HYDROLOGICAL PROCESSES
Hydrol. Process. **21**, 2897–2909 (2007)
Published online 10 January 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/hyp.6507

Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation

M. Janga Reddy and D. Nagesh Kumar*
Department of Civil Engineering, Indian Institute of Science, Bangalore - 560 012, India

Abstract:

A multi-objective particle swarm optimization (MOPSO) approach is presented for generating Pareto-optimal solutions for reservoir operation problems. This method is developed by integrating Pareto dominance principles into particle swarm optimization (PSO) algorithm. In addition, a variable size external repository and an efficient elitist-mutation (EM) operator are introduced. The proposed EM-MOPSO approach is first tested for few test problems taken from the literature and evaluated with standard performance measures. It is found that the EM-MOPSO yields efficient solutions in terms of giving a wide spread of solutions with good convergence to true Pareto optimal solutions. On achieving good results for test cases, the approach was applied to a case study of multi-objective reservoir operation problem, namely the Bhadra reservoir system in India. The solutions of EM-MOPSOs yield a trade-off curve/surface, identifying a set of alternatives that define optimal solutions to the problem. Finally, to facilitate easy implementation for the reservoir operator, a simple but effective decision-making approach was presented. The results obtained show that the proposed approach is a viable alternative to solve multi-objective water resources and hydrology problems. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS multi-objective optimization; particle swarm optimization; elitist-mutation; reservoir operation; hydropower; irrigation; water quality; Pareto optimal solutions

Received 2 December 2005; Accepted 26 June 2006

INTRODUCTION

Most of the water resources and hydrology problems are characterized by multiple objectives and/or goals, which often conflict and compete with one another. Optimization of multi-purpose reservoir systems involves solving multi-objective problems. For example, for a reservoir system having hydropower and flood control as key purposes, the two major objectives can be maximization of the hydropower generation from the reservoir and minimization of flood risk or flood damage. Obviously, these two objectives are in conflict and compete with each other. The higher the level of the reservoir, the more the hydropower generation possible because of the high water head, yet less water storage will be available for flood control purposes and vice-versa. Clearly, one can identify, within the active storage capacity of that reservoir, a Pareto optimum region where the enhancement of the first objective can be achieved only at the expense or degradation of the second, namely flood control (Haimes *et al*., 1990). Also the units of these two objectives are non-commensurable. The first objective, which maximizes the hydroelectric power, is generally measured in units of energy and not necessarily in monetary units, whereas the second objective can be measured in terms of acres of land, livestock, or human lives saved. If the objectives are non-commensurate, the classic methods of optimization cannot be applied easily. Of the several approaches developed to deal with multiple objectives, tradeoff methodologies have shown promise as effective means for considering non-commensurate objectives that are to be subjectively compared in operation determination (Cohon and Marks, 1975). Therefore efficient generation of a set of alternatives for multiple objectives is very important with minimum computational requirements.

Scope for multi-objective optimization using meta-heuristic techniques

Reservoir operation modeling has an exhaustive literature presenting various optimization techniques in order to solve various kinds of problems (Yeh, 1985). However, in order to focus on the goal of this paper, a brief overview is given here. Most of the researchers on reservoir operation problems have tried conventional techniques to generate tradeoffs among multiple objectives. For example, Tauxe *et al*. (1979) have applied a multi-objective dynamic programming (DP) model for analyzing a reservoir operation problem involving three conflicting objectives. Thampapillai and Sinden (1979) and Mohan and Raipure (1992) analyzed the tradeoffs for multiple objective planning through linear programming (LP). To handle multiple objectives, many studies have used either the weighing approach or the constraint method. The constraint method was used for generation of noninferior set and trade-off curves for reservoir operation problems (e.g. Croley and Rao, 1979; Yeh and Becker, 1982; Liang *et al*., 1996 and Wang *et al*., 2005).

* Correspondence to: D. Nagesh Kumar, Department of Civil Engineering, Indian Institute of Science, Bangalore - 560012, India.
E-mail: nagesh@civil.iisc.ernet.in

The conventional optimization methods such as DP, LP, and non-linear programming (NLP) are not suitable to solve multi-objective optimization problems (MOOP), because these methods use a point-by-point approach, and the outcome of these classical optimization methods is a single optimal solution. For example, the weighted sum method will convert the MOOP into a single objective optimization. By using a single pair of fixed weights, only one point on the Pareto front can be obtained. Therefore, if one would like to obtain the global Pareto optimum, all possible Pareto fronts must first be derived. This requires the algorithms to be executed iteratively, so as to ensure that every weight combination has been used. Obviously, it is impractical to reiterate the algorithms continually to exhaust all the weight combinations. Hence the algorithms should have an ability to 'learn' from previous performance to direct the proper selection of weights in further evolutions. Also conventional methods may face problems, if the optimal solution lies on non-convex or disconnected regions of function space (Deb, 2001).

Recently, meta-heuristic techniques such as evolutionary algorithms (EAs) and swarm intelligence techniques are becoming increasingly popular for solving optimization problems. In the recent past, evolutionary techniques have been successfully applied for single objective optimization (Oliveira and Loucks, 1997; Wardlaw and Sharif, 1999; Raju and Nagesh Kumar, 2004) and MOOPs (Yapo *et al.*, 1998; Vrugt *et al.*, 2003; Khu and Madsen, 2005), due to their efficiency and ease in handling non-linear and non-convex relationships of real-world problems. These techniques have some advantages over the classical optimization techniques (Deb, 2001). They use a population of solutions in each iteration and offer a set of alternatives in a single run. They use randomized initialization and stochastic search in their operation. Therefore, they can locate the search at any place over the entire search space and are able to overcome the problems of local optima. Thus population based stochastic search techniques are more appropriate to solve MOOPs. Achieving a well-spread and diverse Pareto solution front is the primary goal of MOOP. Among the elitist multi-objective EAs (MOEAs), strength Pareto EA (SPEA) (Zitzler and Thiele, 1999), Pareto-archived evolutionary strategy (PAES) (Knowles and Corne, 2000) and non-dominated sorting genetic algorithm (NSGA-II) (Deb *et al.*, 2002) have been successfully demonstrated for solving MOOP.

Among the meta-heuristic techniques, until recently particle swarm optimization (PSO) was applied only to single objective optimization tasks. However, the high speed of convergence of the PSO algorithm attracted researchers to develop multi-objective optimization algorithms using PSO (Kennedy and Eberhart, 2001). Also, the PSO seems to have some advantages in terms of the better exploration and exploitation provided by local and global search capabilities of the algorithm. In the present study, a novel approach for multiple-objective PSO (MOPSO) is developed. To demonstrate the efficiency of the proposed approach, results obtained are compared with NSGA-II, and evaluated with standard performance measures that are frequently used for performance evaluation of MOEAs.

The proposed approach for solving a multi-objective decision problem in reservoir operation has a great potential for application, due to its attractive feature of generation of large number of well spread Pareto optimal solutions in a single run. The other approaches suggested in this paper for decision making, provide an opportunity for the reservoir operator to choose the desired alternative from a set of Pareto-optimal solutions.

PARTICLE SWARM OPTIMIZATION

Swarm intelligence is a new area of research, from which the PSO technique has been evolved through a simple simulation model of the movement of social groups such as birds and fish (Kennedy and Eberhart, 2001). The basis of this algorithm is that local interactions motivate the group behaviour, and individual members of the group can profit from the discoveries and experiences of other members. Social behaviour is modeled in PSO to guide a population of particles (so-called swarm), moving towards the most promising area of the search space. The changes of the position of the particles within the search space are based on the social psychological tendency of individuals to emulate the success of other individuals. In PSO, each particle represents a candidate solution. If the search space is D-dimensional, the i^{th} 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 i^{th} particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^T$. Defining g as the index of the global guide of the particle in the swarm, and superscripts denoting the iteration number, the swarm is manipulated according to the following two equations:

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

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

where $d = 1, 2, \ldots, D$; $i = 1, 2, \ldots, N$; N is the size of the swarm population; χ is a constriction factor which controls and constricts the velocity's magnitude; w is the inertial weight, which is often used as a parameter to control exploration and exploitation in the search space; c_1 and c_2 are positive constant parameters called acceleration coefficients; r_1 and r_2 are random numbers, uniformly distributed in [0,1]; Δt is the time step usually set as 1 and n is iteration number.

The successful application of PSO in many single objective optimization problems reflects its effectiveness, and it seems to be particularly suitable for multiobjective optimization due to its efficiency in yielding better quality solutions while requiring less computational time (Kennedy and Eberhart, 2001). The main difficulty in extending PSO to multi-objective problems is to find the best way of selecting the guides for each particle in the swarm. The difficulty is noticeable, as there are no clear concepts of local and global bests that can be clearly identified, when dealing with many objectives rather than a single objective. Recently a few proposals on extensions of PSO technique to multi-objective optimization have been reported (for example, Parsopoulos and Vrahatis, 2002; Hu *et al.*, 2003; Li, 2003; Coello *et al.*, 2004).

In this paper, an efficient method is presented for PSO to solve MOOPs. The approach uses Pareto dominance criteria for selecting non-dominated solutions; an external repository (ERP) for storing best solutions found (elitism); crowding distance operator for creating effective selection pressure among the swarm to reach true Pareto optimal fronts; and incorporates an effective elitist-mutation (EM) strategy for effective exploration of the search space. The proposed elitist-mutated multi-objective particle swarm optimization (EM-MOPSO) algorithm is discussed in detail in the following sections.

MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

Brief concepts of multi-objective optimization are presented first and then the proposed algorithm is explained.

Multi-objective optimization and Pareto optimality

A general MOOP can be defined as: minimize a function $f(x)$, subject to p inequality and q equality constraints.

$$\min.\ f(x) = \{f_1(x) f_2(x) \ldots f_m(x)\}^T \qquad x \in D \tag{3}$$

where $x \in R^n$, $f_i : R^n \to R$ and

$$D = \begin{cases} x \in R^n: & l_i \le x \le u_i, \quad \forall i = 1, \ldots, n \\ & g_j(x) \ge 0, \quad \forall j = 1, \ldots, p \\ & h_k(x) = 0, \quad \forall k = 1, \ldots, q \end{cases} \tag{4}$$

where m is number of objectives; D is feasible search space; $x = \{x_1 x_2 \ldots x_n\}^T$ is the set of n-dimensional decision variables (continuous, discrete or integer); R is the set of real numbers; R^n is n-dimensional hyper-plane or space; and l_i and u_i are lower and upper limits of i-th decision variable.

The MOOP should simultaneously optimize the vector function and produce Pareto optimal solutions. Pareto front is a set of Pareto optimal (non-dominated) solutions, being considered optimal, if no objective can be improved without sacrificing at least one other objective. On the other hand, a solution x^* is referred to as dominated by another solution x, if and only if, x is equally good or better than x^* with respect to all objectives (Haimes *et al.*, 1990).

Elitist-mutated multi-objective particle swarm optimization

The main algorithm consists of initialization of population, evaluation, and reiterating the search on swarm by combining PSO operators with Pareto-dominance criteria. In this process, the particles are first evaluated and checked for dominance relation among the swarm. The non-dominated solutions found are stored in an ERP, and are used to guide the search particles. It uses variable size ERP, in order to improve the performance of the algorithm to save computational time during optimization. If the size of ERP exceeds the restricted limit, then it is reduced by using the crowded comparison operator, which gives the density measure of the existing particles in the function space. Also, an efficient EM strategy is employed for maintaining diversity in the population and for exploring the search space. The combination of these operators helps the algorithm to effectively propagate the search towards true Pareto optimal fronts in further generations.

EM-MOPSO algorithm

The developed EM-MOPSO algorithm can be summarized in the following steps.

Step 1. Initialize population. Set iteration counter $t = 0$.
  1. The current position of the i-th particle X_i is initialized with random real numbers within the specified decision variable range; each particle velocity vector V_i is initialized with uniformly distributed random number in [0,1].
  2. Evaluate each particle in the population. The personal best position P_i, is set to X_i.

Step 2. Identify particles that give non-dominated solutions in the current population and store them in an ERP.

Step 3. $t = t + 1$.

Step 4. Repeat the loop (step through PSO operators):
  1. Select randomly a global best P_g for the i-th particle from the ERP.
  2. Calculate the new velocity V_i, based on Equation (1), and the new X_i by Equation (2).
  3. Repeat the loop for all the particles.

Step 5. Evaluate each particle in the population.

Step 6. Perform the Pareto dominance check for all the particles: if the current local best P_i is dominated by the new solution, then P_i is replaced by the new solution.

Step 7. Set ERP to a temporary repository, *TempERP* and empty ERP.

Step 8. Identify particles that give non-dominated solutions in current iteration and add them to *TempERP*.

Step 9. Find the non-dominated solutions in *TempERP* and store them in ERP. The size of ERP is restricted to the desired set of non-dominated solutions; if it exceeds, use the crowding distance operator to select the desired ones. Empty the *TempERP*.

Step 10. Perform EM operation on specified number of particles.
Step 11. Check for termination criteria; if the termination criterion is not satisfied, then go to step 3; otherwise output the non-dominated solution set from ERP.

The main operators used in this algorithm are explained below.

Variable size external repository

The selection of the global best guide of the particle swarm is a crucial step in a multi-objective PSO algorithm. It affects both the convergence capability of the algorithm as well as maintaining a good spread of non-dominated solutions. As ERP stores non-dominated solutions found in the previous iteration, any one of the solutions can be used as global guide. But we want to ensure that the particles in the population move towards the sparse regions of the non-dominated solutions and speed up the convergence towards the true Pareto optimal region. To perform these tasks, the global best guide of the particles is selected from a restricted variable size ERP. This restriction on ERP is done using the crowding distance operator. This operator ensures that those non-dominated solutions with the highest crowding distance values are always preferred to be in the ERP. The other advantage of this variable size ERP is that it saves considerable computational time during optimization. As the ERP size increases, the computing requirement becomes greater for sorting and crowding value calculations. Thus for effective exploration of the function space, the size is initially set to 10% of maximum ERP, and then the value is increased in a stepwise manner, so that at the start of 90% of maximum iterations, it reaches the maximum size of ERP. Selecting different guides for each particle from a restricted repository allows the particles better exploration of the true Pareto optimal region. This kind of selection is novel and it effectively improves the performance of the algorithm.

Crowding distance assignment operator

This operator is adopted from Deb *et al*. (2002). The crowding distance value of a solution provides an estimate of the density of solutions surrounding that solution. Crowding distance is calculated by first sorting the set of solutions in ascending objective function values. The crowding distance value of a particular solution is the average distance of its two neighboring solutions. The boundary solutions that have the lowest and highest objective function values are given infinite crowding distance values, so that they are always selected. This process is done for each objective function. The final crowding distance value of a solution is computed by adding all the individual crowding distance values in each objective function. For sorting, an efficient quick sorting procedure is used. The pseudo-code of crowding distance computation is given below.

1. Get the number of non-dominated solutions in the ERP

$$l = |ERP|$$

2. Initialize distance.

$$\text{For i} = 1 \text{ to } l$$

$$ERP[\text{i}].\text{dist} = 0$$

3. Compute the crowding distance of each solution.
For each objective m,
Sort using objective value.

$$ERP = \text{sort } (ERP, m)$$

Set the boundary points to a large value so that they are always selected.

$$ERP[1].\text{dist} = ERP[l].\text{dist} = \infty$$

For $i = 2$ to (l-1)

$$ERP[\text{i}].\text{dist} = ERP[\text{i}].\text{dist} + (ERP[\text{i}+1].m - ERP[\text{i}-1].m)/(f_m^{\max} - f_m^{\min})$$

Elitist-mutation operator

To maintain diversity in the population and to explore the search space, a novel strategic mechanism called EM is incorporated into the algorithm. This acts on a predefined number of particles. In the initial phase of this mechanism, it tries to replace the infeasible solutions with the mutated least crowded particles of ERP and at the later phase, it tries to exploit the search space around the sparsely populated particles in ERP along the Pareto fronts. This is a special strategic mechanism, which enhances the performance of MOPSO while extending from traditional PSO algorithm. Thus the EM operator helps to uniformly distribute the non-dominated solutions along the true Pareto optimal front. The pseudo-code of the elitist mutation mechanism is given below.

1. Randomly select one of the objectives from m objectives. Sort the fitness function of particles in descending order and get the index number descending order sorted particles (DSP) for the respective particles.
2. Use crowding distance assignment operator and calculate the density of solutions in the ERP and sort them in descending order of crowding value. Randomly select one of the least crowded solutions from the top 10% of ERP as guide (g).
3. Perform EM on a predefined number of particles ($NM_{\max}$).

Let R_p—be the size of repository; p_{em} - probability of elitist mutation; S_{m} - mutation scale used to preserve diversity; *rand* - uniformly distributed random number $U(0,1)$; *intRnd*(a, b) - uniformly distributed integer random number in the interval [a, b]; *randn* - Gaussian random number *N(0,1)*; and *VR[i]*- range of decision variable i.

For i = 1 to NM_{max}
  l = DSP[i]
  g = $intRnd(1, 0{\cdot}1 \times R_p)$
  For d = 1 to dim
   if ($rand < p_{\mathrm{em}}$)
    X[l] [d] = ERP[g][d] + S_{m}* VR[d]* $randn$
  else
    X[l] [d] = ERP[g][d]
  End For
 End For

If the mutated value exceeds the bounds, then it is limited to the upper or lower bound. The velocity vector of the particle remains unchanged during this EM step.

Constraint handling

In order to handle the constrained optimization problems, this study adopts the constraint handling mechanism proposed by Deb *et al*. (2002). This is a simple, but very effective procedure reported in the literature. In this approach, a solution i is said to be a constrained-dominate solution j if any of the following conditions hold good:

1. Solution i is feasible and solution j is not.
2. Both solutions i and j are infeasible, but solution i has a smaller overall constraint violation.
3. Both solutions i and j are feasible and solution i dominates solution j.

By using all the above steps, the EM-MOPSO approach is coded in user friendly mathematical software package MATLAB 6·5 and is run on PC/WindowsXP/256MB RAM/2GZ computer. The applicability and efficiency of the proposed approach is demonstrated in the following sections.

EM-MOPSO APPLICATION AND PERFORMANCE EVALUATION

In order to demonstrate the efficiency of the proposed EM-MOPSO, it is first tested for a few standard test problems taken from the MOEAs literature and its performance is evaluated with results of NSGA-II.

Test problems

The four test functions considered to test the performance of the proposed algorithm are given in Table I (Deb, 2001). The first test problem (BNH) is a MOOP, with two objectives subject to two constraints. The second test problem (KITA) is a MOOP, with maximization of two objectives subject to three constraints. This problem has non-convexity in its Pareto optimal region. The third test problem (CONSTR) is a MOOP, with two objectives subject to two constraints. This problem has the difficulty that a part of the unconstrained Pareto optimal region is not feasible. Thus the resulting constrained Pareto optimal region is a concatenation of the first constraint boundary and some part of unconstrained Pareto optimal region. The fourth test problem (SRN) is a MOOP, with two objectives subject to two constraints. Here the constrained Pareto optimal set is a subset of the unconstrained Pareto-optimal set, which gives difficulty in finding the true Pareto optimal region for the algorithm.

Sensitivity of EM-MOPSO parameters

The sensitivity analysis of the PSO model is performed with different combinations of each parameter. In this analysis, it is observed that by considering the proper value for the constriction coefficient, the inertial weight does not have much influence on the final result of the model (Nagesh Kumar and Janga Reddy, 2006). So in this study the inertial weight (w) is fixed as 1. Also, it is found that the value of constriction coefficient χ equal to 0·9 yields better results for the given model. After a number of trials, it was found that constant parameters $c_1 = 1{\cdot}0$ and social parameter $c_2 = 0{\cdot}5$ resulted in better quality solutions. The same values are used for all the test problems.

Table I. Test problems used in the study

Problem	Variable bounds	Objective functions	Constraints
BNH	$x_1 \in [0, 5]$ $x_2 \in [0, 3]$	Minimize $f_1(x) = 4x_1^2 + 4x_2^2$ $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$	$g_1(x) = (x_1 - 5)^2 + x_2^2 \le 25$ $g_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \ge 7{\cdot}7$
KITA	$x_i \in [0, 7]$ $i = 1, \ldots, 3$	Maximize $f_1(x) = -x_1^2 + x_2$ $f_2(x) = 0{\cdot}5x_1 + x_2 + 1$	$g_1(x) = x_1/6 + x_2 - 6{\cdot}5 \le 0$ $g_2(x) = 0{\cdot}5x_1 + x_2 - 7{\cdot}5 \le 0$ $g_3(x) = 5x_1 + x_2 - 30 \le 0$
CONSTR	$x_1 \in [0{\cdot}1, 1{\cdot}0]$ $x_2 \in [0, 5]$	Minimize $f_1(x) = x_1$ $f_2(x) = (1 + x_2)/x_1$	$g_1(x) = x_2 + 9x_1 \ge 6$ $g_2(x) = -x_2 + 9x_1 \ge 1$
SRN	$x_i \in [-20, 20]$ $i = 1, 2$	Minimize $f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$	$g_1(x) = x_1^2 + x_2^2 \le 225$ $g_2(x) = x_1 - 3x_2 \le -10$

DOI: 10.1002/hyp

Size of elitist-mutated particles (NM_{max}). The number of particles to be elitist mutated is selected after ensuring that the population does not lose control on search at the cost of exploring for better non-dominated solutions. To experiment with the size of elitist mutated particles, the number of particles is varied as 5, 10, 15, 20, 25 and 30 for the problems having a maximum population of 100. The best results are found for $NM_{max} = 20$ and is kept constant for all the test problems considered in this study.

Probability of elitist mutation (p_{em}). p_{em} is varied from 0 to 0·5 for sensitivity analysis. It is found that best performance occurs at $p_{em} = 0{\cdot}2$, and is kept constant for all the test problems considered in the study.

Elitist-mutation operator mutation scale (S_m). The other parameter used in EM operation is mutation scale (S_m). After various experiments, it is found that a value of S_m in the range of 0·2 to 0·01 gives good performance. This is selected after ensuring that at the initial stage it did not deteriorate the search while exploring the search space or stagnate the search at the end of iterations.

Simulation results

To run the EM-MOPSO algorithm, the following parameters are used: size of population = 100; constant parameters $c_1 = 1{\cdot}0$ and $c_2 = 0{\cdot}5$; inertial weight $w = 1$; constriction coefficient $\chi = 0{\cdot}9$; size of ERP = 100; the size of elitist-mutated particles is set to 20, the value of p_{em} was set to 0·2; and the value of S_m decreases from 0·2 to 0·01 over the iterations. To run the NSGA-II model, the initial population was set to 100, crossover probability to 0·9, and mutation probability to $1/n$ (n is the number of real variables). The distribution index values for real-coded crossover and mutation operators are set to 20 and 100 respectively (Deb *et al.*, 2002). Maximum number of iterations in a run is set to 250 for both the algorithms. The same parameter settings were used for all the problems. To evaluate the performance of the proposed EM-MOPSO algorithm, this study uses two performance measures, set coverage metric (SC) and spacing metric (SP) (Deb, 2001). The details of these performance metrics are presented in APPENDIX- A.

Table II shows the best, worst, mean, variance and standard deviation (SD) values of the two performance metrics (SC and SP) obtained from 10 independent runs using EM-MOPSO and NSGA-II. The set coverage metrics SC (A, B) and SC (B, A) give a measure of how many solutions of A are covered by B and vice versa. Here, the value SC (A, B) = 1 means that all solutions in B are weakly dominated by A, while SC (A, B) = 0 represents the situation when none of the solutions in B are weakly dominated by A. It can be seen that with respect to the SC metric, the average performance of EM-MOPSO is the best for test functions BNH, KITA and SRN, whereas NSGA-II performs best for the CONSTR problem. This metric shows the efficiency of EM-MOPSO in achieving better convergence to true Pareto optimal fronts than NSGA-II. With regard to the spacing metric (SP), as compared to NSGA-II, EM-MOPSO gives smaller SP values for all the test problems considered in the study. The smaller SP indicates that the algorithm gives better distribution of solutions. Thus EM-MOPSO maintains the best distribution of solutions for all the test problems. For illustration purposes, a sample result for each of the test problems considered is shown in the plots in Figure 1. Thus the results obtained clearly show that the proposed method does not have any difficulty in achieving a good spread of Pareto optimal solutions for constrained multi-objective optimization.

Table II. Resulting statistics by EM-MOPSO and NSGA-II for test problems, considered in the study. In SC(A, B), A is EM-MOPSO and B is NSGA-II. Bold numbers indicate the best performing algorithm

Test problem	Statistic	Performance metric			
		Set coverage metric (SC)		Spacing metric (SP)	
		SC(A, B)	SC(B,A)	EM-MOPSO	NSGA-II
	Best	0·1400	0·1200	0·6357	0·6408
	Worst	0·0900	0·0500	0·7559	0·8928
BNH	Mean	**0·1111**	0·0877	**0·6941**	0·7756
	Variance	0·0003	0·0005	0·0015	0·0053
	SD	0·0176	0·0233	0·0385	0·0727
	Best	0·2900	0·2400	0·0374	0·0496
	Worst	0·1200	0·1300	0·4254	0·5117
KITA	Mean	**0·2400**	0·1811	**0·1359**	0·1464
	Variance	0·0015	0·0013	0·0196	0·0227
	SD	0·0394	0·0355	0·1401	0·1507
	Best	0·1600	0·1700	0·0374	0·0372
	Worst	0·0700	0·1000	0·0431	0·0487
CONSTR	Mean	0·1181	**0·1344**	**0·0406**	0·0437
	Variance	0·0008	0·0004	0·0000	0·0000
	SD	0·0281	0·0201	0·0017	0·0041
	Best	0·1400	0·1400	1·0768	1·3402
	Worst	0·0400	0·0400	1·3929	1·7073
SRN	Mean	**0·0978**	0·0944	**1·2439**	1·5860
	Variance	0·0014	0·0009	0·0114	0·0179
	SD	0·0370	0·0305	0·1055	0·1337

CASE STUDY

To demonstrate the efficacy of the proposed approach, the Bhadra reservoir system in India is taken up as a case study for developing optimal reservoir operation policy. Figure 2 shows the location map of the Bhadra reservoir system. The Bhadra dam is located at latitude 13°42′ N and longitude 75°38′20″ E. The Bhadra reservoir is a multi-purpose project providing facilities for irrigation, hydropower generation and meeting water quality requirements downstream. The schematic diagram of the reservoir system is given in Figure 3.

Most of the inflows into the reservoir are received during the monsoon season of 4 months. But the demands are distributed throughout the year. The reservoir provides water for irrigation of 6367 ha and 87 512 ha under

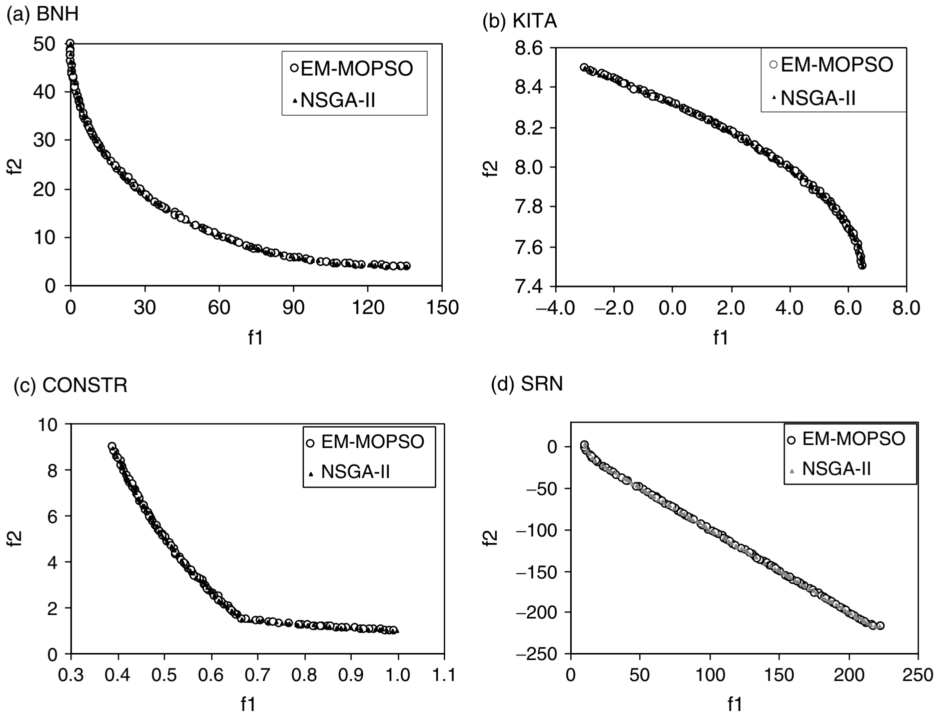

Figure 1. Non-dominated solutions obtained using EM-MOPSO and NSGA-II for the test problems (a) BNH, (b) KITA, (c) CONSTR and (d) SRN

left and right bank irrigation canals respectively. The irrigated area spread over the districts of Chitradurga, Shimoga, Chikmagalur, and Bellary in Karnataka state, comprises predominantly of red loamy soil, except in some portions of the right bank canal area, which consists of black cotton soil. Major crops grown in the command area are paddy, sugarcane, permanent garden, and semidry crops. 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, generating hydropower. The operating head above river bed, ranges from 38·56 m to 54·41 m for the right bank turbine (PH2), and from 36·88 m to 56·69 m for left bank turbine (PH1) and bed turbines (PH3). The mean tail water levels of right bank, left bank, and bed turbines are at 32·736 m, 12·802 m and 6·706 m above the bed level respectively. It can be noted that the water released to left bank and right bank canals goes through turbines only when the water is within the limits of turbine operating range, otherwise it will be released directly for irrigation. Water quality is also a major concern to the reservoir authorities due to continuous development of industries in the downstream region. So the water quality objective requires certain minimum water levels to be maintained in the river downstream.

Salient features of the reservoir are given in Table III. Data pertaining to monthly inflows and other details were collected from water resources development organization (WRDO), Bangalore covering a period of 69 years (from 1930–1931 to 1998–1999). The monthly crop water requirements were calculated using Food and Agricultural Organization (FAO) Penman-Monteith method (Allen *et al.*, 1998).

Model formulation

The objectives of the reservoir operation model are: minimizing the irrigation deficits, maximizing the hydropower generation and maximizing the satisfaction level of water quality. These are conflicting and/or competitive objectives. For example, conflict may arise during dry periods: for minimization of irrigation deficits, more water is to be released to satisfy irrigation demands; while for maximization of hydropower production, higher level of storage in the reservoir is required to produce more hydropower energy; and for maximizing the satisfaction level of water quality, steady release of water is required to meet river water quality requirements downstream. Thus, solving the allocation problems of this reservoir system is interesting from the multi-objective perspective. In order to simplify the water quality objective, in this study it is assumed that if we discharge a prespecified amount of water into the downstream river, the river water quality can be maintained. The water quality demands are chosen after carefully studying the historical data and previous studies on river water quality maintenance. To maintain even distribution of irrigation deficits if any, the irrigation objective is taken as squared deviation of demand to release. The competing objectives of the system are expressed as follows:

Minimize sum of squared deviations for irrigation annually:

$$SQDV = \sum_{t=1}^{12}(D_{1,t} - IR_{1,t})^2 + \sum_{t=1}^{12}(D_{2,t} - IR_{2,t})^2 \quad (5)$$

where $SQDV$ is the sum of squared deviations of irrigation demands from releases. $D_{1,t}$ and $D_{2,t}$ are the

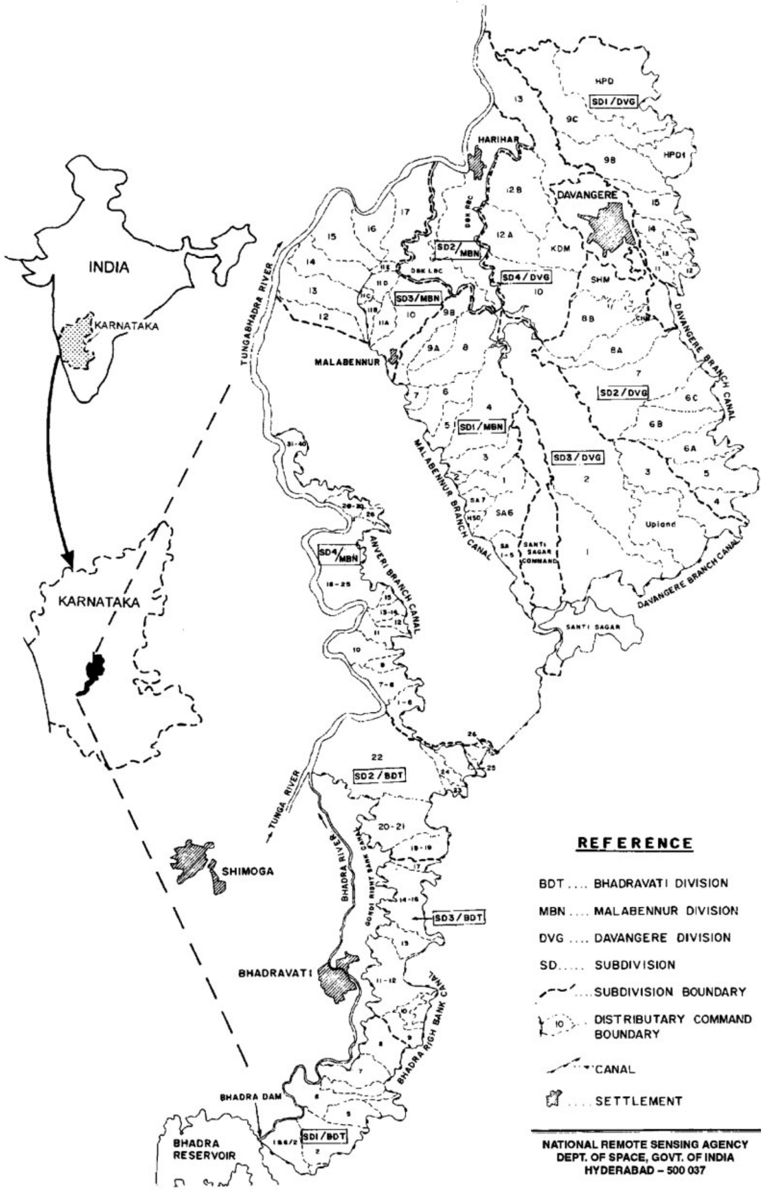

Figure 2. Location map of the Bhadra project command area

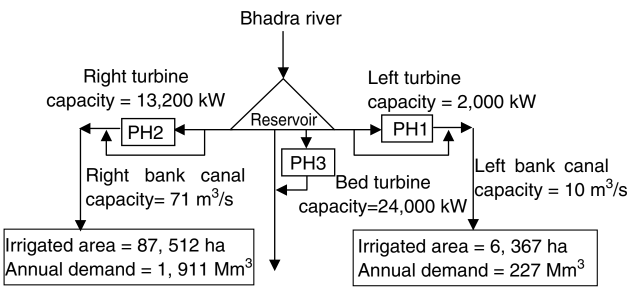

Figure 3. Schematic diagram of the Bhadra reservoir project

Table III. Salient features of Bhadra reservoir system

Description	Quantity
Gross storage capacity	2025 Mm^3
Live storage capacity	1784 Mm^3
Dead storage capacity	241 Mm^3
Average Annual inflow	2845 Mm^3
Left bank canal capacity	10 m^3/s
Right bank canal capacity	71 m^3/s
Left bank turbine capacity (PH1)	2000 Kw
Right bank turbine capacity (PH2)	13 200 Kw
Riverbed turbine capacity (PH3)	24 000 kW

irrigation demands for the left bank canal and right bank canal command areas respectively in period t in Mm^3; $IR_{1,t}$ and $IR_{2,t}$ are the irrigation releases into the left and right bank canals respectively in period t in Mm^3.

Maximize annual hydropower production:

$$P = \sum_{t=1}^{12} p(R_{1,t}H_{1,t} + R_{2,t}H_{2,t} + R_{3,t}H_{3,t}) \tag{6}$$

where P is the total energy produced in M kWh; p is power production coefficient; $R_{1,t}, R_{2,t}$ and $R_{3,t}$ are the releases to left bank, right bank and river bed turbines respectively in period t in Mm^3. $H_{1,t}, H_{2,t}, H_{3,t}$ are the net heads available to the left bank, right bank and bed turbines respectively in meters during period t (here, head is a nonlinear function of initial and final reservoir storage).

Maximize satisfaction level of river water quality:

$$WQ = \min_{\forall t=1,2,\ldots,12}(\lambda_t) \tag{7}$$

where, λ_t is satisfaction level of water quality in period t and is given by,

$$\lambda_t = \begin{cases} 0 & if\ R_{3,t} \le QD_{\min,t} \\ \dfrac{(R_{3,t} - QD_{\min,t})}{(QD_{\max,t} - QD_{\min,t})} & if\ QD_{\min,t} \le R_{3,t} \le QD_{\max,t} \\ 1 & if\ R_{3,t} \ge QD_{\max,t} \end{cases} \tag{8}$$

where, $QD_{\min,t}$ and $QD_{\max,t}$ are the minimum and maximum water demands to maintain water-quality in period t in Mm^3 for the river downstream of the dam.

The optimization is subject to the following constraints:

Storage continuity:

$$S_{t+1} = S_t + I_t - (R_{1,t} + R_{2,t} + R_{3,t} + E_t + O_t) \quad \forall t = 1, 2, \ldots, 12 \tag{9}$$

where S_t = Active reservoir storage at the beginning of period t in Mm^3; I_t = inflow into the reservoir during period t in Mm^3; E_t = the evaporation losses during period t in Mm^3 (here, E_t is a nonlinear function of initial and final storages of period t); O_t = overflow from the reservoir in period t in Mm^3;

Storage limits:

$$S_{\min} \le S_t \le S_{\max} \qquad \forall t = 1, 2, \ldots, 12 \tag{10}$$

where $S_{\min}$ and $S_{\max}$ are the minimum and maximum active storages of the reservoir in Mm^3.

Maximum power production limits:

$$pR_{1,t}H_{1,t} \le E_{1,\max} \qquad \forall t = 1, 2, \ldots, 12 \tag{11}$$

$$pR_{2,t}H_{2,t} \le E_{2,\max} \qquad \forall t = 1, 2, \ldots, 12 \tag{12}$$

$$pR_{3,t}H_{3,t} \le E_{3,\max} \qquad \forall t = 1, 2, \ldots, 12 \tag{13}$$

where, $E_{1,\max}, E_{2,\max}$, and $E_{3,\max}$ are the maximum amounts of power in M kWh, that can be produced (turbine capacity) by the left, right and bed level turbines respectively.

Canal capacity limits:

$$IR_{1,t} \le C_{1,\max} \quad \forall t = 1, 2, \ldots, 12 \tag{14}$$

$$IR_{2,t} \le C_{2,\max} \quad \forall t = 1, 2, \ldots, 12 \tag{15}$$

where, $C_{1,\max}$ and $C_{2,\max}$ are the maximum canal carrying capacities of the left and right bank canals respectively.

Irrigation demands:

$$D_{1\min,t} \le IR_{1,t} \le D_{1\max,t} \qquad \forall t = 1, 2, \ldots, 12 \tag{16}$$

$$D_{2\min,t} \le IR_{2,t} \le D_{2\max,t} \qquad \forall t = 1, 2, \ldots, 12 \tag{17}$$

where, $D_{1\min,t}$ and $D_{1\max,t}$ are minimum and maximum irrigation demands for left bank canal respectively; $D_{2\min,t}$ and $D_{2\max,t}$ are minimum and maximum irrigation demands for right bank canal respectively in time period t.

Water Quality Requirements:

$$R_{3,t} \ge MDT_t \qquad \forall t = 1, 2, \ldots, 12 \tag{18}$$

where, MDT_t = minimum release to meet downstream water quality requirement in Mm^3.

It can be noted that, in this study, under favorable range of reservoir storage for power production, the releases made through power turbines also serve to meet irrigation demands of the left bank and right bank canals during irrigation requirement periods. However, if the reservoir storage is not within the limits of the turbine power production range, then the water is released only to meet irrigation demands through sub-ways to irrigation canals. Since the water released for irrigation is restricted to its demands, any excess water is not a penalizing problem in the first objective function as given in Equation (5).

RESERVOIR OPERATION MODEL APPLICATION AND RESULTS

To apply the EM-MOPSO for reservoir operation model, the following parameters are selected. The initial population of the EM-MOPSO is set to 200; c_1 and c_2 are set to 1·0 and 0·5 respectively; w is set to 1·0; and χ is set to 0·9; the number of non-dominated solutions to be found is set to 200. For the EM step, the size of elitist-mutated particles is set to 30, the value of p_{em} was set to 0·2; and the value of S_{m} decreases from 0·2 to 0·01 over the iterations. Then EM-MOPSO is run for 500 iteration steps. To run the NSGA-II model, the initial population was set to 200, crossover probability to 0·9, and mutation probability to $1/n$ (n is the number of real variables). The distribution index values for real-coded crossover and mutation operators are set to 20 and 100 respectively. NSGA-II is also run for 500 generations. The average monthly inflow into the reservoir computed for each calendar month over a period of 69 years (from 1930–1931 to 1998–1999) is used as inflow data to implement the above model.

Two-objective model

To show the effectiveness of the proposed MOPSO, for solving the reservoir operation problem, first it is applied to irrigation and hydropower as two objectives of the model. The reservoir operation model consists of minimizing irrigation deficits (Equation (5)) and maximizing the hydropower (Equation (6)) subject to satisfying the constraints from Equations (9) to (18).

To check the performance, 10 independent runs were carried out for the two-objective reservoir operation model using both the algorithms. Table IV shows the resulting statistics for both the EM-MOPSO and NSGA-II models. It can be observed that with respect to set coverage metric, the average value of SC(A, B) is higher than the SC(B, A) value (here A is EM-MOPSO and B is NSGA-II). The metric SC(A, B) refers to the percentage of solutions in B that are weakly dominated by solutions of A. Thus in this case, EM-MOPSO is performing better than the NSGA-II. Similarly, for the spacing metric, from Table IV it can be observed that the mean value of SP metric for EM-MOPSO is lower than that for NSGA-II. This indicates that best distribution of Pareto solutions is obtained in EM-MOPSO. For demonstration purposes a sample result corresponding to median value of SC(A,B) is shown in Figure 4. It can be seen that EM-MOPSO is able to generate a set of well-distributed Pareto-optimal solutions. In this case NSGA-II fails to yield the extreme solutions, whereas our proposed method is able to provide extreme solutions comfortably (the minimum squared deviation of irrigation as 0 $(Mm^3)^2$ and maximum hydropower production as 227·845 MkWh). Thus again these results demonstrate that the proposed method is very useful for real life systems application.

Three-objective model

The reservoir operation model for three objectives consists of minimizing the irrigation deficits (Equation (5)), maximizing the hydropower (Equation (6)) and maximizing the satisfaction level of water quality (Equations (7) and (8)) subject to satisfying the constraints from Equations (9) to (18). Figure 5 shows the results of 200 non-dominated solutions obtained using EM-MOPSO after 500 iterations. There are a number of alternatives that can be chosen at various satisfaction levels of the multiple objectives. Depending on the circumstances prevailing under the reservoir system and by analyzing the tradeoff between the multiple objectives, the reservoir operator can make an appropriate decision. The details of decision making for application are presented in the following section.

Table IV. Resulting statistics by EM-MOPSO and NSGA-II for the two-objective reservoir operation model. In SC(A,B), A is EM-MOPSO and B is NSGA-II. Bold numbers indicate the best performing algorithm

Statistic	Performance metric			
	Set coverage metric (SC)		Spacing metric (SP)	
	SC(A, B)	SC(B,A)	EM-MOPSO	NSGA-II
Best	0·2000	0·0400	206·1277	246·2255
Worst	0·8788	0·6350	294·1276	787·7765
Mean	**0.5582**	0·2878	**258.2752**	504·3212
Variance	0·0435	0·0317	1419·9885	32 583·4308
SD	0·2085	0·1780	37·6827	180·5088

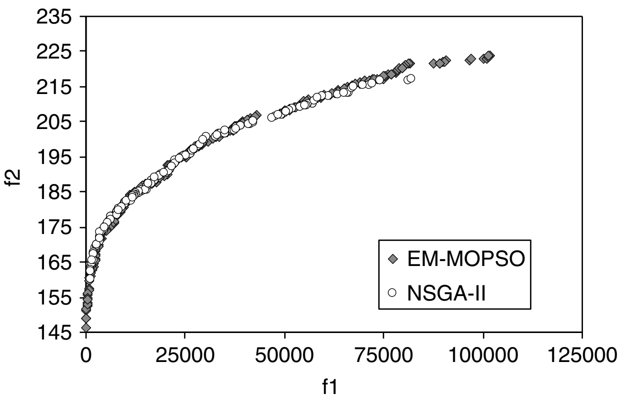

Figure 4. Non-dominated solutions obtained using EM-MOPSO and NSGA-II for two-objective reservoir operation model. The two objectives are f_1- squared deviation of irrigation $(Mm^3)^2$, f_2-hydropower production, M kWh

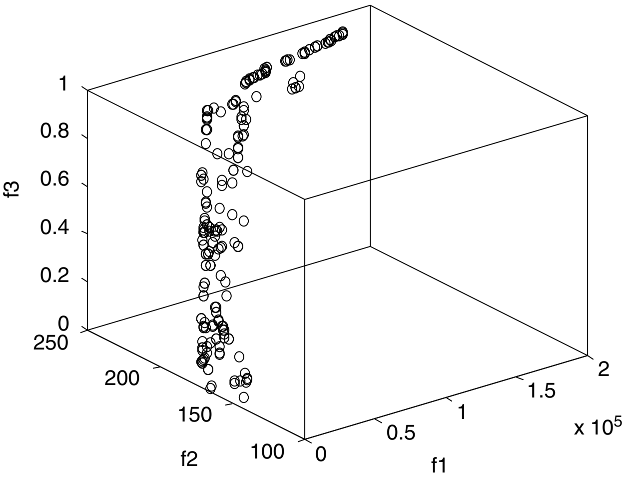

Figure 5. Non-dominated solutions obtained for three-objective reservoir operation model using EM-MOPSO. f_1-sum of squared deficits of irrigation releases $(Mm^3)^2$; f_2-hydropower production, M kWh; f_3-satisfaction level of water quality

Decision making

In any application, for final decision making, the decision maker might be interested in minimum possible number of well representative solutions for further analysis. So it is important that after obtaining many solutions which are true Pareto Optimal with uniform spread and wide coverage, we need to reduce the large set of solutions to a few representative solutions. In order to do that, various clustering algorithms are available. A simple clustering algorithm, which reduces the large number of final Pareto solutions (N) to a few representative solutions ($\overline{N}$), is described here.

Clustering technique

First each solution in ERP is considered to reside in a separate cluster. Thus initially there are N clusters. Thereafter the cluster distances between all pairs of clusters are calculated. Then the two clusters with the minimum cluster distance are combined together to form one big cluster. The procedure is repeated by calculating the cluster distances for all the pairs of clusters obtained by merging the two closest clusters. This process of merging clusters is continued until the number of clusters in the ERP is reduced to $\overline{N}$. Thereafter, in each cluster, the solution with the minimum average distance from other solutions in the cluster is taken as a representative solution for that cluster. The step-by-step procedure of the algorithm is given below (Deb, 2001).

1. Initialize cluster set C; each individual i in ERP constitutes a distinct cluster. i.e. $C_i = \{i\}$, so that $C = \{C_1, C_2, \ldots, C_N\}$
2. If, $|C| \leq \overline{N}$ go to Step 5, otherwise go to Step 3.
3. For each pair of clusters, calculate the cluster-distance by using Equation (19).

$$d_{12} = \frac{1}{|C_1|.|C_2|} \sum_{i \in C_1, j \in C_2} d(i, j) \qquad (19)$$

where the function d_{12} reflects the distance between two individuals i_1 and i_2 (here the distance in objective space is used).

4. Find the pair (i_1, i_2) which corresponds to the minimum cluster-distance. Merge the two clusters C_{i_1} and C_{i_2} together. This reduces the size of C by one. Go to Step 2.
5. Choose only one solution from each cluster and remove all others from the cluster. The solution having the minimum average distance from other solutions in the cluster is chosen as the representative solution of the cluster (centroid method).

To reduce the large number of alternatives, the number of clusters is chosen as 20. According to the cluster algorithm described above, this reduces the large set of non-dominated solutions to a few representative solutions. Figure 6 shows the 20 representative clustered Pareto-optimal solutions for the three-objective reservoir operation model.

To facilitate final decision making, (i.e. to understand how each of the objectives can influence the decision) a

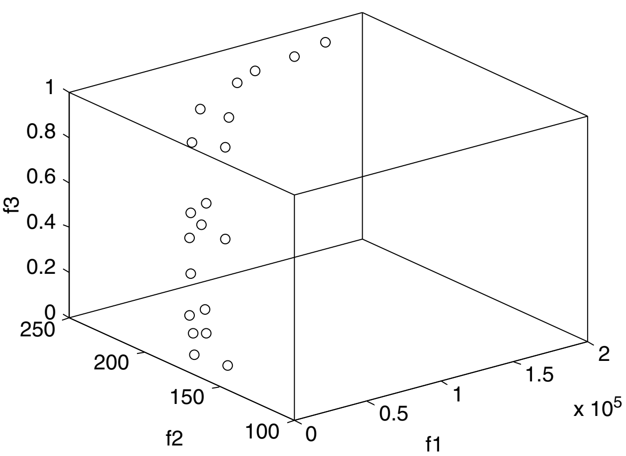

Figure 6. Representative non-dominated solutions obtained from clustering of EM-MOPSO generated solutions. f_1-sum of squared deficits of irrigation releases $(Mm^3)^2$; f_2-hydropower production, M kWh; f_3-satisfaction level of water quality

simple procedure called Pseudo-weight vector approach is employed in this study and the details of the procedure are given below.

Pseudo-weight vector approach

In this approach, a pseudo-weight vector is calculated for each obtained solution. For minimization problems, the approach is described here. From the obtained set of solutions, the minimum $f_i^{\min}$ and maximum $f_i^{\max}$ values of each objective function i are noted. Thereafter the Equation (20) is used to compute the weight w_i for the i-th objective function (Deb, 2001):

$$w_i = \frac{(f_i^{\max} - f_i(x))/(f_i^{\max} - f_i^{\min})}{\sum_{m=1}^{M} (f_i^{\max} - f_m(x))/(f_i^{\max} - f_m^{\min})} \qquad (20)$$

This equation calculates the relative distance of the solution from the worst (maximum) value in each objective function. Thus, for the best solution for the i-th objective, the weight w_i is to be a maximum. The numerator in the right side of the above equation ensures that the sum of all weight components for a solution is equal to one. Once the weight vectors are calculated, a simple strategy is to choose the solution closer to a user-preferred weight vector. For example, if an 80% weightage for f_1 and a 20% weightage for f_2 are desired, the corresponding weight vector closer to that non-dominated solution can be selected for final decision making.

Based on pseudo-weight vector approach, the weights that can be given for each objective are shown in Table V, which gives the objective values for each of the representative Pareto optimal solutions and its respective weight (shown in parenthesis) for each alternative. This provides ease in decision making for policy implementation. After analyzing the available alternatives, based on individual preferences, the final decision can be made. Suppose the reservoir operator decides to implement a policy with weights 0·5, 0·1 and 0·4 for irrigation deficit, hydropower production and water quality satisfaction levels respectively, then a solution closer to that set of weights i.e. 9th solution from the Table V can be selected. The model readily gives the corresponding policy for implementation. Figure 7 shows the corresponding releases to be made into the left bank canal, right bank canal and bed turbine. Figure 8 shows the corresponding initial storages required for ensuring such releases. Thus this kind of analysis can be implemented effectively with the proposed EM-MOPSO technique for derivation of reservoir operation policies.

CONCLUSIONS

In this study, a novel approach for multi-objective optimization based on swarm intelligence principles is proposed and applied to develop efficient operating alternatives for multi-objective reservoir operation. The proposed MOPSO approach combines PSO technique

Table V. Filtered or representative Pareto optimal solutions for the three-objective reservoir operation model. The values in brackets show the pseudo weights obtained for the respective objective

Sl. No.	Sum of squared deficits of irrigation releases $(Mm^3)^2$	Hydropower (M kWh)	Water quality satisfaction level
1	868·18 (0·60)	146·94 (0·02)	0·66 (0·39)
2	32 580·11 (0·56)	199·25 (0·44)	0·03 (0·00)
3	1 56 619·52 (0·00)	232·14 (0·50)	1·00 (0·50)
4	99 357·97 (0·16)	223·42 (0·40)	1·00 (0·44)
5	64 860·54 (0·28)	209·53 (0·36)	0·77 (0·36)
6	40 728·10 (0·31)	202·89 (0·28)	1·00 (0·42)
7	24 705·84 (0·37)	192·85 (0·24)	0·91 (0·39)
8	252·88 (0·93)	144·72 (0·00)	0·11 (0·07)
9	**2796·60 (0·51)**	**161·66 (0·10)**	**0·77 (0·39)**
10	1903·16 (0·68)	161·25 (0·13)	0·31 (0·20)
11	69 078·16 (0·25)	210·89 (0·34)	0·89 (0·40)
12	8392·71 (0·56)	177·37 (0·22)	0·40 (0·23)
13	1 30 367·73 (0·08)	227·08 (0·45)	1·00 (0·47)
14	9315·25 (0·47)	178·42 (0·19)	0·67 (0·33)
15	9851·96 (0·50)	179·74 (0·21)	0·55 (0·29)
16	44 503·89 (0·38)	205·14 (0·37)	0·47 (0·24)
17	3686·70 (0·74)	170·16 (0·22)	0·07 (0·03)
18	19 680·38 (0·58)	189·37 (0·34)	0·16 (0·09)
19	81 005·89 (0·21)	217·66 (0·36)	1·00 (0·43)
20	1542·24 (0·74)	160·49 (0·13)	0·20 (0·13)

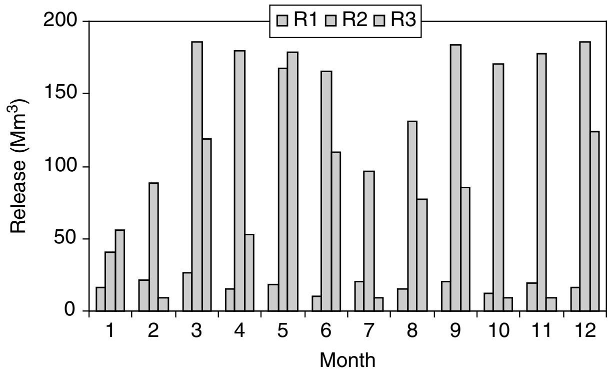

Figure 7. Release policy obtained for selected optimal point, showing releases for left bank canal (R1), right bank canal (R2) and river bed (R3)

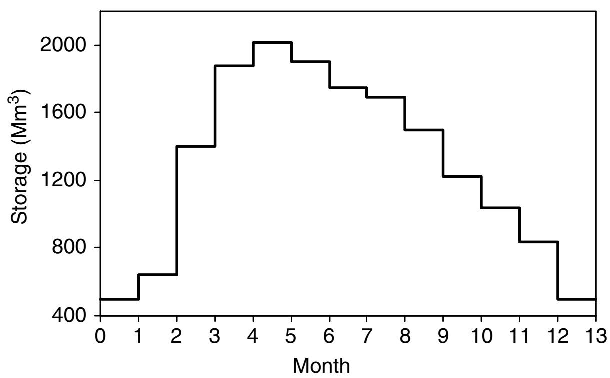

Figure 8. Monthly initial storages to be maintained in the reservoir corresponding to the selected optimal point

with Pareto dominance criteria to evolve non-dominated solutions. It uses a variable size ERP and a crowded comparison operator to promote solution diversity. In addition, a special EM operator is incorporated into the algorithm. This strategic mechanism keeps diversity in the population and consequently helps for effective exploration of Pareto optimal front. The proposed EM-MOPSO was first tested for a few standard test problems from the literature and it was found that the approach is quite robust and very competitive to NSGA-II in terms of yielding a diverse set of solutions along the true Pareto optimal fronts.

On achieving satisfactory performance for test problems, EM-MOPSO is applied to a reservoir operation problem, namely the Bhadra reservoir project. The multiple objectives involve minimization of irrigation deficit, maximization of hydropower and maximization of satisfaction level of downstream water quality requirements. First a two-objective model is solved and EM-MOPSO efficiency is demonstrated by comparing it with the results of NSGA-II. The results obtained clearly show the superiority of the proposed approach. Then the EM-MOPSO approach is extended to a three-objective model and many Pareto optimal solutions are generated. A clustering algorithm is employed to reduce the large set of Pareto optimal solutions to a small number of convenient representative alternatives. To facilitate ease in decision making, a pseudo-weight vector approach is employed. This provides an idea about the relative weight of each alternative and its preference over others. By analyzing the weight combinations, depending on the preference of the reservoir operator, a suitable policy can be implemented.

The main advantages of the proposed EM-MOPSO approach are that it is easy to implement and easy to use, and yet robust in yielding efficient Pareto frontiers. Hence it can be concluded that, for multi-objective water resources and hydrology problems, the proposed technique is a viable tool for multi-objective analysis and decision making, and can be used in any practical situation.

APPENDIX A

Set coverage metric

This metric gives the relative spread of solutions between two sets of solution vectors A and B. The set coverage metric calculates the proportion of solutions in B, which are weakly dominated by solutions of A (Deb, 2001).

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \leq b\}|}{|B|} \tag{21}$$

the value C(A, B) = 1 means that all solutions in B are weakly dominated by A, while C(A, B) = 0 represents the situation when none of the solutions in B are weakly dominated by A. Since the domination operator is not symmetric, i.e. C(A, B) is not necessarily equal to 1-C(B, A), it is necessary to calculate both C(A, B) and C(B, A)

DOI: 10.1002/hyp

to understand how many solutions of A are covered by B and vice versa.

Spacing metric

The spacing metric aims at assessing the spread (distribution) of vectors throughout the set of non-dominated solutions. It is calculated with a relative distance measure between consecutive solutions in the obtained non-dominated set (Deb, 2001):

$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \overline{d})^2} \tag{22}$$

where $d_i = \min_{k\in Q \wedge k\neq i} \sum_{m=1}^{M} |f_m^i - f_m^k|$ and $\overline{d}$ is the mean value of the distance measure $\overline{d} = \sum_{i=1}^{|Q|} d_i \Big/ |Q|$. f_m^k and f_m^i are the values of m objective function for k and i^{th} member in the population. The desired value for this metric is zero, which means that the elements of the set of non-dominated solutions are equidistantly spaced.

REFERENCES

Allen RE, Pereira LS, Raes D, Smith M. 1998. *Crop Evapotranspiration, Guidelines for Computing Crop Water Requirements*, *FAO Irrigation and Drainage Paper 56*. Food and Agricultural Organization of the United Nations: Rome.

Coello CAC, Pulido GT, Lechuga MS. 2004. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **8**: 256–279.

Cohon JL, Marks DH. 1975. A review and evaluation of multiobjective programming techniques. *Water Resources Research* **11**(2): 208–220.

Croley TE II, Rao KNR. 1979. Multi-objective risks in reservoir operation. *Water Resources Research* **15**(4): 1807–1814.

Deb K. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons: Chichester.

Deb K, Pratap A, Agarwal S, Meyarivan T. 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation* **6**: 182–197.

Haimes YY, Tarvainen K, Shima T, Thadathil J. 1990. *Hierarchical Multiobjective Analysis of Large-scale Systems*. Hemisphere publishing co.: New York.

Hu X, Eberhart RC, Shi Y. 2003. Particle swarm with extended memory for multiobjective optimization. *2003 IEEE Swarm Intelligence Symposium Proceedings*. IEEE Service Center: Indianapolis, IN; 193–197.

Kennedy J, Eberhart RC. 2001. *Swarm Intelligence*. Morgan Kaufmann: San Mateo, CA.

Khu ST, Madsen H. 2005. Multiobjective calibration with Pareto preference ordering: an application to rainfall-runoff model calibration. *Water Resources Research* **41**(3): W03004, Doi:10·1029/2004WR003041.

Knowles JD, Corne DW. 2000. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation* **8**: 149–172.

Li X. 2003. A nondominated sorting particle swarm optimizer for multiobjective optimization. In *Proceedings Genetic and Evolutionary Computation—GECCO 2003—Part I*, Vol. 2723, Cantú-Paz E, Foster JA, Deb K, Davi L, Roy R, O'Reilly UM, Beyer HG, Standish RK, Kendall G, Wilson SW, Harman M, Wegener J, Dasgupta D, Potter MA, Schultz AC, Dowsland KA, Jonoska N, .Miller JF (eds.). LNCS: Berlin, Germany; 37–48.

Liang Q, Johnson LE, Yu YS. 1996. A comparison of two methods for multiobjective optimization for reservoir operation. *Water Resources Bulletin* **32**(2): 333–340.

Mohan S, Raipure DM. 1992. Multiobjective analysis of multireservoir system. *Journal of Water Resource Planning and Management ASCE* **118**: 356–370.

Nagesh Kumar D, Janga Reddy M. 2006. Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resource Planning and Management ASCE* (in press).

Oliveira R, Loucks DP. 1997. Operating rules for multireservoir systems. *Water Resources Research* **33**(4): 839–852.

Parsopoulos K, Vrahatis M. 2002. Particle swarm optimization method in multiobjective problems. *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002)*. ACM Press: Madrid; 603–607.

Raju KS, Nagesh Kumar D. 2004. Irrigation planning using genetic algorithms. *Water Resources Management* **18**(2): 163–176.

Tauxe GH, Inman RR, Mades DM. 1979. Multi-objective dynamic programming with application to reservoir. *Water Resources Research* **15**(6): 1403–1408.

Thampapillai DJ, Sinden JA. 1979. Tradeoffs for multiple objectives planning through linear programming. *Water Resources Research* **15**: 1028–1034.

Vrugt JA, Gupta HV, Bastidas LA, Bouten W, Sorooshian S. 2003. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resources Research* **39**(8): 1214, Doi:10·1029/2002WR001746.

Wang YC, Yoshitani J, Fukami K. 2005. Stochastic multiobjective optimization of reservoirs in parallel. *Hydrological Processes* **19**: 3551–3567.

Wardlaw R, Sharif M. 1999. Evaluation of genetic algorithms for optimal reservoir system operation. *Journal of Water Resources Planning and Management ASCE* **125**(1): 25–33.

Yapo PO, Gupta V, Soorishian S. 1998. Multi-objective global optimization for hydrological models. *Journal of Hydrology* **204**: 83–97.

Yeh WW-G. 1985. Reservoir management and operations models: a state-of-the-art review. *Water Resources Research* **21**(12): 1797–1818.

Yeh WW-G, Becker L. 1982. Multiobjective analysis of multreservoir operations. *Water Resources Research* **18**(5): 1326–1336.

Zitzler E, Thiele L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transaction on Evolutionary Computation* **3**: 257–271.