



# Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling

S.-T. Khu, M. G. F. Werner

## ► To cite this version:

S.-T. Khu, M. G. F. Werner. Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling. *Hydrology and Earth System Sciences Discussions*, 2003, 7 (5), pp.680-692. hal-00304915

**HAL Id: hal-00304915**

**<https://hal.science/hal-00304915>**

Submitted on 18 Jun 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling

Soon Thiam Khu<sup>1</sup> and Micha G.F. Werner<sup>2</sup>

<sup>1</sup>University of Exeter, Harrison Building, North Park Road, Exeter, EX4 4QF, UK.

<sup>2</sup>Delft University of Technology, P.O.Box 5048, 2600 GA, The Netherlands

Email for corresponding author: s.t.khu@exeter.ac.uk

## Abstract

Monte-Carlo (MC) simulation based techniques are often applied for the estimation of uncertainties in hydrological models due to uncertain parameters. One such technique is the Generalised Likelihood Uncertainty Estimation technique (GLUE). A major disadvantage of MC is the large number of runs required to establish a reliable estimate of model uncertainties. To reduce the number of runs required, a hybrid genetic algorithm and artificial neural network, known as GAANN, is applied. In this method, GA is used to identify the area of importance and ANN is used to obtain an initial estimate of the model performance by mapping the response surface. Parameter sets which give non-behavioural model runs are discarded before running the hydrological model, effectively reducing the number of actual model runs performed. The proposed method is applied to the case of a simple two-parameter model where the exact parameters are known as well as to a widely used catchment model where the parameters are to be estimated. The results of both applications indicated that the proposed method is more efficient and effective, thereby requiring fewer model simulations than GLUE. The proposed method increased the feasibility of applying uncertainty analysis to computationally intensive simulation models.

**Keywords:** parameters, calibration, GLUE, Monte-Carlo simulation, Genetic Algorithms, Artificial Neural Networks, hydrological modelling, Singapore

## Introduction

When applying a hydrological model to simulate runoff from a particular catchment, modellers must identify a set of parameters such that the model predicts the behaviour of the natural system realistically. This has been the subject of extensive research (Sorooshian and Gupta, 1983, 1995; Kuczera, 1997) and it is commonly suggested that hydrological models are often over-parameterised (Beven, 1989; Jakeman and Hornberger, 1993). The response surface of the objective function to the parameters may also suffer from multi-modality and discontinuity, thus rendering many optimisation methods inefficient (Xiong and O'Connor, 2000). Adding to the difficulty of the concept of a single 'optimal' parameter set is that this optimum set depends very much on the events used in calibration (Beven, 1993) and the length and quality of the data sets used (Sorooshian and Gupta, 1983). Hence, different events may result in very different 'optimal' parameter sets. Equally, the optimum may also depend on the objective function chosen in calibration (Gan *et al.*, 1997).

Monte-Carlo simulation is a robust method for exploring the response surface of the objective function (Melching, 1995) and, based on this technique, Beven and Binley (1992) proposed the Generalised Likelihood Uncertainty Estimation (GLUE) procedure; this rejects the principle of a single optimal parameter set. In GLUE, the likelihood that a given parameter set is a good simulator of the system is a function of the model performance expressed in terms of the objective function chosen. Lamb *et al.* (1998) stated that the definition of likelihood used in GLUE is not to be confused with the traditional statistical definition; rather it is an indication of the relative likelihood that the model is acceptable, given the available data. Parameter sets not considered behavioural because a threshold performance is not achieved are discarded from further analysis. The Bayesian framework within which the procedure has been developed (Romanowicz *et al.*, 1996) effectively allows for the estimation of model uncertainties by constraining prior information on the parameters in the form of prior distributions using the data available. The posterior

parameter distributions which result can be used to make model predictions where the spread, or uncertainty, due to the parameter distributions is shown. Once new data become available, the posterior parameter distributions can be applied again as prior distributions and updated using the new information contained in the additional data.

A practical problem with the GLUE procedure is that, for models with a large number of parameters, the sample size from the respective parameter distributions must be very large to achieve a reliable estimate of model uncertainties (Kuczera and Parent, 1998). The less prior information on the distribution of parameters and acceptable parameter ranges, the higher the number of runs must be. Since hydrological models often have large parameter sets, this can make the application of the GLUE procedure unfeasible due to limited computer resources.

Depending on the prior parameter distributions chosen, the number of redundant model runs may be large. These redundant model runs are runs carried out only to be subsequently discarded due to a performance below the threshold set under the GLUE framework. Considerable saving in computational effort may be achieved if the performance of a given parameter set is estimated without actually running the model. This is especially crucial if the model runs are computationally intensive. Kuczera (1997) proposes a method for identifying hyper-ellipsoid parameter sub-spaces using a probabilistic search method where local and global optima are first identified using a global search algorithm and subsequently the shape of the response surface of the objective function is investigated using gradient-based methods. Although it is shown to reduce the computational effort significantly, the method was applied to a reasonably well-posed problem. Montesinos and Beven (1999) presented the use of a genetic algorithm for identifying parameter sub-spaces that are subsequently sampled and run in the GLUE procedure. Werner and Khu (2001) identified interesting parameter subspaces using a combination of genetic algorithm, Kriging and GLUE; the computational effort could be reduced by as much as 80% without significant loss of information of the interesting sub-spaces. However, Kriging assumes isotropy of the parameter response surface, with the additional drawback that application of Kriging to more than three parameters (such as 4-D mapping) is not trivial.

In this paper, a combination of genetic algorithms (GA) and artificial neural networks (ANN) is applied to identify interesting parameter subspaces for further evaluation using the GLUE procedure. This approach has the same effect as the approach taken in Werner and Khu (2001) but without the assumption on isotropy of the parameter response surface and is more easily applicable to complex response surfaces

with higher dimensions. GA is an efficient way to identify parameter sets above a threshold value of model performance while exploring the response surface of the objective function in the parameter space. ANN is used to map the response surface of the objective function and to estimate the performance of the parameter sets sampled in the GLUE procedure. The proposed method, adopted here as the GAANN procedure, is intended to reduce the number of model evaluations otherwise required in the GLUE procedure.

In this paper, a discussion of the GLUE procedure is followed by a review of the GA and ANN techniques. The proposed approach is then presented in detail and tested using a simple problem where the response to a rainfall event is determined using a Nash Cascade model (Shaw, 1988) with known parameters. It is subsequently applied in parameter estimation for a widely used rainfall runoff model, SWMM, using observed rainfall and runoff from a catchment in Singapore.

## Estimating uncertainties using GLUE

A critical review of the application of physically based distributed modelling (Beven, 1989) with multiple distributed parameters, led to the recognition that, rather than a single globally optimal parameter set, a large number of parameter sets could show equivalent behaviour in terms of the objective function used in calibration. This recognition is the basic principle of the GLUE procedure, where the likelihood that any possible parameter set is a good simulator of the system is expressed in terms of how well the model performs with that parameter set, given the available data. Generally, there are five major steps in the GLUE procedure (Beven and Binley, 1992):

1. Definition of a likelihood measure. This is typically chosen on the basis of the objective function to determine model performance, where the objective function should reflect the use of the model. It is clear that the choice of objective function has significant influence on the results of the procedure, as it influences the likelihoods found for parameter sets and their distribution. Typical examples of objective function are:

- Root Mean Square error;

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (Q_{obs,i} - Q_{sim,i})^2} \quad (1)$$

- Nash-Sutcliffe efficiency (Nash and Sutcliffe, 1970):

$$R^2 = \frac{\sum_{i=1}^m (Q_{obs,i} - Q_{sim,i})^2}{\sum_{i=1}^m (Q_{obs,i} - \bar{Q})^2} \quad (2)$$

where  $Q_{obs,i}$  is the observed discharge at time  $i$ ,  $Q_{sim,i}$  is the simulated discharge  $\bar{Q}$  is the average of the observed discharges.  $m$  is the total number of time steps in the calibration period.

2. For each of the parameters, a prior distribution must be defined. Parameter sets will be sampled randomly from this prior distribution, and it must reflect the prior knowledge on this parameter. In most of the reported applications of the GLUE procedure (Beven and Binley, 1992; Aronica *et al.*, 1998; Uhlenbrook *et al.*, 1999), little prior information was available as to the distribution of each parameter, and a non-informative uniformly distributed prior is selected. The range of the distribution is normally selected based on reasonable values from literature and expert knowledge. Choosing a non-informative prior will have the least effect on the posterior distributions if there is enough observed data available, which is normally not the case. The effort and data required to make a reliable estimate of the posterior distributions is, however, large when compared to that where an informative prior is available.
3. A number of parameter sets ( $N_{MC}$ ) are sampled using the Monte-Carlo technique from the prior parameter distributions. The model is then run using these parameter sets and the model outcome of each run is compared to the observed values using the selected objective function. Based on the value of this objective function, a likelihood value is assigned to the parameter set. As is common with the use of non-informative priors, the distribution of the likelihood function must be normalised to create a proper posterior distribution of likelihoods (Romanowicz *et al.*, 1996).
4. All parameter sets performing below a pre-selected threshold are removed from further analysis. The rationale is that although all the parameter sets may calculate a model response successfully, this response may not be realistic in terms of the behaviour of the system. All parameter sets below the threshold are considered non-behavioural and thus removed by assigning zero likelihood. Clearly, the selection of the value for this distinction between behavioural and non-behavioural can have significant impact on posterior distributions (Melching, 1995). In a general sense, the calibration procedure where an optimal parameter set

is sought, can be seen as the selection of a threshold value such that all parameter sets giving a model efficiency below the optimal value found have a likelihood of zero, and the optimal set a likelihood equal to one (Beven and Binley, 1992). Lees and Wagener (2000) suggested that the threshold value should be selected so that it reflects the expectation of the modeller. It is suggested here to take the argument further and select the value of the threshold independent of the model, model structure and parameter. The selected threshold value should reflect the requirements set for the model with the framework of its practical use. This entails the threshold value dividing behavioural from non-behavioural parameter sets is imposed directly on the fitness in terms of the objective function rather than on the scaled likelihood value.

5. The likelihood values of the behavioural parameter sets are normalised such that the distribution function is again proper. Subsequent predictive model runs using the remaining parameter sets are weighted according to the likelihood value of the parameter used, and from this ensemble the weighted mean and uncertainty bounds of model outcome can be derived.

If a high threshold value is selected, the number of simulation runs retained for analysis is reduced, since all runs lying in the parameter space outside the threshold value are discarded from further analysis. This will result in the method discarding a large number of simulations. Figure 1(a) shows the response surface of the 2-parameter Nash Cascade model (also known as the gamma model), (Shaw, 1988) as an example of an application of the GLUE procedure. The dots represent the randomly generated parameter sets. If the threshold value is set at an efficiency of  $R_{eff} = 0.8$ , the simulations represented by small crosses are the parameter sets retained for further analysis. This shows the relative inefficiency of the GLUE procedure, where only a fraction of the parameter sets run with the model lie within the area of interest, in this case in the order of 20%. The example will be discussed in detail further on in the paper. Similar selections, where large parts of the parameter space are discarded after model evaluation, have been reported in a variety of hydrologic models (van der Perk, 1997; Kuczera, 1997; Campbell *et al.*, 1999).

## Selecting potentially behavioural parameter sets

The main inefficiency of the GLUE procedure is not the generating of the random parameter sets, but comes from

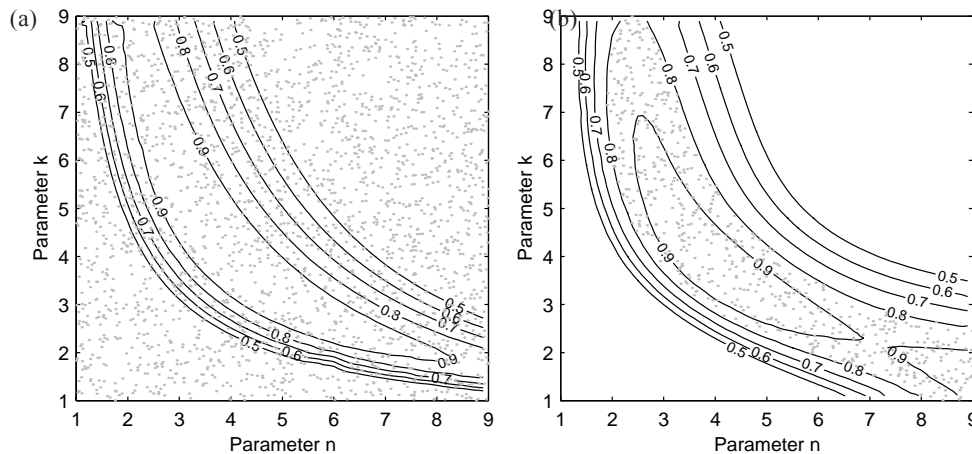


Fig. 1. Contour plot of Nash-Sutcliffe efficiency response surface of the 2-parameter Gamma model. Model performance contours are from: (a) GLUE and (b) mGLUE - GAANN model.

computationally expensive calls to the model. Typically, the number of evaluations required in GLUE (using standard Monte-Carlo sampling) requires tens of thousands of model evaluations. For a simple model requiring moderate computational resources, say 1 minute per simulation, it takes up to 17 days to complete 25000 model simulations, where only about 20% of these simulations are ‘useful’.

There is a number of possible ways to overcome this problem and they are:

1. use a more efficient sampling algorithm, such as Latin hypercube method;
2. replace the computationally intensive model with a faster surrogate model, such as ANN; or
3. use a hybrid of the above 2 techniques.

The proposed method, GAANN, aims to reduce the number of function evaluations and, thereby, to increase the efficiency in the GLUE procedure. Therefore, the authors propose a hybrid method, where part of the hydrological model computation is replaced with a fast surrogate model but the original sampling technique and the GLUE framework remains unchanged. The proposed method serves to identify the contour line representing the ‘threshold’ objective function separating behavioural parameter combinations (sets) from non-behavioural sets in the objective function response surface. The non-behavioural parameter sets are essentially discarded from further analysis. To achieve this, the objective function response surface is mapped based on a small sample of parameter sets using an ANN. This small sample of parameter sets used in the ANN is generated through the application of a genetic algorithm.

## Sampling using Genetic Algorithm

Genetic Algorithm (GA) is an established probabilistic search method in optimisation problems, where the name originates from the analogy of the method with evolutionary theory (Goldberg, 1989). In GA, the value of each parameter is encoded as a binary string of fixed length. The parameter set is then coded by joining the binary strings to form a continuous string called chromosome. The algorithm starts by spawning a group of chromosomes (known as a generation) randomly in the parameter space. The fitness of each of these parameter sets (chromosomes) is evaluated, and this determines the likelihood of an individual surviving into the next generation. Subjecting the existing chromosomes to a process of reproduction (i.e. crossover and mutation) creates the next generation of chromosomes. In crossover, two selected chromosomes with high fitness are intermixed to form two new chromosomes. It is through this process of intermixing (or commonly known as mating) that chromosomes with better fitness emerge. In mutation, a random change occurs in the binary code of a chromosome. This random change causes the algorithm to explore new parameter space. This procedure is carried out until a specified termination criterion is met following the procedure outlined in Fig. 2. Readers are referred to Goldberg (1989) for an extensive overview of GA.

GA gives good results in model parameter optimisation (Wang, 1991; Liong *et al.* 1995) but has to be properly set up to achieve a balance between exploitation and exploration of the parameter space (Goldberg, 1989). Exploitation means that the final solution converges rapidly; exploration means that the search routine has covered a fair amount of parameter space without leaving significant unexplored



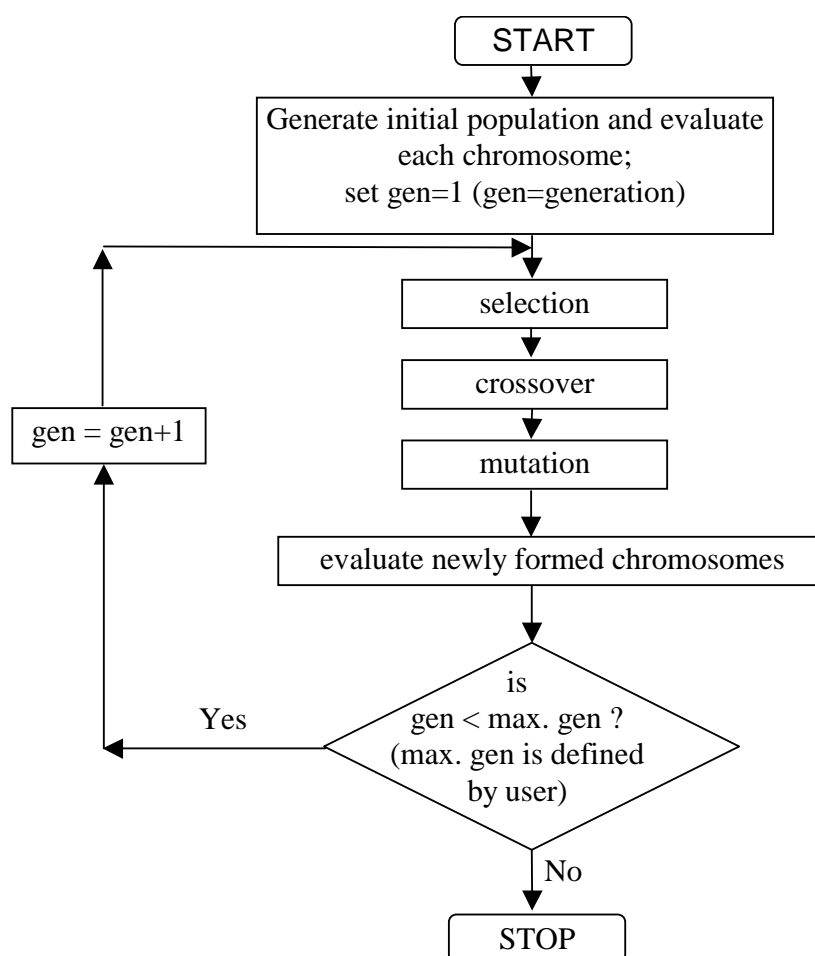


Fig. 2. Flowchart showing the genetic algorithm.

areas. When GA is used as a search routine, the emphasis is on the speed of convergence to the optimal point(s). For the purpose of mapping the response surface of an objective function, the focus should be on locating all regions of interest, and the exploration of the parameter space around the regions of interest.

Montesinos and Beven (1999) used GA to search for interesting subspaces before continuing with the GLUE procedure. They noted that for a high number of GA iterations, there is a tendency for GA to cluster around near optimal points, leading to convergence of the population and narrow uncertainty bounds. This is due to the application of GA as an optimisation technique looking for global optimal parameter sets. The GLUE procedure, however, is founded on the realisation that such a unique optimum does not exist, and exploration should be the key if GA is to be used. The focus on exploitation would cause the GA to converge to some optimal points, hence a possible misrepresentation of the uncertainty bounds.

In this study, GA is applied as a searching technique to locate the interesting parameter sub-spaces, and niching (Goldberg, 1989) is used to avoid premature convergence on local optima, and to find multiple regions of interest. Niching entails that if the Euclidean distance between two parameter sets in a generation is found to fall within the niching radius  $r_n$ , one of the two will be discarded and a new individual randomly sampled from the parameter space. Hence, GA will have a higher probability of exploring all local optima in the parameter space. Using niching, GA is able to locate multiple areas of local optima while finding the region of ‘near’ global optima with only a small number of parameter evaluations. Furthermore, areas in the parameter space with consistent poor performance are sampled comparatively lightly.

## Using ANN as a surrogate model

Artificial Neural Network (ANN) is a computing paradigm designed to mimic natural neural networks in the biological

brain (Hetcht-Nielsen, 1989). ANNs are commonly thought of as universal approximators for function mapping (Hornik *et al.*, 1989). In the area of hydrology and water resources, ANN has been used in many applications (ASCE Task Committee, 2000) such as forecasting water levels or runoff (Karunanithi *et al.*, 1994), prediction of water quality and ecological impact (Rechnagel *et al.*, 1997), regulating operation of treatment plants or networks (Belanche *et al.*, 2000), as a surrogate for computational intensive hydrodynamic models (Ullsch and Roske, 2002), decision variables mapping (Liong *et al.*, 2001), etc.

Although there are many types of ANNs, by far the most widely used is the feed-forward NN or the multi-layer perceptron, which is organised as layers of computing elements (called neurons) connected by weighted connections between layers (Fig. 3). An ANN is commonly divided into three or more layers: an input layer, a hidden layer(s), and an output layer. The input layer contains the input nodes (neurons), i.e. the input variables for the network. The output layer contains the desired output of the system and the hidden layer usually contains a series of nodes associated with transfer functions. The feed-forward network is also known as the (error) back-propagation network because of the method used in its training. Training

is a process of adjusting the connection weights in the network so that the network's response best matches the desired response.

In this study, the mapping capability of ANN is used to reconstruct the surface of the performance of the parameter sets found by GA. A commercially developed ANN shell, Neuroshell 2 (Ward Systems Group Inc., 1993), is used and the 3-layered back-propagation algorithm is applied to train the network.

The inputs to ANN consist of all the parameter sets generated from GA and the outputs are the performance of all these parameter sets. 80% of the data set are randomly chosen for training the ANN and the remaining 20% used as a test for overfitting. After the ANN is trained and tested, the performance of all the parameter sets generated via Monte-Carlo sampling (as in GLUE) can be estimated from the trained network. Parameter sets estimated to be behavioural (by the trained ANN) are selected for running with the numerical simulation model. The number of parameter sets selected is denoted as  $N_{sel}$ . The actual performances of the  $N_{sel}$  sets are subsequently compared with the pre-determined threshold value. Those with actual performance below the threshold are removed and the remaining behavioural parameter sets are denoted as  $N_{beh}$ .

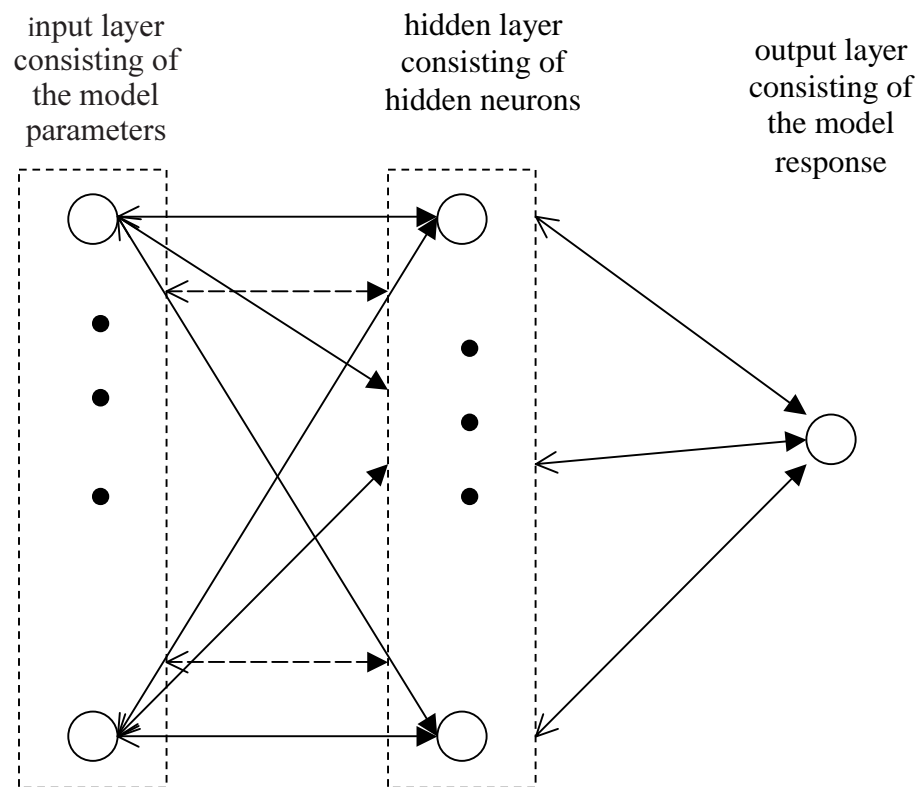


Fig. 3. Schematic diagram of the feed-forward, back propagation architecture of an artificial neural network

## Using GAANN within the GLUE procedure

In this study, GAANN is coupled with the GLUE procedure to reduce the number of simulation runs and increase the sampling efficiency of GLUE. GA is applied independently from the sampling procedure used in GLUE and ANN is used to construct the inverse mapping of the behavioural/non-behavioural space. Application of GAANN in the GLUE procedure (herein known as *mGLUE*) is as follows:

1. Run GA for  $G$  generations with  $P$  individuals in each generation. The fitness of each individual in each generation is stored. The total number of individuals is designed as  $N_{GA}$ .
2. Train the ANN to map the response surface using the fitness of each of the individuals, and the associated parameters;
3. Sample the parameter space using GLUE, resulting in  $N_{MC}$  samples;

4. From the trained ANN, the potential fitness of each of the  $N_{MC}$  parameter sets is estimated. If this is found to be above the threshold the parameter set is retained for analysis with the model; if not, it is discarded. This procedure results in the selection of  $N_{sel}$  parameter sets.
5. Evaluate each of the  $N_{sel}$  parameter sets with the model and compute the ‘actual’ fitness.
6. Remove from the  $N_{sel}$  parameter sets those with ‘actual’ fitness falls below the threshold as in the GLUE procedure described. Likelihood values for the remaining  $N_{beh}$  behavioural parameter sets are assigned as before.

The coupling of this approach with the GLUE procedure is illustrated in the flow chart given in Fig. 4. The success of the integration of the GA depends on the method used to evaluate the potential fitness of each of the  $N_{MC}$  parameter sets. In its most optimal form the GA and subsequent selection procedure will be able to identify the contours

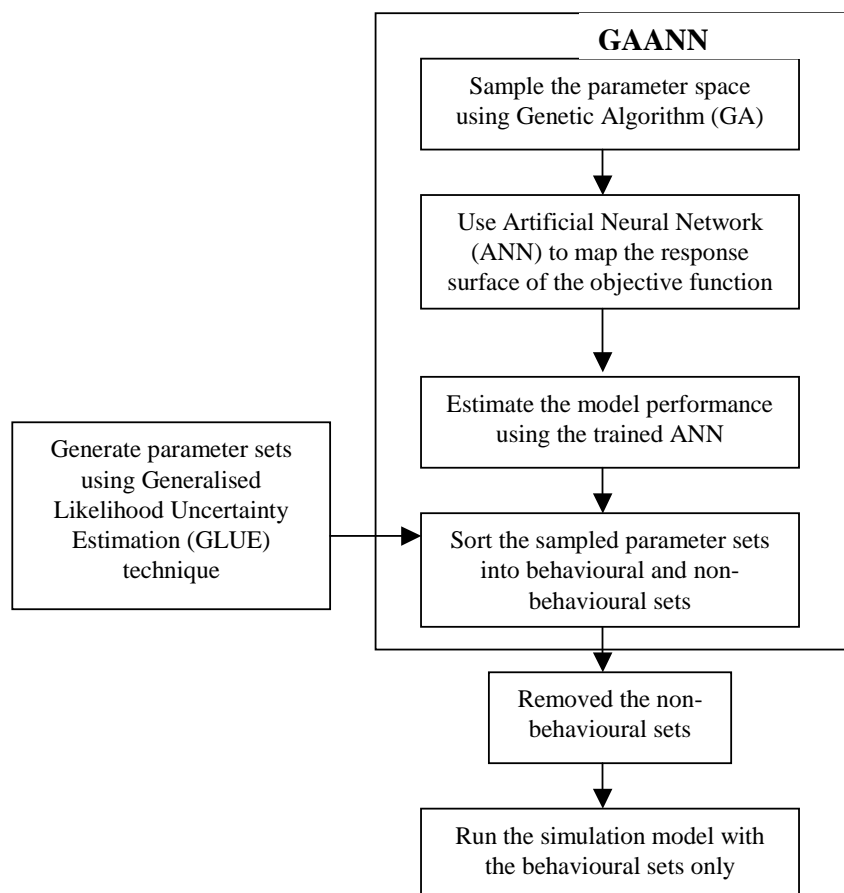


Fig. 4. Flow chart showing *mGLUE* procedure (coupling of *GAANN* procedure in the *GLUE* procedure).



between behavioural and non-behavioural parameter sets such that no runs in step 3 are discarded and  $N_{sel} = N_{beh}$ , while the runs retained for further analysis are the same as those that would have been retained had the normal procedure been carried out.

The performance of the proposed method can be measured using the following indexes:

1. Efficiency of procedure,  $\varepsilon = N_{beh} / N_{tot}$  (3)

2. Effectiveness of GAANN,  $\alpha = N_{beh} / N_{sel}$  (4)

3. Cost of *m*GLUE over GLUE,

$$\gamma = N_{tot}^{mGLUE} / N_{tot}^{GLUE} \quad (5)$$

4. Reliability of *m*GLUE over GLUE,

$$\beta = N_{beh}^{mGLUE} / N_{beh}^{GLUE} \quad (6)$$

## Application I: NASH-CASCADE

The Nash-Cascade, or gamma function model, is a linear model derived from routing rainfall through a cascade of equal linear reservoirs (Shaw, 1988). The instantaneous unit hydrograph (IUH) is given as (Xiong and O'Connor, 2000):

$$U(0, t) = \frac{1}{K\Gamma(n)} \exp\left(-\frac{t}{K}\right) \left(\frac{t}{K}\right)^{n-1} \quad (7)$$

With the Gamma function of  $n$ :

$$\Gamma(n) = \int_0^\infty \exp(-x) x^{n-1} dx \quad (8)$$

The duration  $T$  unit pulse function, or the TUH hydrograph is defined as:

$$se: U(T, t) = [U(1, t) - U(1, t - T)] / T \quad (9)$$

where  $U(1, t)$  is the unit step response:

$$U(1, t) = \frac{1}{K\Gamma(n)} \exp\left(-\frac{\tau}{K}\right) \left(\frac{\tau}{K}\right)^{n-1} \quad (10)$$

and the response to a rainfall event is finally given as the

convolution of step responses, where  $Q_i$  is the outflow at time step  $i$ ,  $R_i$  is the rainfall at timestep  $i$ , and the time interval is  $T$ .

$$Q_i = \sum_{j=1}^m U(T, jT) R_{i-j+1} \quad (11)$$

The gamma function model has two parameters, the linear storage coefficient  $K$  and the number of linear stores  $n$ . Although the second parameter is intuitively taken as an integer value, giving the problem an analytical solution, this need not be the case, and, given an observed runoff and rainfall, the parameters must be estimated by calibration. A gamma model with known parameters  $n = 4$  and  $K = 4$  is applied to an arbitrary rainfall to generate an experimental observed runoff event (Fig. 5).

The GLUE procedure was applied with parameters  $n$  and  $K$  sampled uniformly in the range of 1.0 to 9.0 and the number of Monte Carlo runs ( $N_{MC}$ ) set at 2500. Model performance was evaluated using the Nash-Sutcliffe coefficient, and parameter sets with  $R^2$  below 0.8 were considered non-behavioural. The points in Fig. 1(a) are the randomly generated parameter sets, and contours indicate lines of equal model performance, showing a significant number of parameter sets falling outside the 0.8 contour. Application of the traditional GLUE procedure with  $N_{MC}$  set at 2500 was repeated five times. Table 1 shows that, on average, about 25% of the  $N_{MC}$  runs are found to be behavioural ( $N_{beh}$ ).

In the modified GLUE (*m*GLUE) procedure, 250 GA runs ( $N_{GA}$ ) are used for exploring the parameter space. The population size ( $P$ ) of the GA is 25 and run for a total of 10 generations, giving a total of 250 simulations. The population size need not be too big in this case because the problem is relatively simple for the GA to optimise. The procedure is applied five times to test for consistency using  $N_{MC} = 2500$  and  $N_{GA} = 250$ . Table 1 shows that between 67 and 106 of the 250 runs were found to be behavioural, representing about 34% of the GA runs. The ANN is trained with 80% of these 250 runs and the remaining 20% GA runs are used for verification.

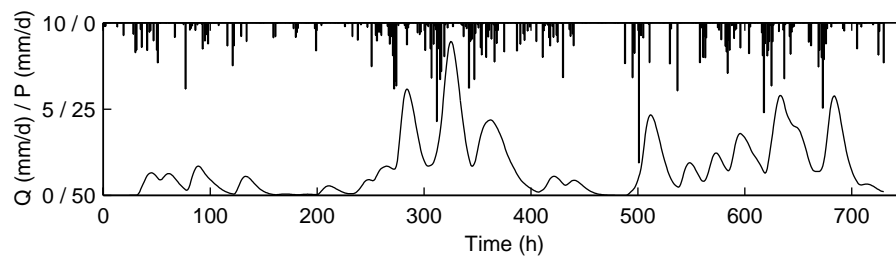


Fig. 5. Rainfall and generated runoff for the Gamma function model with parameters  $n = 4$  and  $K = 4$ .

Table 1. Number of model simulations required in applying GLUE and mGLUE procedure to the Gamma function model

	GLUE procedure	mGLUE procedure
Initial runs required, $N_{MC}$	2500	250
Number of $N_{MC}$ runs that are behavioural	628 (average)	67–106
Number of extra runs required, $N_{sel}$	—	571–723
Total number of runs required, $N_{tot}$	2500	821–973
Number of runs generating behavioural points, $N_{beh}$	628 (average)	502–575

All 2500 randomly sampled parameter sets previously generated in GLUE were fed into the ANN and the corresponding performance in terms of  $R^2$  estimated. Table 1 shows that the ANN selected between 571 and 723 (out of 2500) as behavioural ( $N_{sel}$ ). Out of these selected points, between 501 and 575 points are found to be actually behavioural ( $N_{beh}$ ). This demonstrated that a large portion of the actual model runs generate behavioural points. From the values of the various performance indexes, the mGLUE procedure is a cost effective way to reduce the number of model runs required in GLUE without significantly degrading the reliability of the GLUE procedure.

Fig. 1(b) shows the contours of equal performance for the ANN estimates with the points indicating selected parameter sets. Clearly, the general shape of the objective function response surface, as well as the interaction between parameters, are represented well by the trained network. This indicates that the ANN was able to fit the points that were behavioural.

## Application II: SWMM model

A second application of the procedure to a model with a higher dimensional parameter space is explored. The Storm Water Management Model (SWMM) (Huber and Dickinson, 1988), originally developed by the US Environment Protection Agency (USEPA) is a dynamic rainfall-runoff simulation model, primarily but not exclusively for urban areas, for single-event or long-term (continuous) simulation. Flow routing is performed for surface and sub-surface conveyance and groundwater systems, including the option of fully dynamic hydraulic routing in the Extran block. Non-point source runoff quality and routing may also be simulated, as well as storage, treatment and other best management practices. Readers may consult the USEPA website for more information and free software download. The SWMM Extran runoff block is used in this application and applied to the Upper Bukit Timah catchment in Singapore as shown in Fig. 6. 35% of the 6.11 km<sup>2</sup> catchment

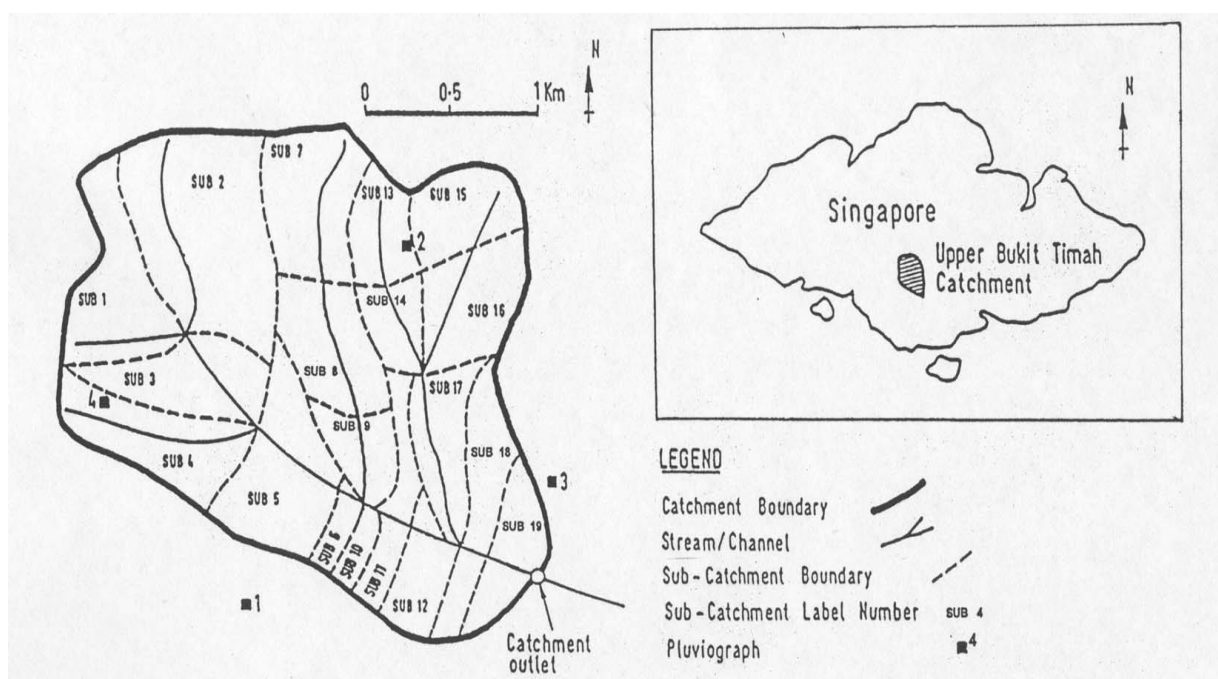


Fig. 6. Upper Bukit Timah catchment in Singapore

is developed, with forests and steep slopes dominating the other 65%. Eight parameters of the model are selected for the study and they are the roughness value for overland flow, depression storage in pervious areas, three Horton infiltration parameters, catchment width, percentage of impervious area, and average catchment slope (Liong *et al.*, 1995). Prior ranges for each parameter are assigned between 50% and 150% of the default values as in Liong *et al.* (1995). The model is applied to the short storm starting March 2nd 1984 (Fig. 7), with model efficiency again being measured through the Nash-Sutcliffe efficiency and behavioural models having efficiency above 0.8.

An initial sensitivity analysis was carried out to determine the number of simulations required by the GLUE procedure where  $N_{mc}$  was varied from 1000 to 100,000. At about 25 000 runs, the estimates for the quantiles of the distribution function were found to become stable with  $N_{beh}$  in the order of 10% of the total randomly generated parameter sets. Five runs using  $N_{mc}=25,000$  were then executed, where the GA and ANN were now used to estimate the performance of the 25,000 runs, and only those with an estimated performance of above 0.8 actually being run.

Using  $N_{GA}$  as 10% of  $N_{MC}$ , ANN gave a good estimate of model performance. The correlation coefficient of the ANN for training and testing were 0.999 and 0.997 respectively. The number of estimated behavioural runs ( $N_{sel}$ ) was remarkably close to 10% of the  $N_{MC}$  runs, with only about 5% waste.

## Discussion of Results

Tables 1 and 2 showed that the number of calls to the hydrological model has been significantly reduced using GA and ANN. Table 3 compares the performance of the *m*GLUE procedure with that of GLUE for the two applications.

The efficiency of the procedure,  $\varepsilon$ , is the actual percentage of useful runs (identified as behavioural) over the total

Table 2. Number of model simulations required in applying GLUE and *m*GLUE procedure to the SWMM model

	GLUE procedure	<i>m</i> GLUE procedure
Initial runs required, $N_{MC}$	25000	2500
Number of extra runs required, $N_{sel}$	—	2425–2503
Total number of runs required, $N_{tot}$	25000	4925–5003
Number of runs generating behavioural points, $N_{beh}$	2586	2306–2354
1% exceedance quantile	16.97	16.90–17.12
2.5 % exceedance quantile	17.16	17.16–17.30
5% exceedance quantile	17.38	17.33–17.48
50% exceedance quantile	19.43	19.34–19.52
95% exceedance quantile	21.65	21.63–21.75
97.5 exceedance quantile	21.95	21.90–22.03
99% exceedance quantile	22.29	22.25–22.36

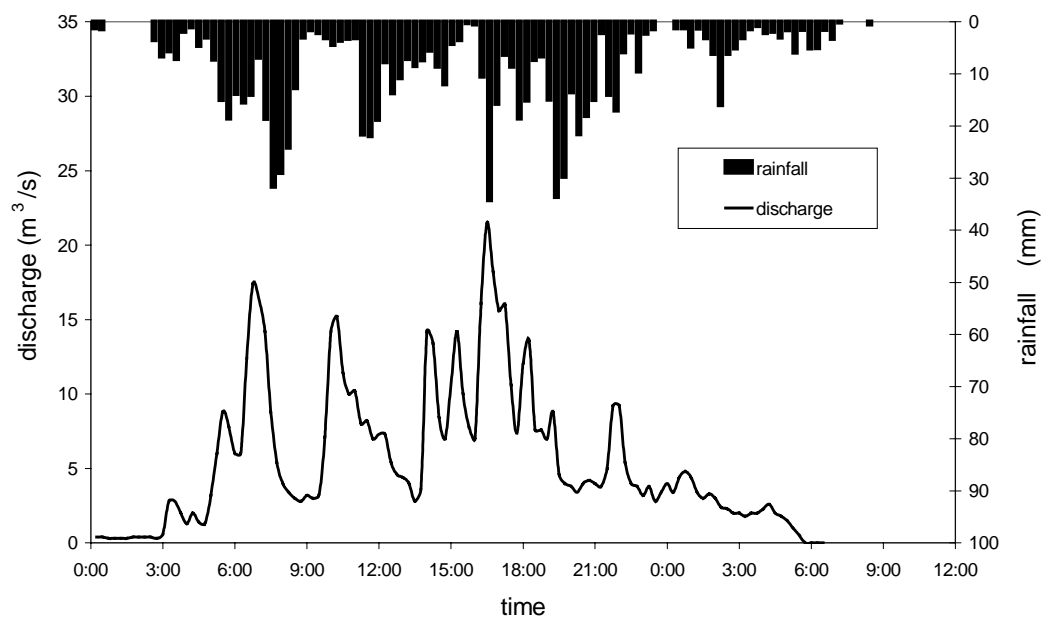


Fig. 7. Hyetograph and hydrograph of storm dated 2<sup>nd</sup> March 1984, Upper Bukit Timah catchment (Singapore).

Table 3. Performance comparison of the GLUE and *m*GLUE procedures

Run	Efficiency $\varepsilon$	Effectiveness of GAANN, $\alpha$	COST of <i>m</i> GLUE over GLUE, $\gamma$	Reliability of <i>m</i> GLUE over GLUE, $\beta$
Gamma model - GLUE	25%	—	—	—
Gamma model - <i>m</i> GLUE	61%	80%	39%	60%
SWMM - GLUE	10%	—	—	—
SWMM - <i>m</i> GLUE	47%	94%	20%	90%

number of simulations required for the procedure. Table 3 shows that the values of  $\varepsilon$  for the Nash-Cascade model is 25% for GLUE and 61% for *m*GLUE; the proposed procedure is, thus, much more efficient in generating behavioural parameter sets than the GLUE procedure. A similar result was obtained for the SWMM model. The efficiencies are 10% and 47% for GLUE and *m*GLUE respectively, due to the complexity and irregularities of the response surface of the SWMM model compared to the results obtained in the Nash-Cascade model discussed earlier.

Although  $\varepsilon$  provides an indication of the efficiency of each procedure, the total number of simulations required in each procedure differs significantly (Tables 1 and 2). Figure 8 shows that, despite this reduction in the number of runs,

the distributions of peak discharges are very similar; the difference between GLUE and *m*GLUE is less than  $0.2 \text{ m}^3 \text{ s}^{-1}$  and this holds even for the tails of the distributions (Table 2). Table 3 shows that the costs,  $\gamma$ , (the ratio of the total number of simulations of each procedure) of GLUE over *m*GLUE for the Gamma function and SWMM models are 39% and 20% respectively. Particularly for more complex response surfaces requiring a greater number of GLUE (Monte Carlo) simulations, the number of actual runs can be reduced significantly by the GAANN procedure; in application 2, the total number of runs is reduced from 25 000 to 5000 ( $\gamma = 20\%$ ).

Having demonstrated that significant savings can be obtained using the *m*GLUE procedure, the reliability and effectiveness of the proposed procedure should now be

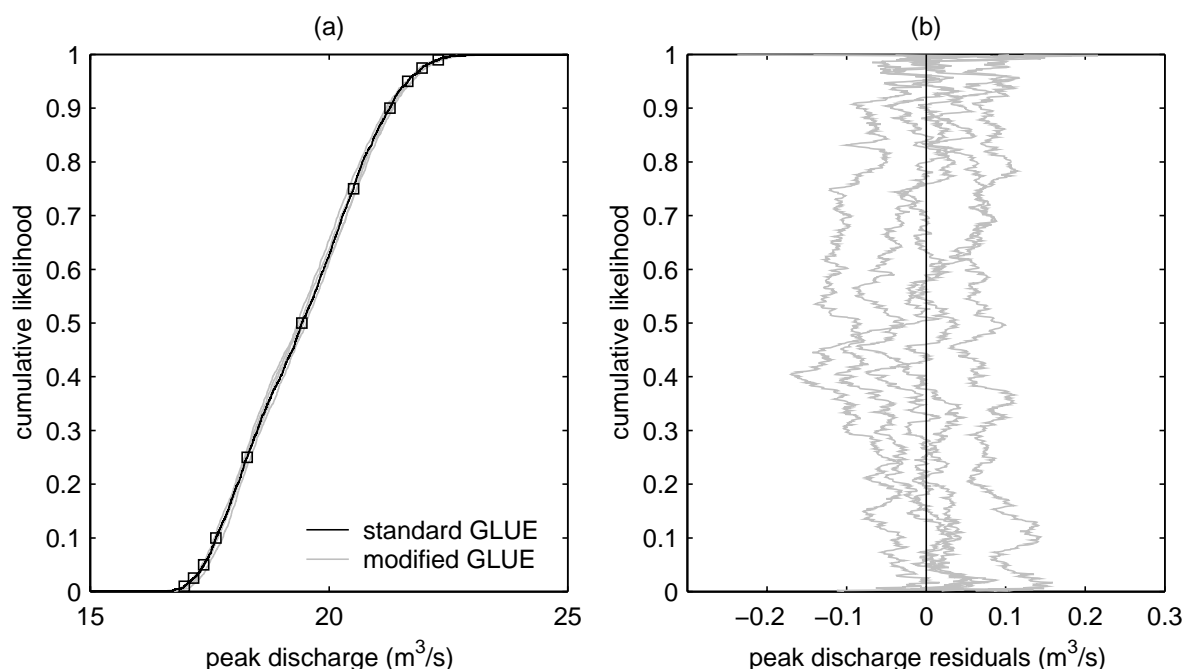


Fig. 8. Cumulative likelihood distribution (cdf) and distribution of residual for the peak discharge of the storm dated 2<sup>nd</sup> March 1984, Upper Bukit Timah catchment (Singapore).

questioned. The reliability of procedure can be calculated by comparing the number of behavioural solution sets selected by both methods. It should be obvious that the number of behavioural sets from GLUE is always larger than *m*GLUE since it uses an approximation to select the behavioural sets from the Monte Carlo generated sets. From Table 3, the reliability of *m*GLUE for the Gamma and SWMM models are 60% and 90% respectively. The higher reliability of the *m*GLUE-SWMM results may be attributed to the effectiveness of GA in obtaining good (behavioural) solution sets using a smaller number of simulations than the Monte Carlo procedure and the high success rate of using ANN to map the response surface of the model simulations. The effectiveness of GAANN for the Gamma and SWMM models are 80% and 94% respectively (Table 3).

With regard to the concern of additional runtime, the proposed procedure was implemented using a collection of MATLAB scripts, including the standard GA toolbox. All steps are currently automatic except for the ANN (due to present lack of availability of the MATLAB-ANN toolbox) but this could be easily automated once the toolbox is purchased. The overheads incurred in GAANN are minimal when compared to the actual runtime of the simulation model.

Finally, there is a slight drawback with the proposed procedure. The efficiencies of *m*GLUE, although higher than GLUE, were lower than expected. This showed that about half the total runs are not being used to determine the posterior distribution functions. The reason was that the runs carried out during the exploration of the objective function response surface (generated by GA) were discarded from the analysis and considered redundant. Although a good number of these runs are behavioural, these cannot be incorporated in the final analysis, as the sampling procedure used in GA is essentially different from the Monte-Carlo sampler used in GLUE. The authors are currently exploring the possibility of using simulated annealing instead of GA as the basis of exploring the response surface. (Note: a simulated annealing type algorithm has been applied successfully lately to uncertainty analysis (Vrugt *et al.*, 2003)).

## Conclusions

A novel method, GAANN, reduces the number of model runs required in the Monte Carlo based sampling of the GLUE procedure. Using a genetic algorithm (GA), the shape of the response surface of the objective function is explored and, then, the structure of the objective function response surface is identified using the universal mapping function, artificial neural network (ANN). This allows for the

estimation of the potential fitness of parameter sets without actually running the hydrological model. Based on whether this estimated potential fitness exceeds a threshold used to distinguish between behavioural and non-behavioural models, the parameter set is either retained for further evaluation with the model, or discarded.

Two applications of GAANN are demonstrated, the Nash-cascade model with known parameters and the catchment model, SWMM, applied to a catchment in Singapore with observations of rainfall and runoff. GAANN reduces significantly the number of runs required to estimate the likelihoods of parameter sets compared to the GLUE procedure. This reduces, significantly, the computational effort involved in investigating model parameter uncertainty. Integration of this method with the GLUE procedure extends the scope of application of uncertainty estimation, without diminishing the robustness of the GLUE procedure or significantly complicating its application.

## Acknowledgements

The authors thank the reviewers, Dr. Hoshin Gupta and Dr. Stefan Uhlenbrook, and a third anonymous reviewer for their valuable comments and suggestions, many which have been taken into account in the amended version of the paper.

## References

- Aronica, G., Hankin, B. and Beven, K., 1998. Uncertainty and equifinality in calibrating distributed roughness coefficients in a flood propagation model with limited data. *Adv. Water Resour.*, **22**, 349–365.
- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000. Artificial neural networks in hydrology. II: Hydrologic applications. *J. Hydrolog. Eng.-ASCE*, **5**, 124–137.
- Belanche, L., Valdes, J.J., Comas, J., Roda, I.R. and Poch, M., 2000. Prediction of the bulking phenomenon in wastewater treatment plants. *Artif. Intell. Eng.*, **14**, 307–317.
- Beven, K.J., 1989. Changing ideas in hydrology. The case of physically based models. *J. Hydrol.*, **105**, 157–172.
- Beven, K., 1993. Prophecy, reality and uncertainty in distributed hydrological modeling. *Adv. Water Resour.*, **16**, 41–51.
- Beven, K. and Binley, A., 1992. The future of distributed models: Model calibration and uncertainty prediction. *Hydrol. Process.*, **6**, 279–298.
- Campbell, E.P., Fox, D.R. and Bates, B.C., 1999. A Bayesian approach to parameter estimation and pooling in nonlinear flood event models. *Water Resour. Res.*, **35**, 211–220.
- Gan, T.Y., Dlamini, E.M. and Biftu, G.F., 1997. Effects of model complexity and structure, data quality, and objective functions on hydrological modeling. *J. Hydrol.*, **192**, 81–103.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison and Wesley, Reading, Mass., USA.
- Hecht-Nielsen, R., 1989. Theory of the back-propagation neural network. *Proc. Int. Joint Conf. Neural Networks*. San Diego, USA. 593–608.



- Hornik, K., Stinchcombe, M. and White, H., 1989. Multi-layer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–66.
- Huber, W.C. and Dickinson, R.E., 1988. *Storm Water Management Model User's Manual, Version 4*, EPA/600/3-88/001a, Environmental Protection Agency, Athens, GA, USA. 595 pp.
- Jakeman, A.J. and Hornberger, G.M., 1993. How much complexity is warranted in a rainfall-runoff model? *Water Resour. Res.*, **28**, 2637–2649.
- Karunaithi, N., Grenney, W.J., Whitley, D. and Bovee, K., 1994. Neural Networks for river flow prediction. *J. Computing Civil Eng.*, **8**, 201–219.
- Kuczera, G., 1997. Efficient subspace probabilistic parameter organization for catchment models. *Water Resour. Res.*, **33**, 177–185.
- Kuczera, G. and Parent, E., 1998. Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The Metropolis algorithm. *J. Hydrol.*, **211**, 69–85.
- Lamb, R., Beven, K. and Myrabo, S., 1998. Use of spatially distributed water table observations to constrain uncertainty in a rainfall-runoff model. *Adv. Water Resour.*, **22**, 305–317.
- Lees, M. and Wagener, T., 2000. *Monte Carlo Analysis Tool (MCAT) User Manual*. Civil and Environ. Eng. Dept., Imperial College of Science, Technology and Medicine, London, UK.
- Liong, S.Y., Chan, W.T. and ShreeRam, J., 1995. Peak-flow forecasting with genetic algorithms and SWMM. *J. Hydraulic Engg., ASCE*, **121**, 613–617.
- Liong, S.Y., Khu, S.T. and Chan W.T., 2001. Derivation of Pareto front with genetic algorithm and neural networks. *J. Hydrol. Eng.*, **6**, 52–61.
- Melching, C.S., 1995. Reliability Estimation. *Computer Models of Watershed Hydrology*, V.P. Singh (Ed.), Water Resources Publications. 69–118.
- Montesinos, P. and Beven, K.J., 1999. *Application of a genetic algorithm as sampler selector within the GLUE approach*. Poster Presentation – European Geophysical Society Assembly, The Hague, The Netherlands.
- Nash, J.E. and Sutcliffe, J.V., 1970. River flow forecasting through conceptual models, Part I: a discussion of principles. *J. Hydrol.*, **10**, 282–290.
- van der Perk, M., 1997. Effect of model structure on the accuracy and uncertainty of results from water quality models. *Hydrol. Process.*, **11**, 227–239.
- Recknagel, F., French, M., Harkonen, P. and Yabunaka, K-I., 1997. Artificial neural network approach for modeling and prediction of algal blooms. *Ecol. Model.*, **96**, 11–28.
- Romanowicz, R., Beven, K. and Tawn, J., 1996. Bayesian Calibration of flood inundation Models. In: *Floodplain Processes*, M. Anderson, D.E. Walling and P.D. Bates, (Eds.). Wiley, Chichester, UK. 333–360.
- Shaw, E.M., 1988. *Hydrology in Practice*. Van Nostrand Reinhold Ltd., London, UK.
- Sorooshian, S. and Gupta, V. K., 1983. Automatic calibration of conceptual rainfall-runoff models: the question of parameter observability and uniqueness. *Water Resour. Res.*, **19**, 260–268.
- Sorooshian, S. and Gupta, V.K., 1995. Model Calibration, *Computer Models of Watershed Hydrology*, V.P. Singh (Ed.), Water Resources Publications, USA. 23–68.
- Uhlenbrook, S., Seibert, J., Leibendgut, C. and Rodhe, A., 1999. Prediction uncertainty of conceptual rainfall-runoff models caused by problems in identifying model parameters and structure. *Hydrolog. Sci. J.*, **4**, 779–797.
- Ulsch, A. and Roske, F., 2002. Self-organizing feature maps predicting sea levels. *Inform. Sciences*, **144**, 91–125.
- Vrugt, J.A., Gupta, H.V., Boughten, W. and Sorooshian, S., 2003. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.*, **39**, 1201, doi:10.1029/2002WR001642
- Wang, Q.J., 1991. The Genetic Algorithm and its application to calibrating conceptual rainfall-runoff models. *Water Resour. Res.*, **27**, 2467–2471.
- Ward Systems Group Inc., 1993. *Neuroshell 2 User's Manual*, 2nd edition.
- Werner, M. and Khu, S.T., 2001. Monte-Carlo-based uncertainty analysis framework using a multi-modal genetic algorithm. *Proc. 29th IAHR Congress: Theme A*, Beijing, China. 425–431.
- Xiong, L. and O'Connor, K.M., 2000. Analysis of the response surface of the objective function by the optimum parameter curve: how good can the optimum parameter values be? *J. Hydrol.*, **234**, 187–207.