International Conference "Water, Environment, Energy and Society" (WEES–2009)
New Delhi, 12–16 January 2009

Reservoir Modelling by Cross Entropy Method

P.G. Jairaj[1] and A.R. Remya
Department of Civil Engineering
College of Engineering, Trivandrum - 695 016, INDIA
E-mail: [1]jairaj_pg@yahoo.com

ABSTRACT: Reservoir modelling problems deals in finding out the optimal reservoir storage volumes and releases to be made in each time period so that the deviation of release values from the demand is minimum thereby maximizing the returns. The present study aims at demonstrating the potential of evolutionary computing algorithm namely Cross Entropy (CE) method for modelling of single reservoir systems. The model development to a single reservoir system operated for satisfying irrigation demands is done. The model was applied to a South Indian Reservoir system. The model development, sensitivity of model parameters in model solution is also explored.

INTRODUCTION

Reservoir modeling problems involves in the determination of optimal storage and release values in different time periods such that the demand is satisfied to the maximum extent without compromising the system constraints. A comprehensive overview of the various conventional techniques for reservoir modeling, with its advantages and limitations is discussed in Yeh (1985). To overcome those limitations, recently meta heuristic techniques like Genetic Algorithm, Simulated Annealing, Ant Colony Optimization, are being used for solving combinatorial optimization problems. These techniques provide a more realistic representation of problem and provide ease in handling the nonlinear and non-convex relationships in the formulation of model.

In the last decade, there has been a wide development and application of evolutionary computing algorithms in various fields of engineering. The Cross Entropy Algorithm is a subset of evolutionary computation, a generic population-based meta heuristic optimization algorithm. The main principle of evolutionary algorithms is reducing the large feasible set of solutions, to smaller sets with in an acceptable computational time. The applications of Cross Entropy method in the modelling of combinatorial optimization problems in the water resources field is limited to water distribution system modeling by Perelman and Ostfeild (2007), Jairaj and Remya (2007). The present study aims at exploring the potential of Cross Entropy (CE) method for modeling a single reservoir system. The concept of cross entropy, details of model development and application to a real world field problem are discussed subsequently.

CONCEPT OF ENTROPY AND CROSS-ENTROPY

In its simplest form, Entropy can be termed as a measure of uncertainty associated with a process. The probability distribution of events, if known provides a certain amount of information. Shannon in the year 1948 defined a quantitative measure of the uncertainty associated with a probability distribution or the information content of the distribution in terms of entropy, called Shannon Entropy (Kapur and Kesavan, 1992) given by the Eqn. 1,

$$H(X) = -k\sum_{1}^{n} p_i \ln p_i \qquad \ldots (1)$$

where $H(X)$ represents the Shannon Entropy corresponding to the random variable X, k is the Boltzmann constant and p_i represents the probability distribution corresponding to the variable x_i.

The uncertainty can be quantified with entropy taking into account all different kinds of available information. Thus entropy is a measure of uncertainty represented by the probability distribution and is a measure of the chaos or of the lack of information about a system. If complete information is available, entropy is equal to zero; otherwise it is greater than zero.

[1]Conference speaker

Cross Entropy is a distance measure from one probability distribution to another. One of the well-known definitions of Cross Entropy is the Kullback-Leibler distance measure, serving to assess the similarity between two probability distributions: the statistical $q(x)$ model and the true distribution $p(x)$. Cross Entropy ($D(P, Q)$) is formulated as in Eqn. 2,

$$D(P, Q) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i} \quad \ldots (2)$$

Looking at Eqn. 2, the Cross Entropy is always greater or equal to the entropy; hence it can be considered as the upper bound of entropy.

Principle of Minimum Cross-Entropy

According to Laplace's principle of insufficient reason, all outcomes of an experiment should be considered equally likely unless there is information to the contrary. Suppose we guess a probability distribution for a random variable X as $Q = \{q_1, q_2, q_3, \ldots, q_n\}$ based on intuition or theory. This constitutes the prior information in terms of a prior distribution. To verify our guess, we take a set of observations $X = \{x_1, x_2, x_3, \ldots, x_n\}$ and compute moments based on these observations. To derive the distribution $P = \{p_1, p_2, p_3, \ldots, p_n\}$ of X, we take all the given information and make the distribution as near to our intuition and experience as possible. Thus, the principle of minimum cross entropy (*POMCE*) is expressed, when the cross entropy, $D(P, Q)$ is minimized, as in Eqn. 3,

$$\text{Minimize } D(P,Q) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i} \quad \ldots (3)$$

On the basis of intuition, experience or theory, a random variable may have an *apriori* probability distribution. The Shannon entropy is maximum when the probability distribution of the random variable is that one which is as close to the *apriori* distribution as possible. This is referred to as the principle of minimum cross entropy, which minimizes the Bayesian entropy (Kullback and Leibler, 1951). This is equivalent to maximizing the Shannon entropy. Here minimizing $D(P, Q)$ is equivalent to maximizing the Shannon entropy.

METHODOLOGY

A single reservoir system with known demands at the reservoir is considered for the model development. In this problem the objective function is to minimize the deviation of release value from the demand value for all time period. The single reservoir system is schematically represented as represented in Figure 1. Thus the objective function is to minimize the deviation of the system release at any time from the demand at that time, subject to the continuity and release constraints. The problem may be mathematically represented as,

$$\text{Minimize } \sum_{t=1}^{12} [D_t - R_t] \quad \ldots (4)$$

$$\text{Subject to: } S_{t+1} = S_t + I_t - R_t - Ovf_t \quad \forall t \quad \ldots (5)$$

$$0 \le S_t \le K_A \qquad \forall t \quad \ldots (6)$$

$$0 \le R_t \le D_t \qquad \forall t \quad \ldots (7)$$

where S_t represent the storage, I_t the inflow value, R_t the release from the reservoir to meet the known demands D_t, and Ovf_t the overflow from reservoir in time period t, K_A is the maximum useful storage capacity of reservoir. The time period considered is one year or 12 months.

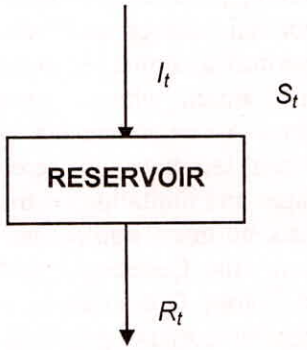

Fig. 1: Single Reservoir System

MODEL APPLICATION

The system considered for model application is the Malaprabha single reservoir system in Karnataka state, South India, which is being used for irrigation purpose. The live storage capacity of the reservoir is 870 mm^3. A constant irrigation demand at the reservoir (incorporating a conveyance loss of 50%) on a monthly basis is made use of in the study. A short term monthly operation of the reservoir system with monthly inflow to reservoir for four years from 1976 to 1978 is made use of in the study.

CROSS ENTROPY ALGORITHM DEVELOPMENT

The Cross Entropy (CE) method is an evolutionary iterative technique based on the concept of rare events, which involves two main stages: (1) generation of a sample of random data (trajectories, vectors, etc.)

according to a specified random mechanism, and (2) parameter updating of the random mechanism, on the basis of the generated data, so as to produce a "better" sample at the next iteration. The procedure of the algorithm is similar to the one used to determine the optimum diameters of a real world water distribution network problem (Jairaj and Remya, 2007).

The steps involved in the algorithm for single reservoir system optimization are described below.

1. The first step is conversion of the deterministic problem into a Stochastic Node Network, by representing the decision variables. The decision variables are numbered continuously covering the entire storage classes with mean value ranging from 5 Mm^3 to 865 Mm^3 for all the twelve time periods. Figure 2 gives the schematic representation of the decision variables.
2. Set the iteration counter $t = 0$ and initialize the probability values for all the decision variables to $p_i = \frac{1}{nf(t_i)}$, where p_i is the probability of the decision variable i and $nf(t_i)$ represents the total number of decision variables in each time period t_i, in this study number of decision variables in each time period is 87, corresponding to the eighty seven storage classes available. Because we have to select one decision variable from the $nf(t_i)$ available options for each time period.
3. Generate N random vectors depending on the probability of decision variables obtained in the last step. In this case the value of N is taken to be equal to the maximum value of $nf(t_i)$, so that all the decision variables appear at least once in the solution. Here random vector corresponds to a system decision or solution, which corresponds to the storage volumes of the reservoir at each of the time periods starting from one to thirteen. It will be an m dimensional vector where m corresponds to the total number of decision variables. Out of these m variables twelve of them will be having a coefficient of one while the rest of them will have zero. Each random vector will be a combination of the twelve decisions (decision variables), each corresponding to one of the twelve time periods.
4. Determine the feasibility of the decision vectors, i.e., check whether the system constraints such as continuity, release and storage constraints etc. are satisfied using Eqns. 5 to 7. The vectors are then arranged in the ascending order of their benefit function values. This is done to separate out the elite sample.
5. Choose a set (say ρ_c) of the top best performing vectors for updating the probability vector $p_{t,j}$ to the probability vector $p_{t+1,j}$. ρ_c corresponds to percentage of the vectors selected and its value varies (between 10% and 20%) but may change as a function of the sample size N. The i^{th} component of $p_{t+1,j}$ is obtained as in Eqn. 8,

$$p_{t+1,i} = \frac{B_{t,i}}{TB_t} \quad \ldots (8)$$

where $p_{t+1,j}$, is the probability of success in the $(t+1)^{th}$ iteration of node i, $B_{t,i}$ is the total number of times node i was chosen out of the best top performance vectors, TB_t at iteration t.
6. In order to avoid early convergence (stopping criteria of decision variable probabilities approaching zero or one) to a local optimum solution, a smoothing parameter α_c is used. The value of smoothing parameter α_c is determined based on sensitivity analysis. The probability is modified as in Eqn. 9,

$$p_{t+1,i} \leftarrow \alpha_c\, p_{t+1,i} + (1-\alpha_c)\, p_{t,i} \quad \ldots (9)$$

Time Period / Storage Class	1	2	3	4					11	12
5	1	88	175	262					871	958
15	2	89	176	263					872	959
25	3	90	177	264	:	:	:	:	873	960
:	:	:	:	:				:	:	:
:	:	:	:	:				:	:	:
855	86	173	260					:	956	1043
865	87	174	261		:	:	:	:	957	1044

Time period No: 12 take the storage class with mean value 15 mm^3

Time period No: 3 take the storage class with mean value 855 mm^3

Fig. 2: Schematic Representation of Decision Variables

7. In this step check whether all the probabilities are approximately equal to zero or one. If yes then the stopping criteria is reached and convert the final vector to its corresponding decisions, which will give the solution to the problem. If the stopping criterion is not reached then continue from step 3 using the new probability.

The coding of the model for the problem is done in *C* language. The model was run till the convergence criteria were met. The results of the sensitivity study, the model solution are discussed subsequently.

RESULTS AND DISCUSSIONS

As explained in the last section a smoothing parameter (α_c) is used to avoid premature convergence of the algorithm. In order to study the significance of the smoothing parameter α_c, a sensitivity study was conducted by varying the value of α_c in the range 20% to 50%. The sensitivity of performance function for some typical values of α_c is shown in Figure 3 and Table 1 respectively. The influence of the parameter α_c on the model solution and the number of iteration to attain the optimum solution are given in Table 1.

From the figure it can be seen that smaller value for smoothing parameter will lead to a very slow convergence of fitness function and higher values lead to faster convergence. Hence number of iterations required to reach stopping criteria will be larger for smaller alpha (α_c) values and vice versa. An optimal value of $\alpha_c = 0.4$, was selected.

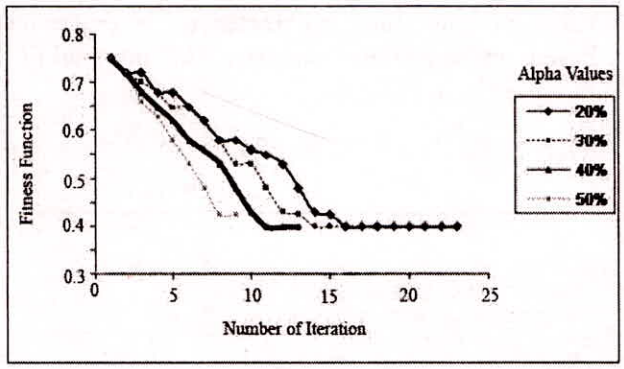

Fig. 3: Sensitivity of Fitness Function to the parameter α_c

Table 1: Influence of Parameter α_c on Model Solution

α_c (%)	No. of Iterations	Optimal Solution
20	23	0.399
30	19	0.399
40	13	0.399
50	9	0.424

The algorithm was implemented as explained earlier using the value of 40% for the smoothing parameter (α_c). At the end of iteration, all those decision variables having probabilities equal to one was combined to produce the optimal system decision or the optimal solution to the problem considered. The decisions variables were transferred to their decisions i.e., to storage classes for the time period, which the decision variable represents. In this problem the near optimal result obtained is having a benefit function value of 0.399, with optimum release values in each period. The optimal release values are plotted against the demand values in each period as shown in Figure 4.

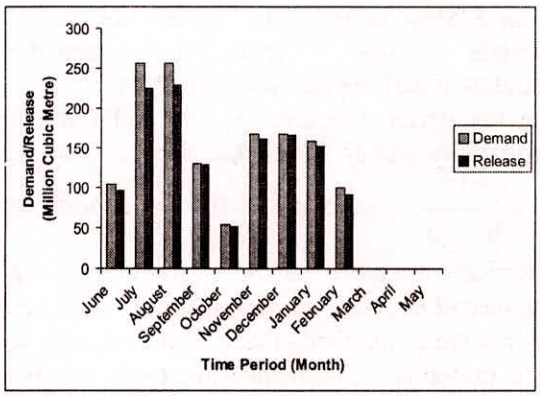

Fig. 4: Optimal Releases for the Single Reservoir System

The algorithm was used to optimize the releases to be made from reservoir in each period (t = 1 to 12) for four consecutive years (1976 to 1979) for the respective inflow values, and constant monthly demands. The optimum release values obtained from solution of model along with the respective demands in each time period (t = 1 to 12) for these four years obtained using CE method is shown in Figure 5.

The cross entropy algorithm requires the fine tuning of only one parameter, and the solution converges quickly. The method can be used for solving other combinatorial optimization problems in the water resources field.

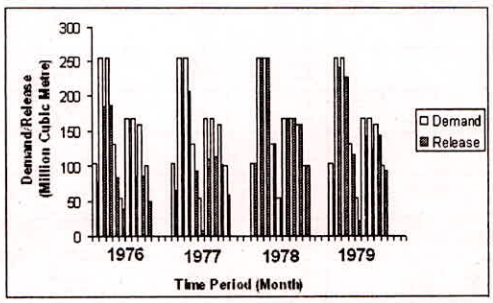

Fig. 5: Optimal Releases for Single Reservoir System for the years (1976–1979)

SUMMARY

The modeling concept of the evolutionary computing algorithm namely the Cross Entropy method, to a single reservoir system was illustrated. The sensitivity study of performance function to the smoothening parameter for the algorithm was also carried out. The cross entropy method is superior to other evolutionary computing algorithms as it requires fine tuning of only one parameter for arriving at a near optimal solution. The number of iterations required to generate an optimal solution are also less thereby making the algorithm easier for applying to various other combinatorial optimization problems.

REFERENCES

Jairaj, P.G. and Remya, A.R. (2007). "Water distribution network optimization using Cross Entropy Method" *Proceedings of National Conference on Hydraulics and Water Resources, HYDRO 2007*, Elite Publishing House Pvt. Ltd., New Delhi, 615–622.

Kapur, J.N. and Kesavan, J.K. (1992). *Entropy optimization principles with applications*, Academic Press Inc.

Kullback, S. and Leibler, R.A. (1951). "On information and sufficiency", *Ann. Math. Statics*, 22, 79–86.

Perelman, L. and Ostfeild, -A. (2007). "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems", *Engineering Optimization*, 39(4), 413–428.

Singh, P.V. (2000). "The entropy theory as a tool for modelling and decision making in Environmental and Water Resources", *Water SA*, 26(1), 1–11.

Yeh, W.W.G. (1985). "Reservoir management and operation models—a state of the art review", *Water Resources Research*, 21(2), 1787–1818.