Multipurpose Reservoir Operation Using Particle Swarm Optimization

D. Nagesh Kumar[1] and M. Janga Reddy[2]

Abstract: This paper presents an efficient and reliable swarm intelligence-based approach, namely elitist-mutated particle swarm optimization (EMPSO) technique, to derive reservoir operation policies for multipurpose reservoir systems. Particle swarm optimizers are inherently distributed algorithms, in which the solution for a problem emerges from the interactions between many simple individuals called particles. In this study the standard particle swarm optimization (PSO) algorithm is further improved by incorporating a new strategic mechanism called *elitist-mutation* to improve its performance. The proposed approach is first tested on a hypothetical multireservoir system, used by earlier researchers. EMPSO showed promising results, when compared with other techniques. To show practical utility, EMPSO is then applied to a realistic case study, the Bhadra reservoir system in India, which serves multiple purposes, namely irrigation and hydropower generation. To handle multiple objectives of the problem, a weighted approach is adopted. The results obtained demonstrate that EMPSO is consistently performing better than the standard PSO and genetic algorithm techniques. It is seen that EMPSO is yielding better quality solutions with less number of function evaluations.

DOI: 10.1061/(ASCE)0733-9496(2007)133:3(192)

CE Database subject headings: Reservoir operation; Optimization; Irrigation; Hydroelectric power generation.

Introduction

In reservoir operation problems, to achieve the best possible performance of the system, decisions need to be taken on releases and storages over a period of time considering the variations in inflows and demands. In the past, various researchers applied different kinds of mathematical programming techniques like linear programming, dynamic programming, nonlinear programming (NLP), etc., to solve such reservoir operation problems. An extensive review of these techniques can be found in Loucks et al. (1981), Yakowitz (1982), Yeh (1985), and Wurbs (1993). But as far as reservoir operation is concerned, no standard algorithm is available, as each problem has its own individual physical and operational characteristics (Yeh 1985).

In the case of multipurpose reservoir operation, the goals are more complex than for single purpose reservoir operation and often involve various problems such as insufficient inflows and larger demands. In order to achieve the best possible performance of such a reservoir system, a model should be formulated as close to reality as possible. In this process, the model is expected to solve problems having nonlinearities and nonconvexities in their domain. For example, a typical hydropower production function is complex, with nonlinear relationships in objectives and constraints. So the linear programming methods cannot be used. The dynamic programming approach faces the additional problem of the curse of dimensionality, whereas the nonlinear programming methods have the limitations of slow rate of convergence, requiring large amounts of computational storage and time compared with other methods (Yeh 1985). Also, often NLP results in local optimal solutions.

In spite of development of many conventional techniques for optimization, each of these techniques has its own limitations. To overcome those limitations, recently metaheuristic techniques are being used for optimization. By using these techniques, the given problem can be represented more realistically. These also provide ease in handling the nonlinear and nonconvex relationships of the formulated model. Genetic algorithms (GAs) (Goldberg 1989) and particle swarm optimization (PSO) (Eberhart and Kennedy 1995) are some of the techniques in this category. These evolutionary algorithms search from a population of points, so there is a greater possibility to cover the whole search space and reaching the global optimum.

GA is one of the population-based search techniques, which works on the concept of "survival of the fittest" (Goldberg 1989). In the field of water resources, in earlier studies, few applications of the GA technique to derive reservoir operating policies have been reported (Oliveira and Loucks 1997; Chang and Chen 1998; Wardlaw and Sharif 1999) and they illustrated the utility of evolutionary techniques for reservoir operation problems. Though GA has many advantages over conventional methods, it also has some drawbacks, such as slow rate of convergence and requiring a large number of simulations to arrive at an optimum solution. Recently researchers reported that PSO is showing some improvement on these issues, which use the local and global search capabilities of the algorithm, to find better quality solutions with less computational time (Salman et al. 2002).

To improve the performance of the standard PSO, in this study, a new strategic mechanism called *elitist mutation* is incorporated

[1]Associate Professor, Dept. of Civil Engineering, Indian Institute of Science, Bangalore 560 012, India. E-mail: nagesh@civil.iisc.ernet.in

[2]Research Scholar, Dept. of Civil Engineering, Indian Institute of Science, Bangalore 560 012, India. E-mail: mjreddy@civil.iisc.ernet.in

Note. Discussion open until October 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on January 19, 2005; approved on April 3, 2006. This paper is part of the ***Journal of Water Resources Planning and Management***, Vol. 133, No. 3, May 1, 2007. ©ASCE, ISSN 0733-9496/2007/3-192–201/$25.00.

in to the algorithm. Using this mechanism, elitist-mutated particle swarm optimization (EMPSO) is propounded to derive operational policies for multipurpose reservoir systems. To test the usefulness of EMPSO for obtaining reservoir operating polices, first it is tested for a hypothetical reservoir system, which was used by earlier researchers. To show its practical utility, EMPSO is then applied to an existing reservoir system, namely the Bhadra reservoir system in Karnataka State, India. To compare the performance of EMPSO, the previous two case studies are also solved using standard PSO and GA techniques and the results are discussed.

Particle Swarm Optimization

The term "swarm intelligence" is used to describe algorithms and distributed problem solvers, which was inspired by the collective behavior of insect colonies and other animal societies. Under this prism, PSO is a swarm intelligence method for solving optimization problems. Particle swarm optimization, originally proposed by Eberhart and Kennedy (1995), is a population-based heuristic search technique, inspired by social behavior of bird flocking. PSO shares many similarities with evolutionary computation techniques such as GA. PSOs are initialized with a population of random solutions and search for optima by updating generations. However, in contrast to methods like GAs, in basic PSO, no operators inspired by natural evolution are applied to extract a new generation of candidate solutions. Instead, PSO relies on the exchange of information between individuals (particles) of the population (swarm). In effect, each particle adjusts its trajectory towards its own previous best position and towards the current best position attained by any other member in its neighborhood (Parsopoulos and Vrahatis 2002). By using the PSO technique, it will be easy to handle nonlinearity and nonconvexity of the problem domain; the search does not depend on the initial population; and overcomes problems of local optima that are common in some conventional nonlinear optimization techniques (Salman et al. 2002). The PSO technique has successful demonstration for numerical optimization and reported that PSO is an attractive alternative for global optimization problems (Parsopoulos and Vrahatis 2002). It has also been found that PSO is outperforming other heuristic search methods such as GAs (Abido 2002; Salman et al. 2002).

PSO Algorithm

If the search space is D-dimensional, the ith individual (*particle*) of the population (swarm) can be represented by a D-dimensional vector, $X_i=(x_{i1},x_{i2},\ldots,x_{iD})^T$. The *velocity* (position change) of this particle can be represented by another D-dimensional vector $V_i=(v_{i1},v_{i2},\ldots,v_{iD})^T$. The best previously visited position of the ith particle is denoted as $P_i=(p_{i1},p_{i2},\ldots,p_{iD})^T$. Defining g as the index of the best particle in the swarm (i.e., the gth particle is the best), and superscripts denoting the iteration number, the swarm is manipulated using the following two equations:

$$v_{id}^{n+1} = \chi[\omega v_{id}^{n} + c_1 r_1^n (p_{id}^n - x_{id}^n) + c_2 r_2^n (p_{gd}^n - x_{id}^n)] \quad (1)$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1} \quad (2)$$

where $d=1,2,\ldots,D$; $i=1,2,\ldots,N$; N=size of the swarm; χ=constriction coefficient; ω=inertial weight; c_1 and c_2=positive constant parameters called acceleration coefficients; r_1,r_2=random numbers, uniformly distributed in [0,1]; and n=iteration number.

```
Begin (Initialization)
        Generate initial swarm X(0)
        Generate initial velocities V(0)
 End
 Set   n = 0 (n is the generation number)
  Repeat
     Begin
     Compute fitness value for each individual of swarm
     End
     Compute PBest(n) and GBest
     Begin (Perform PSO operations)
         Compute V(n +1) from Eq. (1)
         Compute X(n +1) from Eq. (2)
         Perform elitist mutation
      End
         Set   n = n + 1
  Until termination criteria satisfied
```

Fig. 1. Pseudocode of elitist-mutated particle swarm optimization algorithm

The general effect of Eq. (1) is that each particle oscillates in the search space, between its previous best position and the best position of its best neighbor, attempting to find the new best point in its trajectory. Proper fine-tuning of the parameters c_1 and c_2, in Eq. (1), may result in faster convergence of the algorithm, and alleviation of the problem of local minima. The role of inertial weight ω is to control the impact of the previous velocities on the current one. A large inertial weight facilitates global exploration (searching new areas), while a small weight tends to facilitate local exploration. Hence selection of a suitable value for the inertial weight ω usually helps in reduction of the number of iterations required to locate the optimum solution (Parsopoulos and Vrahatis 2002).

In basic PSO algorithms, the constriction factor χ is taken as 1. In later developments of the PSO technique, it is observed that if the particle's velocity is allowed to change without bounds, the swarm will never converge to an optimum, since subsequent oscillations of the particle will be larger. To control the changes in velocity, the constriction factor χ was introduced into the PSO algorithm in Clerc (1999), to force the swarm to converge better. By using Eqs. (1) and (2) the velocities and particle positions are updated repeatedly over the iterations, to get at the optimal solution. Even though PSO is faster in finding quality solutions, compared to other evolutionary computation techniques, it faces some difficulty in obtaining better quality solutions while exploring complex functions. It may face premature convergence and suffer from poor fine-tuning capability of the final solution.

Elitist-Mutated Particle Swarm Optimization

To overcome the problems explained earlier and to improve the performance of the PSO technique, an elitist-mutation strategy is introduced into the algorithm in this study. This elitist process replaces the worst particle solutions by the best solution among the swarm, after performing mutation mechanism on the best particle. This process of random perturbation tries to improve the solution, by maintaining diversity in the population, and explores the new regions in the whole search space. This strategy replaces the position vectors of a predefined number of least ranked particles in the swarm with mutated positions of the global best particle in each iteration. Pseudocode of the EMPSO algorithm is presented in Fig. 1, describing the main steps involved. In the EMPSO methodology, the elitist-mutation step is computed as follows. First the fitness function of particles is sorted in ascending order and the index number for the respective particles is obtained, then elitist-mutation is performed on worst fitness particles in the swarm. Let NM_{max}=number of particles to be elitist-mutated; g=index of global best particle; p_{em}=probability of mutation; ASF=index of sorted population; rand=uniformly distributed random number $U(0,1)$; rand$n$=Gaussian random number $N(0,1)$; and $VR[d]$=range of decision variable d

For i=1 to NM_{max}

  l=ASF[i]

 For d=1 to *dim*

  if (rand$<p_{em}$)

   $X[l][d]=P[g][d]+0.1^{*}\ VR[d]^{*}$rand$n$

  else

   $X[l][d]=P[g][d]$

If the mutated value exceeds the bounds, then it is limited to the upper or lower bound. During this elitist-mutation step, the velocity vector of the particle is unchanged. The step-by-step procedure of EMPSO technique for the reservoir operation problem is explained in the following.

1. *Initialization of position and velocity vectors of the swarm:* For a reservoir operation problem, the search space will be equal to the total number of release combinations. Each decision variable represents a parameter to be optimized in the model. The initial positions of all particles have to be generated randomly within the limits specified for each decision variable. For the reservoir operation problem, the initial searching points are the initial random release values. Each particle is initialized with random position vectors, $X_i(0)$, and random velocity vectors, $V_i(0)$, for all $i=1,2,\ldots,N$.
2. *Initial evaluation of fitness function:* The fitness of each particle is evaluated using the objective function of the problem. Then the best value out of all the fitness values is searched. Two "best" values are defined, the global best and the personal best. The global best P_g is the highest fitness value among all the particles in an entire run (best solution so far), and the personal best P_i is the highest fitness value of a specific particle up to the current iteration.
3. *Modification of each searching point:* Using the global best and the local best of each particle up to the current iteration, the searching point of each particle is modified using the change in velocity as given by Eq. (1) and the positions of the particles are updated using Eq. (2).
4. *Evaluation of fitness function:* After modification of the particle positions, the fitness of each particle is evaluated using the objective function of the problem.
5. *Perform elitist mutation:* The fitness values are ranked in the order of their fitness values. For the selected number of least ranked particles (which is equal to the number of particles to perform elitist mutation in iteration), the respective particle position vectors are replaced with the new position vectors obtained after performing variable-wise mutation on the best particle position vector.

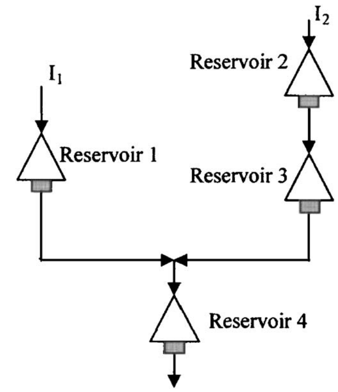

Fig. 2. Four-reservoir system considered for Case Study I [adapted from Larson (1968)]

6. *Updating the global and the local bests:* The P_g and P_i values have to be updated according to the new fitness values. If the best fitness value of a particular particle in the swarm is better than the current P_g, then P_g is to be changed to the value of the searching point of the corresponding particle contributing to this best fitness value. Similarly the local best of other particles in the population should be changed accordingly if the present fitness function value is better than the previous.
7. *Termination criteria:* Repeat steps (3) to (6) until either the pre-set maximum number of iterations is reached or no significant improvement is observed over a prespecified number of iterations.

To compare the performance of EMPSO, the real coded GA developed by Kanpur Genetic Algorithms Laboratory, IIT Kanpur, India (KanGAL, http://www.iitk.ac.in/kangal/soft.htm) is used in this study. The performance of GA is tested for benchmark problems and is performing quite well as reported in the literature (Deb 2000). However, source code of the GA program is modified to suit the present problem and then used to run the model with the same formulation as that used for EMPSO.

The next section presents application of the EMPSO technique for two case studies, one hypothetical and one real life reservoir system.

Case Study I

In this study, a hypothetical four-reservoir system (Larson 1968) is used to test the efficiency of the EMPSO technique, to derive operating policies for the multireservoir system. The four-reservoir system considered for this case is shown in Fig. 2. For this hypothetical reservoir system in the past, Larson (1968) applied dynamic programming with successive approximation (DPSA) algorithm, Heidari et al. (1971) applied discrete differential dynamic programming (DDDP), and Kumar and Singh (2003) applied folded dynamic programming (FDP) to evaluate the performance of their respective proposed techniques. Presently, the EMPSO technique is applied to the same hypothetical system with the same objective function, constraints, and related data as given in Larson (1968). The same reservoir system is also solved using standard PSO and GA techniques to facilitate comparison.

For the EMPSO model, the parameters chosen after thorough sensitivity analyses are constriction coefficient $(\chi)=0.9$; inertia weight $\omega=1.0$; PSO constant parameters $c_1=1.0$, $c_2=0.5$;

Table 1. Performance of Six Models for the Hypothetical Four-Reservoir System

Model	Source	Best fitness value	Number of function evaluations taken
DPSA	Larson (1968)	**401.3**	—
DDDP	Heidari et al. (1971)	**401.3**	—
FDP	Kumar and Singh (2003)	399.06	—
PSO[a]	Presently applied	399.7	748,000
GA[a]	Presently applied	**401.3**	2,279,500
EMPSO[a]	Presently applied	**401.3**	**325,400**

[a]The best solution obtained for ten trial runs with the respective algorithm.

probability of elitist mutation (P_{em})=0.2; size of the swarm N=2,000; maximum number of iterations=500; and the size of elitist-mutated particles=38. For this hypothetical reservoir system, elitist mutation is performed for all the iterations. To run the standard PSO model, the same parameters are used except the elitist-mutation step. For the GA model, the parameters used are as follows: crossover probability (P_c)=0.9 and a variable-wise mutation probability (P_m)=0.2; distribution index for simulated binary crossover=10 and that for mutation operator=100; size of the population=500; and number of generations=5,000.

To check the performance of the proposed technique, the models are run for ten random trials. The performance of six models for the hypothetical four-reservoir system is shown in Table 1, which compares the best fitness value obtained by the earlier techniques and also with presently applied models with respect to the number of function evaluations taken for that solution. The proposed EMPSO technique yielded best solution as 401.3, which is also equal to the global optimal solution reported in the literature. For ten trial runs, the mean fitness values obtained and number of function evaluations taken are 401.18 (447,830), 396.16 (683,400), and 400.84 (2,408,750), respectively, for EMPSO, standard PSO, and GA. These results clearly show that the EMPSO technique is giving the best fitness value, with least number of function evaluations as compared to other models.

Case Study II

To evaluate the practical utility of the proposed technique, an existing reservoir system, namely the Bhadra reservoir system, is taken up as a case study for developing optimal reservoir operation policies. The Bhadra Dam is located at latitude 13°42′ N and longitude 75°38′20″ E in Chikmagalur District of Karnataka State, India. The average annual rainfall in the basin is 2,320 mm with 90% rainfall occurring during the monsoon period (June–November). The Bhadra reservoir is a multipurpose reservoir for irrigation and hydropower generation, in addition to mandatory releases, to maintain water quality downstream.

The reservoir has an active storage capacity of 1,784 $\times 10^6$ m^3 and the water-spread area at full reservoir level is 117 km^2. The reservoir provides water for irrigation of 6,367 and 87,512 ha under left and right bank canals, respectively. Also under this project there are three sets of turbines, one set each on the left bank canal and the right bank canal and the other set at the river bed level of the dam, for generating hydropower. Fig. 3 shows the schematic diagram of the Bhadra reservoir system. The irrigated area is spread over the districts of Chitradurga, Shimoga, Chikmagalur, and Bellary and comprises predominantly of red

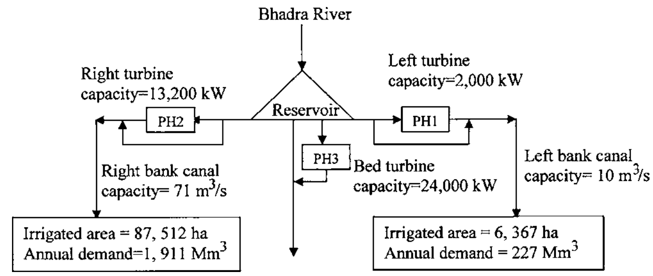

Fig. 3. Schematic diagram of the Bhadra reservoir system

loamy soil, except in some portions of the right bank canal area, which also consists of black cotton soil. The major crops grown in the command area are paddy, sugarcane, permanent garden, and semidry crops.

The data of monthly inflows and other details were collected from the Water Resources Development Organization, Bangalore covering a period of 69 years (from 1930–1931 to 1998–1999). The monthly crop water requirements were calculated using the FAO Penman–Monteith method. In addition to irrigation and hydropower, a mandatory release of 9×10^6 m^3/month to downstream is to be made in each month to meet the water quality requirements.

Mathematical Model Formulation

The two objectives considered in the model are minimization of irrigation deficits and maximization of hydropower generation. These two are conflicting objectives, as one tries to minimize the irrigation deficits, requiring more water to be released to satisfy irrigation demands and the other tries to maximize hydropower production, which requires higher level of storage to be maintained in the reservoir to produce more power. The two competing objectives of the system are expressed as,

Minimize sum of squared deficits for irrigation annually

$$\text{Minimize SQDV} = \sum_{t=1}^{12} (D_{l,t} - Q_{l,t})^2 + \sum_{t=1}^{12} (D_{r,t} - Q_{r,t})^2 \qquad (3)$$

where SQDV=squared deviations of irrigation demand and releases; $D_{l,t}$ and $D_{r,t}$=irrigation demands for the left bank canal and right bank canal command areas, respectively, in period t (10^6 m^3); and $Q_{l,t}$ and $Q_{r,t}$=releases made into the left and right bank canals, respectively, in period t (10^6 m^3).

Maximize annual energy production

$$\text{Maximize } E = \sum_{t=1}^{12} p(Q_{l,t}H_{l,t} + Q_{r,t}H_{r,t} + Q_{b,t}H_{b,t}) \qquad (4)$$

where E=energy produced ($\times 10^6$ kW h); p=power production coefficient; $Q_{b,t}$=release made to riverbed turbine in period t ($10^6 m^3$); and $H_{l,t}$, $H_{r,t}$, $H_{b,t}$=net heads available (m) to left bank, right bank, and bed turbines, respectively, in period t.

Subject to the following constraints:

- *Mass balance equation*

$$S_{t+1} = S_t + I_t - (Q_{l,t} + Q_{r,t} + Q_{b,t} + \text{EVP}_t + \text{OVF}_t) \quad \text{for all } t = 1,2,\ldots,12 \qquad (5)$$

where S_t=reservoir storage at the beginning of period t (10^6 m^3); I_t=inflow to the reservoir during period t (10^6 m^3); EVP_t=evaporation losses during period t (10^6 m^3) (a nonlinear function of initial and final storages of period t); and OVF_t=overflow from the reservoir in period t (10^6 m^3).

- *Storage bounds*

$$S_{\min} \leqslant S_t \leqslant S_{\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{6}$$

where $S_{\min}$ and $S_{\max}$=minimum and maximum storage limits of the reservoir.

- *Maximum power production limits*

$$pQ_{l,t}H_{l,t} \leqslant E_{l,\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{7}$$

$$pQ_{r,t}H_{r,t} \leqslant E_{r,\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{8}$$

$$pQ_{b,t}H_{b,t} \leqslant E_{b,\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{9}$$

where $E_{l,\max}$, $E_{r,\max}$, and $E_{b,\max}$=maximum amount of power (10^6 kW h) that can be produced (turbine capacity) from left, right, and bed level turbines, respectively.

- *Canal capacity limits*

$$Q_{l,t} \leqslant C_{l,\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{10}$$

$$Q_{r,t} \leqslant C_{r,\max} \quad \text{for all } t = 1,2, \ldots ,12 \tag{11}$$

where $C_{l,\max}$ and $C_{r,\max}$=maximum canal carrying capacity limits of the left and right bank canals, respectively (10^6 m^3/month).

- *Irrigation demands*

$$D1_{\min,t} \leqslant Q_{l,t} \leqslant D1_{\max,t} \quad \text{for all } t = 1,2, \ldots ,12 \tag{12}$$

$$D2_{\min,t} \leqslant Q_{r,t} \leqslant D2_{\max,t} \quad \text{for all } t = 1,2, \ldots ,12 \tag{13}$$

where $D1_{\min,t}$ and $D1_{\max,t}$=minimum and maximum irrigation demands for left bank canal, respectively; and $D2_{\min,t}$ and $D2_{\max,t}$=minimum and maximum irrigation demands for right bank canal, respectively, in time period t.

- *Water quality requirements*

$$Q_{b,t} \geqslant \mathrm{MDT}_t \quad \text{for all } t = 1,2, \ldots ,12 \tag{14}$$

where MDT_t=minimum release to meet downstream water quality requirement (10^6 m^3).

- *Steady state storage constraint*

$$S_{13} = S_1 \tag{15}$$

This constraint is required to bring the steady state condition for the reservoir storage, i.e., storage at the end of a year is equal to the initial storage at the beginning of that year.

The reservoir is mainly meant for irrigation, and so priority is given to irrigation. After meeting the irrigation demands, power production is to be maximized. To handle these multiple objectives, in this study a weighted approach is adopted. By giving a larger weight to irrigation and a smaller weight to hydropower generation, it is possible to satisfy the prescribed priorities. In this model a normalized squared deficit is used for irrigation (instead of simpler squared deficits) with the aim to evenly distribute the deficits throughout the season in the case of occurrence of deficits, by giving equal priority to larger and smaller magnitudes of demands of right bank and left bank canals, respectively. To bring both the objectives into same units, the hydropower objective is nondimensionalized. So the final fitness function for the model is as follows:

$$F = k_1 \sum_{t=1}^{12} \left[\left(\frac{D_{l,t} - Q_{l,t}}{D_{l,t}} \right)^2 + \left(\frac{D_{r,t} - Q_{r,t}}{D_{r,t}} \right)^2 \right] + k_2 \sum_{t=1}^{12} \left[\frac{E_{l,\max} - p(Q_{l,t}H_{l,t})}{E_{l,\max}} + \frac{E_{r,\max} - p(Q_{r,t}H_{r,t})}{E_{r,\max}} + \frac{E_{b,\max} - p(Q_{b,t}H_{b,t})}{E_{b,\max}} \right] \tag{16}$$

where k_1 and k_2=constant weights to be chosen based on priority.

So the final model is to minimize F [Eq. (16)] duly satisfying the constraints on reservoir continuity, storage bounds, turbine capacity, canal capacity, irrigation demands, mandatory releases, and steady state storage constraint, i.e., Eqs. (5)–(15).

Results

The EMPSO technique is applied to the model described in the earlier section to obtain the operating policies for the multipurpose reservoir system. In this study to handle the constraints of the problem, simulation and evaluation approach is used, in which at each generation, the decision variables are evaluated for the present solution and then the bounds are checked. If there is any violation in satisfying the constraints, then a penalty is applied by choosing a suitable penalty coefficient. The constant weights to be given for irrigation and hydropower can be decided by the policy maker. In this study, since the irrigation release is having higher priority over hydropower production, for demonstration purpose the constant weights of the fitness function are chosen as k_1=100 and k_2=−1. Before applying the EMPSO technique, the parameters required should be decided. The total number of decision variables of the model is 36 (number of time periods=12 and number of decisions in each period=3), which is equal to the dimension of the problem. The termination criterion is, whether it reaches the maximum number of iterations or if there is no significant improvement in the solution (up to a convergence limit of 0.0001) in 50 consecutive iterations.

The sensitivity analysis of the PSO model is performed with different combinations of each parameter. In this analysis, it is observed that by considering a proper value for constriction coefficient (χ), the inertial weight does not have much influence on the final result of the model. So in this case study the inertial weight (ω) is fixed as 1. Also it is found that the value of constriction coefficient equal to 0.9 is yielding better results for the given model. After a number of trials, it is found that cognitive parameter c_1=1.0 and social parameter c_2=0.5 resulted in better quality solutions. To find the optimal parameters for population size and maximum number of iterations, a thorough sensitivity analysis is carried out for different combinations of the parameter settings. Fig. 4(a) shows the sensitivity for population size, from which it can be observed that the minimum fitness value is at population size of 200. Fig. 4(b) shows the sensitivity for maximum number of iterations and the minimum fitness value is observed for maximum iterations of 500. Similarly to determine the number of particles on which the elitist-mutation strategy is to be performed and the optimal number of iterations after which the elitist-mutation strategy should be started, a thorough sensitivity analysis is carried out. From Fig. 4(c), it can be observed that the number of elitist-mutated particles is equal to 20, which leads to better performance of the model. From Fig. 4(d) it can be observed that for up to 100 iterations there is no significant improve-

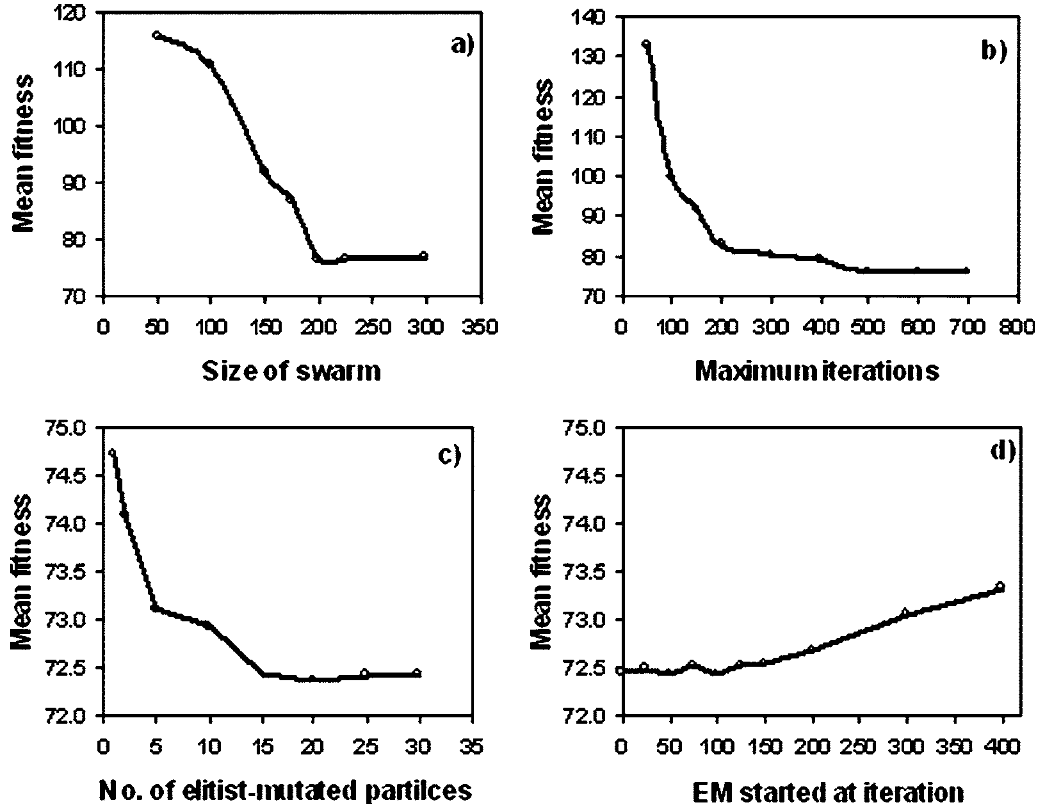

Fig. 4. Sensitivity analysis of EMPSO parameters showing the relationship between: (a) size of the swarm and mean fitness; (b) maximum number of iterations and mean fitness; (c) influence of number of elitist-mutated particles on the mean fitness value of the model; and (d) influence of elitist-mutation (EM) starting iteration on the performance of mean fitness value (mean fitness is the average fitness value of 10 random trials)

ment in the fitness values with the elitist-mutation step, so the elitist-mutation step started iteration is taken as 100. Probability of elitist mutation (P_{em}) is adopted as 0.2. So these parameters are adopted to run the model. The only difference between EMPSO and the standard PSO is that the elitist-mutation strategy is performed in EMPSO. All other parameters adopted are the same for both models.

The same reservoir operation problem is also solved using the real coded GA technique and the results are compared with those by the EMPSO model. After thorough sensitivity analysis, GA model parameters selected are as follows: size of the population, $N=200$; maximum number of generations$=500$; crossover probability, $P_c=0.85$; mutation probability, $P_m=0.05$; the distribution index for simulated binary crossover$=10$ and that for mutation operator$=100$.

To demonstrate the efficiency of the EMPSO model, the model is first applied to a typical year, where the inflow is well represented for the reservoir catchment. Using the values of 200 for population size and 500 for maximum number of iterations, all the three algorithms, namely EMPSO, standard PSO, and GA, were run for 10 different trials. Fig. 5 shows the details of the improvement of fitness values over the iterations, which also details the convergence properties of the EMPSO and PSO techniques. Fig. 6 shows the comparison of fitness values and number of function evaluations required to reach the fitness values for EMPSO, PSO, and GA models. Using the EMPSO model the best fitness value obtained is 72.272, with average fitness value of 72.504 and standard deviation of 0.14; whereas in standard PSO, the best fitness value is 72.972, with 80.26 and 3.85 as mean and standard deviations, respectively, and in GA the best is 72.72, with 73.389 and 0.479 as mean and standard deviations, respectively. For 10 trial runs, the average number of function evaluations taken to get the best solutions is 85,440, 83,280, and 95,660 for EMPSO, standard PSO, and GA, respectively. The GA model is taking more function evaluation numbers compared to other algorithms, and the solution quality is inferior to that of EMPSO. It can be observed that the EMPSO is consistently giving best fitness values over different trials than standard PSO. From these results it is clear that EMPSO is outperforming the standard PSO and GA models with better quality optimal solutions obtained using less number of function evaluations.

To evaluate the performance of the proposed EMPSO in detail,

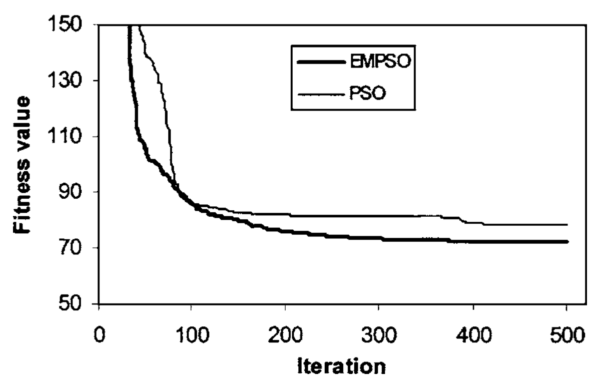

Fig. 5. Improvement of fitness value over the iterations shows the convergence properties for EMPSO and standard PSO

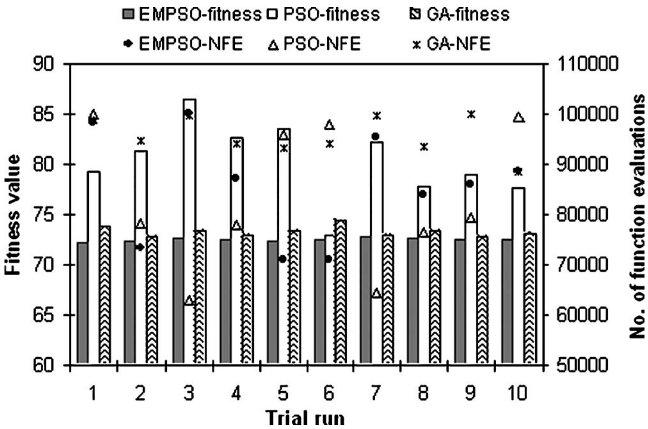

Fig. 6. Comparison of fitness value (nondimensional) and the number of function evolutions taken for EMPSO, standard PSO and GA models for optimal reservoir operation problem for a typical year of 10 random trials

the same model is applied for a good number of years and the results are reported in this section. In the following discussions, it may be noted that comparison is made with only the best solution obtained in ten trials with the respective algorithm. The total annual irrigation deficits (squared deficits) obtained by the releases made to the left and right bank canals over a period of 15 years are obtained as 141,601.94, 147,300.87, and 145,180.93 for the EMPSO, standard PSO, and GA models, respectively. It can be observed that EMPSO is performing better than the standard PSO and GA techniques. Fig. 7 shows the time series plot of monthly irrigation release made for different techniques over a period of 15 years. The other objective of the model is to maximize the hydropower production, for that the average annual hydropower production obtained over a period of 15 years is 1,820.71, 1,802.57, and $1,805.91 \times 10^6$ kW h by the EMPSO, PSO, and GA techniques, respectively. Here also it can be noticed that highest hydropower production is obtained with the EMPSO technique. Fig. 8 shows the time series plot of monthly hydropower production obtained using the EMPSO, PSO, and GA techniques. The time series plot of monthly initial reservoir storage obtained with the EMPSO, standard PSO, and GA techniques over a planning horizon of 15 years is shown in Fig. 9. The causes for superior performance of EMPSO over the other methods in terms of less irrigation deficits and more hydropower production can be summarized as apart from river bed turbine, the water released for irrigation to the left bank canal and right bank canal also generates hydropower. If the water is released in an optimal manner, it yields less irrigation deficits, and also contributes to hydropower. If more water is released through the river bed, it may generate more hydropower, but causes higher irrigation deficits. In objective function higher weightage is given for irrigation, so it will first try to minimize the irrigation deficits, and then come to maximization of hydropower. A best optimal solution should satisfy both these terms. So EMPSO is performing both the tasks well and hence is resulting in lesser irrigation deficits and higher hydropower production. If any overflows occur during a time period, this also affects the final results depending on the model performance. In the overall perspective, it is found that the operating policies obtained by the EMPSO technique are better than those by the standard PSO and GA techniques.

Discussion

In EMPSO methodology the additional parameters involved require little extra effort over that of standard PSO. But the improvement in solution quality is significant with EMPSO. This is achieved by maintaining diversity in the population and effectively exploring the search space throughout the iterations by applying the *elitist-mutation* step. In EMPSO the number of particles to be mutated depends on the size of the swarm, which again depends on the complexity of the problem. From experimental test results, it is observed that $3\sqrt[3]{N}$ (where N=size of the swarm) gives reasonably good results. So it can be set as a default parameter chosen for optimization. Similarly, for elitist-mutation starting iteration, as a default it can be performed for all the iterations. But from experimental test results, it is noticed that in early phases of iterations, this strategy does not have much influence. So after a preselected number of iterations (e.g., after 5–10% of maximum number of iterations), this strategy can be introduced to effectively save computational time and to improve efficiency of the algorithm. The probability of elitist mutation

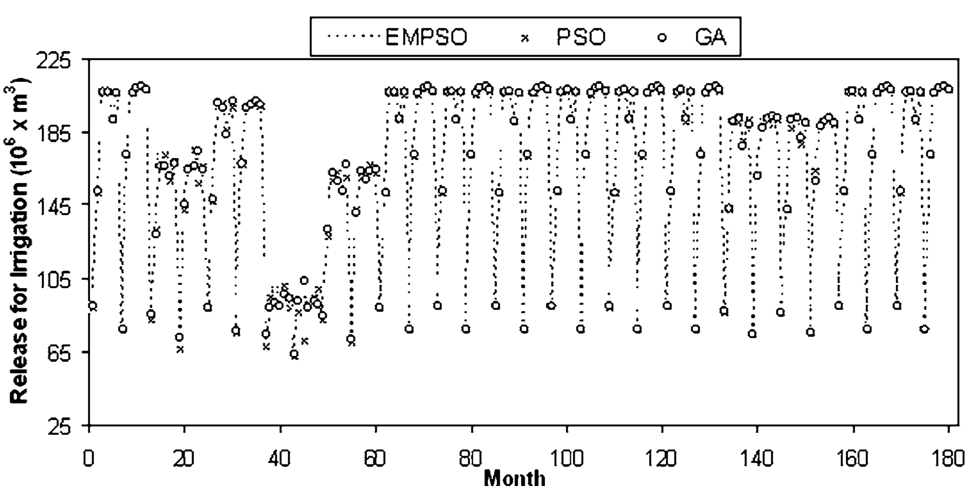

Fig. 7. Time series plot of monthly irrigation release obtained with EMPSO, standard PSO and GA techniques over a period of 15 years (from 1984–1985 to 1998–1999)

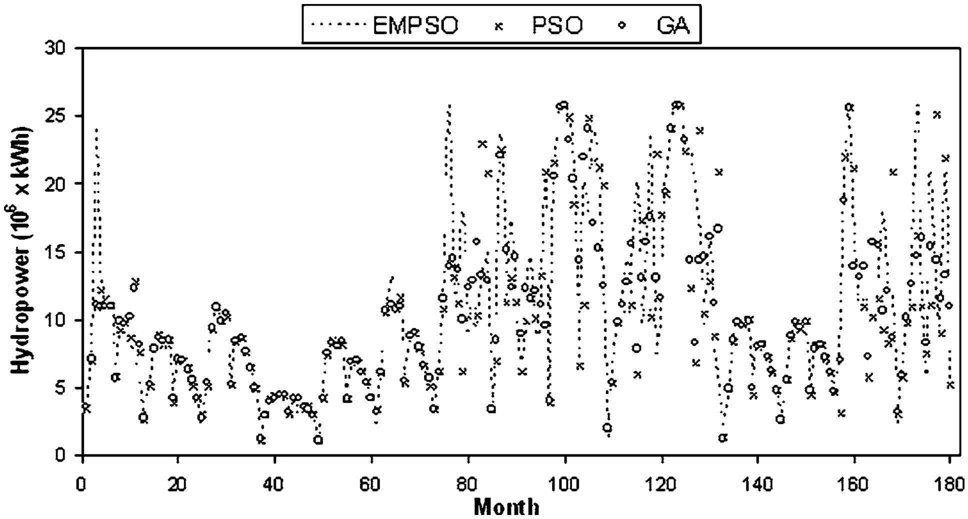

Fig. 8. Time series plot of monthly hydropower generated with EMPSO, standard PSO and GA techniques over a period of 15 years (from 1984–1985 to 1998–1999)

(P_{em}) ranges from 0.01 to 0.5 (depending on the complexity of the decision space).

The optimal solution found by the EMPSO technique is promising. So application of the EMPSO technique for derivation of reservoir operating policies is very much useful for real life problems, since this technique has the following advantages over other optimization techniques:

1. The method does not depend on the nature of the function it maximizes or minimizes. Thus, approximations made in conventional techniques are avoided.
2. EMPSO uses objective function information to guide the search in the problem space. Therefore EMPSO can easily deal with nondifferentiable and nonconvex objective functions.
3. Unlike the traditional methods, the technique is not affected by the initial searching points thus ensuring a quality solution with high probability of obtaining the global optimum for any initial solution.
4. In EMPSO, particle movement uses randomness in its search. Hence, it is a kind of stochastic optimization algorithm that can search a complicated and uncertain area. This makes EMPSO more flexible and robust than conventional methods.
5. The convergence is not affected by the inclusion of more constraints.
6. Unlike GA, EMPSO has the flexibility to control the balance between the global and local exploration of the search space. This property enhances the search capabilities of the PSO technique and yields better quality solutions with fewer function evaluations.
7. Unlike standard PSO, EMPSO is more reliable in giving better quality solutions with reasonable computational time, since the elitist-mutation strategy avoids the premature convergence of the search process to local optima and provides better exploration of the search process.

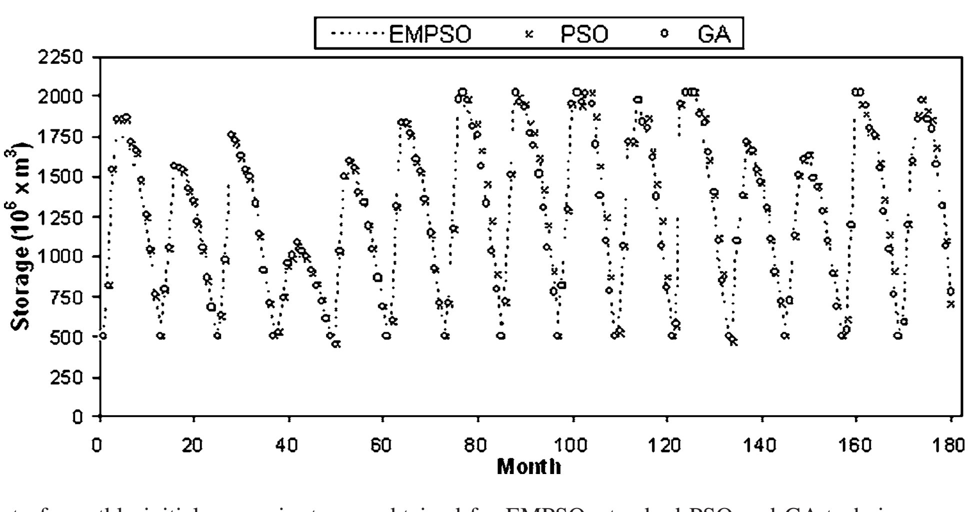

Fig. 9. Time series plot of monthly initial reservoir storage obtained for EMPSO, standard PSO and GA techniques over a period of 15 years (from 1984–1985 to 1998–1999)

Conclusions

In this study, a new swarm intelligence algorithm, namely EMPSO, is propounded to derive operating policies for a multipurpose reservoir system. The technique has global and local exploration capabilities to search for the optimal solutions. In order to overcome the problems of trapping at local optima and to improve the consistency of the standard PSO algorithm in reaching the global optimum, an *elitist-mutation* mechanism is incorporated into the algorithm. The proposed approach is first tested for a hypothetical multireservoir system. The results obtained by EMPSO compared well with results reported by DPSA, DDDP, and FDP. When compared with standard PSO and GA techniques, it is observed that EMPSO is consistently giving better quality solutions, using less number of function evaluations. After achieving satisfactory performance for the hypothetical problem, EMPSO is applied to a realistic problem, the Bhadra reservoir system in India to demonstrate the performance of the proposed technique. First it is applied for a typical year and the efficiency of the EMPSO technique is proved by comparing the obtained results with other algorithms. Then the same model is applied over a longer period of 15 years. It is found that the results obtained by EMPSO are better than those by the standard PSO and GA techniques in terms of minimum irrigation deficits and maximum hydropower production. The main advantages of EMPSO are, easy to implement, requires less number of function evaluations, and yet is efficient in obtaining global optima. The results of this study thus amply demonstrate that the EMPSO technique can be effectively used to solve real life optimization problems with better efficiency.

Notation

The following symbols are used in this paper:

$C_{l,\max}, C_{r,\max}$ = maximum carrying capacity of left and right bank canals;
c_1, c_2 = positive constants;
D = dimension of the solution vector;
$D_{l,t}, D_{r,t}$ = irrigation demands of left and right bank canal command areas, respectively, during time period t;
E = total energy production;
$E_{l,t}, E_{r,t}, E_{b,t}$ = hydropower produced at left bank canal, right bank canal, and bed turbines, respectively, during time period t;
$E_{l,\max}, E_{r,\max}, E_{b,\max}$ = maximum hydropower that can be produced at left bank canal, right bank canal, and bed turbines, respectively;
EVP_t = evaporation losses in period t;
$H_{l,t}, H_{r,t}, H_{b,t}$ = net head available at left bank canal, right bank canal, and bed turbines, respectively, during time period t;
I_t = inflow to the reservoir during time period t;
k_1, k_2 = constant weight coefficients;
MDT_t = mandatory release to be made in period t;
N = size of the swarm;
OVF_t = overflow in period t;
P_c = probability of crossover;
P_{em} = probability of elitist mutation;
P_g = position of the best particle in the swarm;
P_i = best position vector of the ith particle;
P_m = mutation probability;
p = power production coefficient;
p_{id} = best position of the ith particle dth variable;
$Q_{l,t}, Q_{r,t}, Q_{b,t}$ = releases made to left bank canal, right bank canal, and bed turbines, respectively, during time period t;
r_1, r_2 = uniformly distributed random numbers;
S_t = initial storage of reservoir in time period t;
$S_{\min}, S_{\max}$ = minimum and maximum storage limits of reservoir;
SQDV = sum of squared deviation;
V_i = velocity vector of the ith particle;
v_{id} = velocity change of the ith particle dth variable;
X_i = solution vector of the ith particle;
x_{id} = decision value of the ith particle dth variable;
χ = constriction coefficient; and
ω = inertial weight.

Subscripts

d = index of decision variable;
i = index of swarm population;
g = index of the best particle in the swarm; and
t = index of time period.

Superscripts

n = index of iteration number.

References

Abido, M. A. (2002). "Optimal power flow using particle swarm optimization." *Int. J. Electr. Power Energy Syst.*, 24(7), 563–571.

Chang, F. H., and Chen, L. (1998). "Real-coded genetic algorithm for rule-based flood control reservoir management." *Water Resour. Manage.*, 12(3), 185–198.

Clerc, M. (1999). "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization." *Proc., Congress of Evolutionary Computing*, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, eds., Vol. 3, IEEE, New York, 1951–1957.

Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *Comput. Methods Appl. Mech. Eng.*, 186(2–4), 311–338.

Eberhart, R. C., and Kennedy, J. (1995). "A new optimizer using particle swarm theory." *Proc., 6th Symp. on Micro Machine and Human Science*, IEEE Service Center, Piscataway, N.J., 39–43.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wiley, Reading, N.Y.

Heidari, M., Chow, V. T., Kokotovic, P. V., and Meredith, D. D. (1971). "Discrete differential dynamic programming approach to water resources systems optimization." *Water Resour. Res.*, 7(2), 273–282.

Kumar, D. N., and Singh, F. B. (2003). "Folded dynamic programming for optimal operation of multireservoir system." *Water Resour. Manage.*, 17(5), 337–353.

Larson, R. E. (1968). *State increment dynamic programming*, Elsevier, New York.

Loucks, D. P., Stedinger, J. R., and Haith, D. A. (1981). *Water resource systems planning and analysis*, Prentice-Hall, Englewood Cliffs, N.J.

Oliveira, R., and Loucks, D. P. (1997). "Operating rules for multireservoir systems." *Water Resour. Res.*, 33(4), 839–852.

Parsopoulos, K. E., and Vrahatis, M. N. (2002). "Recent approaches to global optimization problems through particle swarm optimization." *Natural Comput.*, 1(2–3), 235–306.

Salman, A., Ahmad, I., and Al-Madani, S. (2002). "Particle swarm optimization for task assignment problem." *Microproc. and Microsyst.*, 26(8), 363–371.

Wardlaw, R., and Sharif, M. (1999). "Evaluation of genetic algorithms for optimal reservoir system operation." *J. Water Resour. Plann. Manage.*, 125(1), 25–33.

Wurbs, R. A. (1993). "Reservoir-system simulation and optimization models." *J. Water Resour. Plann. Manage.*, 119(4), 445–472.

Yakowitz, S. (1982). "Dynamic programming applications in water resources." *Water Resour. Res.*, 18(4), 673–696.

Yeh, W. W.-G. (1985). "Reservoir management and operations models: A state-of-the-art review." *Water Resour. Res.*, 21(12), 1797–1818.