  Journal of Hydroinformatics | 09.4 | 2007

Use of a supercomputer to advance parameter optimisation using genetic algorithms

Achela K. Fernando and A. W. Jayawardena

ABSTRACT

Parameter optimisation is a significant but time-consuming process that is inherent in conceptual hydrological models representing rainfall–runoff processes. This study presents two modifications to achieve optimised results for a Tank Model in less computational time. Firstly, a modified genetic algorithm (GA) is developed to enhance the fitness of the population consisting of possible solutions in each generation. Then the parallel processing capabilities of an IBM 9076 SP2 computer are used to expedite implementation of the GA. A comparison of processing time between a serial IBM RS/6000 390 computer and an IBM 9076 SP2 supercomputer reveals that the latter can be up to 8 times faster. The effectiveness of the modified GA is tested with two Tank Models for a hypothetical catchment and a real catchment. The former showed that the parallel GA reaches a lower overall error in reduced time. The overall RMSE, expressed as a percentage of actual mean flow rate, improves from 31.8% in a serial processing computer to 29.5% on the SP2 supercomputer. The case of the real catchment – Shek-Pi-Tau Catchment in Hong Kong – reveals that the supercomputer enhances the swiftness of the GA and achieves its objective within a couple of hours.

Key words | genetic algorithm, parallel processing computers, parameter optimisation, rainfall–runoff process, tank model

Achela K. Fernando (corresponding author)
The School of Built Environment,
Unitec New Zealand (Te Whare Wananga Owairaka),
Mt. Albert, Auckland,
New Zealand
Tel: +64 981 54321 X 7036
E-mail: *afernando@unitec.ac.nz*

A. W. Jayawardena
Department of Civil & Structural Engineering,
The University of Hong Kong,
Pok Fu Lam Road, Hong Kong,
Hong Kong (SAR)
China

INTRODUCTION

Of the various conceptual models to represent the rainfall–runoff process, the Tank Model, first introduced by Sugawara *et al.* in 1984 (Sugawara *et al.* 1984), is one of the earliest. It is a simple representation of the catchment surface and the underlying system of soil strata by a series of tanks that store the rainfall and subsequently discharge it at a rate proportional to their capacities. Since then, several applications based on the Tank Model and their combinations with other conceptual models have been completed by various researchers (e.g. Jayawardena 1998; Elhassan *et al.* 2001). In a comparative analysis of several conceptual rainfall–runoff models, Franchini & Pacciani (1991) mentioned that the Tank Model, despite its abstract nature of representing the runoff formation without any physical correspondence to the actual phenomena, produces equally good or better results with relative ease compared to other models.

Calibration of the parameters is the main challenge in the development of hydrological models representing rainfall runoff. Use of automatic calibration techniques which enables the hydrologist to rely less on subjective judgement have been reported (Sorooshian & Dracup 1980; James & Burges 1982; Sorooshian & Gupta 1983; Hendrickson *et al.* 1988; Franchini 1996). For the Tank Model, rather than calibration or numerical definition of the parameters characterising the equations which describe a certain phenomenon, it seems to be more appropriate to speak of fine-tuning a mechanism with its own internal structure which emulates the behaviour of a watershed in runoff formation (Franchini & Pacciani 1991). Investigations into

doi: 10.2166/hydro.2007.006

procedures for optimisation of Tank Model parameters have been carried out by Setiawan *et al.* (2003) and Tanakamaru (1995). However, these optimisations are largely limited to the use of long continuous daily rainfall and runoff. Optimising parameters for discrete, shorter events can be different and more time-consuming as there could be numerous permutations and combinations for the values of the model parameters satisfying the objective function, which generally is the model output error.

In this study two significant modifications are made to attempt to achieve optimisation in a reduced time. Firstly, an established genetic algorithm (GA) is modified to improve overall performance and, secondly, the time-consuming computations of the GA, usually implemented serially, are parallelised using a supercomputer with the capacity to perform parallel computations.

Other global optimisation algorithms such as "controlled random search2" (CRS2), "adaptive cluster covering with local search" (ACCOL) and "multiple downhill simplex" (M-SIMPLEX), some of which are much faster and require fewer evaluations of the objective function than GA, were not considered, as the focus of this study is to highlight the benefits of using a supercomputer to parallelise the GA.

BACKGROUND OF TANK MODEL

Figure 1 shows the configuration of a simple Tank Model with 3 tanks (A, B and C) in series. The topmost Tank A receives the rainfall. The surface runoff, sub-surface runoff and the base flow are represented by the lateral discharge from tanks A, B and C, respectively. Tank A has three side outlets to cater for rapid responses for flood situations. Infiltrations are represented by the downward flow from each of the tanks. The total runoff undergoes a channel routing that is represented by a fourth tank D with one bottom and one side outlet. Discharges are proportional to the storage capacity or the available water head in each of the tanks, and the discharge coefficient of the outlets. The variable parameters are the heights of the side outlets, HA_1, HA_2, HA_3, HB, HC and HD and their discharge coefficients A_0, A_1, A_2, A_3, B_0, B_1, C_0, C_1, D_0 and D_1 and the initial storage in each tank $XAIN$, $XBIN$ and $XCIN$, bringing the total number of parameters to be optimised to 19. It has been reported that for modelling runoff in steep Hong Kong catchments, the parameters HA_3, A_3, $XAIN$, $XBIN$ and $XCIN$ can be irrelevant (Jayawardena 1998).

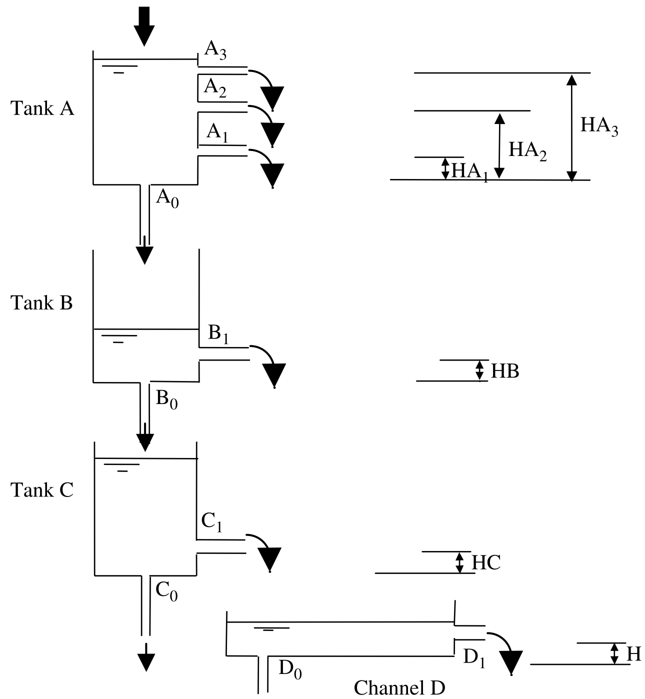

Figure 1 | Representation of tank model.

GENETIC ALGORITHM AND REPRESENTATION OF TANK MODEL PARAMETERS

John Holland (Holland 1975) is the founder of the field of Genetic Algorithms (GAs) which was inspired by the natural evolution of biological species. The adaptive nature of the GA lends itself to be applied to problems that require progressive modification such as parameter optimisation.

The GAs operate on a coding of the tank model parameters, rather than on the parameters themselves. Each parameter is encoded into a string of finite length made up of binary numbers. These strings are then concatenated to form one long string that is regarded as one individual or a structure; several such individuals compose a population. The "fitness" of an individual is equivalent to the value of the objective function determined collectively by the structure (made up of one possible set of model parameters). Genetic rules are then applied to the whole population, with a selection procedure that has a guided randomness leading the structures in the subsequent iterations increasingly towards the optimum. The basis for this is Holland's Schema theory (Holland 1993). details of which can be found in Goldberg (1989).

Each iteration of the algorithm is expected to produce a population of structures superior in fitness to the former. The fitness of the population as well as the "best so far" structure that corresponds to the minimum error (the discrepancy between the actual and model-predicted run-off) is recorded over the iterations.

Wang (1991) applied a GA to calibrate a conceptual rainfall–runoff model. In Wang's work (Wang 1991), and in this study, a constant string length has been used for all the parameters.

The GA proposed by Wang (1991) begins by arbitrarily generating an initial population of m sets of strings representing m possible parameter sets. The objective function is computed for each set. Each set is then given a ranking based on its fitness such that the fittest set assumes the highest rank and the most unfit the lowest. Each set is assigned a probability for being chosen for the reproduction process. For a population of m, the average probability is $1/m$. Wang assigned the value of C times the average probability, C/m, to the fittest set where $C > 1$. He suggested a probability distribution for the jth individual, p_j, in the form of

$$p_j = p_1 + \frac{p_m - p_1}{m-1}(j-1)$$

where p_m is the highest probability corresponding to the highest ranking set and p_1 corresponds to the lowest ranking one. The summation of all probability values should be equal to unity, i.e. $\sum_{j=1}^{m} p_j = 1$, and therefore the probability of the lowest ranking individual is $(2-C)/m$. To ensure non-negativity for the probabilities $C \geq 2$ and Wang assigned value of 2 to C. The genetic operations are then carried out as follows.

(i) Two distinct sets, SET_1 and SET_2, are selected from the population of m at random according to the probability distribution p_j, $j = 1,2, \ldots, m$. Two bit positions k_1 and k_2 are selected at random giving all the bit positions the same chance. If $k_1 > k_2$ they are interchanged.

(ii) A new set is formed by taking the values of the bits from k_1 to $k_2 - 1$ of the SET_1 coding and the values of the bits from k_2 to the end and from 1 to $k_1 - 1$ from the SET_2 coding. Occasionally a bit value of the newly formed set is changed from 0 to 1 or vice versa.

Steps (i) and (ii) are repeated until m new sets of the next generation are formed. The whole process is repeated until a prescribed number of generations have been reproduced. The best set so far is recorded during the entire process.

METHODOLOGY

In this study the above algorithm is modified with the aim of preventing the most unfit sets from taking any part in the regeneration. This is achieved by changing the probability distribution such that the probability of the lowest ranking one-eighth of the population being involved in the regeneration process be zero as can be expressed below:

$$p_j = 0 \quad \text{for } j = 1, 2, \ldots, (m/8)$$

$$p_j = p_{(m/8)+1} + \frac{p_m - p_{(m/8)+1}}{[m-(m/8)-1]}[j-(m/8)-1]$$

$$\text{for } j = (m/8)+1,\ (m/8)+2,\ \ldots, m$$

where $p_m = C/[m-(m/8)]$, $p_{[(m/4)+1]} = (2-C)/[m-(m/8)]$.

Figure 2 shows the two probability distributions in graphical form. It may appear that the diversity of the group

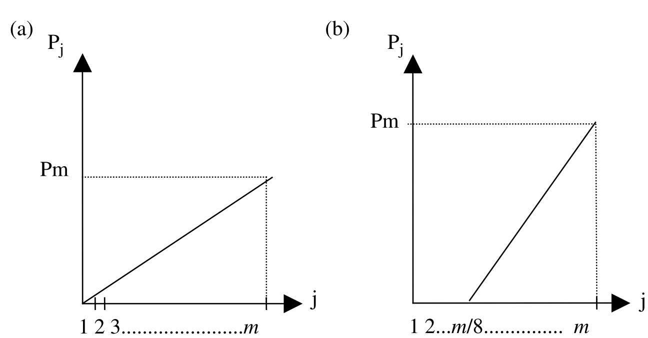

Figure 2 | (a) Probability distribution (Wang 1991). (b) Proposed probability distribution.

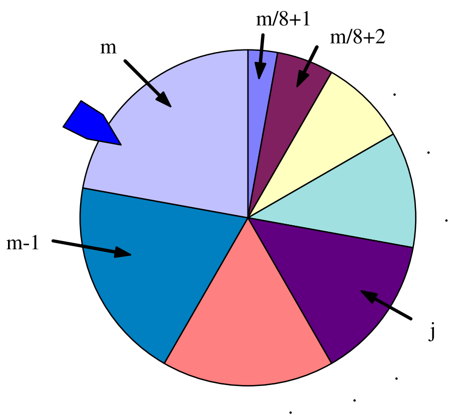

Figure 3 | Probability of the reproducing structures.

is restrained by this. If the value of m is large enough, however, it is expected that the proposed distribution would enhance the fitness of the group. Moreover, it appears to agree with reproduction patterns of nature where some of the most unfit individuals cease to exist before any reproduction is possible. The proportion of the total group thus restrained from regenerating, however, is not known. One-eighth was chosen in this study. Other proportions could also be tested to gauge its sensitivity to the overall performance of the algorithm.

As a result of this proposed probability distribution, one-eighth of the group having the lowest ranking becomes extinct following each iteration. The rest of the group take part in the regeneration, forming a new group of size m. Figure 3 shows a disk divided according to the probabilities of the reproducing sets. It illustrates that the higher ranking structures have higher chances of being pointed at (and thus chosen for regeneration) by the pointer when the disk stops after a random rotation. A total of $(m \times 2)$ such rotations will determine which pairs of structures undergo genetic operations to produce the new group.

Natural genetic evolution is slow and takes millions of years to bring about a significant change in a species. Similarly, its artificial counterpart, the genetic algorithm, requires a large number of iterations to complete an optimisation it is required to perform. The time required for these lengthy computations is extremely long compared to other conventional methods. To be able to compete with existing techniques of optimisation and to be an efficient method in its own right, the implementation of GA must somehow be accelerated. The SP2 supercomputer installed in the University of Hong Kong offered an ideal environment to test the viability of the GA. With its swiftness and parallel computing facility, the SP2 supercomputer provides the GA programmer with the flexibility to parallelise those computations in the GA that need not necessarily be performed in a serial manner.

The major steps of a sequence in an evolution and its counterparts in the context of GA as applied to optimising the parameters of a Tank Model are shown in Figure 4. One complete iteration involves steps II to IV.

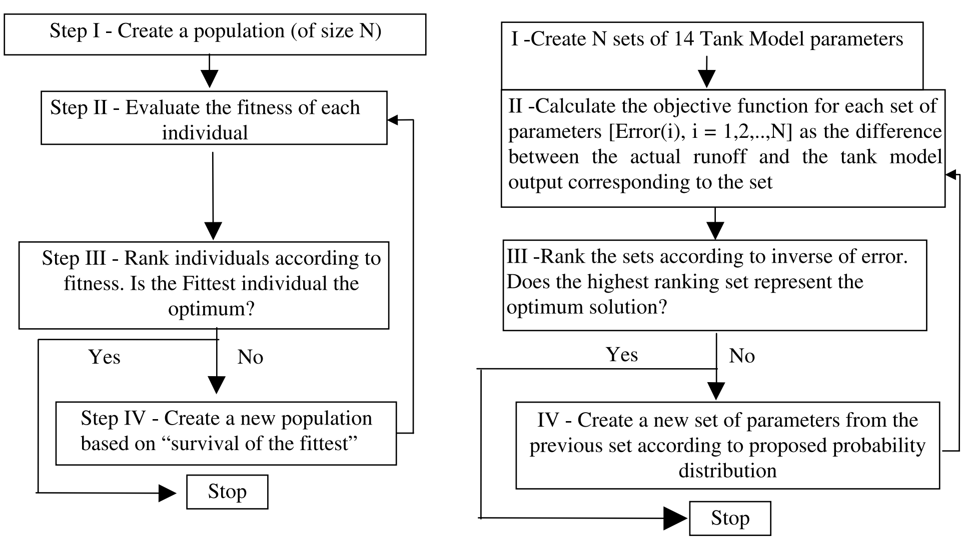

Figure 4 | Major steps in the genetic computation procedure: (a) in general genetic terms, (b) as applied to Tank Model parameter optimisation.

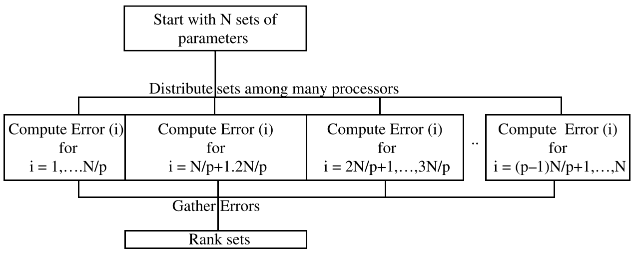

Figure 5 | Parallel computation of the objective function (error) with *p* processors.

In the conventional serial computer all the steps will be carried out in sequence. However, it can be observed that Step II in the above sequence can be implemented in a parallel manner as the objective function (or the error) corresponding to one set of parameters is independent of that for the other sets. Considering the large number of sets (of the order of hundreds), and the lengthy and repetitive nature of computing the error (based on the Tank Model output and the "actual" runoff), a significant time-saving can be expected if this step or group of such steps can be carried out simultaneously, rather than sequentially.

Ideally, the objective function for all the N sets of parameters would be computed simultaneously by N numbers of processors. Distributing the task among p ($p \ll N$) processors would still be much faster than performing it serially. This is what can be accomplished in the SP2 supercomputer by distributing the task of computing the objective function among many processors, as shown in Figure 5.

IBM RS/6000 390 computer

An IBM RS/6000 390 computer was used in this study for the serial implementation of the GA. This has 128 MB memory with 8 GB disk storage space. The choice of this machine was based on the fact that it allowed the user to access the IMSL (International Mathematical and Scientific Library) subroutines which form an essential part of the GA code.

IBM 9076 SP2 supercomputer

This computer, installed in Hong Kong University, has 48 IBM P2SC RISC processors, each with a 160 MHz CPU, 2 GB local disk storage and capable of performing 640 MFLOPS. The processors are connected by high-performance switches. The communications between the processors follow a message passing (MP) model. The message passing model is defined as a set of processors having only local memory, communicating by sending and receiving messages. The transfer of data between the processors requires cooperative operations to be performed by each processor. The MP model has full parallel functionality. MPI (Message Passing Interface) is the *de facto* standard MP library.

The GA code written for the IBM RS/6000 390 computer had to be altered to include the MPI commands to be run in the IBM 9076 SP2 supercomputer. Care was taken to use the most relevant MPI commands that were necessary for the error-free, efficient communication among the processors.

RESULTS AND DISCUSSION

To test the effectiveness of GA in finding suitable parameters for the Tank Model and also to gauge the improvement achieved by using the SP2 supercomputer to parallelise the GA, two catchments, one hypothetical and one real, were considered. The actual values of the Tank Model parameters were set *a priori* for the hypothetical catchment, while for the real one they were unknown.

Hypothetical catchment

A set of parameters was assumed for a hypothetical catchment. Four real rainfall events recorded in Hong Kong were used as input to simulate the events in the Tank Model and the "actual" runoff of this hypothetical catchment was computed. The assumed parameters and the catchment characteristics are shown in the first three columns of Table 1.

A string length of 16 was used to encode each parameter. Assumptions were made regarding the ranges within which the parameters were expected to lie. They are shown in the last two columns of Table 1. Wang (1991) used a population size of 100. In this study $m = 96$, a figure divisible by 8, was used (the reason for this being that the number of processors in the SP2 supercomputer used for parallelisation was 8 and the proportion of the population to become extinct after each iteration was one-eighth). The 12 ($= 96/8$) lowest ranking individuals were excluded

Table 1 | Details of assumed parameters in the hypothetical catchment

Parameter number (i)	Parameter	Assumed value (cm)	Minimum value of the parameter (x_i) (cm)	Maximum value of the parameter (y_i) (cm)
1	HA_1	5	0	7
2	HA_2	10	7	15
3	HA_3	15	15	15
4	A_0	0	0	1
5	A_1	0.2	0	1
6	A_2	0.3	0	1
7	A_3	0	0	0
8	HB	5	0	10
9	B_0	0.4	0	0.5
10	B_1	0.2	0	0.5
11	HC	2	0	40
12	C_0	0.1	0	0.5
13	C_1	0.2	0	0.5
14	HD	15	0	30
15	D_0	0.01	0	0.5
16	D_1	0.5	0	0.5
17	XAIN	0	0	0
18	XBIN	0	0	0
19	XCIN	0	0	0

from each generation and the remaining 84 were chosen according to the probability distribution shown in Figure 2(b) to reproduce a new population of 96. The objective function used was the total squared error between the actual and computed runoff for all four events. A mutation rate of 0.001 was applied.

The parameter optimisation was carried out on three platforms, namely, serial conventional, serial SP2, and parallel SP2 platforms. On the SP2 platform, 4 and 8 parallel processors, respectively, were tested. In this experiment, which involved four rainfall-runoff events, Events 1 and 3 were used for parameter optimization.

Table 2 summarises the time required for the implementation of the GAs. Several runs were carried out and the time refers to the average time. It shows that, compared to the conventional serial implementation, the parallel implementations with 4 and 8 processors are 27 and 53 times faster, respectively.

In terms of optimization results, Figures 6(a, b) compare and contrast the final values of the Tank Model parameters corresponding to the lowest value of the objective function from serial computation with SP2 parallel computations with 8 processors.

Figure 7 shows the comparison between the actual and model-predicted hydrographs for the four events for the above parameter values. By visual comparison, the actual and model-predicted hydrographs closely follow each other. However, a quantitative measure of the discrepancy, expressed in terms of RMSE (Root Mean Squared Error) expressed as a percentage of the actual mean and tabulated below (Table 3), shows that the performance of each Tank Model calibrated in the various methods varies.

A careful observation shows that parallel GA produces the lower error overall in a shorter time. This is predictable given that the parallel GA performs a higher number of iterations, albeit in shorter time. Although the errors appear small, the parameters obtained seem somewhat different from the "actual" values. One reason that can be given for this observation is that the sensitivity of these parameters to the outcome of the Tank Model may not be so significant that a wide range of values can result in an equal or very similar output. Another reason may be that the inter-relationships among the parameters, if there are any, have not been

Table 2 | Summary of GA implementation time

Feature	Serial	SP2 Serial	SP2 Parallel	SP2 Parallel
String length	16	16	16	16
Iterations	2000	2000	5000	5000
No. of processors	–	–	4	8
Time required (h:min)	13:10	03:50	02:23	01:14

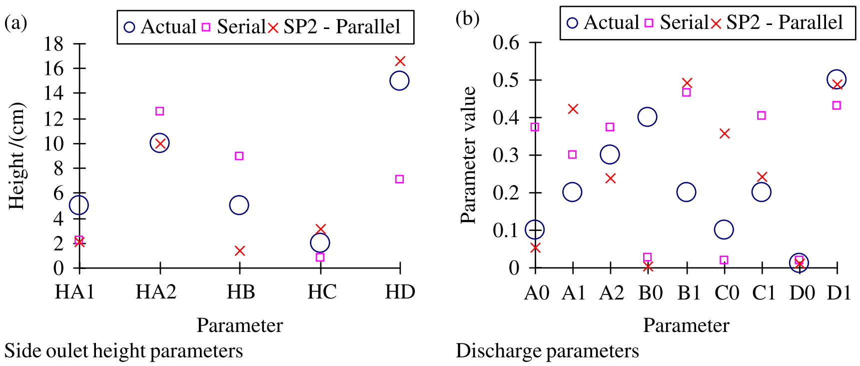

Figure 6 | Actual and GA-optimised values of the Tank Model parameters.

included as part of the objectives to be met and that allows more freedom for the parameters to "wander" from their actual values-to-be. It will be useful to incorporate such knowledge of any interdependence of the parameters, if such knowledge exists, into the GA optimisation procedure.

It is also possible that the discrepancies in the input data, such as noise, erroneous records and spatial variations in the rainfall, can affect the accuracy and execution times of the algorithms and their implementations. However, in this hypothetical case, there is no room for such noise.

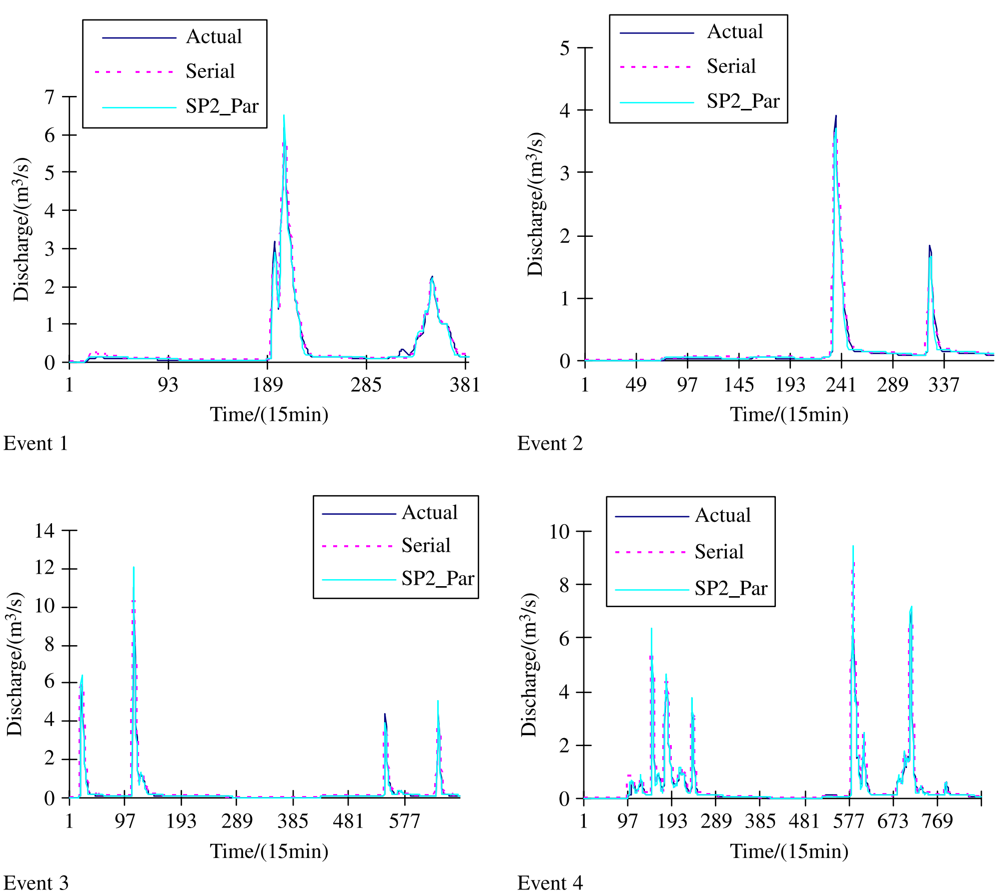

Figure 7 | Actual and calibrated Tank Model outputs using serial and parallel GA.

Table 3 | Root mean squared error expressed as a percentage of the actual mean

	Serial	SP2 – Parallel
Event 1	17.4	16.9
Event 2	41.3	33.1
Event 3	40.8	41.6
Event 4	21.5	21.9
Overall	31.8	29.5

The real case to be presented later suffers from such effects as a consequence of inaccurate data.

The observations reveal that minimising the objective function using a GA can be done effectively. Implementing a parallel GA, instead of a serial one, enables results to be obtained more swiftly i.e. within a reasonable time. The type of GA used here with its sole objective to minimise the error between the actual and computed runoff appears to be ideal for problems where the actual values of the parameters are not so significant, which is the case in the Tank Model.

Next, two SP2 parallel runs were completed with Wang's probability distribution function to compare with those from the proposed distribution. The cumulative objective function for the population at the end of each iteration, using all four events, was plotted and a linear trend line fitted to the values. The equations of the fitted lines are as shown in Table 4 where y is the sum of the objective functions (cumulative error) and x is the number of iterations. As indicated by the slope of the lines, the proposed distribution gives marginally steeper negative slopes. While this experiment alone is not conclusive, this is indicative of marginally improved fitness of the population as a whole during evolution.

Table 4 | The equations of the linear trend lines for population fitness

Simulation	Equation
Wang – Run 1	$y = -0.0059x + 54093$
Wang – Run 2	$y = -0.0974x + 57521$
Proposed – Run 1	$y = -0.0481x + 54196$
Proposed – Run 2	$y = -0.1091x + 56470$

Table 5 | Parameter ranges for the Tank Model for Shek Pi Tau (SPT) Catchment

Parameter number (i)	Parameter	Minimum value of the parameter (x_i) (cm)	Maximum value of the parameter (y_i) (cm)
1	HA_1	0	20
2	HA_2	20	40
3	HA_3	40	80
4	A_0	0	1
5	A_1	0	1
6	A_2	0	1
7	A_3	0	1
8	HB	0	40
9	B_0	0	1
10	B_1	0	1
11	HC	0	40
12	C_0	0	1
13	C_1	0	1
14	HD	0	40
15	D_0	0	1
16	D_1	0	1

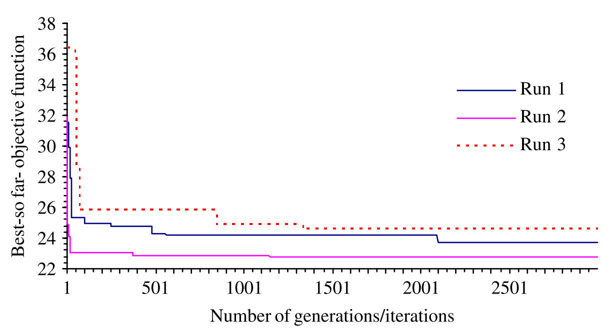

Figure 8 | The variation of the best-so-far objective function.

Table 6 | Optimum values of the parameters of the tank model obtained by the proposed GA

Tank model component	Discharge coefficient of the bottom outlet	Height of lateral outlets (cm)	Discharge coefficients of the lateral outlets
Tank A	$A_0 = 0.9338$	$HA_1 = 17.45$	$A_1 = 0.5832$
		$HA_2 = 39.82$	$A_2 = 0.9734$
		$HA_3 = 63.23$	$A_3 = 0.8247$
Tank B	$B_0 = 0.0217$	$HB = 0.4$	$B_1 = 0.0258$
Tank C	$C_0 = 0.0030$	$HC = 6.1$	$C_1 = 0.0063$
Channel D	$D_0 = 0.3600$	$HD = 67.93$	$D_1 = 0.55$

Shek Pi Tau Catchment – Hong Kong

Rainfall and runoff data for four flood events that occurred in the Shek Pi Tau catchment in Hong Kong in May 1983, August 1983, May 1984 and August 1985, each lasting for 4, 4, 7 and 9 days, respectively, were used to calibrate and test the Tank Model representing that catchment. The area of the catchment is $27.92\,km^2$. The parameters of the model and the ranges used for their variations are shown in Table 5.

Figure 8 shows the variation of the objective function during the parallel implementation of the proposed GA for three separate computer runs with 8 processors. The parameters for the lowest objective function, namely Run 2, are tabulated in Table 6.

Since the actual parameters, if such exist, are not known in this case, it is not possible to make any comparison.

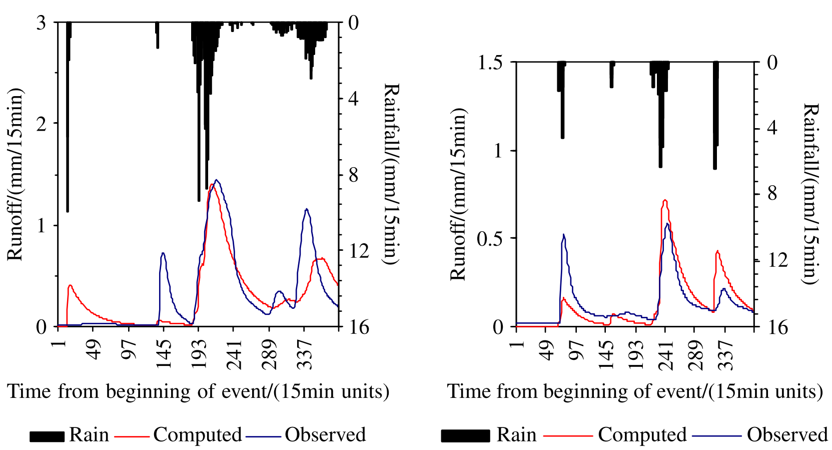

Figure 9 | Observed and GA-optimised Tank Model estimated hydrographs for calibration events for Shek Pi Tau.

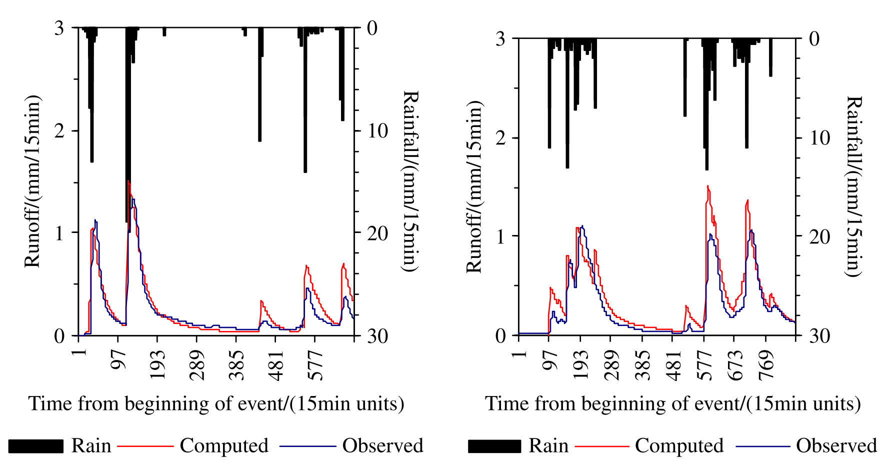

Figure 10 | Observed and GA-optimised Tank Model estimated hydrographs for validation events for Shek Pi Tau.

Table 7 | Results of the event simulations using the Tank Model parameterised by the proposed GA

Flood event	Total rainfall (mm)	Total observed runoff (mm)	Total computed runoff (mm)	RMS error as % of the observed mean runoff
Calibration-1	175.7	129.8	115.67	67.6
Calibration-2	71.8	49.0	46.0	73.8
Validation-1	217.3	148.89	155.5	51.2
Validation-2	366.4	235.1	286.4	57.5

Figures 9 and 10 illustrate the calibration and validation hydrographs, respectively, which show that the observed and the computed runoff for the storm events match fairly well. The discrepancies may be attributed to the data used for this study. From the hydrograph for Calibration Event 1 it is apparent that the rainfall of high intensity at the beginning of the storm event does not produce a peak in the observed hydrograph but a low intensity (of the order of 1–2 mm/15 min) which generates a disproportionate peak flow. In Event 2, however, a high intensity rainfall at the outset of the flood event has produced a peak. These mixed signals have probably confused the parameter optimisation technique and it has failed to capture this strange behaviour of the catchment. It may be that parameters that perfectly represent these data do not exist.

In terms of total flow rates and flood volumes, Table 7 summarises the performance of the Tank Model for the four events.

CONCLUSIONS

The conclusions from this study are:

1. The proposed modified GA with the exclusion of a portion of the weakest individuals in a population appears to marginally enhance the overall fitness of the population. However, further research with different proportions of extinction as well as larger numbers of iterations should be attempted to conclusively declare the superiority of one distribution over the other.
2. When a GA is used for parameter optimisation, a considerable amount of time is required to execute the algorithm over a reasonable number of iterations. This may appear prohibitive on computer platforms of conventional configuration. However, given access to an IBM 9076 SP2 supercomputer, it was possible to parallelise part of the lengthy process, thereby reducing the time to acceptable limits. This proves that GA is a viable method to achieve the objective within hours using the SP2. While not many institutions have access to a supercomputing facility, this study highlights the achievable computation time-saving using one.
3. The values of the parameters obtained at the end of the GA are somewhat different from the "actual" ones although the difference between the resulting runoff is very low. The optimisation is aimed at only minimising the total error, subject to the limits assigned to the parameters. If any inter-relationship among the parameters is known that too can, and should, be tailored into the optimisation objectives.

REFERENCES

Elhassan, A.M., Goto, A. & Mizutani, M. 2001 Combining a tank model for simulating regional groundwater flow in an alluvial fan. *Transaction of Japanese Society of Irrigation Drainage and Reclamation Engineering* **215**, 21–29.

Franchini, M. 1996 Use of genetic algorithm combined with a local search method for the automatic calibration of conceptual rainfall-runoff models. *Hydrol. Sci. J.* **41** (1), 21–39.

Franchini, M. & Pacciani, M. 1991 Comparative analysis of several conceptual rainfall-runoff models. *J. Hydrol.* **122** (1–4), 161–219.

Goldberg, D. E. 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addision Wesley, Reading, MA.

Hendrickson, J. D., Sorooshian, S. & Brazil, L. E. 1988 Comparison of Newton-type and direct search algorithms for calibration of conceptual rainfall-runoff models. *Water Resour. Res.* **24** (5), 691–700.

Holland, J. H. 1975 *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI.

Holland, J. H. 1993 *Adaptation in Neural and Artificial Systems*. MIT Press, Cambridge, MA.

James, L. D. & Burges, S. J. 1982 Selection, calibration and testing of hydrologic models. In *Hydrologic Testing of Small Watersheds* (ed. C. T. Haan, H. P. Johnson & D. L. Brakensiek), American Society of Agricultural Engineers, St. Joseph, MI. pp. 437–471.

Jayawardena, A.W. 1998 Streamflow simulation using Tank Model: application to Hong Kong catchments. *Hong Kong Engineer* July, 33–36.

Setiawan, B. I., Fukuda, T. & Nakano, Y. 2003 Developing procedures for optimization of tank model's parameters. In *Agricultural Engineering International: the CIGR Journal of Scientific Research and Development*, **5**, Available online: http://cigr-ejournal.tamu.edu/submissions/volume5/LW%2001%20006%20Setiawan.pdf.

Sorooshian, S. & Dracup, J. A. 1980 Stochastic parameter estimation procedures for hydrologic rainfall-runoff models: correlated and heteroscedastic error cases. *Water Resour. Res.* **16** (2), 430–442.

Sorooshian, S. & Gupta, V. K. 1983 Automatic calibration of conceptual rainfall-runoff model parameters: the question of parameter observability and uniqueness. *Water Resour. Res.* **19** (1), 260–268.

Sugawara, M., Watanabe, I., Ozaki, E. & Katsuyama, Y. 1984 Tank model with snow component. *Research notes of the National Centre for Disaster Prevention* No. 65, November, Science and Technology Agency, Japan.

Tanakamaru, H. 1995 Parameter estimation for the tank model using global optimization. *Transactions of the Japanese Society of Irrigation, Drainage and Reclamation Engineering* **178**, 103–112 (in Japanese).

Wang, Q. J. 1991 The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Wat. Resour. Res.* **27**(9), 2467–2471.