal structure to be adopted with data of 1956-1965. It is interesting to find how the obtained model will be confronted with these implicitly used data. The answer to this “how” would help us to understand if any unconscious memory exists in the obtained model. Table (2), presents the error indices for six different scenarios. As mentioned, we consider 3 scenarios of input variables and the program is run by either considering elitism or without it. Consequently six different models are achieved. This table also represent the optimize structures obtained by evolutionary structure identification. Four error indices are used for comparing the performance of these different models. They are Root Mean Square of Errors (RMSE), Percentage of Volume Error (%VE), Percentage of error in Maximum Flow (%MF) and Correlation Coefficient (CORR). These measures are considered in Training phase (T), i.e., 1956-1965; input identification phase (V1), i.e., data of 1948-1950; structure identification phase (V2), i.e., data of 1951-1955; and an unseen verification data (V3), i.e., data of 1966-1975.

Table (2): Error indices and performance evaluation of six evolutionary neural networks describing the rainfall-runoff process in Leaf River

Scenario	6par	6par- elitism	Float par	Float par- elitism	12par	12par- elitism
<b>Structure</b>	[6 9 14 1]	[6 7 12 1]	[6 11 10 1]	[8 11 14 1]	[12 4 10 1]	[12 6 8 1]
<b>Error indices</b>						
<i>RMSE (T)</i>	8.4685	9.5001	9.2033	8.1351	10.0244	9.1476
<i>RMSE (V1)</i>	73.3091	59.4810	65.1426	76.6021	41.1112	30.1811
<i>RMSE (V2)</i>	42.9551	33.6021	32.1009	43.2971	25.9411	27.6795
<i>RMSE (V3)</i>	37.8533	32.7684	32.9890	52.0487	31.6302	33.5924
<i>%VE (T)</i>	-0.0212	-0.1452	0.0562	-0.2529	0.0067	0.0096
<i>%VE (V1)</i>	11.4486	-2.9578	-9.3137	-0.1448	-5.7981	4.0563
<i>%VE (V2)</i>	8.5920	-7.6882	-5.5639	-3.0525	-5.6246	2.0029
<i>%VE (V3)</i>	25.7441	22.3847	16.6512	27.0211	14.9439	30.6113
<i>%MF (T)</i>	0.2487	0.3779	0.0022	0.1257	0.0088	0.4606
<i>%MF (V1)</i>	0.0218	0.1546	1.03e-008	0.0624	7.76e-005	0.6740
<i>%MF (V2)</i>	0.1242	0.1309	4.33e-012	9.23e-004	0.0750	8.10e-004
<i>%MF (V3)</i>	2.5257	2.4e-004	0.0417	0.0214	0.0999	43.1647
<i>CORR (T)</i>	0.9916	0.9894	0.9901	0.9923	0.9882	0.9902
<i>CORR (V1)</i>	0.5936	0.8076	0.8225	0.7141	0.9299	0.9353
<i>CORR (V2)</i>	0.5337	0.8000	0.8351	0.7112	0.8626	0.8212
<i>CORR (V3)</i>	0.7379	0.8149	0.8249	0.6069	0.8303	0.8460

Considering the results of training phase in these scenarios, it can be realized that the evolutionary MLP parameter identification is quite successful. However it must be consider that the computational time increases significantly. This is the

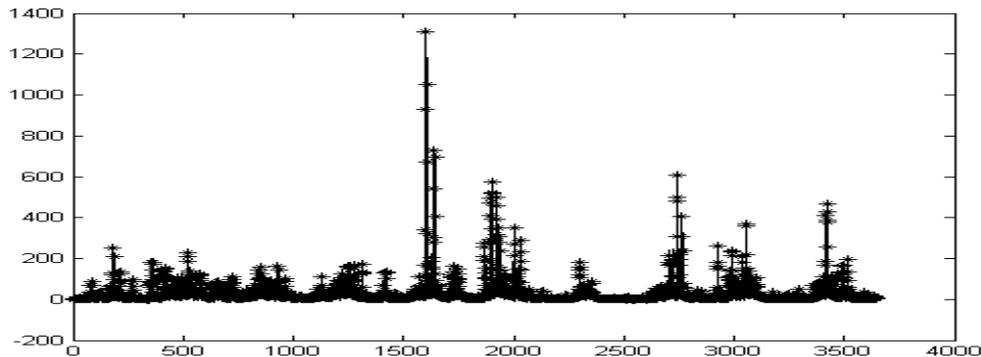
inherent problem of evolutionary-based search methods. So the modeler may need some justifications to apply it. In regard to the fact that the rainfall-runoff modeling is an offline one, it may be reasonable to apply the method to guarantee a well-done calibration.

Considering the results of 6par and 6par-elitism scenarios, it can be realized that reducing the number of input variables yields inaccurate modeling. So consideration of the other input variables is justified. Applying elitism, the results may improve; however, 12par and Float par, the scenario in which the input variables are identified by evolutionary process, show better predictions.

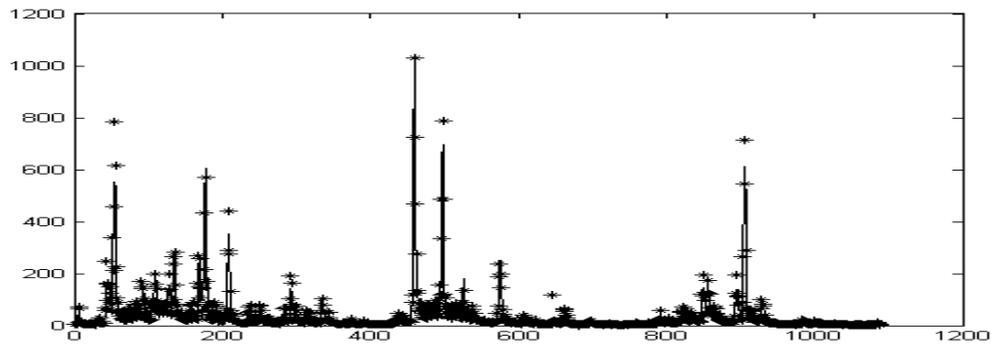
In spite of the fact that evolutionary input identification procedure should result to improve approximation, the results of testing phases represent else. However it must be noted that the search space in Float par scenario is increased exponentially. Considering total states of  $2^8$  in 12par scenarios and  $2^{20}$  in Float par scenarios, it can consider that the evolutionary process of structure identification in Float par scenarios can not converge to optimal genotypes. The result of Float par (without applying elitism) can be considered as local optima, comparing the results of Float par-elitism scenario. The results may be improved by increasing the number of population in each generation and/or number of generation. However it must be noted that the evolution of structure via genetic process requires much computational effort than parameter identification by genetic process. In this study, the genetic process of structure identification with 50 generation and 50 individuals in each generation needs 48 hours to be terminated. So continuing the evolutionary structure identification may not be justified.

The results of 12par scenarios show a better approximation among the others. It can be concluded that the evolution structure identification process has converged to optimal structure. The behavior of achieved structures show a good approximation capability in training and V1 testing sequences. However the results of V2 and V3 sequences are less accurate. The results of V2 testing period can be improved by considering its data in training procedure.

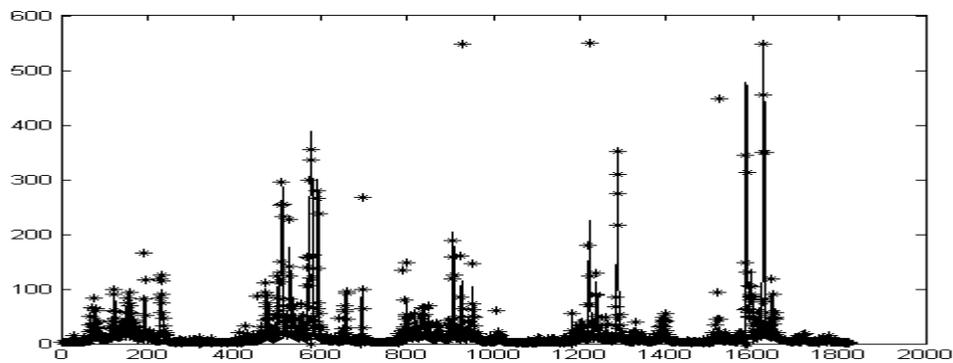
In this study, Mean Square of Errors (MSE) is applied as the training objective function. Previous studies show that consideration of this objective leads to a good prediction in regular events, but it converges to inaccurate results when confronting with extreme ones. Figure (2) shows the prediction of the best-achieved model, i.e., 12par-elitism scenario, in considered sequences. Applying multi-objective training function may result to improve accuracy, which can be considered as further research. In addition, the concept of co-evolution, the optimization of structure and parameters at the same time, which has more natural justification, can be applied as an interesting topic.



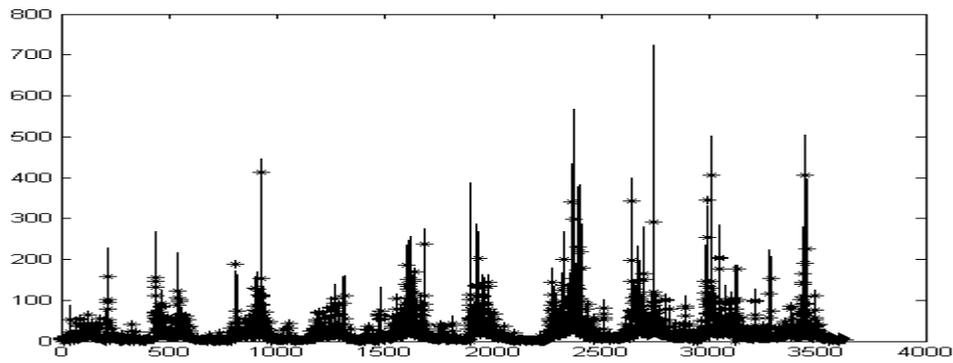
(a)



(b)



(c)



(d)

Figure (2): The behavior of achieved structure, i.e., [12 6 8 1] MLP neural network in describing the rainfall-runoff process in Leaf River; (a) training phase (1956-1965), (b) input identification phase (1949-1950), (c) structure identification phase (1951-1955), and (d) testing phase (1966-1975)

## 6. Conclusion

In this study, a rigid framework in hybrid soft computing-based paradigms, evolutionary neural networks, is applied as a modeling procedure for describing the rainfall-runoff process. A method based on correlation analysis is presented for input

identification of MLP networks. The structure of MLP is obtained through a binary code genetic algorithm and the parameter optimization of MLP is calculated with hybridization of genetic algorithm and backpropagation. An inspired procedure based on the behavior of natural systems is performed for modeling construction. The simulated results based on Leaf River data, show that the application of soft computing-based systems can produce an intelligence behavior in describing the rainfall-runoff process and may have the great potentials in new era of hydrologic modeling.

## 7. Acknowledgements

The authors would like to thank Professor S. Sorooshian and Professor K. L Hsu for providing the data set of Leaf River Basin and their recent publications. They do not hesitate to thank Professor Jorge A. Ramirez for his concern and great attention. This article is dedicated to Professor Jose D. Salas for his great researches in hydrologic modeling and the one, whom this conference is held on his honor.

## 8. References

- [Achela, et. al., 1998] Achela, D., Fernando, K., Jayawardena, W., "Runoff forecasting using RBF networks with OLS algorithm", *Journal of hydrologic engineering*, VOL. 3, NO. 3, pp. 203-209, 1998.
- [Anamala, et. al., 2000] Anamala, J., Zhang, B., Govindaraju, R., S.: "Comparison of ANNs and empirical approaches predicting watershed runoff", *Journal of water resources planning and management*, VOL. 126, NO. 3, pp. 56-66, 2000.
- [Badossy and Disse, 1993] Bardossy, A., Disse, M., "Fuzzy rule-based models for infiltration", *Water resources research*, VOL. 29, NO. 2, pp. 373-382, 1993.
- [Bardossy and Duckstein, 1995] Bardossy, A., Duckstein, L.: *Fuzzy rule-based modeling with applications to geophysical, biological & engineering systems*. CRC Press, Boca Raton, 1995.
- [Chang & Chen, 2001] chang, F., J., Chen, Y., C.: "A counter-propagation fuzzy-neural network modeling approach to real time stream flow prediction", *Journal of hydrology*, VOL. 245, pp. 153-164, 2001.
- [Coley, 1999] Coley, D., A., *An introduction to genetic algorithms for scientists and engineers*, World Scientific, Singapore, New jersey, London, Hong Kong, 1999.
- [Cooper, et. al., 1997] Cooper, V., A., Nguyen V., T., V., Nicell, J., A.: "Evaluation of global optimization methods for conceptual rainfall-runoff model calibration", *Water science technology*, VOL. 36, NO. 5, pp. 53-60, 1997.
- [Demuth & Beale, 2000] Demuth, H., Beale, M., *Neural Network Toolbox for Use with MATLAB®*, MathWorks, Inc., 2000.
- [Franchini, 1996] franchini M., "Use of genetic algorithm combined with local search method for the automatic calibration of conceptual rainfall-runoff models", *Hydrological sciences*, VOL. 41, NO. 1, February 1996.
- [Furundzic, 1998] Furundzic, D.: "Application example of neural networks for time series analysis: rainfall-runoff modeling", *Signal Processing*, VOL. 64, pp. 383-396, 1998.
- [Gorzalczany, 2002] Gorzalczany, M. B., *Computational Intelligence Systems and Applications*, Physica-Verlag Company, Heidelberg, Germany, 2002
- [Han, et. al., 2000] Han, D., Cluckie, I., D., Karbassioun, D.: " A fuzzy logic approach to river flow modeling", *Proceedings of stochastic hydraulics 2000*.
- [Hsu, et. al., 1995] Hsu, K.,-L., Gupta, H., V., Sorooshian, S.: "Artificial neural networks modeling of the rainfall-runoff process", *Water resources research*, VOL. 31, NO. 10. pp. 2517-2530, 1995.

- [Hsu et. al. 2002] Hsu, K., S. Sorooshian, H.V. Gupta, X. Gao, and B. Imam, "Hydrologic Modeling and Analysis Using A Self-Organizing Linear Output Network," International Environmental Modeling and Software Society Conference, Vol. 2, p172-177, Lugano, Switzerland, June 24-27, 2002.
- [Hsu et. al. 2002] Hsu, K., S. Sorooshian, H.V. Gupta, X. Gao, and B. Imam, "Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis" *Water Resources Research*, VOL. 38, NO. 12, 1302, pp 1-16. 2002.
- [Imrie, et. al., 2000] Imrie, C., E., Durucan, S., Korre, A.,: "River flow prediction using artificial neural networks: generalization beyond the calibration range", *Journal of hydrology*, VOL. 233, pp. 138-153, 2000.
- [Jang, et. al., 1997] Jang, J.,-S., R., Sun, C., T., Mizutani, E.,: *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice Hall, Upper Saddle River, NJ, 1997.
- [Karunanithi, et. al., 1994] Karunanithi, N., Grenney, W., J., Whitley, D., Bovee, K.,: "Neural networks for river flow prediction", *Journal of computing in civil engineering*, VOL. 8, NO. 2, pp. 201-220, 1994.
- [Khu, et. al., 2001] Khu S., T., Liong, A.,-Y., Babovic, V., Madsen, H., Muttil, N., "Genetic programming and its application in real-time runoff forecasting", *Journal of the American water resources association*, Vol. 37, NO. 7, pp. 439-451, 2001.
- [Liong, et. al., 2001] Liong, S.,-Y., Khu, S.,-T., Chan, W.,-T.,: "Derivation of pareto front with genetic algorithm and neural network", *Journal of hydrologic engineering*, VOL. 6, NO. 1, pp. 52-61, 2001.
- [Mason, et. al., 1996] Mason, J., C., Price, R., K., Tem'Me A.,: "A neural network model for rainfall-runoff using radial basis function", *Journal of hydraulic research*, VOL. 36, NO. 4, pp. 537-548, 1996.
- [Minns & Hall, 1996] Minns, A., W., Hall, M., J.,: "Artificial neural networks as rainfall-runoff models", *Hydrological sciences*, VOL. 41, NO. 3, pp. 399-417, 1996.
- [Ndiritu & Daniell, 2001] Ndiritu, J., G., Daniell, T. M.,: "An improved genetic algorithm for rainfall-runoff model calibration and function optimization", *Mathematical and computer modeling*, VOL. 33, pp. 695-706, 2001.
- [Ozelkan & Duckstein, 2001] Ozelkan, E. C., Duckstein, L.,: "Fuzzy conceptual rainfall-runoff models", *Journal of hydrology*, VOL. 253, pp. 41-68, 2001.
- [Shamseldin, 1997] Shamseldin, A. Y., *Application of neural networks technique to rainfall-runoff modeling*, *Journal of Hydrology*, VOL. 199, 1997, pp. 272-294.
- [Sorooshian, et. al., 1993] Sorooshian, S., Duan, Q., Gupta, V., K., "Calibration of rainfall-runoff models: application of global optimizations to Sacramento Soil Moisture Account model", *Water resources research*, VOL. 29, NO. 4, pp. 1185-1194, 1993.
- [Wang, 1997] Wang X. J., *A course in fuzzy systems and control*, Prentice Hall Inc., Upper Saddle River, NJ, 1997.
- [Woolhiser, 1996] Woolhiser D., A.,: "Search for physically based runoff model: a hydrologic El Dorado?", *Journal of hydraulic engineering*, VOL. 122, NO. 3, pp. 122-129, 1996.
- [Xiong, et. al., 2001] Xiong, L., Shamseldin A., Y., O' Connor, K., M.,: "A non linear combination of the forecasts of rainfall-runoff models by the first order Takagi-Sugeno fuzzy system", *Journal of hydrology*, VOL. 254, 196-217, 2001.
- [Zhu & Fujita, 1994] Zhu., M., L., Fujita M.,: "Comparisons between fuzzy reasoning and neural network methods to forecast runoff discharge". *Journal of hydroscience and hydraulic engineering*, VOL. 12, NO. 2, pp. 131-141, 1994.