CS(AR) 11/97-98

CALIBRATION OF A CATCHMENT MODEL USING SCE-UA ALGORITHM

NATIONAL INSTITUTE OF HYDROLOGY
JALVIGYAN BHAWAN
ROORKEE - 247 667

1997-98

CONTENTS

Page No.

PREFACE
ABSTRACT

1.0 INTRODUCTION 1
1.1 Calibration of conceptual models 2
1.2 Scope of this report 4

2.0 THE PROBLEM OF MULTIPLE OPTIMA IN CRR MODELS 5
2.1 Sampling methods 5
2.1.1 Uniform random sampling (URS) method 5
2.1.2 Exhaustive gridding (EG) method 6
2.2 Appraisal of sampling methods 7
2.3 Globe search procedures 10
2.3.1 Adaptive random search method 10
2.3.2 Combined ARS/simplex method 11
2.3.3 Multistart simplex procedure 11
2.3.4 Genetic algorithm 12
2.4 Conclusions 16

3.0 THE SHUFFLED COMPLEX EVALUATION OPTIMIZATION METHOD 18
3.1 The shuffled complex evaluation algorithm 19
3.2 Competitive complex evolution algorithm 21

4.0 APPLICATION OF SCE-UA METHOD 24
4.1 Input data used 24
4.2 Discussion of results 25
4.2.1 Sensitivity of accuracy criteria 27
4.2.2 Optimal duration of data for calibration 28

5.0 CONCLUSIONS 29

TABLES 9
FIGURES 8
REFERENCES 32
APPENDICES 33

LIST OF TABLES

T-2.1 Five major characteristics complicating the optimization problem in CRR model calibration 9

T-4.1 Lower and upper bounds of various parameters 25

T-4.2 Comparison of volumes of observed and simulated discharges for Kolar basin 30

T-4.3 Variation of calibration with accuracy criterion 31

T-4.4 Duration of calibration data and corresponding optimized parameters 31

LIST OF FIGURES

Fig. 2.1 Response surface for two parameters of the CRR model 8

Fig. 3.1 Flowchart of the SCE-UA algorithm 20

Fig. 3.2 Flowchart of the CCE strategy of the SCE-UA algorithm 22

Fig. 4.1 Plot of observed and simulated hydrographs for Kolar basin for calibration period 26

LIST OF APPENDICES

A.1 Description of the model used 33

B.1 Variables controlling a run of SCE algorithm 35

ABSTRACT

An important step in application of a conceptual model to a catchment is model calibration. The objective of a calibration is to determine the model parameters such that an acceptable match is obtained between the observed and the computed discharge hydrographs. Two approaches are followed for calibration of a conceptual model -- manual using trial and error and automatic using an optimization algorithm.

The aspects of the conceptual models which cause problems during automatic calibration are : 1) Interdependence between the model parameters, 2) Indifference of the objective function to the values of the inactive parameters, 3) Discontinuities of the response surface, and 4) The presence of local optima.

The degree of complexity of model plays a significant role in model calibration phase. The difficulties encountered during the calibration are closely connected to the number of parameters typical of the model and to the greater or lesser ease of visualizing the various parameters. There are differing views in the literature about what constitutes the adequate data for model calibration. Both the right kind (the data which activate all the model parameters) and right duration of data are needed for a good calibration. Besides ensuring that the data are error free, one has to be careful about the duration of the calibration period also, the periods of extreme events should be suitably incorporated.

Duan et al. (1992) have presented a new algorithm, named Shuffled Complex Evolution Method (SCE-UA). It has been claimed that this method has a very high probability of finding the global optima. The main aim of this report was to apply the SCE-UA algorithm for the calibration of a CRR model. The model reported by Jain (1993) was used to simulate the response of a basin of size 820 km^2. The results show that the algorithm is able to converge to the global optimum when the computations are started from a number of initial points. The main conclusion of the study is that the SCE-UA algorithm is a global optimization method and is able to converge to the global optimum parameters when different initial values of parameters are used. The computational requirements for calibrating a CRR model are quite reasonable and thus the algorithm is computationally efficient.

***

1.0 INTRODUCTION

A catchment is a complex system where various physical, chemical, and biological processes take place and govern the movement of water. In practice it is difficult to model all these processes and some simplifications have to be made either in the representation of the system or in the processes involved or both. The most common simplification made is spatial lumping and replacement of various components of the hydrological cycle by conceptual storages. It amounts to saying that the catchment system and its inputs and responses can be represented using the dimensions of depth and time. The within catchment variations of inputs and parameters are ignored. As pointed out by Blackie and Eeles(1985), due to this spatial averaging, the lumped model concept can be considered adequate only for small homogeneous catchments. However, in practice they have been applied to sufficiently big and heterogeneous catchments. The computational requirements of these models are moderately small vis-a-vis the computational speeds of a typical computer now-a-days.

The conceptual rainfall runoff (CRR) modelling lies intermediate between the physically based models and the black box models. Generally the term *conceptual* is used to describe models which rely on simple arrangement of a relatively small number of interlinked conceptual elements, each representing a segment of land phase of hydrologic cycle. The most commonly used element in a conceptual model is the storage. Each of these unequal sized storage usually has one input and one or more outputs and represents a catchment storage like detention, soil moisture etc. The linear reservoirs and channels are used for routing. The modelling basically consists of a set of rules which govern moisture flow from one element to another. Since this is a non-iterative accounting procedure, these models are computationally efficient and pose very small computational requirements in terms of CPU time and memory.

The CRR models were initially developed for small homogeneous areas. However, they have been successfully applied to basins having wide variations in topography and vegetation and catchment area of the order of thousands of sq. km. The input data requirements for these models are quite modest and can be easily met with. Blackie & Eeles (1985) provide excellent discussion on philosophy and applications of these models.

The main advantages of using a lumped conceptual model are the following : a) The CPU time requirement of conceptual models are quite small. The computational efficiency of these models results from the fact that these models make use of several storages to represent the movement of water through different elements of the catchment. The model is nothing but a logical procedure to regulate the inputs and withdrawals of water from these storages. b) The requirement of other computer resources for these models are also very small and hence some of these can be used on micro computers as well. c) In rainfall-runoff studies, normally only one variable, i.e., discharge at a defined location is of interest. This is the major output from most of such models.

1.1 Calibration of Conceptual Models

An important step in application of a conceptual model to a catchment is model calibration. The objective of a calibration is to determine the model parameters such that an acceptable match is obtained between the observed and the computed discharge hydrographs. Basically two approaches are followed for calibration of a conceptual model -- manual using trial and error and automatic using an optimization algorithm. The parameters obtained from automatic calibration may be further fine tuned manually to achieve an improved match from the point of view of interest. According to Sorooshian and Gupta(1983), the purpose of calibration may be: 1) To obtain a unique and conceptually realistic parameter set which closely represents our understanding of the physical system, or 2) to obtain a parameter set which gives the best possible fit between the model-simulated and the observed hydrograph. However, from the point of view of physical modelling, a method which emphasizes both aspects is desirable.

Four aspects of the CRR models which cause problems during automatic calibration were listed by Johnston and Pilgrim (1976) as : 1) Interdependence between the model parameters, 2) Indifference of the objective function to the values of the inactive parameters, 3) Discontinuities of the response surface, and 4) The presence of local optima. Sorooshian and Gupta(1983) identified three areas which hinder the accurate calibration of the CRR models : 1) model structure representation, 2) data and their associated measurement errors, and 3) imperfect representation of the physical process by the model. The data which are used in calibration may not represent the entire range of hydrologic events that the catchment may experience and the consequent lack of activation of parameters leads to differing sensitivities of the response surface & poor convergence properties. This problem is accentuated since the optimum parameters are to be found in a high dimensional parameter space. The threshold parameters, cross-correlation between parameters and auto-correlation and hetrescedascity in the residuals also cause difficulties, Beven and Binley (1992).

Franchini and Pacciani(1991) pointed out that automatic calibration, rather than capitalizing on prior knowledge intrinsic of the model, ignores it and thus emphasizes the uncertainty inherent in every statistical analysis. The degree of complexity of model plays a significant role in model calibration phase. The difficulties encountered during the calibration were found to be closely connected to the number of parameters typical of the model and to the greater or lesser ease of visualizing the various parameters. Significantly different models produced basically equivalent results, with calibration time generally proportional to the complexity of the structure of the model. They further pointed out that an excess of schematization causes loss of the link with the physics of the problem and of the possibility of taking advantage of the prior knowledge of the geomorphological nature of the watershed. These findings also support the age old wisdom of going for the simpler approach to a problem if the results are comparable.

Franchini et al. (1998) compared three algorithms for global optimization of CRRs -- the Genetic Algorithm coupled with Sequential Quadratic Programming (GA-SQP), the pattern search method coupled with SQP (PS-SQP) and the SCE-UA method. Two types of data set were used in the analysis, the first generated using the model and hypothetical rainfall (having no model and data error and termed as theoretical data set) and the second was actual observed data of a real catchment. When the data of a single catchment was used, it was found that the SCE-UA algorithm was the most reliable as it converged to the exact solution in all runs when the theoretical data set was used and was the most consistent when the real world data were used. The performance of other two algorithms was found to be inferior. The authors also used data of a complex basin (consisting of three sub-basins and the gauging station located at the outlet of third sub-basin) and it was found that none of the algorithms converged to the exact solution in the theoretical case. However, the SCE-UA algorithm systematically converged nearest to the exact solution as compared to the others. The authors concluded that the SCE-UA algorithm is the best while the other two are equivalent in performance.

The results of their analysis also showed that the optimization algorithm is only one of the factors which affect the calibration of a CRR model. The other important factors are: 1) The conceptual base and structure of the model, 2) The quality and amount of information contained in the data set used for calibration, 3) The selection of the objective function used in the optimization procedure, and 4) The definition of the feasible space of the parameters. Further, all these aspects are highly interactive. It may be added that the importance of the above aspects has also been highlighted in several other papers which have been referred earlier in this section.

Regarding the factor number 2 in the above paragraph, there are differing views in the literature about what constitutes the *adequate data* for model calibration. Sorooshian and Gupta(1983) point out that use of more and more data is not necessarily the answer. What is required is the right kind of data. The term right kind of data implies the data which activate all the model parameters. Clearly, both the right kind and right duration of data are needed for a good calibration. Besides ensuring that the data are error free, one has to be careful about the duration of the calibration period too. While deciding it, one must ensure that the periods of extreme events have been suitably incorporated.

1.2 Scope of this Report

Duan et al. (1992) have presented a new algorithm, named Shuffled Complex Evolution Method (SCE-UA). It has been claimed that this method has a very high probability of finding the global optima. The objective of this report is to apply this approach for calibration of the CRR model reported by Jain (1993). Duan et al. (1992) have given a comprehensive discussion on the problems associated with calibration of CRR models. The same is summarized here.

***

2.0 THE PROBLEM OF MULTIPLE OPTIMA IN CRR MODELS

The problem of CRR model calibration has its own peculiar characteristics. This chapter discusses the main features of the problem. The major source of difficulty of the problem is due to the existence of many optimal solutions. To tackle this problem, it is necessary to obtain global information about parameter sensitivity and the structure of the objective function response surface so that appropriate algorithm can be devised.

2.1 SAMPLING METHODS

Because of the highly complex and nonlinear nature of CRR models, a theoretical analysis to obtain detailed information about the objective function response surface is very difficult. Therefore, computational methodologies that employ information sampled from the entire parameter space, taking care to adequately span the space and provide a sufficient density of coverage can be used to get the desired information.

Two procedures that are helpful in obtaining the requisite information are : (1) do a uniform random sampling of the parameter space; and (2) select a grid spacing and do exhaustive gridding of the parameter space. Both these require large computer resources. These are discussed below.

2.1.1 Uniform Random Sampling (URS) Method

The method of uniform random sampling is a primitive probabilistic approach to global optimization. In this method, a pre-specified number of points (say N) is sampled at random from the feasible parameter space using a uniform probability distribution. The objective function value is computed at each point, and the point with the best (or minimum in case of CRR models) objective function value is taken as an estimate of the optimum. The N sampled points contain important information about the nature and structure of the objective function response surface. This information can be extracted by constructing graphical projections of the sampled points after having arranged them in order of increasing objective function value. Two useful graphical projections are: (1) X-Y plot of the distance of each point from the optimum normalized by the parameter range (NORD) versus objective function value; and (2) X-Y plot of parameter value (PARV), versus objective function value.

The URS method was used by Duan et al. (1992) to study the SLS response surface of the SIXPAR model (a CRR model having six parameters) by sampling 10,000 points from the feasible parameter space. The plots all show a broad spread of points from the very beginning of the ordered data set, which indicates very poor sensitivity of the objective function to parameter variation over the entire feasible parameter space. An interesting aspect was that of the two "best" points obtained from the sample, the second point was far from the "true" values, while having a function value virtually indistinguishable from the first. This behaviour could possibly be due to insufficient coverage of the feasible space, model structural factors, the characteristics of the data set, or some combination of the three. However, because it was known that the sensitivity of the model output to the percolation parameters is small relative to that of the other parameters of the model, and that a significant degree of parameter interdependence exists (see Gupta and Sorooshian, 1983), the study was repeated with the percolation parameters fixed at their true values. There was no significant change in the results.

2.1.2 Exhaustive Gridding (EG) Method

The method of exhaustive gridding is a deterministic approach to global optimization. First, the number of discretization units for each parameter range is chosen, thereby specifying the number of grid points and their location. Now the function value is computed at each grid point and is compared with those of all immediate neighbouring points. The location and function value of those points for which the function value is less or equal to that of all its neighbours are recorded. These points are part of either a global or a local optimum. This method is computationally very intensive. For example, a two-dimensional problem with a grid size of 100*100 requires 10,000 function evaluations and 78,804 function comparisons. So the method is not efficient to find the global minimum. Moreover, as we move to higher-dimensional subspaces, the computational burden rapidly increases.

To examine the two-dimensional subspaces, a grid size of 100 x 100 was used by Duan et al. (1992) for each pair of parameters. In addition to the case of "perfect" data (no streamflow error), the influence of two types of streamflow data error was examined. In case of homoscedastic error, the error variance does not change with the magnitude of streamflow but for heteroscedastic error, the error variance changes with the magnitude of streamflow. The homoscedastic error variance value used was 25% of the mean streamflow value, while the heteroscedastic error variance used was 25% of the true streamflow value. Besides the SLS objective function, the heteroscedastic maximum likelihood error (HMLE) objective function developed by Sorooshian et al. (1983) was used so that the properties of the two objective functions could be compared.

It was found that some of the subspaces contain quite a large number of local optima.

The percolation parameters were found to be associated with larger numbers of local optima than the other four parameters. The introduction of errors into the streamflow data generally increases the number of local optima though many local optima were found to be present on the response surface even when the data were not corrupted with errors. When the error is heteroscedastic, the choice of objective function influences the number of optima in each subspace. The randomness in the data error significantly influences the behaviour of the response function, with the number of local optima varying with seed value.

The objective function values obtained at each grid point were used to construct mesh surface plots. One such plot for the model reported by Jain (1993) is shown in Fig. 2.1 which shows the presence of local optima. These plots show the comparative sensitivities of the parameters, the regions of roughness of the response surface, and the locations of abrupt changes in slope. In general, the response surface can be quite steep when far from the global optimum, but is relatively insensitive to the parameters near the global optimum. Many local optima appear near the edges of the parameter bounds, while others are scattered in the general region of the global optimum. In addition to isolated stationary points, many of the local optima appear in clusters. The derivatives vary in a discontinuous manner over the feasible space and due to this, the derivative-based (Newton or quasi-Newton type) search algorithms for parameter estimation are not likely to be very successful.

The studies reported in the literature indicate that the number of local optima in the three-dimensional subspaces increases rapidly. This poses additional problems for local search optimization procedures. A mixture of the following three basic patterns of the distribution of local optima are usually observed: (1) sparsely scattered local optima, (2) dense clusters of local optima, and (3) line optima. A great many of the local optima may not be located close to the global optimum.

2.2 APPRAISAL OF SAMPLING METHODS

Based on the above analysis Duan et al. (1992) have identified following five major characteristics which complicate the calibration problem in CRR models.

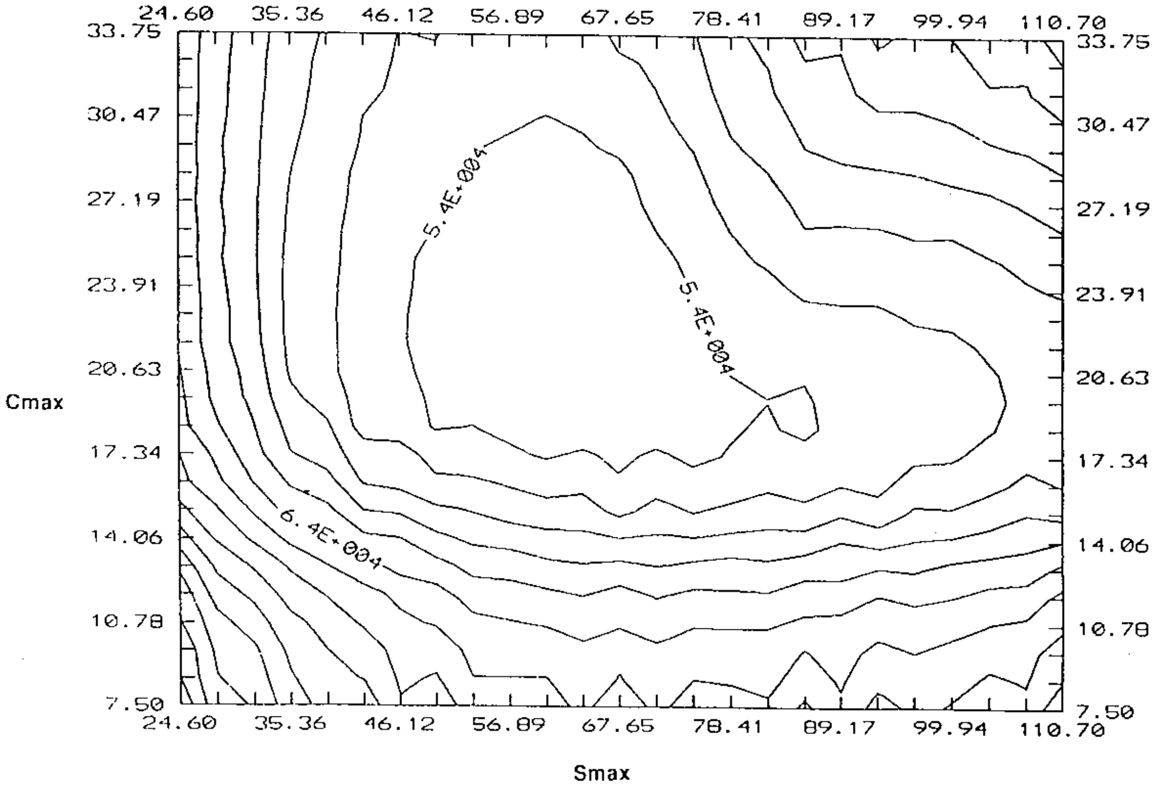


Fig. 2.1 Response surface for two parameters of the CRR model

Major Characteristics Complicating the Optimization Problem in CRR Model Calibration

SN	Characteristic	Reason for Complication
1	Regions of attraction	More than one main convergence region
2	Minor local optima	Many small "pits" in each region
3	Roughness	Rough response surface with discontinuous derivatives
4	Sensitivity	Poor and varying sensitivity of response surface in region of optimum, and nonlinear parameter interaction
5	Shape	Non-convex response surface with long curved ridges

The most important of the above five characteristics is that the structure of multiple optima exists on at least two scales. At the "large" scale, there is more than one broad "region of attraction" into which a search strategy may converge. Such regions can be seen very clearly in the EG mesh surface plots, while the URS method is able to detect such regions in multiple parameter spaces. However, at the "small" scale, each major region of attraction contains numerous local minima (stationary points where the first derivatives are zero and the Hessian matrices are positive definite or positive semi-definite). These minor optima occur both close to and at various distances from the best solution. The minor local optima are not detectable using the URS method. They are also not normally visible on mesh surface and contour plots and can only be detected by numerical analyses of the gridded data.

The large number of minor optima is the most probable reason for the inability to find unique "optimal" parameter values. In a local search procedure, when a line optimum or a stationary point is encountered, the optimization procedure will generally stop. Local searches with small step size will be unable to run the maze of minor optima, inevitably failing to reach the global optimum. Because many of the minor optima can be found quite far from the global optimum, the search may terminate without even finding an approximate solution. In addition to the presence of optima at different scales, the objective function surface in the multi-parameter space is not smooth and has discontinuous derivatives that vary in an unpredictable manner through the parameter space. This is the reason why derivative-based local optimization methods do not performed well. Furthermore, it indicates that, for any global optimization to be successful, it must not depend on smooth and continuous derivatives either. Finally, the response surface in the region of the global optimum is not necessarily convex; it exhibits widely varying degrees of sensitivity to the model parameters, and indicates the existence of a great deal of nonlinear parameter interaction and compensation.

2.3 GLOBAL SEARCH PROCEDURES

The methods for finding the global solutions to multi-optima problems may be classified as deterministic or probabilistic. The deterministic methods can provide a guarantee of success. However these methods require that the function satisfy certain restrictive conditions (e.g., continuity, differentiability to second order etc.) that cannot be guaranteed for CRR models. Further, they are typically inefficient and slow in converging to the optimum. Probabilistic methods involve the evaluation of the function at a random sample of points in the feasible parameter space, followed by subsequent manipulations of the sample using a combination of deterministic and probabilistic rules. The probabilistic methods can guarantee convergence only in a probabilistic sense. However, they are quite efficient in practice and have the major advantage that they do not usually impose restrictive conditions on the nature of the function. Many such methods can be employed when the function is discontinuous and when derivative information is difficult or impossible to obtain. This makes them potential candidate for the optimization of CRR models. In the following discussion, some probabilistic global optimization methods have been discussed.

2.3.1 Adaptive Random Search Method

The uniform random sampling (URS) approach discussed previously is a probabilistic optimization method which does not beneficially use any information about the nature of the response surface obtained during sampling to direct the search. For this reason, several strategies have been proposed to guide the random search adaptively toward the region of the global optimum. One such strategy is the Adaptive Random Search (ARS) method. The first step in applying the ARS method is to specify a feasible space in which the automatic search is to be conducted. To define the feasible space, the upper and lower bounds on each of the parameters are given. These bounds can be estimated by analysis of the hydrologic data, from knowledge of the physiographic characteristics of the watershed and by prior experience of working with the model. The ARS strategy is as follows :

1. Choose a focal point. This point can be the "best" point obtained in the preliminary process of defining the parameter space, or it can be some arbitrary point such as the centroid of the feasible space.

2. Generate a set of N points randomly distributed in the entire feasible space (for

example, according to a uniform or normal distribution) and centered on the focal point. Store the location of the point with the best function value.

3. Repeat step 2 a pre-specified number of times, on the ith time using the initial parameter range divided by 10^i anrl centered on the focal point to restrict the search space. Each time, store the location of the point with the best function value.

4. Compare all the stored points and determine the point with the best function value. Redefine this point to be the new focal point. Record the range level in which this point was found.

5. Repeat steps 2-4 until the best point is found in the smallest range level a user-specified successive number of times (say three). This point is chosen as the optimal parameter set.

A review of the optimization literature indicates that the ARS strategy works well for simple problems. The ARS algorithm, however, has been found to be neither effective nor efficient for the problem of calibration of CRR models. It has been reported that after about 1000-5000 function evaluations, the marginal benefit of further sampling is quite small. This is because the probability of finding a lower function value diminishes as the search proceeds. Thus the ARS algorithm is not much suited for CRR models.

2.3.2 Combined ARS/Simplex Method

As an improved strategy, a random search strategy could be used to obtain an initial point from which a local search procedure could then be started, Ibbitt and O'Donnell [1971]. They mentioned that the use of the outcome of a stochastic search as the starting point for a deterministic search indicates that a sequential stochastic deterministic. search technique would be very powerful, particularly for poor initial parameter values. Using this strategy, the failure rate reduces considerably with only marginal increases in the average number of function evaluations. However, the failure rate still remains high.

2.3.3 Multistart Simplex Procedure

A method for dealing with multiple optima is to run several trials of a local search optimization method from different starting points in the feasible space. Johnston and Pilgrim (1976) suggested that a set of parameter values should not be accepted as an optimum until a number of attempts to make further improvements have been made, e.g. by using another optimization method starting from this set of parameter values or numerical trials around these values. Several different starting points and more than one optimization methods should be used. This global search strategy of repeating the search from many different locations is called a *multistart procedure*. The validity of such an approach can be demonstrated by the following arguments. To have good confidence in the results of any probabilistic optimization procedure, it is necessary to have a relatively small failure probability on the problem of interest. Let us say that we run a procedure once on a problem beginning from some randomly selected location in the feasible search space, and the probability of failure is P_f. If we then rerun the procedure r times from r independent randomly selected locations, the overall failure probability will decrease according to the equation $P_f(r) = P_f(1)^r$ and this will tend to zero as r becomes large. For example, if P_f is 0.65 and if $r = 12$, we will get a failure rate of less than 1 in 100.

The efficiency of the multistart procedure varies nonlinearly with P_f. The number of restarts r required to achieve an overall failure probability of $P_f(r)$ is given by $r = \ln\{P(r)\}/\ln\{P_f(1)\}$. A very large number of restarts is not required for single-start failure probabilities $P_f(1)$ of less than approximately 0.8. However, as $P_f(1)$ increases above 0.8, the number of restarts required rapidly increases thus making the procedure impractical.

The results of the multistart simplex (MSX) show that from a single-start failure probability of approximately 65%, the failure rate with 12 restarts falls to 1 in 100. This result is very encouraging, in view of the difficult nature of the problem. However, the number of function evaluations required to achieve this feat (approximately 10,500) is still high. A complex CRR model when calibrated with several years of data requires quite significant amounts of computer time for even one function evaluation. Moreover, due to the fact that a CRR model may have many more optimizable parameters than the SIXPAR model, the initial failure probability $P_f(1)$ is likely to be much higher than 0.65, so that the number of restarts required would be much larger than 12. It is, therefore, desirable that the efficiency of the search procedure be improved.

2.3.4 GENETIC ALGORITHM

The genetic algorithm (GA) is a search procedure based on the mechanics of natural selection and natural genetics, which combines an artificial survival of the fittest with genetic operators abstracted from nature. It has been applied to a number of problems including search, optimization and machine learning.

The genetic algorithm differs from other search methods in that it searches among a population of points and works with a coding of the parameter set rather than the parameter values themselves. It uses probabilistic rather than deterministic transition rules. In this algorithm, a population of m points are chosen initially at random in the search space. The objective function values are calculated at all points and compared. From these m points, two points are selected randomly, giving better points higher chances. The selected two points are subsequently used to generate a new point in a certain random manner with occasionally added random disturbance. This is repeated until m new points are generated. The generated population of points are expected to be more concentrated in the vicinity of optima than the original points. The new population of points, which can again be used to generate another one and so on, yields points more and more concentrated in the vicinity of the optima.

Wang (1991) showed that a GA when coupled to a standard local search method can provide an efficient and robust means for calibrating a CRR model. Franchini (1996) modified the GA proposed by Wang and applied it for calibration of CRR models. A description of the Wang's genetic algorithm modified by Franchini (1996) is given below. The three modifications are : (i) a different parameter coding system; (ii) a different criterion for merging two distinct points; and (iii) a different criterion for the definition of an offspring population. The first modification replaces the binary coding system proposed by Wang with one based on real numbers. This modification has a minor effect on the characteristics of the genetic algorithm: it simply offers the possibility of programming directly with the real rather than binary numbers. Hydrologists are more familiar with the real numbers. By contrast, the last two modifications have a major and positive effect on the efficiency of the procedure in finding the general "optima/minimum" of the objective function.

Consider the function $f = f(x_1, x_2, ..., x_n)$ subject to $a_i \leq x_i \leq b_i$, $i = 1, 2, 3, ..., n$. The aim is to define the set or vector of parameters which produces the general minimum of the given function in the selected domain. Thus, the range of each parameter is discretized into M points and the discretization interval is :

$$\Delta x_i = (b_i - a_i) / (M - 1) \qquad ...(2.1)$$

The jth value of the ith parameter can easily be defined as :

$$x_{ij} = a_i + (j - 1) \Delta x_i; \qquad i = 1, 2, 3, ..., n; \quad j = 1, 2, 3, ..., M \qquad ...(2.2)$$

Therefore, the code of each value of the ith parameter can be represented by j, while M^n represents the total number of points in the search space among which the global minimum is sought. The procedure can be implemented by the following steps :

1. Set the counter i_{eval} to 1.

2. Select m distinct points ($m \leq M$) randomly in the search space, i.e. consider the first of the n parameters; generate an integer j value at random ($1 \leq j \leq M$); calculate the value of the first parameter using equation (2.2); do the same for the other parameters. These steps identify the first point in the search space. Repeat this procedure m times.

3. Find the objective function value for each point.

4. Rank the points so that their function values are in descending order.

5. Assign a probability value P_k to each point ($k = 1, 2, 3, \ldots, m$), giving higher probability to points with a lower function value. The worst point after ranking has probability p_l while the best point has probability P_m. The other points have a probability ranging from p_l to p_m. Wang (1991) suggested that a linear relationship should be used :

   $$p_k = p_l + (p_m - p_l)*(k - 1) / (m - 1) \qquad \ldots(2.3)$$

   The average probability value for all points is l/m. The probability of the best point can be defined as p_m = C/m while the probability of the worst point can be defined as p_l = (2 - C)/m. Wang (1991) suggested that the value C = 2 should be used. Smaller values of C can be used, resulting in a more robust but slower search.

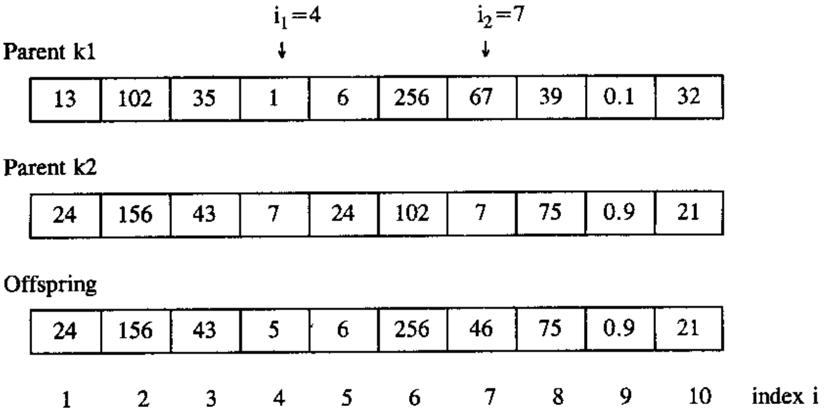
6. Select two points (two parents) k_1 and k_2 at random from the m points already generated, according to the probability distribution, P_k ($k = 1, 2, 3, \ldots, m$).

7. Select two integer values i_1 and i_2 ($1 \leq i \leq n$). If $i_1 > i_2$, their values are exchanged. Each of these two integer values should be considered as the index of a parameter in the corresponding vector parameter.

8. Create a new point according to the following rule : from the point k_1 take all the parameters whose indices i in the vector parameter are $i_1 < i < i_2$; from the point k_2 take all the parameters whose indices i are $i > i_2$ and $i < i_1$. In the new point, the two parameters with indices $i = i_1$ and $i = i_2$ are calculated by selecting at random a value between the corresponding values (extremes included) observed at the

two starting points (in the two parents) k_1 and k_2 (see Fig. 2.2).

9. Occasionally, with a small probability p*, produce an alteration of the newly created point. To do this, choose one or more parameters at random. For each parameter of the new point, generate a code j ($1 \leq j \leq M$) at random, then compute the new value of the parameter using eq (2.2). Wang (1991) suggested the value 0.01 for p*.



index i	1	2	3	4	5	6	7	8	9	10
Parent kl	13	102	35	1	6	256	67	39	0.1	32
Parent k2	24	156	43	7	24	102	7	75	0.9	21
Offspring	24	156	43	5	6	256	46	75	0.9	21

Fig. 2.2 Generation of Offspring in Genetic Algorithm

10. Repeat steps 6-9 m times so that new m points are generated. These new m points represent the offspring population and substitute the old ones but the best point of the previous populations (the parents) is temporarily maintained as a (m + l)th point.

(a) Find the objective function value for each of the new m points.

(b) Rank the new m points so that their function values are in descending crder.

If the "best" point of the offspring is represented by an objective function value lower than the corresponding "best" point of the parents, this latter value is permanently abandoned and the offspring population comprises the previously identified m points. Otherwise, the worst point of the offspring is rejected and the m points of the new population comprise m - 1 points of the offspring with the mth point represented by the best point of the preceding population.

11. Increase the counter i_{eval} by one and repeat steps 5-10. The best point found so far is always recorded. The search is terminated when steps 5-10 have been repeated n_{eval}

times, i.e. when $i_{eval} = n_{eval}$. If the preselected population consists of m points, the number of times the objective function is calculated is mofe = n_{eval} x m (mofe = maximum objective function evaluations). The parameter mofe can be used as a stopping criterion in place of n_{eval}.

Note that the steps described above reflect natural genetics in some respects : in fact, for any animal species, the DNA chain of an "individual" is a mixture of the DNA chains of its parents. Furthermore, fit parents are likely to produce fit offspring and better performing individuals produce more offspring. The combination of selection and reproduction steadily improves the performance level of the better individuals. In any case the "individual" with the best adaptation remains in the population at the expense of the weaker "individual", until an individual with superior adaptation replaces "it".

The above algorithm was applied for calibration of the Xinanjiang rainfall-runoff model by Franchini (1996). The seven parameters of the model were optimized. Out of 10 runs of optimization, each starting from a different set of randomly selected initial points in the search space, eight runs were able to locate the global peak. Although the other two runs located two other peaks, the objective values of these two peaks were only marginally higher than, and practically indistinguishable from, the objective function value of the global peak. Thus all 10 runs were regarded as successful by them. The number of objective function evaluations was about 6100. This number appears to be quite high.

2.4 CONCLUSIONS

As discussed above, the problem of calibration of CRR models is unique and the available methods are not capable of providing a satisfactory answer. The need is to design an optimization procedure which can overcome these various difficulties. Such an algorithm should have the following properties :

- The algorithm must be globally based and possess global convergence properties.
- The algorithm must be able to avoid being trapped by minor optima, and it must not require the availability of explicit analytic expressions for the objective function in terms of its parameters or for the derivatives.
- It must be robust in the presence of parameter interaction and non-convexity of the objective function surface.
- The algorithm must be efficient in the presence of high dimensionality because CRR models usually involve a large number of parameters.

The Shuffled Complex Evolution Method (SCE-UA) developed by Duan et al. (1992) has been claimed to have a very high probability of finding the global optima. The same is described in the next chapter.

***

3.0 THE SHUFFLED COMPLEX EVOLUTION OPTIMIZATION METHOD

A strategy based on the use of multiple simplexes which start from random locations of the search space, has certain desirable properties that enable it to overcome the various difficulties encountered on the response surface of a CRR model. A source of inefficiency in the method is that each simplex search operates completely independently, with no sharing of information. This is analogous to giving the same difficult problem to 12 identically capable people and asking them to solve it without conferring with each other. A more efficient strategy would clearly be for them to spend some time working independently or in small groups and to get together now and then to share information about their progress.

Based on this notion of sharing information and on concepts drawn from genetic algorithm, Duan et al. (1992) have developed an optimization strategy called the *Shuffled Complex Evolution (SCE-UA)* method. It has been claimed that this strategy is a global optimization algorithm. The theory of the method is discussed in the following.

The SCE-UA approach treats the global search as a process of natural evolution. The sampled points constitute a population. The population is partitioned into several communities (complexes), each of which is permitted to evolve independently (i.e., search the space in different directions). After a certain number of generations, the communities are mixed and new communities are formed through a process of shuffling. This procedure enhances survivability by sharing of the information (about the search space) which is gained independently by each community.

Each member of a community (complex) is a potential parent with the ability to participate in a process of reproduction. A subcomplex selected from the complex is like a pair of parents, except that a subcomplex may consist of more than two members. To ensure that the evolution process is competitive, it is necessary that the probability that "better" parents contribute to the generation of offspring is higher than that of "worse" parents. The use of a triangular probability distribution ensures this competitiveness. Nelder and Mead's [1965] procedure is applied to each subcomplex to generate most of the offspring. This strategy uses the information contained in the subcomplex to direct the evolution in an improvement direction. In addition, offsprings are introduced at random locations of the feasible space under certain conditions to ensure that the process of evolution does not get trapped by unpromising regions. This is analogous to mutation in response to stress that can occur in biological evolution. Each mutation also helps to increase the amount of information stored in the sample. Finally, each new offspring replaces the worst point of the current subcomplex. This ensures that every parent gets at least one chance to contribute to the reproduction process before being replaced or discarded. Thus, none of the information contained in the sample is ignored.

The processes of competitive evolution and complex shuffling inherent in the SCE-UA algorithm help to ensure that the information contained in the sample is efficiently and thoroughly exploited. They also help to ensure that the information set does not become degenerate. These properties endow the SCE-UA method with good global convergence properties over a broad range of problems. Thus, given a pre-specified number of function evaluations (fixed level of efficiency), the SCE-UA method has a high probability of succeeding in its objective of finding the global optimum.

The computations begin with a population of points sampled randomly from the feasible space. The population is partitioned into several communities, each containing (2n+1) points where n is the dimension of the problem. Each community is made to evolve based on a statistical "reproduction" process that uses the "simplex" geometric shape to direct the search in an improvement direction. At periodic stages in the evolution, the entire population is shuffled and points are reassigned to communities to ensure information sharing. As the search progresses, the entire population tends to converge toward the neighbourhood of global optimum, provided the initial population size is sufficiently large. The algorithm is discussed in more detail below.

3.1 THE SHUFFLED COMPLEX EVOLUTION ALGORITHM

The SCE algorithm of Duan et al. (1992) combines the strengths of the simplex procedure of non-linear optimization, the concepts of controlled random search, competitive evolution and the concept of complex shuffling. The algorithm is reproduced below and is illustrated in Figure 3.1.

1. To initialize the process, select $p \geq 1$ and $m \geq n + 1$, where p is the number of complexes, m is the number of points in each complex, and n is the dimension of the problem. Compute the sample size $s = pm$.

2. Generate a sample as follows. Sample s points $x_1, \ldots\ldots, x_s$, in the feasible space $\Omega \subset \mathbb{R}^n$. Compute the function value f_i at each point x_i. In the absence of prior information, a uniform sampling distribution can be used.

3. Rank the points as follows. Sort the s points in order of increasing function value. Store them in an array $D = \{x_i, f_i, i=1,\ldots, s\}$, so that $i = 1$ represents the point with the

smallest function value.

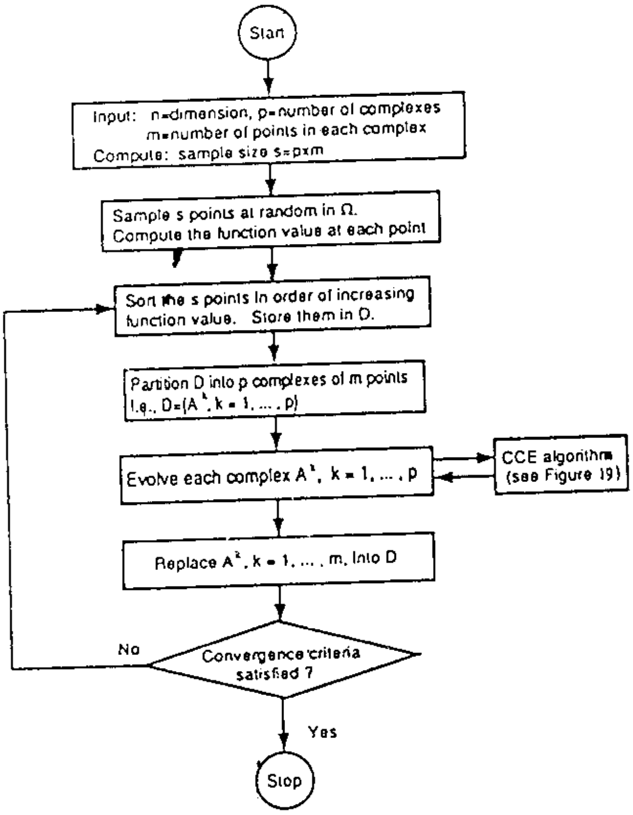

Start
Input: n=dimension, p=number of complexes
m=number of points in each complex
Compute: sample size s=pxm
Sample s points at random in Ω.
Compute the function value at each point
Sort the s points in order of increasing function value. Store them in D.
Partition D into p complexes of m points
i.e., D={A^k, k = 1, ..., p}
Evolve each complex A^k, k = 1, ..., p
CCE algorithm
(see Figure 19)
Replace A^k, k = 1, ..., m, into D
Convergence criteria satisfied ?
No
Yes
Stop

Fig.3.1 Flowchart of the SCE-UA algorithm given by Duan et al. (1992)

4. Partition D into p complexes A^1, ..., A^p, each containing m points, such that $A^k = \{x^k_j, f^k_j | x^k_j = x_{k+p(j-1)}, f^k_j = f_{k+p(j-1)}, j = 1, ..., m\}$.

5. Evolve each complex A^k, k = 1, ..., p, according to the competitive complex Evolution (CCE) algorithm outlined separateiy.

6. Shuffle the complexes as follows. Replace A^1, ..., A^p into D, such that D = {A^k, k = 1, ..., p}. Sort D in order of increasing function value.

7. Check convergence. If the convergence criteria are satisfied, stop; otherwise, return to step 4.

3.2 COMPETITIVE COMPLEX EVOLUTION ALGORITHM

The competitive complex evolution (CCE) algorithm required for the evolution of each complex in step 5 of the shuffled complex evolution method is presented below and is illustrated in Figure 3.2:

1. To initialize the process, select q, α, and β, where $2 \leq q \leq m$, $\alpha \geq 1$ and $\beta \geq 1$.

2. Assign weights as follows. Assign a trapezoidal probability distribution to A^k, i.e.,

$$p_i = [\ \{2(m+1-i)\}\ /\ \{m(m+1)\}\ ],\ i = 1, ..., m \qquad ...(3.1)$$

The point x^k_1 has the highest probability ρ_1 =2/m+1. The point x^k_m has the lowest probability ρ_m = 2/m(m+1).

3. Select parents by randomly choosing q distinct points u_1 ..., u_q, from A^k, according to the probability distribution specified above (the q points define a "subcomplex"). Store them in array B = {u_i , v_i , i = 1, ..., q }, where v_j is the function value associated with point u_j. Store in L the locations of A^k which are used to construct B.

4. Generate offspring according to the following procedure :

(a) Sort B and L so that the q points are arranged in order of increasing function value and compute the centroid g using the expression :

$$g = \frac{1}{q-1}\sum_{j=1}^{q-1} u_j \qquad ...(3.2)$$

(b) Compute the new point r = 2g - u_q (reflection step).
(c) If r is within the feasible space Ω, compute the function value f_r and go to step d; otherwise compute the smallest hypercube $H \subset \mathbb{R}^n$ that contains A^k, randomly generate a point z within H, compute f_z, set r = z and set f_r = f_z (mutation step).

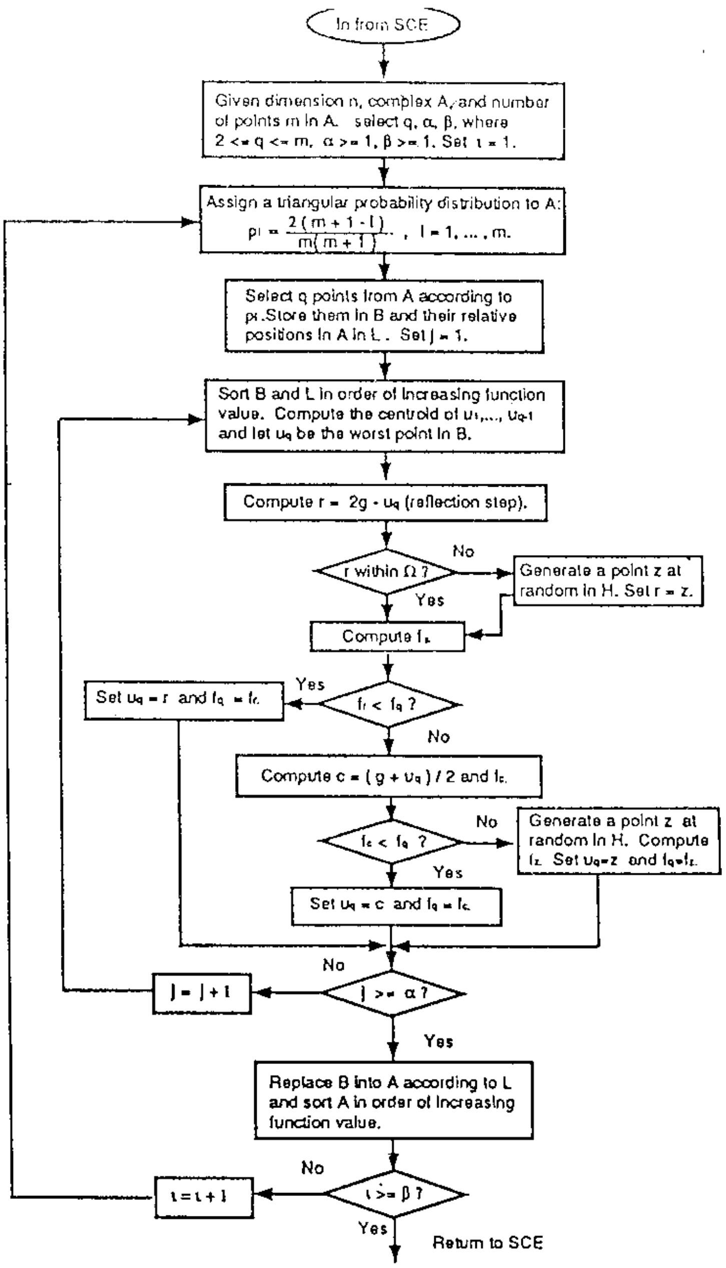

In from SCE
Given dimension n, complex A, and number of points m in A. select q, α, β, where 2 <= q <= m, α >= 1, β >= 1. Set ι = 1.
Assign a triangular probability distribution to A: pi = 2(m + 1 - i) / m(m + 1), i = 1, ... , m.
Select q points from A according to pi. Store them in B and their relative positions in A in L. Set j = 1.
Sort B and L in order of increasing function value. Compute the centroid of u1,..., uq-1 and let uq be the worst point in B.
Compute r = 2g - uq (reflection step).
r within Ω ?
No
Yes
Generate a point z at random in H. Set r = z.
Compute fr
Yes
fr < fq ?
No
Set uq = r and fq = fr
Compute c = (g + uq) / 2 and fc
fc < fq ?
No
Yes
Generate a point z at random in H. Compute fz. Set uq=z and fq=fz.
Set uq = c and fq = fc
No
j >= α ?
j = j + 1
Yes
Replace B into A according to L and sort A in order of increasing function value.
No
ι >= β ?
ι = ι + 1
Yes
Return to SCE

Fig.3.2 Flowchart of the CCE strategy of the SCE-UA algorithm given by Duan et al. (1992)

(d) If $f_r < f_q$, replace u_q by r and go to step f; else compute $c = (g + u_q)/2$ and f_c (contraction step).
(e) If $f_c < f_q$, replace u_q by c, go to step f; otherwise randomly generate a point z within H and compute f_z (mutation step). Replace u_q by z.
(f) Repeat steps (a) to (e) α times, where $\alpha \geq 1$ is a user-specified parameter.

5. Replace parents by offspring as follows : Replace B into A^k using the original locations stored in L. Sort A^k in order of increasing function value.

6. Iterate by repeating steps 2-5 β times, where $\beta \geq 1$ is a user-specified parameter which determines how many offspring should be generated (how far each complex should evolve).

The algorithm developers have recommended the values $m = (2n + 1)$, $q = (n + 1)$, $\alpha = 1$, and $\beta = (2n + 1)$ and the only variable to be specified by the user is the number of complexes p.

The results of an application of this algorithm are presented in section 4.0.

***

4.0 APPLICATION OF SCE-UA METHOD

The SCE-UA method described in previous chapter was applied for the calibration of the CRR model used by Jain (1993). This model is briefly described in Appendix A. This model has ten parameters. Out of these, six parameters represent characteristics of the conceptual catchment storages and control the movement of water through them. The remaining four parameters are time constants of various reservoirs and affect the shape of the discharge hydrograph. A two stage process is recommended for calibration of this model. The first stage involves matching the volumes of observed and simulated hydrographs on monthly basis. The main parameters affecting it are S_{max}, C_{max}, FC, F_{inf}, C_{int} and Ewf. In the second stage, the shape of the simulated hydrograph is matched with the shape of the observed hydrograph by fine tuning the various parameters and time constants of linear reservoirs. This approach gives flexibility to the modeller to adjust the model parameters in light of the objectives of the study, e.g., whether peak flow modelling is more important or low flow modelling.

The code for the SCE-UA algorithm was provided by Prof. Hoshin Vijay Gupta of University of Arizona, USA, who is one of the developers of this algorithm. The code provided by him included the main programme and subroutines for the SCE algorithm. A subroutine is needed for the CRR model being used and one such routine for the SIXPAR model was also provided. To use the proposed model, two subroutine were needed to be written. The first routine was necessary to read the initial data necessary for simulation. This data included the details about simulation period, initial value of the various model parameters, and initial conditions at the beginning of the simulation. The second subroutine is called by the SCE algorithm a number of times with different parameter values. At each call, this subroutine simulates the catchment response with the trial parameter set, checks if any constraints are being violated and computes the value of the objective function.

All the programming was done in Fortran and double precision accuracy was used for real numbers. The computations were carried out on an IBM-compatible Pentium machine.

4.1 INPUT DATA USED

The data of Kolar sub-basin of Narmada basin was used in this study. The hourly rainfall and discharge data for the period 1983-88 were available. The same data were used in the calibration study. However, as the computational requirements increase with an increase in the period of calibration, the data for the monsoon period of 1983-85 were used for most of the calibration runs in this study.

The lower and upper bounds of the search region for the various parameters are required to be given as an input. These bounds can be set-up based on expected range from "physical point-of-view", the experience of the modeller, the characteristics of the catchment etc. The bounds should be specified such that they cover the entire range in which the true value of the parameters is likely to fall and may actually over-span the range in which the parameters may lie. The feasible range of the parameters specified in the present case is given in the following table. It can be seen that a rather extensive range has been specified.

Table 4.1 Lower and Upper Bounds of Various Parameters

Parameter	Lower bound	Upper bound
S_{max}	10.0	400.0
C_{max}	10.0	900.0
FC	0.05	0.95
F_{inf}	0.05	0.95
C_{int}	0.05	0.95
Ewf	0.05	0.95

4.2 DISCUSSION OF RESULTS

The SCE algorithm was used to optimize above six model parameters given in Table 4.1. The developers of this algorithm have claimed that this method is a global optimization method. To see whether the algorithm really converges to the global optimum when the initial values of the parameters are different, a number of computer runs were taken. The various combination of initial values of the model parameters were given as input in these runs. These combinations included the extreme values of one or more model parameters. In this way, about 16 calibration runs were taken. It was found that in each case, the algorithm converged at the same point. The corresponding parameter values were : S_{max} = 79.88, C_{max} = 368.09, FC = 0.413, F_{inf} = 0.477, C_{int} = 0.947 and Ewf = 0.088. The value of the objective function (sum of squares between observed and computed discharge) was 4922.99. The number of shuffling loops was 20 and the number of function evaluations was 814 when an accuracy (see appendix B) equal to 0.01 was adopted. The volumes of the observed and simulated discharges for the period 1983 to 1985, on monthly basis are given in Table 4.2. The observed and simulated hydrographs for these years have been plotted in Fig. 4.1.

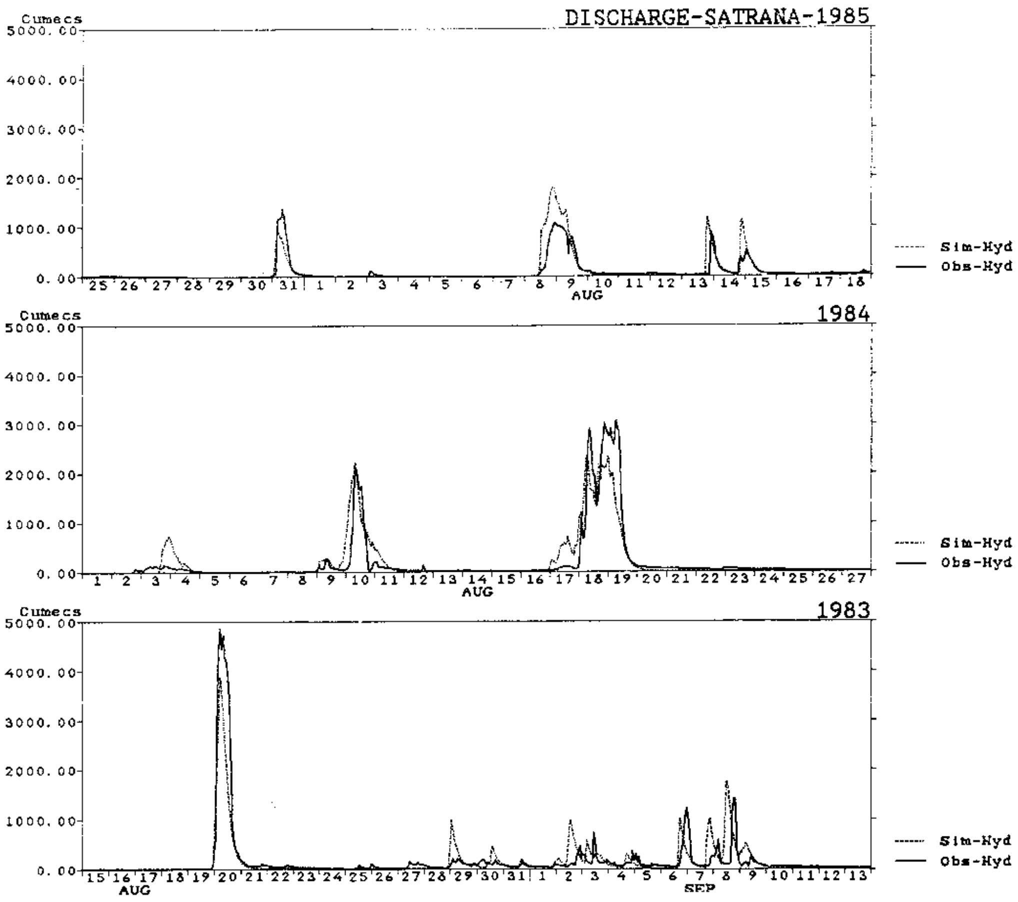


Fig.4.1 Plot of observed and simulated hydrographs for kolar basin for calibration period

The optimal parameters show a minor change when the accuracy criteria is changed slightly but they remain in the vicinity of the same point. Thus the above parameter set can be considered to be the *optimal point* and it can be concluded that this method is indeed a global optimization method. The number of function evaluations needed to arrive at the optimum was not very large and the calibration exercise can be completed in a reasonable time on a PC.

4.2.1 Sensitivity of Accuracy Criteria

The number of function evaluations and thereby the computation burden of an iterative scheme depends upon the accuracy requirements. Evidently, the number of function evaluations goes up with increase in accuracy requirement. To get an idea about the increased computational requirements, the model calibration was carried out with different values of required accuracy. The results of these runs are given in Table 4.3. In order to save computational time, in these runs only four parameters were optimized and the remaining two were set at their optimal value. For all the runs in this table, all other model parameters as well as other controlling variables, except the accuracy requirements, were kept unchanged. It is seen that the number of function evaluations rapidly increases as accuracy requirement is made more stringent although there is insignificant change in the optimized value of the parameters and there is very slight improvement in the final value of the objective function. The final value of the optimized parameters also changes only marginally.

The results of this table can be used as guideline to set the accuracy requirements while calibrating a CRR model. It can be seen that the number of function evaluations almost doubles from 319 to 604 when the accuracy requirement is changed from 1.0 to 0.001: However, the corresponding change in objective function is only from 4927.89 to 4923.01 which is insignificant. The parameter values also do not change significantly. It is apparent from this table that the response surface is more or less flat in the vicinity of the optimal point and this is a common phenomenon associated with CRR models.

In case a long calibration period is used and the computational requirements are very high, the modeller may choose less stringent accuracy requirement to keep the computations in manageable limits. It is pertinent to note that the developments in information technology have lead to availability of powerful personal computers at affordable prices. Looking at the trends in this industry, it is certain that within a period of one to two years, a typical user will have access to machines on which the calibration of a CRR model can be carried out within reasonable times.

4.2.2 Optimal Duration of Data For Calibration

A question which is commonly raised related to calibration of CRR models is : what is the optimal duration of data which should be used in such computations. Jain (1993) mentions that both light quantity as well as quality of data are necessary and the input data should be selected in such a fashion that all the model parameters are activated during the calibration time. Yapu et al. (1996) also examined this aspect and their results for a basin of about 2000 sq. km. suggest that approximately 8 years of calibration data are necessary to obtain parameters that are relatively insensitive to the period selected. Further they found that the benefits of using more than 8 years of data may be only marginal. It was also reported that the parameter identifiability can be significantly improved by selecting the wettest area of data for calibration.

Experiments were made with the current set-up in which the calibration was carried out by using the different lengths of the input data. In these runs, all other conditions were kept unchanged except the duration of input data. The results of these runs have been given in Table 4.4. It is seen that there is wide variation in the optimum parameters when the duration of data is increased from one year. The parameters appear to have somewhat stabilized at the data length of 5 years. The change in the optimum value of the parameters is small when the duration of the data is increased from 5 years to 6 years. As the data for only 6 years were available, the analysis could not be carried out further but the results seem to confirm the findings of Yapu et al. (1996) given above. Thus at least about 7 to 8 years of calibration data appear to be necessary to obtain parameters that are representative for a catchment.

***

5.0 CONCLUSIONS

The main aim of this report was to apply the SCE-UA algorithm of Duan et al. (1992) for the calibration of a CRR model. The model reported by Jain (1993) was used to simulate the response of a basin of size 820 km^2. The results show that the algorithm is able to converge to the global optimum when the computations are started from a number of initial points. The initial points were selected such that the extreme points of all the parameters were selected. The number of function evaluations needed to arrive at the optima was also quite reasonable.

The main conclusions of the study are as follows :

1 The SCE-UA algorithm is a global optimization method and is able to converge to the global optimum parameters when different initial values of parameters are used.

2. The computational requirements for calibrating a CRR model are quite reasonable and thus the algorithm is computationally efficient.

3 The user has complete control on the calibration through a large number of parameters which control the execution of the algorithm.

***

Table 4.2

Comparison of Volumes of Observed and Simulated Discharges for Kolar Basin

Year	Month	Rainfall (mm)	Observed Discharge (mm)	Simulated Discharge (mm)
1983	7	270	29	8
1983	8	548	361	376
1983	9	382	248	292
1983	10	10	37	29
1984	6	141	10	0
1984	7	141	20	0
1984	8	851	592	594
1984	9	27	53	21
1984	10	4	23	11
1985	6	139	0	0
1985	7	293	76	78
1985	8	386	218	238
1985	9	181	60	69
1985	10	118	40	44

Table 4.3

Variation of Calibration with Accuracy Criterion

Accuracy	Optimized Parameters				Objective Function	No. of Loops	No. of Funct. Eval.
	S_{max}	C_{max}	FC	F_{inf}			
1.0	79.36	365.84	0.415	0.481	4927.89	11	319
0.5	79.36	365.84	0.415	0.481	4927.89	11	319
0.2	78.43	362.26	0.422	0.516	4949.54	13	381
0.1	78.43	362.26	0.422	0.516	4949.54	13	381
0.05	78.43	362.26	0.422	0.516	4949.54	13	381
0.01	79.88	368.15	0.413	0.477	4922.97	16	472
0.005	79.88	368.17	0.415	0.477	4923.32	18	537
0.001	79.92	368.26	0.415	0.477	4923.01	20	604

The initial values of the parameters were 10.0, 20.0, 0.1 and 0.1 respectively. The parameters C_{int} and Ewf were set at 0.947 and 0.088.

Table 4.4

Duration of Calibration Data and Corresponding Optimized Parameters

Length of Data (Years)	Optimized Parameters				No. of Loops	No. of Funct. Eval.
	S_{max}	C_{max}	FC	F_{inf}		
1	32.13	659.67	0.739	0.223	21	636
2	22.27	636.10	0.324	0.380	17	504
3	79.88	368.15	0.413	0.477	16	472
4	72.55	380.45	0.412	0.943	16	477
5	77.79	380.40	0.337	0.945	16	460
6	77.55	378.38	0.368	0.944	16	465

REFERENCES

Beven,K., and A. Binley, "The future of distributed models : model calibration and uncertainty prediction", Hydrological Processes, 6, 279-298, 1992.

Blackie, J.R, and C.W.O. Eeles. Lumped catchment models. in Hydrological Forecasting. M.G. Anderson and T.P. Burt (ed). John Wiley and Sons. New York, 1985.

Duan, Q., Sorooshian, S. and Gupta, V.K., Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour. Res. 28(4): 1015-1031, 1992.

Duan, Q., Sorooshian, S. and Gupta, V.K., Optimal use of the SCE-UA global optimization method for calibrating watershed models. J. Hydrol., 158: 265-284, 1994.

Franchini,M., and M. Pacciani, "Comparative analysis of several conceptual rainfall-runoff models", Journal of Hydrology, 122, 161-219, 1991.

Franchini,M., G. Galeati and S. Berra, "Global optimization techniques for the calibration of conceptual rainfall-runoff models", Hydrological Sciences Journal, IAHS, 43(3), 443-458, 1998.

Gupta,V.K., and S. Sorooshian, "The automatic calibration of conceptual catchment models using derivative-based optimization algorithm", Water Resources Research, 21(4), 473-485, 1985.

Ibbitt,R.P., and T. O'Donnell. Fitting methods for conceptual catchment models, Journal of Hydraulic Division, ASCE, 97(HY9), 1331-1342, 1971.

Jain, S.K., "Calibration of conceptual models for rainfall runoff simulation", Hydrological Sciences Journal, IAHS, 38(5), 431-441, 1993.

Johnston,P.R., and D.H. Pilgrim. Parameter optimization for watershed models, Water Resources Research, 12(3), 477-486, 1976.

Sorooshian,S., and V.K. Gupta, Automatic calibration of conceptual rainfall-runoff models : The question of parameter observability and uniqueness, Water Resources Research, 19(1), 260-268, 1983.

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

Yapo, H.V. Gupta, and S. Sorooshian, Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data, Journal of Hydrology, 181, 23-48, 1996.

***

A.1 DESCRIPTION OF THE MODEL USED

In the CRR model used in the present study, the catchment is represented with the help of three storages. The first storage, termed as surface storage, represents the water stored on the surface and top few cms of soil of the catchment. It has a maximum storage capacity given by S_{max} (mm). The second storage represents the catchment soil moisture storage and has a maximum water holding capacity given by C_{max} (mm). The third storage represents the ground water zone. It may be mentioned that most of the existing models make use of similar arrangement of linear reservoirs with varying degrees of complexity. The box diagram of the model structure is given in Fig. A.1.

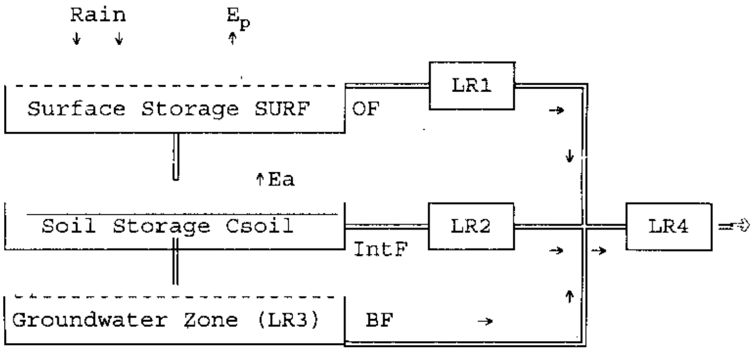


Fig. A.1 Structure of the Conceptual Model Used in this Study

The rainfall is input to the surface storage. The water leaves this storage through evaporation, infiltration or overland flow. The moisture content of this storage at any time is denoted by SURF. If SURF > E_p (potential evaporation in mm/hr), the actual evapotranspiration is at the potential rate else evapotranspiration (ET) takes place from the lower storage at a lesser rate. The infiltration of water from this storage to the soil storage takes place at the rate INF :

$$\begin{aligned} INF &= (1 - C_{soil}/C_{max}) * F_{inf} \quad \text{if SURF} > 0 \\ &= 0 \quad \text{otherwise.} \end{aligned} \qquad \text{... (A.1)}$$

where F_{inf} is a factor(mm/hr) controlling the infiltration rate. When $C_{soil} = C_{max}$, INF is zero. One may visualize that in this event the surface and the soil moisture storages have merged and the downward movement of moisture is computed as described below.

If at any instant SURF > S_{max}, the excess water over S_{max} flows as overland flow (OF). The OF is routed through a linear reservoir LR1 with time constant K_o.

The water which infiltrates from the surface storage enters the soil storage. The outflow from this storage can take place through ET losses, interflow or recharge to the groundwater zone. If the surface storage is empty, ET takes place from the soil storage at a rate E_a given by

$$E_a = C_{soil}/C_{max} * E_p \qquad \ldots(A.2)$$

and C_{soil} is depleted by E_p*dt where dt is the length of computation interval in hour. If SURF < E_p *dt, the actual ET is SURF+E_a *dt. The maximum value of E_a is E_p.

If the contents of soil storage exceed a threshold denoted by FC, water flows out of it as interflow and recharge to groundwater. The excess moisture available for these two is:

$$Exw = (C_{soil}/C_{max} - FC)*Ewf \qquad \text{if } C_{soil}/C_{max} > FC \qquad \ldots(A.3)$$

where Ewf is a factor(mm/hr) controlling the volume of excess water. The volume of interflow is given as :

$$IntF = Exw * C_{int} \qquad \ldots(A.4)$$

and recharge to groundwater is

$$RECH = Exw * (1 - C_{int}) \qquad \ldots(A.5)$$

where C_{int} is a dimensionless coefficient which controls how much of the excess moisture goes as recharge and how much as interflow. The interflow is routed through a linear reservoir LR2 with time constant K_i. The ground water zone behaves as a linear reservoir whose time constant is K_G. The moisture comes out of it as the baseflow (BF). The flow coming out of the reservoirs LR1, LR2 and LR3 is combined and then routed through a linear reservoir, LR4, to yield the discharge from the catchment, denoted by TF.

The input to the model consists of the values of model parameters, rainfall and potential evaporation data for the simulation period, the time step size and the initial contents of various storages.

***

Appendix B

B.1 VARIABLES CONTROLLING A RUN OF SCE ALGORITHM

The various variables controlling a run of SCE algorithm are described below :

MAXN : Maximum number of trials allowed before optimization is terminated. The purpose of MAXN is to stop an optimization search before too much computer time is expended. MAXN should be set large enough so that optimization is generally completed before MAXN trials are performed. MAXN was set at 5000 in this study.

KSTOP : Number of shuffling loops in which the criterion must improve by the specified percentage or else optimization will be terminated. Recommended value is 15.

PCENTO : Percentage by which the criterion value must change in the specified number of shuffling loops or else optimization is terminated. (Decimal equivalent: Percentage/100 used). Varied systematically in the runs.

NGS : Number of complexes used for optimization search. Minimum value is 1. Recommended value is between 2 and 20 depending on the number of parameters to be optimized and on the degree of difficulty of the problem.

ISEED : Random seed used in optimization search. Recommended valus is any large integer.

NPG : Number of points in each complex. NPG should be greater than or equal to 2. The default value is equal to (2 * number of optimized parameters +1).

NPS : Number of points in each sub-complex. NPS should be greater than or equal to 2 and less than NPG. The default value is equal to (number of optimized parameters + 1).

NSPL : Number of evolution steps taken by each complex before next shuffling. Default value is equal to NPG.

MINGS : Minimum number of complexes required for optimization search, if the number of complexes is allowed to reduce as the optimization search proceeds. The default value is equal to NGS.

INIFLG : Flag on whether to include the initial point in the starting population.

***

STUDY GROUP

Director	:	Dr. S. M. Seth
Scientists	:	Dr. S. K. Jain Mr. M. K. Goel
Scientific Staff	:	Mr. P. K. Agarwal
Typing	:	Mrs. Mahima Gupta