Applying Genetic Programming to Model Rainfall-Runoff

P.A. Whigham
CSIRO Land and Water, P.O. Box 1666,Canberra, A.C.T. 2601,AUSTRALIA
Peter.Whigham@cbr.clw.csiro.au
P.F. Crapper
CSIRO Land and Water, P.O. Box 1666,Canberra, A.C.T. 2601,AUSTRALIA
Peter.Crapper@cbr.clw.csiro.au

Abstract Genetic Programming is an inductive form of machine learning that evolves a computer program to perform a task defined by a set of presented (training) examples. An initially random population of programs, generated using a formal grammar, evolves as a population over a number of generations using the genetic operators of crossover and mutation. Each generation programs are selected for crossover, mutation and reproduction based on their relative fitness with the training examples. This technique mimics aspects of Darwinian natural selection and drives the system towards discovering more useful programs. Genetic Programming has been successfully applied to problems that are complex, non-linear and where the size, shape and overall form of the solution are not explicitly known in advance. This paper describes the application of a grammatically-based Genetic Programming system to discover rainfall-runoff relationships for two vastly different catchments. A context-free grammar is used to define the search space for the mathematical language used to express the evolving programs. A daily time series for of rainfall-runoff is used to train the evolving population. A deterministic lumped parameter model, based on the unit hydrograph, is compared with the results of the evolved models. The favourable results of the Genetic Programming approach show that machine learning techniques are potentially a useful tool for developing hydrological models, especially when surface water movement and water losses are poorly understood.

1. INTRODUCTION

The major objective of rainfall-runoff modelling is to predict the runoff of a catchment from the rainfall incident on the catchment. The response of the catchment (especially Australian catchments) is highly capricious depending not only on the catchment characteristics (e.g. topography, area), vegetation characteristics and antecedent conditions, but the meteorological conditions (e.g. areal distribution of rainfall) in a highly non-linear and unpredictable fashion. Developing models that describe this relationship may help in understanding the overall behaviour of the catchment and support the development of more process-based models and catchment classification schemes.

This paper will compare two different approaches to predicting rainfall-runoff relationships. One approach uses an evolutionary machine learning algorithm which evolves programs defined in a formal language; the other approach uses a (more traditional) deterministic modelling framework based on the unit hydrograph.

1.1 Machine Learning

The field of evolutionary computation has been widely studied since the 1960's. This form of machine learning is characterised by the use of a population of objects that compete to perform some specified task. Using biological analogies, the population of possible solutions are modified in two main ways:

- *Mutation* of an individual, which is an asexual operation causing (normally) a small change in the individual, and
- *Mating* between individuals, which mixes the parent representations to create a child.

Individuals are given a *fitness* measure, which is defined by their performance against a training set of examples relating input and output patterns. Selection of individuals for mutation and mating are based in a proportional way on their *fitness*. This selection mechanism (similar to Darwinian natural selection) drives the population towards better individuals and therefore better solutions.

1.2 Bias and Learning

Bias may be defined as the factors that influence a learning system to favour certain hypotheses or strategies. The application of a learning system always involves some form of bias. Bias may be introduced in any of the following areas:

- The problem representation.
- The operators used to search the representation space.
- The structural constraints of the representation.
- The search constraints when manipulating the representation.
- The criterion used to evaluate proposed solutions.

The use of bias in machine learning has been promoted for many years when knowledge is available that can help narrow the search space of the problem. For example, Lenat [1984] stated "All our experiences in AI research have led us to believe that for automatic programming, the answer lies in *knowledge*, in adding a collection of expert rules which will guide code synthesis and transformation".

When knowledge about the structure and form of good solutions is known there should be the opportunity to define this knowledge explicitly. This is described as background knowledge, and includes bias of the language and how particular parts of the search space are to be explored.

1.3 Rainfall-Runoff Modelling

One of the traditional approaches to hydrograph modelling is to use the concept of the Instantaneous Unit Hydrograph (IUH). The IUH can be defined as the hydrograph produced by the instantaneous application of a unit depth of rainfall to a catchment. The shape of the IUH is similar to a single peak hydrograph with a rapid rise and a slower decay. The fundamental assumption in the IUH model is that the precipitation input is equal to the integrated streamflow output. The non-linear relationship between rainfall and streamflow has led to the development of *effective rainfall*, which is determined by applying a non-linear filter to the raw rainfall data. This effective rainfall is then equated with the integrated streamflow for the specified catchment.

The IHACRES model applied in this paper is based on IUH principles. The model defines a unit hydrograph for total streamflow by defining separate unit hydrographs for the quickflow and the slowflow components. The model is defined by six parameters, four of which are determined directly from the raw rainfall, streamflow and temperature (or a surrogate), while the other two (the non-linear parameters) are calibrated using a trial and error search procedure, optimising the model to fit the observed rainfall-runoff relationship. Additional details about the model are contained in Jakeman et al.(1) and Littlewood and Jakeman(2).

1.4 Format of the paper

This paper will proceed as follows: Genetic Programming (GP) will be introduced as a form of program induction using evolution. Formal grammars are then shown to be a useful method for defining language structure and may be neatly described in the same framework as GP. Grammars are shown to be able to be used as generators for sentences of structured languages and may be manipulated using a crossover operator that maintains the language structure. Additionally, a grammar is shown to be able to express both language and search bias. Finally, a grammatical GP system is applied to two catchments with vastly different rainfall-runoff behaviour. The results are compared with the deterministic model IHACRES.

2. GENETIC PROGRAMMING

The field of program induction, using a tree-structured approach, was first clearly defined by John Koza [1992]. This field, named Genetic Programming (GP), evolved a solution in the form of a Lisp program using an evolutionary, population-based, search algorithm which extended the fixed-length concepts of Genetic Algorithms defined by Holland [1992]. The structures were defined as a combination of functions (arity > 0) and terminals (0-arity functions) which combined to form Lisp programs. The following steps summarise the search procedure used with GP.

1. Create an initial population of programs, randomly generated as compositions of the function and terminal sets.

2. WHILE termination criterion not reached DO

   a) Execute each program to obtain a performance (fitness) measure representing how well each program performs the specified task.
   b) Use a fitness proportionate selection method to select programs for reproduction to the next generation.
   c) Use probabilistic operators (crossover and mutation) to combine and modify components of the selected programs.

3. The fittest program represents a (partial) solution to the problem.

GP has been applied successfully to many problems, however the lack of structure in the possible combinations of functions and terminals meant that there was no way to define bias in terms of program construction or how the programs were modified. This lack of explicit biasing mechanisms has probably slowed the wider application of GP to real-world problems.

3. FORMAL GRAMMARS AND LEARNING

A formal grammar is a production system which defines how nonterminal symbols may be transformed to create terminal sentences of a language. A grammar is represented by a four-tuple (N, Σ, P, S), where N is the alphabet of nonterminal symbols, Σ is the alphabet of terminal symbols, P is the set of productions and S is the designated start symbol.

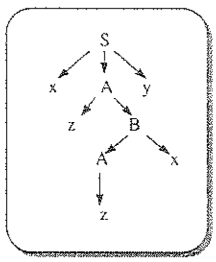

Figure 1. A Derivation Tree

3.1 Derivation Steps and Derivation Trees

A derivation step represents the application of a production to some string which contains a nonterminal. In general, a series of derivation steps may be represented by a syntax tree or derivation tree. As shown in Figure 1, the derivation steps

$$S \Rightarrow xAy \Rightarrow xzBy \Rightarrow xzAxy \Rightarrow xzzxy \quad (1)$$

may be represented as a tree. The genetic operators crossover and mutation are applied directly to these trees.

3.2 How do we use the grammar ?

A grammar G may be used as a generator of sentences which are a part of the language L(G). A limit on the depth of the derivation tree created from the grammar is necessary to ensure that the generation process halts (there is also the practical issues of implementing the generation on a finite machine and being able to execute the created programs). There are two steps involved in using G to generate random sentences from L(G). Initially each production $P \in G$, is labelled with the minimum depth of derivation tree that can be created from this production to produce a string of terminal symbols. This min-depth-tree value is used to guide the selection of productions when randomly creating sentences from L(G), limited by some maximum depth of derivation tree.

Each derivation tree is evaluated against the test data to obtain a fitness measure. Selection of programs (derivation trees) for crossover and mutation are based on their relative fitness. Each generation, the population is transformed using the genetic operators of crossover and mutation to give the next population. This process continues until some maximum number of generations have passed, or an acceptable solution has been discovered.

3.3 Genetic Operators

This section describes how a derivation tree is modified from one generation to the next. Since mutation is not used in the examples, only the crossover operator will be described.

3.3.1 Crossover

Crossover applies 2 (parent) individuals and creates 2 (offspring) individuals. Crossover is defined by two parameters: the probability of crossover occurring and the nonterminal $A \in N$ where crossover will be attempted. Given two derivation trees d_1 and d_2 the steps involved are (see Fig. 2):

- Randomly select a nonterminal site A_1 from d_1 and A_2 from d_2.
- Swap the derivation trees below A_1 and A_2 thereby creating two new derivation trees.
- Insert these trees into the next-generation population.

If d_1 or d_2 do not contain the nonterminal A then no crossover is possible and the operation is aborted.

The benefit of using the derivation trees to represent the population now becomes clear; by defining crossover to swap subtrees at the same nonterminal guarantees that the language defined by the grammar is maintained.

4. LANGUAGE AND SEARCH BIAS

The language L(G) is shaped by the grammar G. Since this grammar is declaratively defined the language bias for the learning system is clearly stated and easily changed without changing underlying functions of the system. This promotes the exploration of new language constructions which become apparent during the evolution of partial solutions. For example, a combination of terms that consistently appear in partial solutions may be explicitly stated as part of the grammar. Thus an incremental approach to developing a solution may be performed in a declarative manner.

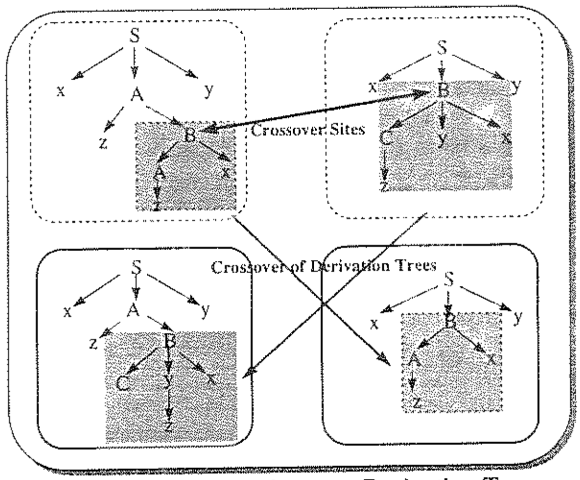

Figure. 2. Crossover between Derivation Trees

An additional language bias may be introduced when defining the grammar. Each production is labelled with a weighting factor, which biases the probability of each production being selected during the generation of the initial population. This allows background knowledge about which terminal symbols are likely to be most useful in developing a solution.

Search bias is introduced by allowing the crossover and mutation sites (i.e. the nonterminals) to be individually specified. For a particular setup, a number of crossover and mutation operations may be defined. Each crossover and mutation is defined as a nonterminal with a probability of occurrence. The ability to specify which nonterminals are used for these operators allows a declarative specification of where the majority of the search for better solutions will be performed. It also promotes the exploration of the search space in an interactive sense by the user. This paper will not pursue the goal of achieving the best possible solution. Rather, the paper is being used to demonstrate the appropriateness of this technique to rainfall-runoff modelling, recognising the potential for further discovery and improvements. The system described in this paper has been previously defined by Whigham [1995, 1996] and is entitled CFG-GP.

5. CATCHMENT DESCRIPTIONS

In order to test the two modelling approaches two very different catchments were chosen. The first catchment was the Teifi catchment at Glan Teifi in Wales. The Teifi catchment is a rural catchment draining 893.6 km^2 with an average annual rainfall of 1368 mm. This station was maintained and operated by the UK Environment Agency and the data can be obtained from the Institute of Hydrology. Compared with the Namoi catchment, the number of rain days per annum is much greater at Teifi but the maximum daily rainfall is only about half the value for the Namoi. The other major difference is that runoff percentages are very much higher at Teifi. For the calibration period the runoff percentage was 66.7% and for the simulation period the runoff percentage was 74.95%.

The second catchment was located within the Namoi River catchment in northeastern New South Wales. This catchment was chosen to be as different as possible from the Namoi catchment. Using the Department of Land and Water Conservation (the gauging authority) naming convention the catchment is referred to as 419030 or the Manilla Rv at Barraba (30° 23' 24" S and 150° 37' 08" E). This catchment is of 568 km^2 and drains the southern part of the Nandewar Range. Within the catchment there are three reliable long-term raingauge stations with average annual rainfalls of 686mm, 704mm and 727mm. These stations have a reasonable spread of location and altitude and the average of the three values has been used as the catchment rainfall. More sophisticated techniques do exist for determining catchment rainfall but given that there were only three rainstations such sophisticated approaches are inappropriate (Fleming P. M. pers comm). For very large events (greater than 100mm) there was a strong relationship between rainfall and runoff but as the size of the event decreased the relationship between rainfall and runoff became more random. The rainfall in this part of the country is strongly summer dominated, which influenced the selection of the calibration and simulation periods. The calibration run was done from 13 November 1965 to 10 March 1966 and the simulation run was done from 4 November 1966 to 13 March 1967. In spite of this catchment having a comparatively high rainfall (by Australian standards at least) and our selection of the high rainfall months, the runoff percentage over the calibration period was only 6.12% and the simulation period was 8.24%.

6. CFG-GP SETUP

The grammar, G_{exp}, used by CFG-GP to develop the rainfall-runoff models allowed general mathematical functions to be evolved, and was defined as follows.

G_{exp} =

{S,
N = {EQU, NL, EXPN},
Σ = {+,-,*,/,r0,r1,r2,r3,r4,r5,av5,av10,av15,av20, av25, av30, av40, av50,av60,av100,$\Re$},
P =
{
S → + EQU NL
NL → * EQU EXPN
EXPN → exp EQU
EQU → + EQU EQU | - EQU EQU
EQU → * EQU EQU | / EQU EQU
EQU → r0 | r1 | r2 | r3 | r4 | r5
EQU → av5 | av10 | av15 | av20 | av25
EQU → av30 | av40 |av50 | av60 | av100
EQU → $\Re$
}
}

The terminal symbols r0,r1,..r5 represent the rainfall for the current day up to the last 5 days rain. The av5, av10, .. av100 terminals are the average rainfall for the last 5, 10, ... 100 days, respectively. The terminal $\Re$ is a random floating point number between -10.0 and 10.0 which is generated for each occurrence of $\Re$ when the initial population is created. The grammar has a structural bias to form equations that are composed of a linear component and a non-linear (exponential) component. This is shown by the production S → + EQU NL, which forces all programs to have the minimal structure of A + B * exp(C), where A, B and C are climate variables.

CFG GP Parameter	Setting
POPULATION SIZE	1000
GENERATIONS	50
GRAMMAR	G_{exp}
MAX. TREE DEPTH	15
CROSSOVER ⊗ = {EQU}	90%
FITNESS MEASURE	Minimise RMSE

Table 1. CFG-GP Parameter Settings for evolving a simple rainfall-runoff Model

Table 1 shows the CFG-GP setup parameters which were used to develop both catchment models. The fitness measure used to evaluate the performance of each program during calibration was the root mean square error (RMSE). This approach treated each point, irrespective of magnitude, as equally important. Further work of interest would be to study the affects of different metrics for the fitness measurement, based on various criterion. A likely outcome of this work would be a classification for catchments based on which type of metric best evolves an overall rainfall-runoff relationship.

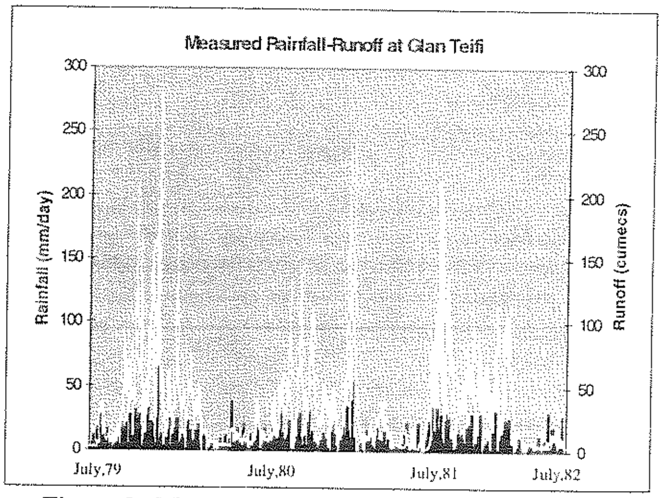

Figure 3. Measured Rainfall and Runoff at Teifi

7. RESULTS

7.1 The Glan Teifi Catchment

The measured rainfall and streamflow, for the Teifi catchment between 1979 and 1982, is shown in Figure 3 (rainfall events are shown in black). The simulated streamflows determined by IHACRES and CFG-GP are shown in Figures 4 and 5. A visual comparison with the measured streamflow (Figure 3), indicates that both approaches have captured the basic response of the catchment, however IHACRES appears to have better represented the extreme streamflow events. The root mean square error (RMSE) for IHACRES was 0.46 and for CFG-GP was 0.47. The CFG-GP equation for the Teifi catchment was defined as follows.

+(+(**r1**,+(**av40**,*(**av10**,**av100**))),*(*(**av5**, +(-17.121983,*(**av5**,**av40**))),exp(-3.739896))) (2)

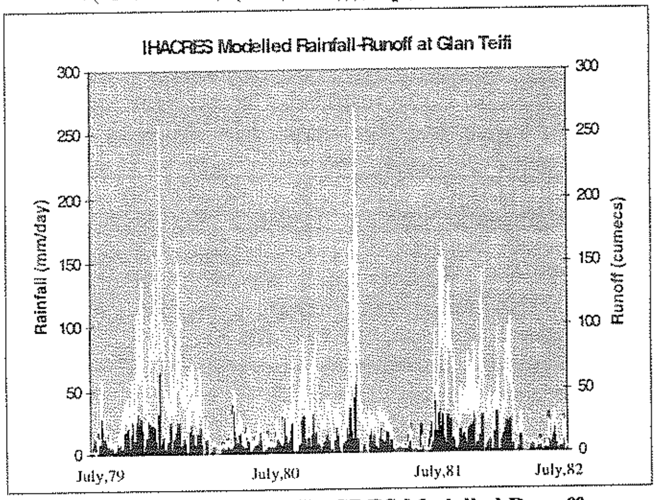

Figure 4. Rainfall and IHACRES Modelled Runoff

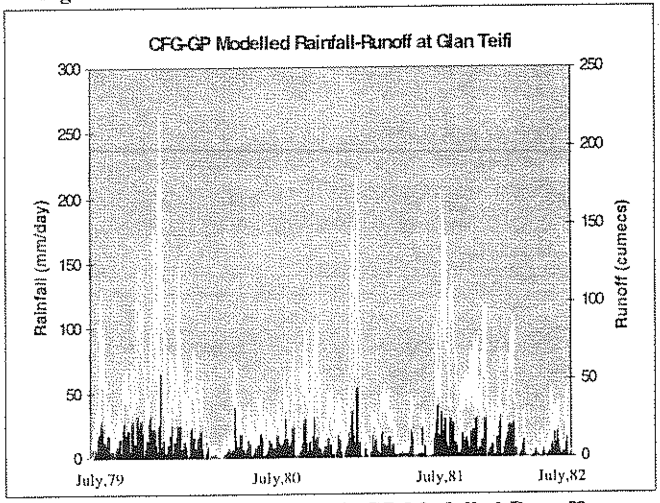

Figure 5. Rainfall and CFG-GP Modelled Runoff

Equation (2) shows that the catchment was influenced by antecedent conditions that could extend for several months into the past (the **av100** variable represents the average rainfall for the last 100 days). Note also that the constant exponential expression in equation (2), namely **exp(-3.739896)**, means that the resultant equation is a linear function of **r1**,**av5**, **av10**, **av40**, and **av100**.

7.2 The Manilla River at Barraba, N.S.W.

The measured rainfall and subsequent streamflow for the simulated period in the Barrada catchment are shown in Figure 6. As can be seen for large events, there is a strong relationship between rainfall and runoff. For smaller events, there is not a significant relationship between the rainfall and runoff. The simulated streamflows determined by the two approaches are shown in Figures 7 and 8. The RMSE for the IHACRES approach was 0.54 and for CFG-GP was 0.50. For the purposes of our comparison these results are similar. The evolved equation found by CFG-GP was:

+(/(/(/(/(**r0**,-1.911790),-0.474622),-4.164400), -(/(-1.542888,5.944119),-(**av10**,**r0**))), *(**r0**,exp(/(0.251564,**av10**)))) (3)

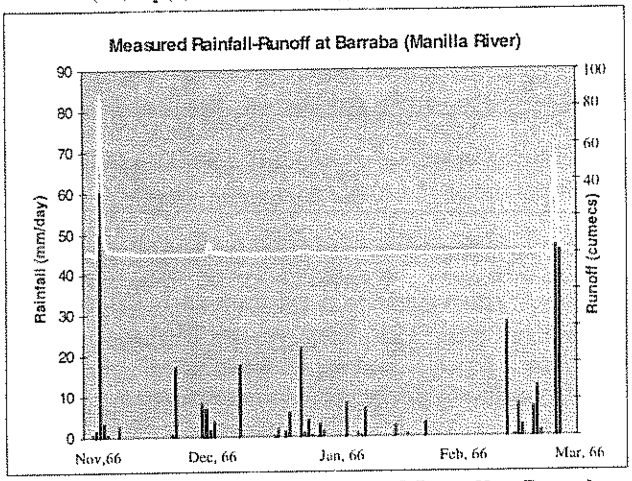

Figure 6. Measured Rainfall and Runoff at Barraba

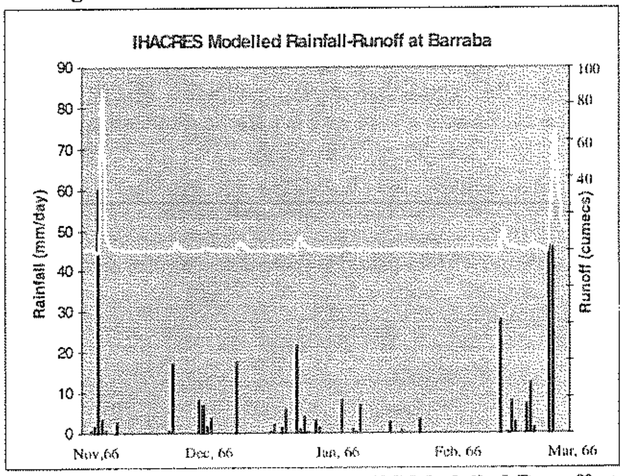

Figure 7. Rainfall and IHACRES Modelled Runoff at Barraba

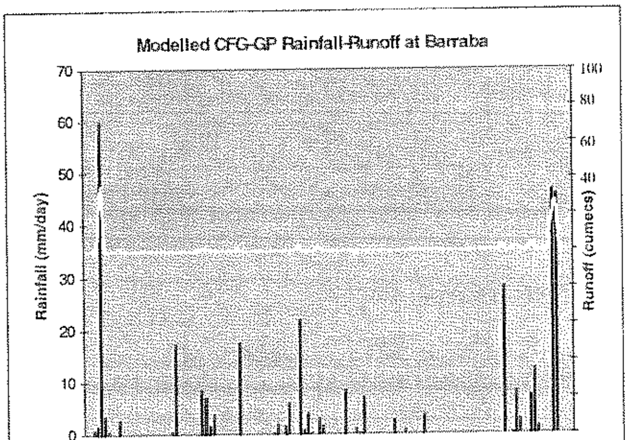

Figure 8. Rainfall and CFG-GP Modelled Runoff at Barraba

It is worth noting that equation (3) uses the current days rainfall (**r0**), and the average of the last 10 days rainfall (**av10**). Additionally, (3) has the nonlinear term (**exp**(/(0.251564,**av10**)), which is a function of **av10**. A comparison of equations (2) and (3) shows that the two catchments have been modelled in very different ways. The Welsh catchment has been modelled using long term averages in a linear combination, whereas the Australian catchment has been modelled using short average times and the current day in a nonlinear fashion. This would suggest that the underlying processes that are driving the water movement throughout both catchments are quite different.

When an attempt was made to calibrate over different consecutive seasons for the Barraba data, the IHACRES model was not able to find coefficients to suit all seasons, and therefore could not converge. This also accounts for the short calibration and simulation periods. However the CFG-GP approach, because it makes no assumption about underlying relationships, was able to be calibrated over successive seasons. When CFG-GP was calibrated using this longer period (1000 days) the resultant model achieved significantly better results on the original simulated data set (RMSE = 0.27). The response of this modelled streamflow is shown in Figure 9. The evolved equation was:

+(**exp**(+(/(**exp**(-4.874963),-0.796608),
/(/(**r0**,-1.864706),+(-3.018028,-4.418388) (4)
))),*(-3.240420,**exp**(-1.181253)))

The interesting comparison between equations (3) and (4) is that using the larger dataset for calibration resulted in a solution that was a nonlinear function of (**r0**). No average rainfall value was found to be useful. This implies that the Barrada catchment has a very quick response between rainfall and runoff, and no significant seasonal signal.

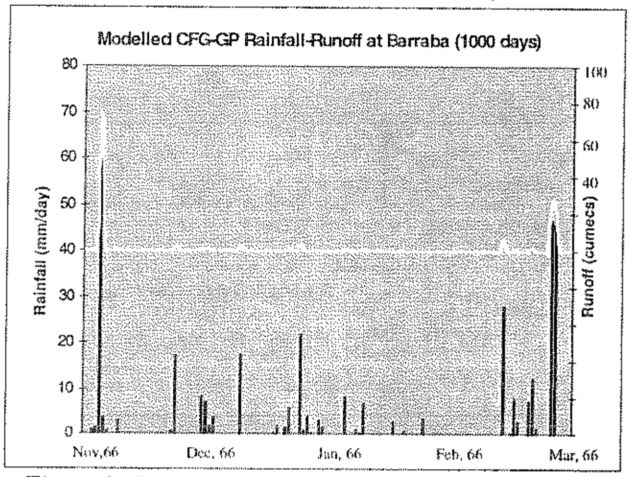

Figure 9. Rainfall and CFG-GP Modelled Runoff based on a calibration period of 1000 days.

8. CONCLUSION

In the present work we have compared the results obtained with a deterministic lumped parameter model, based on the unit hydrograph approach, with those obtained using a stochastic machine learning model.

For the Welsh catchment the results between the two models were similar. Since rainfall and runoff were highly correlated the deterministic assumption underlying the IHACRES model was satisfied. Therefore IHACRES could achieve a satisfactory correlation between simulated and unseen data. It is also interesting to note that for this catchment the runoff ratio was approximately 70% which suggests that a relationship does indeed exist between the rainfall and runoff. The CFG-GP approach does not require any causal relationships but achieved similar results. The behaviour of the studied Australian catchment is very different from the Welsh catchment. The runoff ratio was very low (7%) and hence the *a priori* assumptions of IHACRES (and other deterministic models) were a poor representation of the real world. This was demonstrated by the inability of IHACRES to use more than one seasons data for calibration purposes and only able to use data from a high rainfall period. Since the CFG-GP approach did not make any assumptions about the underlying physical processes, calibration periods over more than one season could be used. These led to significantly improved generalisations for the modelled behaviour of the catchment.

In summary, either approach worked satisfactorily when rainfall and runoff were correlated. However, when this correlation was poor, the CFG-GP had some advantages because it did not assume any underlying relationships. In these circumstances the use of evolutionary algorithms warrants further consideration.

9. REFERENCES

Holland, J.H. Adaptation in Natural and Artificial Systems. MIT Press, second edition, 1992.

Jakeman, A. J., Littlewood, I. G. and Whitehead, P.G. "Computation of the instantaneous unit hydrograph and identifiable component flows with application to two small upland catchments. Journal of Hydrology, 117, 275-300, 1990.

Koza, J.R. *Genetic Programming: on the programming of computers by means of natural selection*. A Bradford Book, MIT Press, 1992.

Lenat, D. The role of heuristic's in learning by discovery. In R.S. Michalski, J. Carbonell and T. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach*, Chap. 9, Springer-Verlag, 1984.

Littlewood, I. G. and Jakeman, A. J. (1994) "A new method of rainfall-runoff modelling and its applications in catchment hydrology. In: P. Zannetti (ed) Environmental Modelling (Volume II), Computational echanics Publications, Southampton, UK, 143-171.

Whigham, P.A.. Inductive Bias and Genetic Programming. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages. 461-466. UK:IEE, Sept., 1995.

Whigham, P.A. Grammatical Bias for Evolutionary Learning. PhD Thesis, School of Computer Science, University College, University of New South Wales, Australia, 1996.