Performance evaluation of elitist-mutated multi-objective particle swarm optimization for integrated water resources management

M. Janga Reddy and D. Nagesh Kumar

ABSTRACT

Optimal allocation of water resources for various stakeholders often involves considerable complexity with several conflicting goals, which often leads to multi-objective optimization. In aid of effective decision-making to the water managers, apart from developing effective multi-objective mathematical models, there is a greater necessity of providing efficient Pareto optimal solutions to the real world problems. This study proposes a swarm-intelligence-based multi-objective technique, namely the elitist-mutated multi-objective particle swarm optimization technique (EM-MOPSO), for arriving at efficient Pareto optimal solutions to the multi-objective water resource management problems. The EM-MOPSO technique is applied to a case study of the multi-objective reservoir operation problem. The model performance is evaluated by comparing with results of a non-dominated sorting genetic algorithm (NSGA-II) model, and it is found that the EM-MOPSO method results in better performance. The developed method can be used as an effective aid for multi-objective decision-making in integrated water resource management.

Key words | decision-making, multi-objective optimization, particle swarm optimization, water resources management

M. Janga Reddy
Department of Civil Engineering,
Indian Institute of Technology Bombay,
Mumbai 400076,
India

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

INTRODUCTION

Most of the water resource systems serve multiple purposes and involve several conflicting goals. Efficient use of water for different stakeholders imposes considerable complexity and often leads to multi-objective optimization. In India, reservoirs are the major control structures storing surface water, supplying water for various purposes such as drinking water, irrigation, hydropower, flood control, environmental safety, etc. Owing to the scarcity of water resources in different regions of the country, there is a greater need to consider the different uses of water together and develop integrated water resources management (IWRM) models, so that the developed models can guide water managers in efficient utilization of the available resources. In this scenario, with conflicting goals, obtaining optimal solutions to integrated water management

problems is always a challenging task. In a multi-objective environment, in order to perceive the effect of a particular decision on the performance of individual goals, first, various alternatives need to be generated and then decisions need to be made. To solve multi-objective problems, 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 operational domains (Haimes *et al.* 1990).

In the past, for optimization of water resource systems, classical methods such as linear programming (LP), dynamic programming (DP) and nonlinear programming (NLP) have been widely applied to solve various types of problems and they were also used to generate the optimal

tradeoffs between multiple objectives in reservoir operation (Tauxe *et al.* 1979; Thampapillai & Sinden 1979; Liang *et al.* 1996). However, these conventional optimization methods are not suitable to solve multi-objective optimization problems, because these methods use a point-by-point approach, and the outcome is a single optimal solution. The enumerative based DP technique poses severe computational problems for a multi-purpose multi-reservoir system due to the increase in the number of state variables and the corresponding discrete states. In this method, a linear increase in the number of state variables causes an exponential increase in the computational time requirement. So, when DP is applied to larger dimensional problems it has the major problem of the curse of dimensionality. Also, the LP and NLP have essential approximation problems while dealing with discontinuous, non-differentiable, non-convex or multi-model objective functions (Deb 2001).

Recently, there has been increasing interest in biologically motivated adaptive systems for solving optimization problems. Apart from Evolutionary Algorithms (EAs), such as genetic algorithms and genetic programming, swarm intelligence (SI) algorithms are also very promising and are receiving wider attention because of their flexibility and effectiveness for optimizing complex systems. Among swarm intelligence algorithms, ant colony optimization (ACO) (Dorigo 1992), particle swarm optimization (PSO) and honey-bees mating optimization (HBMO), are some of the important population-based search and optimization methods.

Inspired by the foraging behavior of real ants in finding the shortest paths between food sources and their nest, Dorigo (1992) proposed a stochastic search algorithm, namely ant colony optimization, for solving hard combinatorial optimization problems. Whereas honey-bee mating optimization was inspired by the process of mating in real honey-bees (Abbass 2001; Haddad & Mariño 2007). Eberhart & Kennedy (1995), inspired by the social behavior of bird flocking or fish schooling, proposed the PSO as a population-based heuristic search technique for solving continuous optimization problems.

The swarm optimization techniques have also found successful applications in water resources. For example, application of ACO algorithms: Abbaspour *et al.* (2001)

employed ACO algorithms to estimate hydraulic parameters of unsaturated soils; Maier *et al.* (2003) used ACO algorithms to find a near global optimal solution to a water distribution system; Nagesh Kumar & Janga Reddy (2006) used ACO techniques for optimizing the reservoir release policies for a multi-purpose reservoir system. Application of honey-bees mating optimization (HBMO) algorithms: Haddad *et al.* (2006) applied HBMO for single reservoir operation and it is also applied for optimal control and operation of an irrigation pumping station (Haddad & Mariño 2007). Application of PSO algorithms: Nagesh Kumar & Janga Reddy (2007) applied the PSO technique for optimal operation of a multipurpose reservoir system and found very good performance as compared to the genetic algorithms.

The swarm optimization algorithms have some special features, such as flexible operators, not needing the use of gradients, ease in tackling mixed-integer problems, combinatorial problems, etc. However, it is necessary to put together different heuristic operators to make an effective search and they have to be tuned properly so as to make a balance between the conflicting aspects present in an optimization algorithm, namely exploitation of available resources and exploration of search space. In this study, one such swarm intelligence algorithm is improved and adapted for multi-objective optimization in water resource systems and its performance is evaluated by applying it to a real world case study.

Recent studies in multi-objective optimization problems (MOP) suggests that the conventional approaches often fail to yield true Pareto optimal solutions when the objective function is non-convex and consists of disconnected Pareto solutions, and they require human expertise and a good number of simulation runs in order to get sufficient trade-off solutions. In contrast, the population-based multi-objective evolutionary algorithms (MOEAs) are able to overcome those drawbacks and evolve a wider Pareto front in a single run without significant extra computational time over that of a single objective optimizer (Deb *et al.* 2002). In recent years, MOEAs are being widely used in diverse fields of real-world applications. For example, applications in water resources include operation and management of water distribution networks (Halhal *et al.* 1997; Prasad & Park 2004; Farmani *et al.* 2006), groundwater monitoring

(Reed *et al.* 2001, 2003), reservoir systems operation (Janga Reddy & Nagesh Kumar 2006, 2007a), etc.

More recently, utilizing the basic principles of the single-objective particle swarm optimization method, Janga Reddy & Nagesh Kumar (2007b) have developed an efficient and effective multi-objective algorithm, namely the elitist-mutated multi-objective particle swarm optimization (EM-MOPSO) algorithm and have tested its performance for several numerical optimization problems including engineering design problems. It was found that the EM-MOPSO results in superior performance to that of a standard multi-objective genetic algorithm technique, NSGA-II. In this paper, the performance of EM-MOPSO is further evaluated for a multi-objective water resources optimization problem. In the following sections, the details of the particle swarm principles and the working procedure of EM-MOPSO are presented.

PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) technique has evolved from a simple simulation model of the movement of social groups such as birds and fish (Kennedy & Eberhart 2001). The basis of this algorithm is that local interactions motivate the group behavior, and individual members of the group can profit from the discoveries and experiences of other members. Social behavior is modeled in PSO to guide a population of particles (the so-called swarm) and help the search to move towards the most promising area of the search space. The changes to 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 ith individual (particle) of the population (swarm) can be represented by a D-dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^{\mathrm{T}}$. The velocity (position change) of this particle can be represented by another D-dimensional vector, $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^{\mathrm{T}}$. The best previously visited position of the ith particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^{\mathrm{T}}$. 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:

$$v_{id}^{n+1} =$$

$$\chi\left[wv_{id}^n + c_1\mathrm{rand}()(p_{id}^n - x_{id}^n)/\Delta t + c_2\mathrm{rand}()\left(p_{gd}^n - x_{id}^n\right)/\Delta t \right] \tag{1}$$

$$x_{id}^{n+1} = x_{id}^n + \Delta t\, v_{id}^{n+1} \tag{2}$$

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 magnitude of the velocity; 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; rand() is a random number generator function, in [0,1]; Δt is the time step usually set as 1 and n is the iteration number.

ELITIST-MUTATED MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

First, brief concepts of multi-objective optimization are presented and then the EM-MOSPO algorithm is explained.

Multi-objective optimization

A general multi-objective optimization problem (MOP) can be defined as: minimize a function $f(x)$, subject to p inequality and q equality constraints:

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

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

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

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

Pareto optimality

The MOP solutions are usually derived through non-domination criteria, where the MOP should simultaneously optimize the vector function and produce Pareto optimal solutions. A 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 as good or better than x^* with respect to all objectives.

EM-MOPSO algorithm

The description of the EM-MOPSO algorithm is based on Janga Reddy & Nagesh Kumar (2007b). The main algorithm consists of initialization of the 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 external repository (ERP), and are used to guide the search particles. A variable size ERP is used in order to improve the performance of the algorithm and to save computational time during optimization. If the size of ERP exceeds the specified limit, then it is reduced by using the crowding distance assignment operator, which gives the density measure of the existing particles in the function space. Also an efficient elitist-mutation strategy was used for maintaining diversity in the population and for more intensive exploring of the search space. Optimal combination of various operators helps the multi-objective algorithm to find the true Pareto optimal front. The main operators used in this algorithm are as follows.

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 (Janga Reddy & Nagesh Kumar 2007b). As ERP stores the non-dominated solutions found in the previous iteration,

any one of the solutions can be used as a global guide. But it is necessary that the particles in the population move towards the sparse regions of the non-dominated solutions and also that it should speed up the convergence towards the true Pareto optimal region. To perform these tasks, the global best guide of the particles is selected from the restricted variable size ERP. This restriction on ERP is done using a crowding distance operator. This operator provides for those non-dominated solutions with the highest crowding distance values to be always preferred to remain 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 the sorting and crowding value calculations. Thus, for effective exploration of the function space, the size is initially set to 10% of the maximum ERP, and then the value is increased in a stepwise manner, so that at the stage of 90% of maximum iteration, it reaches the maximum size.

Elitist-mutation operator

To maintain diversity in the population and to explore the search space, a strategic mechanism called elitist-mutation is used in this methodology (Janga Reddy & Nagesh Kumar 2007a). This acts on a predefined number of particles. In the initial phase, this mechanism tries to replace the infeasible solutions with the mutated least-crowded particles of ERP and at a later phase, it tries to exploit the search space around the sparsely populated particles in ERP along the Pareto fronts. Thus the elitist-mutation operator helps to uniformly distribute the non-dominated solutions along the true Pareto optimal front. The steps involved in the elitist-mutation mechanism are given below.

1. Randomly select one of the objectives from m objectives. Sort the fitness function of the particles in descending order and get the index number (DSP) for the respective particles.
2. Use the crowding distance assignment operator and calculate the density of solutions in the external repository (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 elitist mutation on predefined number of particles (NM_{max}).

4. If the mutated value exceeds the bounds, then it is limited to the upper or lower bound. It may be noted that the velocity vector of the particle remains unchanged during this elitist-mutation step.

The pseudo-code of the elitist mutation operator is presented in Table 1.

The EM-MOPSO algorithm can be summarized in the following steps (Janga Reddy & Nagesh Kumar 2007*a*).

Step 1. Initialize population. Set iteration counter $t := 0$.

a. The current position of the *i*th particle X_i is initialized with random real numbers within the range of the specified decision variable; each particle velocity vector V_i is initialized with a uniformly distributed random number in [0,1].

b. Evaluate each particle in the population. The personal best position P_i is set to X_i.

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

Step 3. $t := t + 1$.

Step 4. Repeat the loop (step through *PSO* operators):

a. Select randomly a global best P_g for the *i*th particle from the ERP.

b. Calculate the new velocity V_i, based on Equation (1) and the new x_i by Equation (2).

c. Perform the PSO operations for all particles in the iteration.

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 the 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 this, use crowding distance operator to select the desired ones. Empty TempERP.

Step 10. Perform elitist-mutation 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.

In order to handle the constrained optimization problems, this study adopts the constraint handling mechanism proposed by Deb *et al.* (2002). To use all the steps mentioned above, the EM-MOPSO approach is coded in the user-friendly mathematical software package MATLAB 7.0 and is run on a PC/WindowsXP/512 MB RAM/2 GHz computer. To compare and evaluate the performance of the EM-MOPSO, a standard multi-objective genetic algorithm technique, namely NSGA-II (Deb *et al.* 2002), is also employed for the developed reservoir operation model.

Table 1 │ Pseudo-code for elitist mutation operator

```
For i = 1 to NMmax
l = DSP[i]
  g = intRnd(1, 0.1 * Rp)
  For d = 1 to dim
    if (rand < pem)
      X[l] [d]  =  ERP[g][d] + Sm  *  VR[d]  * randn
    else
      X[l] [d]  =  ERP[g][d]
  End For
End For
```

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

CASE STUDY DESCRIPTION

To evaluate the performance of EM-MOPSO for water resource management problems, a case study of the Hirakud reservoir project, located in Orissa state, India is considered. The project is situated at latitude 21°32′N and longitude 83°52′E. The location of the Hirakud dam in the Mahanadi river basin is shown in Figure 1. The reservoir has a live storage capacity of 5,375 Mm^3 (million cubic

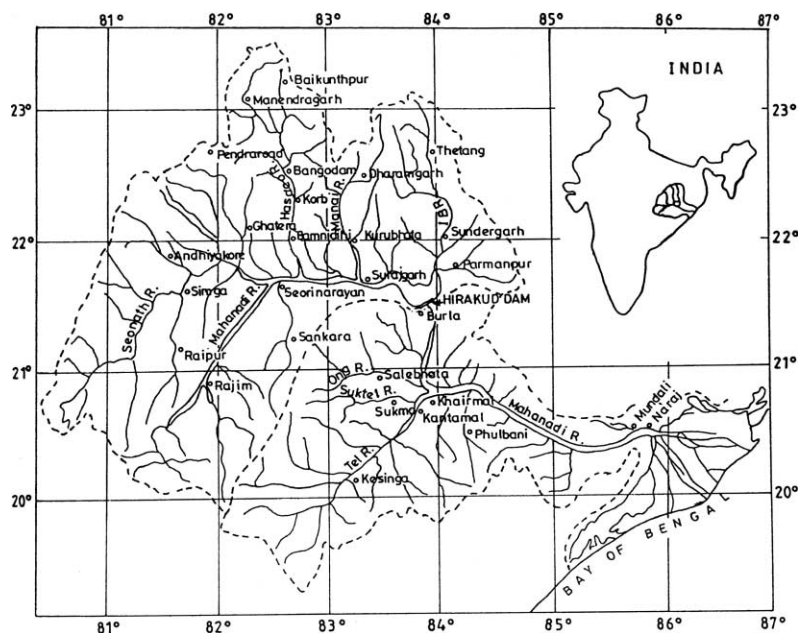**Figure 1** | Location map of Hirakud dam in Mahanadi Basin.

meters) and a gross storage of $7,189\,\text{Mm}^3$. The Hirakud project is a multi-purpose scheme and the water available in the dam is used in the following order of priority: for flood control, drinking water, irrigation and power generation. Since the drinking water requirement is very small in quantity, this is neglected in this particular model formulation. Water levels begin rising in July with the beginning of the monsoon season in the region, and begin declining in October, at the end of the season. During the monsoon season, the project provides flood protection to $9,500\,\text{km}^2$ of delta area in the districts of Cuttack and Puri. The project provides irrigation for 155,635 ha in the wet season (Kharif) and for 108,385 ha in the dry (Rabi) season in the districts of Sambalpur, Bargarh, Bolangir and Subarnpur. The water released through the powerhouses after power generation irrigates a further 436,000 ha of command area in the Mahanadi delta. The installed capacity of power generation is 259.5 MW from the powerhouse at Burla (PH-I) located on the right bank and 72 MW from the powerhouse at Chiplima (PH-II) located 22 km downstream of the dam. PH-I generates energy by utilizing water discharged directly from the Hirakud dam. Then the utilized water passes to PH-II through a power channel to generate further power at Chiplima. The reservoir inflow, utilization pattern and details of the dam were collected from the Department of Irrigation, Government of Orissa, India. The historic inflow data was available for 36 years from 1958 – 1993. For model formulation and operation, a time interval of 10 days is adopted over a year.

MODEL FORMULATION

The multiple purposes of the reservoir system causes a multi-objective problem, of minimizing flood risk, maximizing hydropower production and minimizing irrigation deficits in a year, subject to various physical and technical constraints. Among them, the flood control objective of a dam is in conflict with the other objectives of irrigation and hydropower generation. While for irrigation and hydropower, the reservoir has to be filled up as soon as it could be done and the level retained to be as high as possible, flood control requires a low water level and also quick depletion of the reservoir after a flood. As flood control is the major goal of the project, it is given high priority, compared to the other objectives during the monsoon season. From the historical time series of inflows and flood-prone periods, the reservoir authority adopts a set of safe guidelines to minimize flood risk in the downstream area and avoid losses to the maximum extent possible. To manage this goal,

the model incorporates the flood rule curve restrictions as constraints, so that the required priority is achieved. The model is formulated for ten daily operations, with the objectives of maximizing hydropower production (f_1) and minimizing the annual sum of squared deficits of irrigation release from demands (f_2). They are expressed as follows:

$$f_1 = \sum_{t=1}^{NT} (P_{1t} + P_{2t}) \qquad (5)$$

where

$$P_{i,t} = k_i{}^* RP_{i,t}{}^* H_{i,t} \qquad (6)$$

$$f_2 = \sum_{t=1}^{NT} [\min(0, IR_t - ID_t)]^2 \qquad (7)$$

subject to the following constraints:

$$S_{t+1} = S_t + I_t - RP_t - IR_t - EVP_t - OVF_t \quad \forall t \qquad (8)$$

$$S_t^{\min} \le S_t \le S_t^{\max} \quad \forall t \qquad (9)$$

$$RP_t^{\min} \le RP_{i,t} < TC_i \quad \forall t; \quad i = 1, 2 \qquad (10)$$

$$IR_t^{\min} \le IR_t \le IR_t^{\max} \quad \forall t \qquad (11)$$

where $P_{i,t}$ is the hydropower produced in MkWh in the ith powerhouse ($i = 1,2$) during t time period ($t = 1,2,\ldots,36$); NT = total number of time periods; k_i is power coefficient; $RP_{i,t}$ is the amount of water released to turbines during period t; $H_{i,t}$ is the average head available during period t and is expressed as a nonlinear function of the average storage during that period; IR_t is irrigation release in period t; ID_t is maximum irrigation demand in period t; P_t is the total hydropower produced in period t ($P_{1t} + P_{2t}$); $RP_t^{\min}$ is minimum release to meet downstream requirements; S_t is initial storage volume during time period t; I_t is inflow into the reservoir; EVP_t is the evaporation losses (a nonlinear function of the average storage); OVF_t is the overflow from the reservoir; $S_t^{\min}$ and $S_t^{\max}$ are minimum and maximum storages allowed in time period t, respectively. $IR_t^{\min}$ and $IR_t^{\max}$ are minimum and maximum irrigation releases, respectively, in time period t; TC_i is the turbine capacity of power plant i ($i = 1,2$).

In addition to the above constraints (Equations (8)–(11)), it is to be ensured that the storage at the end of the last period of the year is greater than or equal to the initial storage of the first period of the next year.

RESULTS AND DISCUSSION

The sensitivity analysis of the PSO model is performed with different combinations of each parameter. During this analysis, it is observed that by considering proper values for the constriction coefficient (χ), the inertial weight (w) does not have much influence on the final result of the model. So in this study the inertial weight w is fixed as 1. It is observed that the value of the constriction coefficient χ equal to 0.9 yields better results for the given model. In this analysis, it is also found that the cognitive parameter $c_1 = 1.0$ and social parameter $c_2 = 0.5$ result in better quality solutions. For running the reservoir operation model, the initial population of the EM-MOPSO is set to 200; the number of non-dominated solutions to be found is set to 200. For the elitist-mutation step, the size of the 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. The EM-MOPSO is run for 500 iterations.

To compare the performance of the EM-MOPSO, a standard MOEA, NSGA-II, is also applied to the developed reservoir operation model. 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 EM-MOPSO and NSGA-II are applied to the developed model. A sample result of a typical run is shown in Figure 2. Here f_1 the first objective (annual sum of irrigation deficits) is a minimization type, and f_2 the second objective (annual hydropower production) is a maximization type objective. Both the models have generated large numbers of solutions and it can be seen that the Pareto optimal front is showing a nonlinear relationship between the two objectives.

To check the performance of the EM-MOPSO and NSGA-II models, 20 independent runs were carried out for the reservoir operation model using both algorithms. Two performance measures, viz. set coverage metric (SC) and
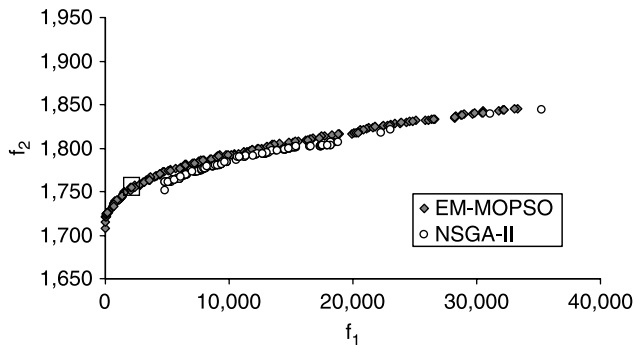
Figure 2 │ Pareto optimal solutions obtained for the Hirakud reservoir operation problem by EM-MOPSO and NSGA-II, for a typical run (f_1 = annual squared deficit for irrigation ($\times 10^6 \, \mathrm{m}^3)^2$ and f_2 = hydropower production $\times 10^6$ kWh).

spacing metric (SP) (Deb 2001) have been used to evaluate the quality of the solutions. The set coverage metric gives the relative convergence and domination of solutions between two sets of solution vectors U and V. The SC(U, V) metric calculates the proportion of solutions in V, which are weakly dominated by solutions of U. SC(U, V) = 1 means that all solutions in V are weakly dominated by U, while SC(U, V) = 0 represents the situation when none of the solutions in V are weakly dominated by U. The spacing metric (SP) aims at assessing the spread (distribution) of vectors throughout the set of non-dominated solutions. This indicates how far the generated non-dominated solutions are closer and equidistantly spaced. The desired value for this SP metric is zero, which means that the elements of the set of non-dominated solutions are very well distributed along the Pareto front.

Table 2 shows the resulting performance statistics for both the EM-MOPSO and NSGA-II models based on 20

independent runs for both algorithms. It can be observed that, with respect to the set coverage metric, the average value of SC(U, V) is higher than the SC(V,U) value (here U is EM-MOPSO and V is NSGA-II). The metric SC(U, V) shows the percentage of solutions in V that are weakly dominated by solutions of U. Thus, in this case, EM-MOPSO is performing better than NSGA-II. Regarding the spacing metric, it can be observed that the mean value of the SP metric for NSGA-II is lower than for EM-MOPSO. This indicates that distribution of Pareto optimal solutions is closer equidistantly distributed in NSGA-II than in EM-MOPSO. However, Deb *et al.* (2002) mentioned that it is always desired to have better performance in respect of the set coverage metric first, since this metric tells about which algorithm is achieving better convergence to true Pareto optimal solutions. Thus EM-MOPSO results in better performance. In Figure 2 also it can be seen that EM-MOPSO results in a wide spread of Pareto optimal solutions, with better convergence as compared to NSGA-II.

Thus the EM-MOPSO algorithm can solve the problem for different kinds of complexities such as non-convex, disconnected Pareto front, multiple solutions, and evolves a widespread Pareto front in a single run without much significant extra computational effort to that of the single-objective optimizer. This will help the decision-maker to analyze the trade-off solutions and to implement a suitable policy incorporating the preferences of the various stakeholders.

Decision-making

The operating policy corresponding to each noninferior solution is called a satisfactory operating policy and it can be discriminated from the optimal operating policy of the single-objective optimization. There are many ways to select the final compromising solution. However, this may require the decision-maker's analysis and interpretation. In this study for final decision-making, the Tchebycheff metric-based compromise programming approach (Deb 2001) is adopted. The method of compromise programming picks up a solution which is minimally located from a given reference point. From the generated solutions, first we have to fix a distance metric $d(f, z)$ and a reference point z for this purpose. Then

Table 2 │ Results of EM-MOPSO and NSGA-II for the reservoir operation model. In SC(U, V), U is EM-MOPSO and V is NSGA-II. Bold numbers indicate the best performing algorithm

| | Performance metric | | | |
| | Set coverage metric | | Spacing metric | |
Statistic	SC(U, V)	SC(V, U)	SP(U)	SP(V)
Best	1.0000	0.2757	100.13	12.22
Worst	0.0000	0.0000	903.75	419.96
Mean	**0.5106**	0.0853	354.18	**124.81**
Variance	0.1950	0.0083	84047.73	28336.74
SD	0.4416	0.0912	316.30	168.34

the Tchebycheff metric is computed by

$$\text{Tchebycheff metric}: d(f,z) = \max_{m=1}^{M} \frac{(|f_m(x) - z_m|)}{\max_{x \varepsilon S}(f_m(x) - z_m)} \qquad (12)$$

where S is the entire search space; z_m is a reference solution for the mth objective function. The reference point comprises of the individual best objective function values, $z = (f_1^*, f_2^*, \ldots, f_M^*)^T$. Since this solution is non-existent, the decision-maker is interested in choosing a feasible solution which is closest to this reference solution. So the solution which has a smaller metric value is the desired one.

Using the Tchebycheff metric approach, the best compromised solution is found at a point (2048.58, 1754.39) on the EM-MOPSO-generated Pareto front (Figure 2). The results of this solution give the compromised decision for the optimal reservoir operation. For each alternative solution, the model gives detailed results. The decisions at reservoir level include reservoir releases for irrigation and hydropower in each time period, and other variables of interest are the storages, evaporation losses and overflows for each time period over a year. Figure 3 shows the corresponding ten daily water release policies for irrigation and hydropower purposes, and also the generated hydropower over a year. Figure 4 shows the reservoir storage policy for that corresponding Pareto optimal solution.

On applying the EM-MOPSO technique to a case study of the Hirakud reservoir project in India, it is found that the
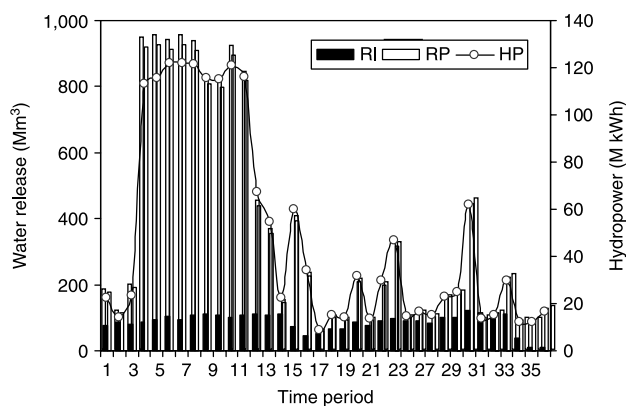

Figure 3 | Model results for reservoir operation using EM-MOPSO, showing the ten daily release policies and power generated, corresponding to the compromised optimal solution. RI = release to irrigation; RP = release to hydropower production; HP = hydropower.
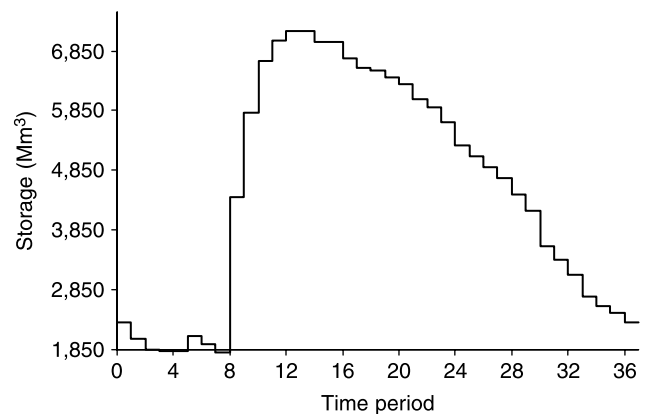

Figure 4 | The ten daily storage operating policy (corresponding to the compromised optimal solution) obtained for reservoir operation using EM-MOPSO.

method is effectively exploring the complex search space of the reservoir operation model, and is providing a wide spread of Pareto optimal solutions by simultaneously evolving the water release policies for different purposes. The EM-MOPSO approach generates a large number of Pareto optimal solutions in a single run and makes it easy for the decision-maker to choose the desired alternative as per individual preferences. Thus the swarm intelligence-based multi-objective algorithm, EM-MOPSO, can be used effectively to aid decision-making for multi-objective problems in integrated water resource management.

CONCLUSIONS

In this paper a multi-objective swarm intelligence algorithm, namely elitist-mutated multi-objective particle swarm optimization (EM-MOPSO), is presented for generating efficient Pareto optimal solutions in the operation and management of water resources. The EM-MOPSO approach uses several efficient operators for effective generation of Pareto optimal solutions, such as Pareto dominance criteria for selecting non-dominated solutions, an external repository (ERP) for storing the best solutions found, a crowding distance operator for creating effective selection pressure among the swarm to reach true Pareto optimal fronts, and incorporates an effective elitist-mutation strategy for intensive exploration of the search space. The developed method is applied to an integrated water resource management problem, a case study of optimal

operation of a multipurpose reservoir system in India. The efficiency of the EM-MOPSO approach is evaluated by comparing it with a standard multi-objective evolutionary algorithm, NSGA-II, and it is found that EM-MOPSO provides a wide spread of Pareto optimal solutions with better convergence than NSGA-II. Thus this study demonstrates the potential of the advanced computational technique, EM-MOPSO, for solving multi-objective decision problems in integrated water resource management and concludes that EM-MOPSO is an effective approach, which can guide water managers in multi-criterion decision-making, and helps in better utilization of the available water resources in the system.

REFERENCES

Abbaspour, K. C., Schulin, R. & van Genuchten, M. T. 2001 Estimating unsaturated soil hydraulic parameters using ant colony optimization. *Adv. Water Res.* **24**(8), 827–841.

Abbass, H. A. 2001 Marriage in honeybees optimization (MBO): a haplometrosis polygynous swarming approach. In: *Proc. Congress on Evolutionary Computation, CEC2001, Seoul, Korea, May* pp 207–214 doi: 10.1109/CEC.2001.934391.

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

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002 A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197.

Dorigo, M. 1992 *Optimization, Learning and Natural Algorithms*. PhD Thesis, Politecnico di Milano, Milan.

Eberhart, R. C. & Kennedy, J. 1995 A new optimizer using particle swarm theory. In *Proc 6th Symposium on Micro Machine and Human Science*, pp. 39–43. IEEE, IEEE Service Center, Piscataway, NJ.

Farmani, R., Walters, G. & Savic, D. 2006 Evolutionary multi-objective optimization of the design and operation of water distribution network: total cost vs. reliability vs. water quality. *J. Hydroinf.* **8**(3), 165–179.

Haddad, B. O., Afshar, A. & Mariño, M. A. 2006 Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. *Water Res. Manage.* **20**, 661–680.

Haddad, O. B. & Mariño, M. A. 2007 Dynamic penalty function as a strategy in solving water resources combinatorial optimization problems with honey-bee mating optimization (HBMO) algorithm. *J. Hydroinf.* **9**(3), 233–250.

Haimes, Y. Y., Tarvainen, K., Shima, T. & Thadathil, J. 1990 *Hierarchical Multiobjective Analysis of Large Scale Systems*. Hemisphere Publishing, New York.

Halhal, D., Walters, G. A., Ouazar, D. & Savic, D. A. 1997 Water network rehabilitation with a structured messy genetic algorithm. *J. Water Res. Plan. Manage.* **123**(3), 137–146.

Janga Reddy, M. & Nagesh Kumar, D. 2006 Optimal reservoir operation using multi-objective evolutionary algorithm. *Water Res. Manage.* **20**(6), 861–878.

Janga Reddy, M. & Nagesh Kumar, D. 2007a An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Eng. Optimiz.* **39**(1), 49–68.

Janga Reddy, M. & Nagesh Kumar, D. 2007b Multi-objective differential evolution with application to reservoir system optimization. *J. Comput. Civil Eng.* **21**(2), 136–146.

Kennedy, J. & Eberhart, R. C. 2001 *Swarm Intelligence*. Morgan Kaufmann, San Mateo, CA.

Liang, Q., Johnson, L. E. & Yu, Y. S. 1996 A comparison of two methods for multiobjective optimization for reservoir operation. *Water Res. Bull.* **32**, 333–340.

Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y. & Tan, C. L. 2003 Ant colony optimization for design of water distribution systems. *J. Water Res. Plan. Manage.* **129**(3), 200–209.

Nagesh Kumar, D. & Janga Reddy, M. 2006 Ant colony optimization for multipurpose reservoir operation. *Wat. Resour. Manage.* **20**(6), 879–898.

Nagesh Kumar, D. & Janga Reddy, M. 2007 Multipurpose reservoir operation using particle swarm optimization. *J. Water Res. Plan. Manage.* **133**(3), 192–201.

Prasad, T. D. & Park, N. S. 2004 Multiobjective genetic algorithms for design of water distribution networks. *J. Water Res. Plan. Manage.* **130**(1), 73–82.

Reed, P., Minsker, B. S. & Goldberg, D. E. 2001 A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data. *J. Hydroinf.* **3**, 71–89.

Reed, P., Minsker, B. S. & Goldberg, D. E. 2003 Simplifying multiobjective optimization: an automated design methodology for the nondominated sorted genetic algorithm-II. *Water Resour. Res.* **39**(7), 1196.

Tauxe, G. H., Inman, R. R. & Mades, D. M. 1979 Multi-objective dynamic programming with application to reservoir. *Water Resour. Res.* **15**, 1403–1408.

Thampapillai, D. J. & Sinden, J. A. 1979 Tradeoffs for multiple objectives planning through linear programming. *Water Resour. Res.* **15**, 1028–1034.