**Jairaj, P. G.; Remya, A. R.**

# Reservoir Modelling by Entropy and Ant Colony Algorithms

*Proceedings of ninth International Conference on Hydro-Science and Engineering (ICHE 2010), IIT Madras, Chennai, India.*
*2 - 5 August 2010*

# RESERVOIR MODELLING BY ENTROPY AND ANT COLONY ALGORITHMS

Jairaj.P.G.[1] and Remya. A.R[2]

***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 a minimum thereby maximizing the returns. The present study aims in demonstrating the potential of heuristic evolutionary computing algorithms namely Ant Colony Optimization (ACO) and Cross Entropy (CE) methods for modelling of single reservoir systems. The model development to a single reservoir system operated for satisfying irrigation demands was done. The model was applied to a South Indian Reservoir system. The model development, sensitivity of model parameters in model solution was also explored*

***Keywords :*** *Optimization; Ant Colony; Cross Entropy; reservoir operation*

## INTRODUCTION

Reservoir modelling 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). 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.

An overview of evolutionary computing techniques for optimal reservoir operation is presented in Janga Reddy and Nagesh Kumar (2009). The Ant Colony Optimization (ACO) and Cross Entropy Algorithms are a subset of evolutionary computation, a generic population based meta-heuristic optimization algorithm. The present study aims at exploring the potential of two heuristic procedures Ant Colony Optimization (ACO) and Cross Entropy (CE) methods for modeling a single reservoir system. The concepts and details of model development and application to a real world field problem are discussed subsequently.

[1] Assistant Professor, Department of Civil Engineering, College of Engineering, Trivandrum, Thiruvananthapuram-695016, India. Email: dr_jairaj@rediffmail.com. (Member IAHR)
[2] Former Post Graduate Student, Department of Civil Engineering, College of Engineering, Trivandrum, Thiruvananthapuram-695016, India. Email: remyaark2@yahoo.com

## METHODOLOGY

The features of model development using the ant colony method and cross entropy method to a single reservoir system are as follows.

### Problem Definition

A single reservoir system with known demands at the reservoir is considered for model development. The objective function considered is to minimize the deviation of the system release in any period from the demand at that time period, subject to system constraints. The problem may be mathematically represented as :

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

Subject to: $$S_{t+1} = S_t + I_t - R_t - Ovf_t \quad \forall t \quad (2)$$

$$0 \le S_t \le K_A \quad \forall t \quad (3)$$

$$0 \le R_t \le D_t \quad \forall t \quad (4)$$

where $S_t$ : storage, $I_t$ : inflow, $R_t$ : release from reservoir to cater the known demands $D_t$, and $Ovf_t$ : overflow from reservoir in time period *t*, and $K_A$ : useful storage capacity of reservoir.

### ACO Model for the System

Since ACO method is a problem dependent application, before applying the ACO algorithm, the following preparation steps were done.

1. An appropriate representation of the problem as a graph or a similar structure easily covered by ants, which facilitates the incremental construction of possible solutions, using a probabilistic transition rule to move from one state *i* to a neighboring state *j*.
2. Selection of heuristic information $\eta_{ij}(t)$ that provides the problem specific knowledge to be used by the search process to move from node *i* to node *j*.
3. Defining an appropriate fitness function to be optimized for the problem
4. Selection of proper pheromone updating rules, which best suites the given problem.

The preliminary steps in ACO implementation are as follows.

The first step is to convert the problem into a graph format or as a set of nodes, which can be easily traced by ants. For this the storage volume was discretized. The heuristic information used in this problem was determined by considering the criterion as minimum deviation of release from the demand. The heuristic information for any time period is given by Eq. 5.

$$\eta_{ij}(t) = \frac{1}{(R_t - D_t)^2 + c} \quad (5)$$

where $R_t$ and $D_t$ represents release and demand values at any time period *t*. $\eta_{ij}$ represents the heuristic information when an ant moves from the $i^{th}$ node to $j^{th}$ node.

The fitness function is defined as a measure of the goodness of the generated solution according to the defined objective function. For this study the sum of total deviation of the release from demand was used as the fitness function, and represented by Eq. 6.

$$Fitness\ Function = \sum_{t=1}^{12} \left[ \frac{D_t - R_t}{D_{t\max}} \right] \tag{6}$$

***Algorithm***

After the preliminary steps the algorithm implementation was carried out as follows:

1. The iteration counter is incremented. The starting nodes for all the ants were determined; the ants select a particular node (which determines the time period and storage class for the reservoir) at random. The starting time period need not be the equal to one. It can be any value ranging from 1 (initial period) to thirteen (end of period).

2. Do the following steps until all ants have developed solutions or completed one full cycle. For moving from one node to another the ant increments the time period and checks the following conditions and takes the necessary steps.
   (a) If the next time period is equal to the initial time period then increment the ant counter and proceeds forward with the next ant.
   (b) If the above condition is not true then the ant uses the available heuristic information and the pheromone concentration in the path between the $i^{th}$ node and $j^{th}$ node or it selects a node at random from a set of feasible nodes based on the system constraints. This is achieved by the state transition rule Eq. 7.

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{u \in J^k(i)} \{[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}\}} & if\ j \in J^k(i) \\ 0 & if\ j \notin J^k(i) \end{cases} \tag{7}$$

where $\eta_{ij}(t)$ represents heuristic information about the problem i.e., the heuristic value of path $ij$ at time $t$ according to the measure of the objective function; $\tau_{ij}(t)$ represents the total pheromone deposited on path $ij$ at time $t$, $J^k(i)$ represents the allowable moves for ant $k$ from node $i$; $\alpha$ and $\beta$ are parameters that determine the relative importance of the pheromone trial with respect to the heuristic information. The state transition rule is as follows: the next node $j$ that an ant $k$ chooses to go is given as,

$$J = \begin{cases} \max\limits_{u \in J^k(i)} [\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta} & if\ q \le q_0 \\ J & if\ q > q_0 \end{cases} \tag{8}$$

where $q$ is a random number uniformly distributed in [0,1]; $q_0$ is a tunable parameter ($0 \le q_0 \le 1$); $J \in J^k(i)$ is a node randomly selected according to the probability distribution given by Eq. 7.

3. In this step the fitness function of the ants were determined. For this the release value in each time period was calculated from the storage values prescribed by the nodes along the path traversed by the ant. If at any time period $t$ the ant travels from storage class $S_t$ to a storage class $S_{t+1}$, and then the release at time period $t$ is calculated using Eq. 2. With the release values the fitness function was calculated using Eq. 6.

4. Among the ants, the one which produces the maximum value for benefit function was chosen as the iteration best ant. The pheromone concentrations along the route or links (connecting 2 nodes) followed by the best ant are updated using local pheromone updation rule Eq. 9.

$$\tau_{ij}(t) \xleftarrow{step} (1-\psi).\tau_{ij}(t-1) + \psi.\tau_0 \tag{9}$$

where $\tau_0$ the initial value of pheromone; $\psi$ a tunable parameter. Local updating is very useful to avoid premature convergence of the solution and helps in exploring the new search space, for the problems where the starting node is fixed.

5. Check whether the iteration counter has reached the maximum value (stopping criteria). Then stop the algorithm and the best ant route so far obtained will form the solution set, otherwise move to step (6).

6. Check whether the iteration counter is a multiple of five. If yes, then a global pheromone updating is carried out for the ant route traversed by the global best ant (ant which gives the best solution so far in the algorithm) using Eq. 10.

$$\tau_{ij}(t) \xleftarrow{iteration} (1-\rho).\tau_{ij}(t-1) + \rho.\Delta\tau_{ij} \tag{10}$$

where $\rho$ varies from zero to one is a persistence parameter that controls the pheromone decay; $\Delta\tau_{ij}$ is increment in pheromone trial, given by Eq. 11 as

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{F_{gb}} & if\ (i,j) \in global\ best\ tour \\ 0 & otherwise \end{cases} \tag{11}$$

where, $F_{gb}$ is fitness function corresponding to global best tour within all the past iterations.

Various parameters of the algorithm like the initial pheromone concentration, number of iterations, number of ants and pheromone-updating parameters etc. were fixed after sensitivity studies. Implementation of the algorithm is done using program developed in C language.

**Cross-Entropy Model of System**

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 $Q = \{q_1, q_2, q_3, \ldots\ldots\ldots\ldots\ q_n\}$ 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 $X = \{x_1, x_2, x_3, \ldots\ldots\ldots\ldots\ x_n\}$ and compute moments based on these observations. To derive the distribution $P = \{p_1, p_2, p_3, \ldots\ldots\ldots\ldots\ p_n\}$ 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 Eq. 12.

$$\text{Minimize } D(P,Q) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i} \qquad (12)$$

***Cross Entropy algorithm***

The Cross Entropy (CE) method is an 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 algorithm steps 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, for all the time periods.

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 $t_i$,

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 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 Eqs. 2 to 4. 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 $\rho_c$) of the top best performing vectors for updating the probability vector $p_{t,j}$ to the probability vector $p_{t+1,j}$. $\rho_c$ 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 Eq. 13.

$$p_{t+1,i} = \frac{B_{t,i}}{TB_t} \tag{13}$$

where $p_{t+1,,j}$, is the probability of success of node $i$, $B_{t,i}$ is the number of times node $i$ was chosen out of the best 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 $\alpha_c$ is used. The value of smoothing parameter $\alpha_c$ is determined based on sensitivity analysis. The probability is modified as in Eq. 14.

$$p_{t+1,i} \leftarrow \alpha_c \; p_{t+1,i} + (1-\alpha_c)\, p_{t,i} \tag{14}$$

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 values.

.

The coding of the model for the problem was done in C language. The results of the sensitivity study, the model solution are discussed subsequently.

## 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 time period 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 considered in the study. More specific details pertaining to the study are available in Remya (2007).

## RESULTS AND DISCUSSION

### ACO Method

As explained in the previous section a large number of tunable parameters are involved in this algorithm and for the best performance of the algorithm they need to be fine-tuned. The results of the sensitivity study are discussed below.

***Sensitivity of parameters***

To start with the following values were assigned to parameters: $\alpha$ = 2, Number of ants = 15, Number of Iterations = 25, $q_0$ = 0.5, $\tau_0$ = 1, $\psi$ = 0.5, $\rho$ = 0.5 and the value of the parameter $\beta$ was varied. Based on model solution for values of β given in Table 1, $\beta$ =1 was selected.

**Table 1. Influence of Parameter $\beta$ on ACO Model Performance**

| $\beta$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 1 | 0.824 | 0.767 | 0.909 |
| 2 | 0.855 | 0.824 | 0.909 |
| 3 | 0.856 | 0.824 | 0.909 |

The parameter $\alpha$, represents the weightage to be given to the pheromone concentration in determining the probability of node selection. From the results of analysis in Table 2, it can be seen that for α = 2, with specified $\beta$ = 1, gives the best performance.

**Table 2. Influence of Parameter $\alpha$ on ACO Model Performance**

| $\alpha$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 1 | 0.828 | 0.813 | 0.909 |
| 2 | 0.824 | 0.767 | 0.909 |
| 3 | 0.863 | 0.846 | 0.909 |

The sensitivity of system to the parameter $q_0$ determines whether to explore or exploit the solution space. The values of other parameters used are $\alpha$ = 2, $\beta = 1$, and all other parameters assigned above, the model performance with different $q_0$ values are given in Table 3. The best value of $q_0$ is found to be equal to 0.6.

**Table 3. ACO Model Performance for Parameter $q_0$**

| $q_0$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 0.5 | 0.824 | 0.767 | 0.909 |
| 0.6 | 0.814 | 0.720 | 0.884 |
| 0.7 | 0.854 | 0.798 | 0.884 |
| 0.8 | 0.868 | 0.824. | 0.884 |

The local pheromone updating parameter $\psi$ was then determined with the other parameters as $\alpha$ = 2, $\beta = 1$, $q_0$ = 0.6, and all other parameters initially assigned. The model performance to different $\psi$ values are provided in Table. 4, and the value $\psi$ = 0.6 was selected.

**Table 4. ACO Model Performance for Parameter $\psi$**

| $\psi$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 0.4 | 0.824 | 0.743 | 0.884 |
| 0.5 | 0.814 | 0.720 | 0.884 |
| 0.6 | 0.798 | 0.660 | 0.884 |
| 0.7 | 0.788 | 0.680 | 0.859 |
| 0.8 | 0.762 | 0.743 | 0.842 |

The model performance to the tunable parameter $\tau_0$ with the best parameter values of $\alpha$ = 2, $\beta$ =

*1*, $\psi = 0.6$, $q_0 = 0.6$, and all other parameters assigned above, the model performance evaluated is given in Table. 5. The best performance was obtained for $\tau_0 = 1$, and selected.

**Table 5. Influence of Parameter $\tau_0$ on ACO Model Performance**

| $\tau_0$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 1 | 0.798 | 0.660 | 0.884 |
| 2 | 0.813 | 0.743 | 0.884 |
| 3 | 0.822 | 0.743 | 0.859 |

The model performance to different $\rho$ values with specified values of $\alpha = 2$, $\beta = 1$, $\psi = 0.6$, $q_0 = 0.6$, $\tau_0 = 1$ and all other parameters assumed above is shown in Table 6. The best solution is obtained for a $\rho$ value equal to 0.7.

**Table 6. Influence of Parameter $\rho$ on ACO Model Performance**

| $\rho$ | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 0.4 | 0.801 | 0.694 | 0.884 |
| 0.5 | 0.798 | 0.660 | 0.884 |
| 0.6 | 0.767 | 0.625 | 0.884 |
| 0.7 | 0.743 | 0.618 | 0.824 |
| 0.8 | 0.746 | 0.682 | 0.798 |
| 0.9 | 0.750 | 0.719 | 0.798 |

The model performance with varied number of ants and all other parameters assigned above is shown in Table 7, and the best performance is reached when the number of ants is 45. Similarly, it was seen that an increase in the number of iteration above 25, no further improvement of the fitness function value is observed.

**Table 7. Influence of Number of Ants on ACO Model Performance**

| No:of Ants | Mean | Best Solution | Worst Solution |
|---|---|---|---|
| 15 | 0.743 | 0.618 | 0.824 |
| 25 | 0.732 | 0.561 | 0.884 |
| 35 | 0.726 | 0.441 | 0.884 |
| 45 | 0.686 | 0.399 | 0.884 |
| 55 | 0.678 | 0.399 | 0.909 |
| 75 | 0.672 | 0.399 | 0.909 |
| 100 | 0.672 | 0.399 | 0.909 |

Having done the sensitivity analysis, algorithm parameters for the best solution: $\beta = 1$, $\alpha = 2$, $q_0 = 0.6$, $\tau_0 = 1$, $\psi = 0.6$, $\rho = 0.7$, Number of Ants = 45, Number of Iterations = 25 were arrived at.

***Model solution***

The values of parameters from sensitivity study were used to find the optimal reservoir operation policy for the system. The model solution gave the optimal solution of 0.399, for the set of inflows and demands for the system. The algorithm was used to optimize the releases in each period ($t$ = 1 to 12) for four consecutive years (1976 to 1979) using the corresponding inflow values and constant demand values for the system. The optimal release values and the demands for twelve monthly periods considered in the study is shown in Figure 1. Even though the algorithm gives optimal solution, it requires the fine tuning of so many parameters.

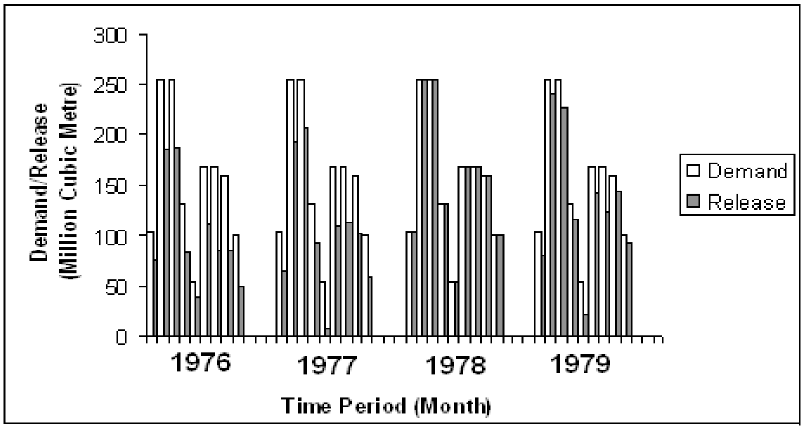


**Fig. 1. Optimal Releases using ACO Method**

**CE Method**

The CE model discussed above was run for the system selected in the study. At the end of iteration, all those decision variables having probabilities equal to one was combined to produce 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. As explained in the CE algorithm, a smoothing parameter ($\alpha_c$) is used to avoid premature convergence of the algorithm. In order to study the significance of the smoothing parameter $\alpha_{c,}$ a sensitivity study was conducted by varying the value of $\alpha_c$ in the range 20% to 50%. The influence of the parameter $\alpha_c$ on the model solution and the number of iteration to attain the optimum solution are given in Table 7. 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 ($\alpha_c$) values and vice versa. An optimal value of $\alpha_c$ = 0.4, was selected.

**Table 7. Influence of Parameter $\alpha_c$ on Model Solution**

| $\propto_c$ (%) | No. of Iterations | Optimal Solution |
|---|---|---|
| 20 | 23 | 0.399 |
| 30 | 19 | 0.399 |
| 40 | 13 | 0.399 |
| 50 | 9 | 0.424 |

***Model Solution***

For the system selected in the study, the near optimal solution obtained is 0.399, for the specified inflw and demand. 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 is shown in Figure 2. The cross entropy algorithm requires the fine tuning of only one parameter, and the solution converges quickly.

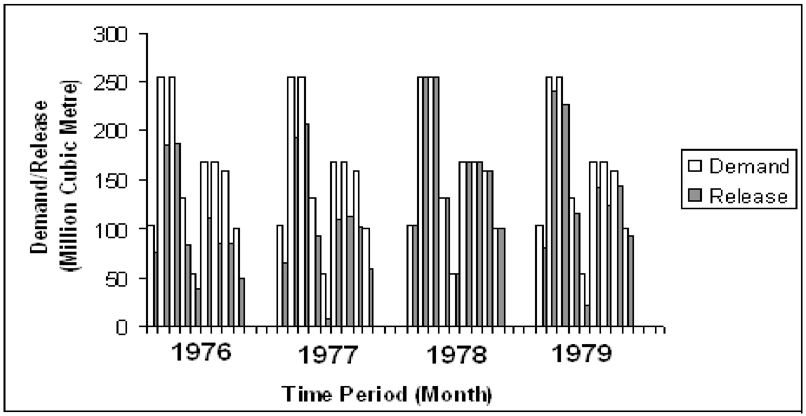


**Fig. 2. Optimal Releases using CE Method**

## CONCLUSIONS

The modeling concept using two heuristic computing algorithms namely Ant Colony method and the Cross Entropy method, to a single reservoir system was illustrated. The model was applied to an existing single reservoir system in South India. The sensitivity study of model parameters in model performance function were also carried out. Both the algorithms performed equally well, and gave the same result. While the ACO method requires the fine tuning of large number of parameters in arriving at the solution, the CE method has got only one parameter and hence much simpler in implementation. The number of iterations required to generate an optimal solution are also less thereby making the CE algorithm easier for application to various combinatorial optimization problems in water resources field.

## REFERENCES

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

Janga Reddy, M. and Nageskh Kumar, D. 2009. Evolutionary Computing in Optimal Reservoir Operation *International Conference WEES 2009*, Allied Publishers, New Delhi, 999-1010.

Remya, A.R. 2007. Evolutionary algorithms for modelling Water Distribution Networks and Reservoir Systems. Master's Thesis, University of Kerala.

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