Multiobjective Differential Evolution with Application to Reservoir System Optimization

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

Abstract: Many water resources systems are characterized by multiple objectives. For multiobjective optimization, typically there can be no single optimal solution which can simultaneously satisfy all the goals, but rather a set of technologically efficient noninferior or Pareto optimal solutions exists. Generating those Pareto optimal solutions is a challenging task and often difficulties arise in using the conventional methods. In the optimization of reservoir systems, most of the times there is interdependence among one or more decision variables. Recently, it is emphasized that the evolutionary operators used in differential evolution algorithms are very much suitable for problems having interdependence among the decision variables. This paper utilizes this aspect and presents an efficient and effective approach for multiobjective optimization, namely multiobjective differential evolution (MODE) algorithm with an application to a case study in reservoir system optimization. The developed MODE algorithm is first tested on a few benchmark test problems and validated with standard performance measures by comparing them with the nondominated sorting genetic algorithm-II. On achieving satisfactory performance for test problems, it is applied to generate Pareto optimal solutions to a multiobjective reservoir operation problem. It is found that MODE provides many alternative Pareto optimal solutions with uniform coverage and convergence to true Pareto optimal fronts. The results obtained show that the proposed MODE can be a viable alternative for generating optimal trade-offs in multiobjective optimization of water resources systems.

DOI: 10.1061/(ASCE)0887-3801(2007)21:2(136)

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

Introduction

Many real-world systems involve multiobjective optimization in their operation. Multiobjective optimization problems (MOPs) require the simultaneous optimization of several noncommensurable and often competitive/conflicting objectives. Because of the multiple conflicting objectives, it is not possible to find a single optimal solution, which will satisfy all the goals. Instead, the solution exists in the form of alternative trade-offs, also known as the Pareto optimal solutions. For example, a reservoir system, which serves multiple purposes, involves multiobjective optimization in its implementation. In the past, many researchers have used classical optimization techniques such as linear programming, dynamic programming, and nonlinear programming to solve the multiobjective problems, by adopting a weighted approach or a constrained approach, without considering all the objectives simultaneously (Croley and Rao 1979; Haimes and Hall 1974; Tauxe et al. 1979; Thampapillai and Sinden 1979; Liang et al. 1996).

Despite having some useful applications for single objective optimization, most of the traditional methods are not suitable approaches for multiobjective optimization, because these methods use a point-by-point search approach, and the outcome for which is a single optimal solution. They often fail in yielding true Pareto optimal solutions, when the objective function is nonconvex and consists of disconnected Pareto fronts (Deb 2001). They also require human expertise and a number of simulation runs in order to get the trade-off behavior of solutions and often it is difficult to obtain entire Pareto optimal solutions, especially for problems of a large scale. Typically, real-world problems contain complex search spaces, where the quality of decision alternatives is evaluated through simulation using sophisticated computer models. This usually prevents conventional optimization techniques from being applicable. The main goal of this paper is to present an efficient multiobjective optimization algorithm, which will overcome the problems explained previously and demonstrate its applicability and efficiency through a case study in water resources systems optimization.

Recent studies emphasized that evolutionary algorithms are attractive alternatives for solving MOPs and are used for solving many practical problems because they are independent of the problem representation (Deb 2001). Typically, the population based evolutionary multiobjective optimization (EMO) methods are able to evolve good Pareto fronts in a single run without significant extra computational time over that of a single objective optimizer which can find just one solution on the front. Hence, treating a multiobjective real-world problem as it really should be treated, i.e., "without simplifying and changing it to a single objective pseudoreal problem" is becoming a viable, fast, and effective alternative. In recent years, this is becoming more evident in diverse fields of real-world applications. For example, EMOs have been used in the water resource projects of designing and

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

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

Note. Discussion open until August 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on January 9, 2006; approved on May 8, 2006. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 21, No. 2, March 1, 2007. ©ASCE, ISSN 0887-3801/2007/2-136–146/$25.00.

operation of reservoir management, water distribution networks, and groundwater monitoring (Janga Reddy and Nagesh Kumar 2006; Halhal et al. 1997; Devi Prasad and Park 2004; Reed et al. 2001).

Traditional genetic algorithms that use low mutation rates and fixed step sizes may face trouble with problems having interdependent relationships between decision variables and may require more number of function evaluations (Salomon 1996). Many water resources and hydrological problems might involve interdependence relationships among the decision variables of the optimization model. Interdependencies among the decision variables can be effectively handled, by properly rotating the coordinate system of the given function as it is done in differential evolution (Price 1999). Differential evolution has all the desired properties necessary to handle complex problems with interdependencies between input parameters, without the implementation complexity and computation cost of some of the self-adaptive evolutionary computation techniques, such as evolutionary strategies (Price 1999). They maintain correlated self-adapting mutation step sizes in order to make timely progress in optimization. In reservoir operation, there is often interdependence among one or more decision variables. Thus, the evolutionary operators used in differential evolution algorithm are very much suitable to tackle these problems. This paper utilizes this aspect and presents a novel approach for multiobjective optimization, namely, multiobjective differential evolution (MODE) to generate operational trade-offs to reservoir operation problems.

Background

Multiobjective Optimization

A general formulation for multiobjective optimization problem can be described as

$$\text{Minimize } f(x)=\{f_1(x),f_2(x),\ldots,f_m(x)\},\quad x\in S \tag{1}$$

where $f_i(x)$ $(i=1,2,\ldots,m)$=scalar objective function which maps decision variable x into the objective space $f_i=\mathbf{R}^n\rightarrow\mathbf{R}$.

The n-dimensional variable x is constrained to lie in a feasible region (S) and R=set of real numbers. The feasible region is constrained by J-inequality and K-equality constraints, i.e.

$$S=\{x:g_j(x)\geqslant 0,\quad h_k(x)=0,\quad j=1,2,\ldots,J,\quad k=1,2,\ldots,K\} \tag{2}$$

In MOP, the desired goals are often conflicting against each other and it is not possible to satisfy all the goals at a time. Hence, it gives a set of noninferior solutions also known as *Pareto optimal* solutions (Cohon 1978; Goioechea et al. 1982; Deb 2001). The definition of these Pareto optimal concepts is given as follows.

Let $x=(x_1,\ldots,x_k)$, and $y=(y_1,\ldots,y_k)$, be two vectors. Then, x dominates y if and only if, (1) $x_i\leqslant y_i$ for all $i=1,\ldots,m$; and (2) $x_i<y_i$ for at least one i.

This property is known as *Pareto dominance* and it is used to define *Pareto optimal* points. In other words, a solution, x, of the MOP is said to be Pareto optimal if and only if, there does not exist another solution y, such that $f_i(y)$ dominates $f_i(x)$ for all $i=1,2,\ldots,m$. The set of all Pareto optimal solutions of an MOP is called *Pareto optimal set* and is denoted as Q^*. The set, $\text{PF}^*=\{[f_1(x),\ldots,f_m(x)]^T|x\in Q^*\}$, is called *Pareto front*.

Differential Evolution

Differential evolution (DE) is a recent optimization technique in the family of evolutionary computation. It is proposed as a variant of genetic algorithms to achieve the goals of robustness in optimization and faster convergence to a given problem. DE differs from other evolutionary algorithms in the mutation and recombination phase. Unlike some metaheuristic techniques such as genetic algorithms and evolutionary strategies, where perturbation occurs in accordance with a random quantity, DE uses weighted differences between solution vectors to perturb the population (Storn and Price 1995).

The general convention used for different variants of DE is DE/α/β/γ. Here DE stands for differential evolution algorithm, α represents a string denoting the vector to be perturbed; β=number of difference vectors considered for perturbation of α; and γ=type of crossover being used (exp=exponential; bin=binomial). Here the perturbation can be made either in the best vector of the previous generation (best) or in any randomly chosen vector (rand). Similarly for perturbation, either single or two vector differences can be used. For perturbation with a single vector difference, out of the three distinct randomly chosen vectors, the weighted vector differential of any two vectors is added to the third one. Similarly for perturbation with two vector differences, five distinct vectors other than the target vector are chosen randomly from the current population. Out of these, the weighted vector difference of each pair of any four vectors is added to the fifth one for perturbation. In binomial crossover, the crossover is performed on each of the decision variables whenever a randomly picked number between 0 and 1 is within the crossover constant (CR) value. As earlier studies reported, that both variants of crossover operators having similar kind of performance (Storn and Price 1997; Price 1999; Onwubolu and Davendra 2006), so in this study we evaluated only bin variants of DE. A brief description of the algorithm is presented below.

DE Algorithm

Let $S\subset R^n$ be the search space of the problem under consideration. Then, the DE algorithm utilizes NP (population size), n-dimensional vectors

$$X_i=(x_{i1},\ldots,x_{\text{in}})^T\in S,\quad i=1,\ldots,\text{NP}$$

as a population for each iteration, called a generation, of the algorithm. The initial population is usually taken to be uniformly distributed in the search space. At each generation, two operators, namely mutation and crossover are applied on each individual, thus producing the new population. Then, a selection phase takes place, where each individual of the new population is compared to the corresponding individual of the old population, and the best between them is selected as a member of the population in the next generation (Storn and Price 1995).

According to the *mutation* operator, for each individual, $X_i^{(G)}$, $i=1,\ldots,\text{NP}$, at generation G, a mutation vector

$$V_i^{(G+1)}=(v_{i1}^{(G+1)},v_{i2}^{(G+1)},\ldots,v_{in}^{(G+1)})^T$$

is determined using one of the equations from Eqs. (3)–(7) (Storn and Price 1997). Here the choice of equation, dictates the variant of DE to be used in the model application. Therefore, use of Eqs. (3)–(7) lead to different variants of DE, such as DE/rand/1/bin, DE/best/1/bin, DE/rand-to-best/1/bin, DE/rand/2/bin, and DE/best/2/bin, respectively

```
Generate initial population of size NP
Do while
    For each individual, j, in the population
        Generate three random integers, r1, r2, r3 ∈ (1, NP), with r1 ≠ r2 ≠ r3 ≠ j
        Generate a random integer i_rand ∈ (1, n)
        For each parameter i
                    { x_{i,r3} + F * (x_{i,r1} − x_{i,r2})    if rand(0,1) < CR  or i = i_rand
            x'_{i,j} = {
                    { x_{i,j}                                  otheriwse
        End For
        Replace x_j with the child x'_j, if x'_j is better
    End For
Until the termination condition is satisfied
```

Fig. 1. Pseudocode of the differential evolution algorithm (DE/rand/1/bin)

$$V_i^{(G+1)} = X_{r1}^{(G)} + F[X_{r2}^{(G)} - X_{r3}^{(G)}] \quad (3)$$

$$V_i^{(G+1)} = X_{\text{best}}^{(G)} + F[X_{r1}^{(G)} - X_{r2}^{(G)}] \quad (4)$$

$$V_i^{(G+1)} = X_i^{(G)} + F[X_{\text{best}}^{(G)} - X_i^{(G)}] + F[X_{r1}^{(G)} - X_{r2}^{(G)}] \quad (5)$$

$$V_i^{(G+1)} = X_{\text{best}}^{(G)} + F[X_{r1}^{(G)} - X_{r2}^{(G)}] + F[X_{r3}^{(G)} - X_{r4}^{(G)}] \quad (6)$$

$$V_i^{(G+1)} = X_{r1}^{(G)} + F[X_{r2}^{(G)} - X_{r3}^{(G)}] + F[X_{r4}^{(G)} - X_{r5}^{(G)}] \quad (7)$$

where $X_{\text{best}}^{(G)}$=best individual of the population at generation G; $F>0$=real parameter, called mutation constant, which controls the amplification of the difference between two individuals so as to avoid search stagnation; and r_1, r_2, r_3, r_4, and r_5, are mutually different integers, randomly selected from the set $\{1,2,\ldots,i-1,i+1,\ldots,\text{NP}\}$.

Following the mutation phase, the *crossover* operator is applied on the population. For each mutant vector, $V_i^{(G+1)}$, an index $\text{rnbr}(i) \in \{1,2,\ldots,n\}$ is randomly chosen, and a trial vector

$$U_i^{(G+1)} = [u_{i1}^{(G+1)}, u_{i2}^{(G+1)}, \ldots, u_{\text{in}}^{(G+1)}]^T$$

is generated, with

$$u_{ij}^{(G+1)} = \begin{cases} v_{ij}^{G+1} & \text{if } [\text{rand } b(j) \leqslant \text{CR}] \text{ or } [j = \text{rnbr}(i)] \\ x_{ij}^{G} & \text{if } [\text{rand } b(j) > \text{CR}] \text{ and } [j \neq \text{rnbr}(i)] \end{cases} \quad (8)$$

where, $j=1,2,\ldots,n$; rand $b(j)=j$th evaluation of a uniform random number generator within $[0,1]$ and CR=user defined crossover constant in the range $[0,1]$ (Storn and Price 1997). In other words, the trial vector consists of some of the components of a randomly selected individual of the population, i.e., the individual with index rnbr(i).

To decide whether the vector $U_i^{(G+1)}$ should be a member of the population of the next generation, it is compared to the corresponding vector $X_i^{(G)}$. Thus, if f denotes the objective function under consideration, then

$$X_i^{(G+1)} = \begin{cases} U_i^{G+1} & \text{if } f(U_i^{G+1}) < f(X_i^G) \\ X_i^G & \text{otherwise} \end{cases} \quad (9)$$

Thus, each individual of the trial vector is compared with its parent vector and the better one is passed to the next generation, so the elitism (the best individuals in the population) is preserved. These steps are repeated until specified termination criterion is reached. The pseudocode for DE algorithm (DE/rand/1/bin) is given in Fig. 1.

This DE technique has proved significantly faster and robust for numerical optimization (Storn and Price 1997) and is also capable of optimizing all integers, discrete and continuous variables, and can handle nonlinear objective functions with multiple nontrivial solutions (Onwubolu and Davendra 2006). The ability to provide efficient solutions for complex problems and simpler operations of DE is very much attractive and encouraging to develop MODE technique.

Multiobjective Differential Evolution

Recently, researchers attempted to extend DE technique to multiobjective optimization and showed that DE can be an attractive alternative for multiobjective numerical optimization (Abbas and Sarker 2002; Madavan 2002; Xue 2003; Parspolous et al. 2004; Robič and Filipič 2005). In this study, the MODE is developed by integrating nondominated sorting, ranking, and crowding distance assignment procedures (Deb et al. 2002) with DE. This also maintains an external archive to maintain the best noninferior solutions explored over the generations. The details of MODE methodology are described in the following.

MODE Methodology

Handling of multiple objectives with DE, poses certain difficulties in its implementation. Besides preserving a uniformly spread front of nondominated solutions in the process of reaching true Pareto optimal solutions, it is also necessary to take decision on when to replace the parent with the candidate solution. To achieve the multiobjective optimization goals, the MODE methodology combines Pareto-dominance principles with DE and uses elitism in its evolution.

The main algorithm consists of initialization of population, evaluation, Pareto-dominance selection, performing DE operations, and reiterating the search on population to reach true Pareto optimal solutions. In this process, the members are first evaluated and checked for dominance relation. If the new member dominates the parent, then it replaces the parent. If the parent dominates the candidate, the new member is discarded. If the parent and new member both are nondominated to each other, then these two are added to a temporary population (tempPop). This step is repeated for all members of the population. Thereafter, in order to select the population for next generation, the tempPop is reduced to NP by using nondominated ranking and crowding distance assignment procedures. Apart from that, this study uses *nondominated elitist archive* (NEA) to store the best solutions found so far over the generations. The size of NEA can be set to any desirable number of nondominated solutions. To maintain the consistency and for easiness in comparison with nondominated sorting genetic algorithm-II (NSGA-II), in this study, the size of NEA is set to NP. In case, the size of NEA exceeds NP, then a crowding operator is used to select the sparse individuals to create effective selection pressure toward true Pareto optimal solutions. The selection of best in DE algorithm is made by randomly choosing a solution from the elite archive, NEA. The proposed MODE methodology can be summarized in the following steps.

1. Input the required DE parameters. Initialize all the vector populations, randomly in the limits of specified decision variables.
2. Evaluate each member of the population. Identify individuals that give nondominated solutions in the current population and store them in NEA. Set generation counter, $G := 0$.
3. Perform mutation and crossover operations (as explained for

Table 1. Test Problems Used in This Study

Problem	Variable bounds		Objective functions and constraints
KUR	$x_i \in [-5,5]$ $i=1,\ldots,3$	Minimize	$f_1(x)=\Sigma_{i=1}^{n-1}(-10\exp(-0.2\sqrt{x_i^2+x_{i+1}^2}))$ $f_2(x)=\Sigma_{i=1}^{n}(\lvert x_i\rvert^{0.8}+5\sin(x_i^3))$
ZDT3	$x_i \in [0,1]$ $i=1,\ldots,30$	Minimize	$f_1(x)=x_1$ $f_2(x)=g(x)[1-\sqrt{x_1/g(x)}-(x_1/g(x))\sin(10\pi x_1)]$ $g(x)=1+9(\Sigma_{i=2}^{n}x_i)/(n-1)$
ZDT4	$x_1 \in [0,1]$ $x_i \in [-5,5]$ $i=2,\ldots,10$	Minimize	$f_1(x)=x_1$ $f_2(x)=g(x)[1-\sqrt{x_1/g(x)}]$ $g(x)=1+10(n-1)+\Sigma_{i=2}^{n}[x_i^2-10\cos(4\pi x_i)]$
ZDT6	$x_i \in [0,1]$ $i=1,\ldots,10$	Minimize	$f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x)=g(x)[1-(f_1(x)/g(x))^2]$ $g(x)=1+9[(\Sigma_{i=2}^{n}x_i)/(n-1)]^{0.25}$
CONSTR	$x_1 \in [0.1,1.0]$ $x_2 \in [0,5]$	Minimize	$f_1(x)=x_1$ $f_2(x)=(1+x_2)/x_1$
		Subject to	$g_1(x)=x_2+9x_1 \geqslant 6$ $g_2(x)=-x_2+9x_1 \geqslant 1$
SRN	$x_i \in [-20,20]$ $i=1,2$	Minimize	$f_1(x)=(x_1-2)^2+(x_2-1)^2+2$ $f_2(x)=9x_1-(x_2-1)^2$
		Subject to	$g_1(x)=x_1^2+x_2^2 \leqslant 225$ $g_2(x)=x_1-3x_2 \leqslant -10$

single objective DE) on all the members of the population, i.e., for each parent P_i

a. Select distinct vectors randomly from the current population (primary vector) other than the parent vector (i.e., randomly select $r_1, r_2, r_3 \ldots \in \{1,2,\ldots,n\}$, such that $r_1 \neq r_2 \neq r_3 \ldots \neq j$);
b. Calculate new mutation vector using one of the expressions from Eqs. (3)–(7);
c. Modify the mutated vector by binary crossover with the parent, using crossover probability CR [Eq. (8)]; and
d. Restrict the variables to its boundaries, if any variable is outside the lower or upper bound.

4. Evaluate each member of the population. Check for dominance with its parents. If the candidate dominates the parent, the candidate replaces the parent. If the parent dominates the candidate, the candidate is discarded. Otherwise, the candidate is added to a temporary population (tempPop).
5. Add the latest solution vectors (current population) to the tempPop. Then use the nondominated sorting and crowding assignment operators to select the individuals to next generation. Store the nondominated solutions in nondominated elite archive, NEA. If NEA size exceeds the desired number of Pareto optimal set, then select desired number of the least crowded members with the help of crowding assignment operator. Empty the tempPop.
6. Increment the generation counter, G to $G+1$ and check for termination criteria. If the termination criterion is not satisfied, then go to Step 3; otherwise output the nondominated solution set from NEA.

The details of the operators for nondominated ranking and crowding distance assignment can be found in Deb et al. (2002). Following the steps explained earlier, the algorithm is coded in MATLAB 6.5 (The MathWorks Inc., U.S.A.) and are executed on a 1.4 GHz, 512 MB RAM, Pentium 4 PC.

The developed MODE is first evaluated through few standard test problems to ensure that the algorithm is performing well for different types of complexity and then applied to a case study in reservoir system optimization. To evaluate the efficiency of MODE, the model results are compared with the results obtained using the Nondominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al. 2002). The C-programming code for NSGA-II is obtained from the web site of Kanpur Genetic Lab (KanGAL, ⟨http://www.iitk.ac.in/kangal/soft.htm⟩).

Test Problems and Results

The test problems considered in this study are only those which are reported as difficult to be solved by general MOP solvers (Deb et al. 2002). Therefore, there is a better chance to prove the efficiency of the MODE. The details of test problems considered are given in Table 1. In that, out of six, the first four problems involve unconstrained optimization (KUR, ZDT3, ZDT4, and ZDT6) and the other two are constrained optimization problems (CONSTR and SRN). All these test problems have different levels of complexity like convexity, nonconvex, and disconnected Pareto optimal solutions. The test problems KUR and ZDT3 are having disconnected objective space; ZDT4 is having complexity with too many local optimal solutions, whereas ZDT6 is having nonconvex Pareto optimal front with low density of solutions near Pareto front. CONSTR and SRN are both having complexity with

Table 2. Performance Evaluation of Different Variants of MODE with respect to NSGA-II, Showing Mean and Standard Deviation Values for Set Coverage Metric

Test case	Statistic	SC(A,R)	SC(R,A)	SC(B,R)	SC(R,B)	SC(C,R)	SC(R,C)	SC(D,R)	SC(R,D)	SC(E,R)	SC(R,E)
KUR	Mean	0.1190	0.1190	0.1360	0.1000	**0.1365**	0.1050	0.1120	0.1280	0.1110	0.1420
	SD	0.0213	0.0357	0.0357	0.0245	0.0250	0.0255	0.0235	0.0449	0.0370	0.0432
ZDT3	Mean	0.1320	0.0000	**0.1390**	0.0010	0.0130	0.3798	0.0810	0.0096	0.0690	0.0394
	SD	0.0432	0.0000	0.0498	0.0032	0.0116	0.0835	0.0335	0.0093	0.0251	0.0251
ZDT4	Mean	0.5370	0.0000	0.3710	0.1050	**0.5410**	0.0020	0.1030	0.2680	0.0000	0.8833
	SD	0.2469	0.0000	0.2710	0.3114	0.2475	0.0063	0.1024	0.2131	0.0000	0.0583
ZDT6	Mean	0.8950	0.0000	0.9070	0.0000	**0.9080**	0.0000	0.8870	0.0000	0.8940	0.0000
	SD	0.0643	0.0000	0.0591	0.0000	0.0391	0.0000	0.0574	0.0000	0.0659	0.0000
CONSTR	Mean	0.1620	0.1082	0.1420	0.1170	**0.1440**	0.1150	0.1020	0.2030	0.1000	0.2130
	SD	0.0343	0.0284	0.0358	0.0149	0.0412	0.0387	0.0346	0.0414	0.0327	0.0400
SRN	Mean	**0.1040**	0.0720	0.0940	0.0600	0.0880	0.0610	0.0830	0.0920	0.0900	0.0810
	SD	0.0369	0.0225	0.0295	0.0231	0.0494	0.0213	0.0211	0.0270	0.0333	0.0242

Note: The result are based on ten independent runs for the respective algorithms. Here A=MODE/best/1/bin; B=MODE/rand/1/bin; C=MODE/rand-to-best/1/bin; D=MODE/best/2/bin; E=MODE/rand/2/bin; and R=NSGA-II. Bold numbers indicate the best performing algorithm.

constrained optimization and non-smooth Pareto optimal fronts.

For any multiobjective problem the main goals in optimization are, minimizing the distance of the Pareto front produced by an algorithm with respect to the true Pareto front (i.e., convergence to true Pareto-optimal solution set) and maximizing the spread of solutions found (i.e., maintenance of diversity among the generated set of solutions). Also it should ensure that the generated solutions to be uniformly distributed along the true Pareto front. To test the performance of the MODE, two performance metrics have been used, viz., set coverage metric and spacing metric (Deb 2001). The details of these metrics are given in the following.

Set Coverage Metric

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

$$\mathrm{SC}(U,V) = \frac{|\{v \in V | \exists\, u \in U : u \leq v\}|}{|V|} \tag{10}$$

the value $\mathrm{SC}(U,V)=1$ means that all solutions in V are weakly dominated by U, whereas $\mathrm{SC}(U,V)=0$ represents the situation when none of the solutions in V are weakly dominated by U. As the domination operator is not symmetric, i.e., $\mathrm{SC}(U,V)$ is not necessarily equal to $1-\mathrm{SC}(V,U)$. Therefore, it is necessary to calculate both $\mathrm{SC}(U,V)$ and $\mathrm{SC}(V,U)$ to understand how many solutions of U are covered by V and vice versa.

Spacing Metric

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

$$\mathrm{SP} = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \tag{11}$$

where $d_i=\min_{k\in Q \wedge k\neq i}\sum_{m=1}^{M}|f_m^i - f_m^k|$ and $\bar{d}$=mean value of the distance measure $\bar{d}=\sum_{i=1}^{|Q|} d_i/|Q|$; f_m^i and f_m^k=values of mth objective function for ith and kth members in the population; and Q=Pareto optimal solution set. The desired value for this metric is zero, which means that the elements of the set of nondominated solutions are equidistantly spaced.

To apply the MODE algorithm, the parameters of F and CR are to be decided, for that a thorough sensitivity analysis is carried out. For mutation constant (F), the sensitivity is evaluated for different values of F, such as 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. For CR, the sensitivity is evaluated for different values of CR, such as 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. During this analysis, the following parameters are found to give good results. The initial population was set to 100; CR=0.3; and F=0.5. The maximum number of generations carried out is set to 250. For NSGA-II, the initial population was set to 100, crossover probability to 0.9, and mutation probability to $1/n$ (n=number of real variables). The SBX and real parameter mutation were set to 20 and 20, respectively. This algorithm is also run for 250 generations. Thus, both the algorithms are compared for the same number of function evaluations.

In order to find the best performing variant for the MODE algorithm, different variants of MODE are evaluated. The computations of different MODE variants can be recognized by using different equations for mutation operator, viz., Eqs. (3)–(7) for MODE/rand/1/bin, MODE/best/1/bin, MODE/rand-to-best/1/bin, MODE/rand/2/bin, and MODE/best/2/bin, respectively. The MODE algorithm variants have been applied to the above-described test problems. For each test problem, all the algorithms were run for ten independent trial solutions. Different algorithms are denoted as A=MODE/rand/1/bin; B=MODE/best/1/bin; C=MODE/rand-to-best/1/bin; D=MODE/rand/2/bin; E=MODE/best/2/bin, and R=NSGA-II.

In Tables 2 and 3, performance of different variants of MODE are compared with NSGA-II, showing the mean and standard deviation (SD) values for set coverage metric (SC) and spacing metric (SP), respectively. It is to be noted that these results are compared based on model performance in a run. The set coverage metric, $\mathrm{SC}(A,R)=1$, means that all solutions in R are weakly dominated by A, whereas $\mathrm{SC}(A,R)=0$ represents the situation when none of the solutions in R are weakly dominated by A. So the higher value of SC gives indication of better performance. From Table 2, as far as the SC metric is concerned, MODE performance is superior to NSGA-II, as it can be observed that all the six (KUR, ZDT3, ZDT4, ZDT6, CONSTR, and SRN) test cases are having higher SC values in MODE variants as compared to NSGA-II. Table 3 shows the spacing metric (SP) statistics for all

Table 3. Spacing Metric Values for Different Variants of MODE and for NSGA-II

Test case	Statistic	SP(A)	SP(B)	SP(C)	SP(D)	SP(E)	SP(R)
KUR	Mean	0.0937	0.0944	0.0910	0.0921	0.0897	**0.0823**
	SD	0.0020	0.0034	0.0048	0.0066	0.0019	0.0223
ZDT3	Mean	0.0105	0.0094	0.0092	0.0101	0.0103	**0.0079**
	SD	0.0016	0.0007	0.0006	0.0009	0.0010	0.0009
ZDT4	Mean	**0.0071**	0.0077	0.0080	0.0089	1.7218	0.0076
	SD	0.0004	0.0032	0.0014	0.0035	2.6112	0.0007
ZDT6	Mean	0.0222	0.0054	**0.0050**	0.0273	0.0051	**0.0050**
	SD	0.0532	0.0005	0.0006	0.0689	0.0003	0.0007
CONSTR	Mean	0.0436	0.0428	**0.0430**	0.0526	0.0657	0.0465
	SD	0.0029	0.0026	0.0044	0.0081	0.0224	0.0032
SRN	Mean	1.3108	1.4669	**1.2670**	1.2805	1.2675	1.5969
	SD	0.1625	0.2777	0.1000	0.0980	0.0992	0.1307

Note: Here A=MODE/best/1/bin; B=MODE/rand/1/bin; C=MODE/rand-to-best/1/bin; D=MODE/best/2/bin; E=MODE/rand/2/bin and R=NSGA-II. Bold numbers indicate the best performing algorithm.

the algorithms. This SP metric gives an idea of how uniformly a solution set is distributed along the Pareto optimal front. For spacing metric small values of SP are desirable. Generally this metric becomes important only when the generated solutions lie in true Pareto optimal frontiers. However, it can be observed that MODE variants are resulting in good distribution of solutions with smaller SP values. Bold numbers in Tables 2 and 3 indicate the best performing algorithm for each test case. From the previous results (Tables 2 and 3), it can be observed that, MODE/rand-to-best/1/bin variant is performing superior to all other variants of MODE.

In order to demonstrate the working of the algorithm, results of a typical simulation run with MODE/rand-to-best/1/bin algorithm and NSGA-II are shown in Fig. 2. The visual illustration also clearly demonstrates that the developed methodology is efficient and is able to achieve true Pareto optimal solutions for all the test problems. Next, we test the MODE algorithm to evaluate its performance on a complex, real-world problem.

Case Study Description

To demonstrate the efficiency of the proposed methodology for reservoir operation problems, a case study of Hirakud Reservoir project in Orissa state, India, is considered. The project is situated at latitude 21°32′ N and longitude 83°52′ E. The index map of Mahanadi River Basin showing the location of Hirakud dam is presented in Fig. 3. The reservoir has live storage capacity of $5{,}375\times10^6$ m^3 and a gross storage of $7{,}189\times10^6$ m^3. The Hirakud Reservoir is a multipurpose project, which serves for flood control, drinking water, irrigation, and for power generation; the water is used in the decreasing order of priority of this sequence. Since the drinking water requirement is a very small quantity, this quantity is neglected in this particular model formulation. Water levels begin rising in July, the beginning of monsoon season in the region, and begin declining in October, at the end of the season. During monsoon season, the project provides flood protection to 9,500 km^2 of delta area in the districts of Cuttack and Puri. The project provides irrigation for 155,635 ha in wet season (Kharif) and for 108,385 ha in dry (Rabi) season in the districts of Sambalpur, Bargarh, Bolangir, and Subarnpur. The water released through the powerhouses after power generation, irrigates further 436,000 ha of command area in Mahanadi delta. Installed capacity of power generation is 259.5 MW from powerhouse at Burla (PH-I) located at the right bank and 72 MW from powerhouse at Chiplima (PH-II) located at 22 km downstream of the dam. The PH-I generates energy by utilizing water discharged directly from the Hirakud Dam. Then the utilized water passes to the PH-II through a power channel to generate further power at Chiplima.

Orissa state is having plenty of water during the wet season, so there is greater possibility for hydropower improvement in that season. Net energy production is high during the monsoon period and low during the dry season. Many times, unless the region experiences unusually heavy rain in the dry season, power generation would not be possible in that season. Over a period of 36 years the average annual inflow is 3.36×10^6 ha m. The reservoir inflow, utilization pattern, and dam details were collected from the Department of Irrigation, Government of Orissa. The historic inflow data were available for 36 years, from 1958 to 1993. The model formulation and operation is for a time interval of ten daily periods in a year. So the ten daily data for 36 years is used in this study.

Model Formulation

The multiple objectives of the reservoir system are minimizing flood risk, maximizing hydropower production, and minimize irrigation deficits in a year, subject to various physical and technical constraints. Among them, flood control objective of a dam is in conflict with the other objectives of irrigation and hydropower generation. Although for irrigation and hydropower, the reservoir has to be filled up as soon as it could be done and the level retained at as high as possible, flood control requires low water level and also quick depletion of reservoir after a flood. As flood control is the major goal of the project, it should be given high priority, compared to other objectives during monsoon season. From historical time series of inflows and flood prone periods, the reservoir authority adopts 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 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 maximize hydropower production (f_1) and minimize the annual sum of squared deficits of irrigation release from demands (f_2). They are expressed as follows:

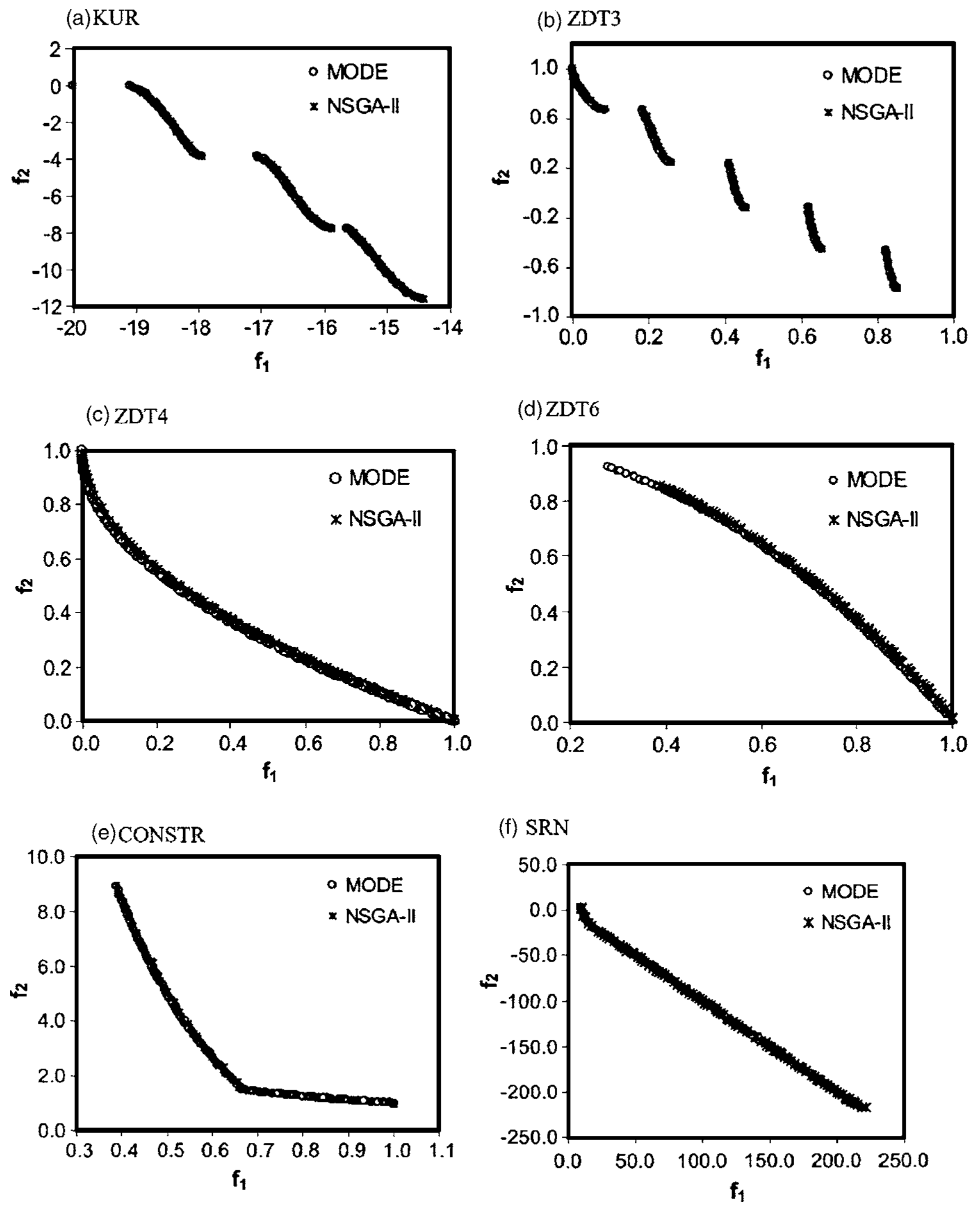

Fig. 2. Nondominated solutions obtained for test problems using MODE and NSGA-II, shows the result of a typical run

$$\text{Maximize } f_1 = \sum_{t=1}^{\text{NT}} (P_{1,t} + P_{2,t}) \tag{12}$$

where

$$P_{i,t} = k_i \text{RP}_{i,t} H_{i,t} \quad \text{for } i = 1,2, \text{and } \forall t \tag{13}$$

$$\text{Minimize } f_2 = \sum_{t=1}^{\text{NT}} [\min(0, \text{ID}_t - \text{IR}_t)]^2 \tag{14}$$

Subject to the following constraints:

$$S_{t+1} = S_t + I_t - \text{RP}_{1,t} - \text{IR}_t - \text{EVP}_t - \text{OVF}_t \quad \forall t \tag{15}$$

$$S_t^{\min} \leqslant S_t \leqslant S_t^{\max} \quad \forall t \tag{16}$$

$$\text{RP}_{i,t}^{\min} \leqslant \text{RP}_{i,t} < TC_i \quad \forall t; \quad i = 1,2 \tag{17}$$

$$\text{IR}_t^{\min} \leqslant \text{IR}_t \leqslant \text{IR}_t^{\max} \quad \forall t \tag{18}$$

where, $P_{i,t}$=hydropower produced ($\times 10^6$ kWh) in the ith power house ($i=1,2$) during period t ($t=1,2,\ldots,36$); NT=total number of time periods; k_i=power coefficient; $\text{RP}_{i,t}$=amount of water released to ith turbine, during period t; $H_{i,t}$=average head available during period t and is expressed as a nonlinear function of the average storage during that period; IR_t=irrigation release in period t; ID_t=maximum irrigation demand in period t; $\text{RP}_{i,t}^{\min}$ =minimum release to be made to meet hydropower requirements; S_t=initial storage volume during time period t; I_t=inflow into the reservoir; EVP_t=evaporation losses (is a nonlinear function of the average storage); OVF_t=overflow from the reservoir; $S_t^{\min}$ and $S_t^{\max}$=minimum and maximum storages allowed in time period t, respectively; $\text{IR}_t^{\min}$ and $\text{IR}_t^{\max}$=minimum and maximum irrigation releases, respectively, in time period t; and TC_i=turbine capacity of power plant i ($i=1,2$).

In addition to the previous constraints, it is to be ensured that end storage of the last period of the year is greater than or equal

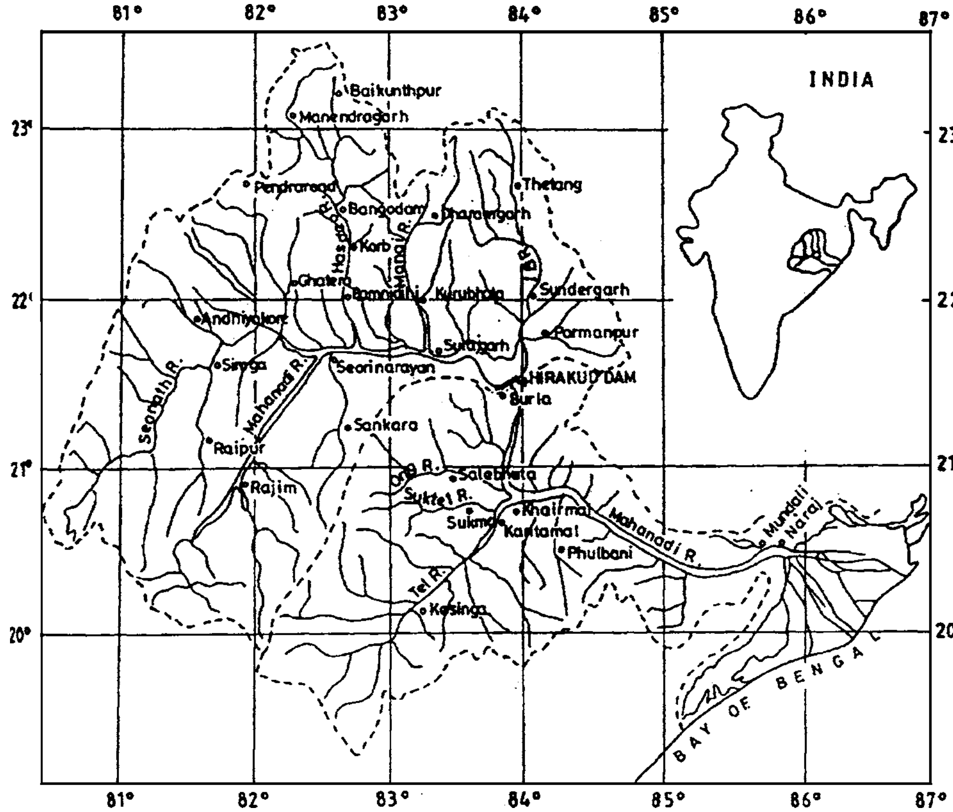

Fig. 3. Location map of the Hirakud Dam in the Mahanadi Basin

to the initial storage of the first period of the next year. This reservoir operation model contains a total of 108 decision variables, i.e., releases to irrigation (IR_t) and releases to hydropower ($RP_{1,t}$ and $RP_{2,t}$) for all $t=1,2,\ldots,36$.

Model Application and Results

To apply the MODE algorithm, the following parameters are used. The initial population was set to 200; CR=0.3; mutation constant $F=0.5$; and maximum number of generations=500. For NSGA-II, the initial population was set to 200, crossover probability=0.9, and mutation probability=$1/n$ (n=number of real variables). The SBX and real parameter mutation were set to 10 and 20, respectively. This algorithm is also run for 500 generations.

The water stored in the reservoir is to be released to meet both the irrigation and hydropower demands. As there is no single optimal solution which can simultaneously satisfy both goals, the developed MODE is intended to find a set of noninferior solutions to function as the decision-making information for decision makers. From test problems, it is found that MODE/rand-to-best/1/bin is performing better than the other variants of MODE. So the reservoir operation model is solved using both MODE/1/rand-to-best/bin (hereafter, for convenience it is referred as MODE) and NSGA-II algorithms. The developed MODE has generated various efficient alternatives (Pareto optimal solutions) in a single run. Table 4 shows the performance of the MODE, as compared with NSGA-II results. It can be observed that the MODE is resulting in higher values of SC metric compared to NSGA-II. This indicates that, both NSGA-II and MODE find a set of Pareto optimal solutions, but MODE has a better coverage of the Pareto front than NSGA-II, hence giving more choices to the decision maker. Fig. 4 gives the results of MODE, shows the improvement attained in noninferior solutions convergence to true Pareto optimal solutions over the generations.

Fig. 5 shows the results of best noninferior solutions obtained in a typical run using MODE and NSGA-II. It can be clearly seen that MODE is achieving better noninferior solutions (Pareto optimal solutions) compared to NSGA-II. Thus for this reservoir operation problem, the complexity involved in its modeling is well captured by the MODE. This can be attributed to the operators used in differential evolution, selection criteria used for new generations and proper matching of all other operators. Also the rotationally invariant MODE is effectively utilizing the interde-

Table 4. Results of MODE and NSGA-II for the Two-Objective Hirakud Reservoir Operation Model, Showing the Best, Worst, Mean, Variance, and Standard Deviation Values for Performance Measures of Set Coverage Metric and Spacing Metric

	Performance metric			
	Set coverage metric		Spacing metric	
Statistic	SC(A,R)	SC(R,A)	MODE	NSGA-II
Best	0.0250	0.0000	185.74	12.22
Worst	1.0000	0.1761	726.66	419.96
Mean	**0.7556**	0.0454	425.15	**124.81**
Variance	0.1253	0.0040	29,428.46	28,336.74
SD	0.3539	0.0632	171.55	168.34

Note: In SC(A,B), A=MODE/rand-to-best/1/bin; and R=NSGA-II. The results are based on ten independent runs for both the algorithms. Bold numbers indicate the best performing algorithm.

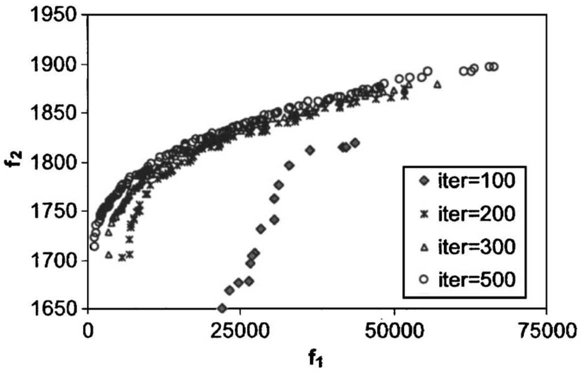

Fig. 4. A typical run of MODE show the improvement in nondominated solutions over the generations for Hirakud reservoir operation problem. [f_1=annual squared deficit for irrigation $(\times 10^6\ \text{m}^3)^2$ and f_2=hydropower production $\times 10^6$ kWh]

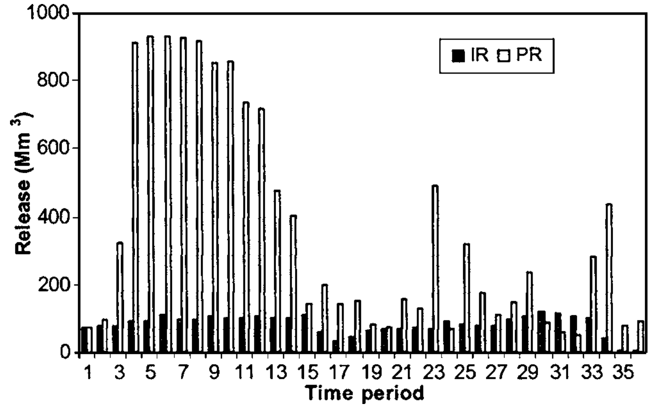

Fig. 6. Ten daily release policy (corresponding to compromised optimal solution) obtained for reservoir operation problem using MODE (IR=release to irrigation and PR=release to power production)

pendence relationships among the variables and exploring the efficient Pareto frontiers in each generation, thus achieving superior performance to that of NSGA-II.

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 decision maker's analysis and interpretation. To demonstrate the final decision making, compromise programming approach (Deb 2001) is adopted in this study. The method of compromise programming picks 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 the Tchebycheff metric is computed by

$$d(f,z) = \max_{m=1}^{M} \frac{[|f_m(x) - z_m|]}{\max\limits_{x\varepsilon S}[f_m(x) - z_m]} \tag{19}$$

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

Using the previous approach, the best compromised solution (BCS) is found and is shown as BCS in Fig. 5. The results of the corresponding alternative solution give the compromised decision for optimal reservoir operation. Fig. 6 shows the corresponding ten daily release policies for irrigation and hydropower over a year and the corresponding storage policy is shown in Fig. 7. It is also noticed that, BCS for NSGA-II generated solutions (Pareto optimal front), produces slightly different optimal decision policy to that of MODE result. However, both are in similar range. These results suggest that MODE is an appropriate and effective optimization tool for identifying reservoir management trade-offs. Further, on the case study on which MODE was tested, it outperformed a commonly used multiobjective genetic algorithm formulation (NSGA-II). So the MODE can be judged as an effective optimization tool for multiobjective reservoir operation decision making.

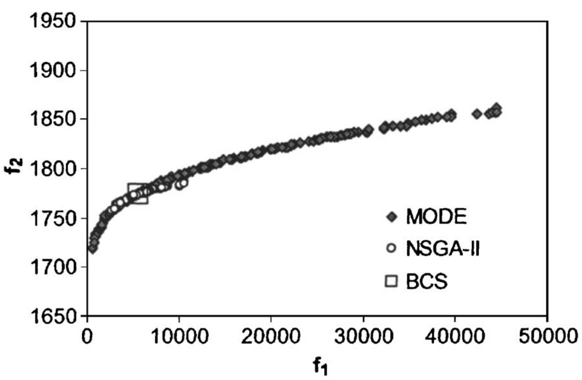

Fig. 5. Nondominated solutions obtained for Hirakud Reservoir operation problem using MODE and NSGA-II, shows the result of a typical run. Here BCS is the best compromised solution obtained for MODE solution set [f_1=annual squared deficit for irrigation $(\times 10^6\ \text{m}^3)^2$ and f_2=hydropower production $\times 10^6$ kWh].

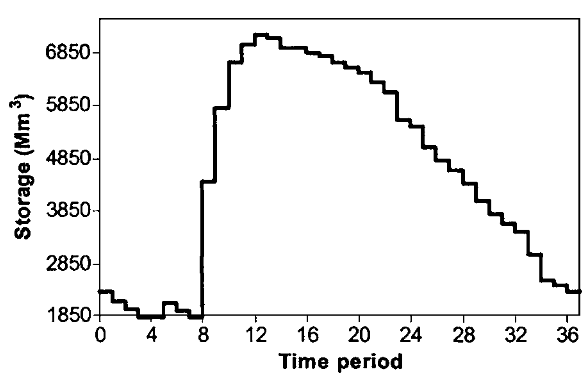

Fig. 7. Ten daily storage operating policy (corresponding to compromised optimal solution) obtained for reservoir operation problem using MODE

Discussion

Measure of MODE Complexity

The proposed MODE approach is simple to implement, yet efficient in yielding true Pareto optimal solutions. The computational complexity of MODE is also reasonable. In this approach, in addition to the objective function computations, the computational complexity of the algorithm is mainly involved in the nondominated comparison of the members in the population, sorting, and crowding distance computation. If there are m objective functions and N number of solutions in the population, then the objective function computation has $O(mN)$ computational complexity. In this study, the computational time required in the reservoir operation simulation model is small; therefore, the computational effort required for sorting and other operations is also important. The costly part of crowding distance computation is sorting the solutions in each objective function. If there are K solutions in the NEA, sorting the solutions in the external repository has $O(mK \log K)$ computational complexity. If the population and the NEA have the same number of solutions, say N, the computational complexity for the nondominated comparison is $O(mN^2)$. Therefore, the overall complexity of the MODE is less than or equal to $O(mN^2)$, which is in well agreement with the latest versions of multiobjective evolutionary algorithms (MOEAs). For example, the overall computational complexity of NSGA-II is $O(mN^2)$ (Deb et al. 2002). Thus, MODE is fast enough to compete with the latest versions of MOEAs and superior to older versions of MOEAs [e.g., NSGA-I has a computational complexity of $O(mN^3)$].

Conclusions

This study presents a novel approach for solving multiobjective reservoir system optimization problems using DE. The proposed methodology for MODE combines Pareto dominance criteria with DE for nondomination selection and crowded distance comparison operator for promoting solution diversity, and incorporates elitism in its evolution to improve the performance of the algorithm. First, different variants of MODE are tested by applying to standard test problems in EMO literature, and their efficiency is evaluated with standard performance measures by comparing with the results of NSGA-II. It is found that the MODE/rand-to-best/1/bin variant is resulting in better performance among all other variants of MODE. To have practical significance, then the developed MODE is applied for generation of optimal trade-offs for a multiple objective reservoir operation problem, namely Hirakud Reservoir project. The optimization involves minimization of flood risk, maximization of hydropower production, and minimization of irrigation deficits while properly evaluating other constraints. The MODE resulted in many Pareto optimal solutions in a single run, by specifying the reservoir releases and storage policy for each solution. The interdependence among the decision variables is better exploited using MODE. It is also found that the performance of MODE is better than NSGA-II for the reservoir system optimization problem. Thus, the obtained results suggest that the MODE approach is robust, and converging to the true Pareto optimal front with a good solution spread and coverage.

The main advantages of MODE algorithm are, the method is relatively simple and easy to implement with few parameters to be fine-tuned and can handle interdependence relationships among the decision variables effectively, thus resulting in fast and efficient Pareto optimal solutions. Finally, it is suggested that the developed MODE algorithm can be used as an efficient alternative technique to solve the multi objective optimization problems.

Notation

The following symbols are used in this paper:

best = the best solution of previous generation;
bin = binomial crossover operator;
CR = crossover constant;
D = dimension of the solution vector;
DE = differential evolution;
DP = dynamic programming;
$\bar{d}$ = mean value of the distance measure;
E = total energy production;
EVP_t = evaporation losses in period t;
exp = exponential crossover operator;
F = mutation constant;
f_m = mth objective function;
$H_{i,t}$ = net head available for ith turbine during time period t;
I_t = inflow to the reservoir during time period t;
ID_t = irrigation demand in time period t;
IR_t = irrigation release in time period t;
$\mathrm{IR}_t^{\min}$, $\mathrm{IR}_t^{\max}$ = minimum and maximum irrigation releases in time period t;
LP = linear programming;
MODE = multiobjective differential evolution;
MOEA = multiobjective evolutionary algorithm;
MOP = multiobjective optimization problem;
NEA = nondominated elite archive;
NLP = nonlinear programming;
NSGA = nondominated sorting genetic algorithm;
NT = total number of time periods;
NP = population size;
n = dimension of decision vector;
OVF_t = overflow in period t;
$P_{i,t}$ = hydropower produced from ith turbine in time period t;
PF^* = Pareto optimal front;
Q^* = Pareto optimal set;
$\mathrm{RP}_{i,t}$ = release made to ith turbine in period t;
r_1, r_2, r_3, r_4, r_5 = mutually different random integers;
rand = randomly chosen solution from previous generation;
rnbr = random integer;
S_t = initial storage of reservoir in time period t;
$S_t^{\min}$, $S_t^{\max}$ = minimum and maximum storage limits of the reservoir;
SC = set coverage metric;
SD = standard deviation;
SP = spacing metric;
tempPop = temporary population;
TC_i = ith turbine capacity;
U_i^G, V_i^G, X_i^G = solution vector of the ith individual in generation G;
u_{id}, v_{id}, x_{id} = decision value of the ith solution dth variable;
x, y = vector of decision variables;
z = reference solution;
α = string denoting the vector to be perturbed;

β = number of difference vectors considered for perturbation of α; and

γ = crossover type;

References

Abbass, H. A., and Sarker, R. (2002). "The Pareto differential evolution algorithm." *Int. J. Artificial Intelligence Tools*, 11(4), 531–552.

Cohon, J. L. (1978). *Multiobjective programming and planning*, Academic, New York.

Croley, T. E., II, and Rao, K. N. R. (1979). "Multi-objective risks in reservoir operation." *Water Resour. Res.*, 15(4), 1807–1814.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, Wiley, Chichester, U.K.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). "A fast and elitist multi-objective genetic algorithm: NSGA-II." *IEEE Trans. Evol. Comput.*, 6(2), 182–197.

Devi, Prasad T., and Park, N. S. (2004) "Multiobjective genetic algorithms for design of water distribution networks." *J. Water Resour. Plann. Manage.*, 130(1), 73–82.

Goicoechea, A., Hansen, D. R., and Duckstein, L. (1982). *Multi-objective decision analysis with engineering and business applications*, Wiley, New York.

Haimes, Y. Y., and Hall, W. A. (1974). "Multiobjectives in water resources systems: The surrogate worth trade-off method." *Water Resour. Res.*, 10(4), 615–624.

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

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

Liang, Q., Johnson, L. E., and Yu, Y. S. (1996). "A comparison of two methods for multiobjective optimization for reservoir operation." *Water Resour. Bull.*, 32(2), 333–340.

Madavan, N. K. (2002). "Multiobjective optimization using a Pareto differential evolution approach." *Congress on Evolutionary Computation (CEC'2002)*, Vol. 2, IEEE Service Center, Piscataway, N.J., 1145–1150.

Onwubolu, G., and Davendra, D. (2006). "Scheduling flow shops using differential evolution algorithm." *Eur. J. Oper. Res.*, 171(2), 674–692.

Parsopoulos, K. E., Tasoulis, D. K., Pavlidis, N. G., Plagianakos, V. P., and Vrahatis, M. N. (2004). "Vector evaluated differential evolution for multiobjective optimization." *Proc., IEEE 2004 Congress on Evolutionary Computation*, Portland, Ore., 204–211.

Price, K. V. (1999). "An introduction to differential evolution." *New ideas in optimization*, D. Corne, M. Dorigo, and F. Glover, eds., McGraw-Hill, London, 79–108.

Reed, P., Minsker, B. S., and 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. Hydroinformatics*, 3(2), 71–89.

Robič, T., and Filipič, B. (2005). "DEMO: Differential evolution for multiobjective optimization." *Proc., 3rd Int. Conf. on Evolutionary Multi-Criterion Optimization, EMO 2005*, 520–533.

Salomon, R. (1996). "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: A survey of some theoretical and practical aspects of genetic algorithms." *BioSystems*, 39(3), 263–278.

Storn, R., and Price, K. (1995). "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces." *Technical Rep. No. TR-95–012*, International Computer Science Institute, Berkley, Calif.

Storn, R., and Price, K. (1997). "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces." *J. Global Optim.*, 11, 341–359.

Tauxe, G. H., Inman, R. R., and Mades, D. M. (1979). "Multi-objective dynamic programming with application to reservoir." *Water Resour. Res.*, 15(6), 1403–1408.

Thampapillai, D. J., and Sinden, J. A. (1979). "Tradeoffs for multiple objectives planning through linear programming." *Water Resour. Res.*, 15(5), 1028–1034.

Xue, F., Sanderson, A. C., and Graves, R. J. (2003). "Pareto-based multiobjective differential evolution." *Proc., 2003 Congress on Evolutionary Computation (CEC'2003)*, Vol. 2, IEEE, New York, 862–869.