

**Provided for non-commercial research and educational use only.
Not for reproduction or distribution or commercial use.**



This article was originally published by IWA Publishing. IWA Publishing recognizes the retention of the right by the author(s) to photocopy or make single electronic copies of the paper for their own personal use, including for their own classroom use, or the personal use of colleagues, provided the copies are not offered for sale and are not distributed in a systematic way outside of their employing institution.

Please note that you are not permitted to post the IWA Publishing PDF version of your paper on your own website or your institution's website or repository.

Please direct any queries regarding use or permissions to hydro@iwap.co.uk

MOPSO algorithm and its application in multipurpose multireservoir operations

E. Fallah-Mehdipour, O. Bozorg Haddad and M. A. Mariño

ABSTRACT

The main reason for applying evolutionary algorithms in multi-objective optimization problems is to obtain near-optimal nondominated solutions/Pareto fronts, from which decision-makers can choose a suitable solution. The efficiency of multi-objective optimization algorithms depends on the quality and quantity of Pareto fronts produced by them. To compare different Pareto fronts resulting from different algorithms, criteria are considered and applied in multi-objective problems. Each criterion denotes a characteristic of the Pareto front. Thus, ranking approaches are commonly used to evaluate different algorithms based on different criteria. This paper presents three multi-objective optimization methods based on the multi-objective particle swarm optimization (MOPSO) algorithm. To evaluate these methods, bi-objective mathematical benchmark problems are considered. Results show that all proposed methods are successful in finding near-optimal Pareto fronts. A ranking method is used to compare the capability of the proposed methods and the best method for further study is suggested. Moreover, the nominated method is applied as an optimization tool in real multi-objective optimization problems in multireservoir system operations. A new technique in multi-objective optimization, called warm-up, based on the PSO algorithm is then applied to improve the quality of the Pareto front by single-objective search. Results show that the proposed technique is successful in finding an optimal Pareto front.

Key words | multi-objective particle swarm optimization (MOPSO), multipurpose, multireservoir systems, optimization problems

E. Fallah-Mehdipour (corresponding author)

O. Bozorg Haddad

Department of Irrigation and Reclamation Engineering, Faculty of Agriculture Engineering and Technology, College of Agriculture and Natural Resources, University of Tehran, Karaj, Tehran, Iran
E-mail: Falah@ut.ac.ir

M. A. Mariño

Department of Land, Air and Water Resources, Department of Civil and Environmental Engineering, and Department of Biological and Agricultural Engineering, University of California, 139 Veihmeyer Hall, University of California, Davis, CA 95616-8628, USA

INTRODUCTION

Application of a suitable multi-objective optimization technique is important in determining a compromise between different objectives. Traditional multi-objective methods attempt to find a set of nondominated solutions using mathematical programming (Balcer & Fontane 2006). Both the weighting and ϵ -constraint methods are commonly used as traditional techniques that can produce nondominated solutions without any information from a decision-maker. In these methods, multi-objective problems are transformed to single-objective ones. Thus, solutions directly depend

on weights in the weighting method and constraints in the ϵ -constraint method. In contrast, only one solution can be detected per optimization effort (Parsopoulos *et al.* 2004).

The increased complexity of engineering problems and especially in the field of water resources management has led to more applications of evolutionary algorithms. The latter are stochastic search methods that simulate natural biological evolution and/or the social behavior of species. Evolutionary algorithms present a set of nondominated solutions/Pareto fronts for multi-objective problems. Kennedy & Eberhart (1995)

developed particle swarm optimization (PSO), an evolutionary algorithm based on the social behavior of birds to reach a destination. In the water resources area, Chuanwen & Bompard (2005) applied a self-adaptive chaotic PSO algorithm for short-term, hydroelectric-system scheduling in a deregulated environment. Suribabu & Neelakantan (2006) used the PSO algorithm to design a water distribution pipeline network in which EPANET was applied as a simulation model. Matott *et al.* (2006) identified the PSO algorithm as an effective algorithm for solving pump-and-treat optimization problems. Izquierdo *et al.* (2008) applied the PSO algorithm to the optimization of a wastewater collection network while Montalvo *et al.* (2008) applied the PSO algorithm to design a water supply system.

Development of the PSO algorithm in multi-objective optimization is a contemporary development in single-objective problems. Hu & Eberhart (2002) presented multi-objective optimization based on the PSO algorithm using a dynamic neighborhood to find local optima (*Lbest*) for each particle in bi-objective optimization problems. The algorithm searches for *Lbest* which is used as the global best (*Gbest*) in the PSO algorithm by finding the nearest particles as the neighbors of the current particle. The particle best (*Pbest*) is the best position in a particle's history when a new position dominates the current position. Parsopoulos & Vrahatis (2002) proposed a vector-evaluated PSO (VEPSO), based on the concept of vector-evaluated GA (VEGA) to solve bi-objective problems. VEPSO uses one swarm for each objective and the best particle of each swarm is used as the *Gbest* to determine particle velocities. Parsopoulos & Vrahatis (2002) developed a method in which the optimal Pareto front was extracted at the end of the search process. Thus, the number of Pareto-front members depends on the swarm size. In contrast, Coello & Lechuga (2002) presented a method in which an external repository is maintained as the nondominated solutions. To find *Gbest* in the PSO algorithm, the best position in a repository which has the smaller value of density is selected by performing a procedure similar to roulette-wheel selection. On the other hand, to find the *Pbest* in the PSO, if the new position dominates the other, a new position is selected as the *Pbest*; if no one dominates the others, one of them is randomly selected as the *Pbest*. Fieldsend & Singh (2002) presented a multi-objective PSO (MOPSO) which uses a nondominated tree for selecting *Lbest* for each particle. The

proposed method was compared with a Pareto archive evolutionary strategy (PAES) on bi-objective problems. Mostaghim & Teich (2003) introduced the sigma method, in which the sigma vector for each particle is the gradient of a line connecting that particle with the origin. To solve bi-objective problems, Mostaghim & Teich (2004) proposed a method to cover gaps between nondominated solutions in the Pareto front by dividing a swarm into sub-swarms. Abido (2007) presented a two-level of nondominated solutions approach to detect the optimal Pareto front. Yang *et al.* (2009) proposed an algorithm for choosing *Gbest* in which the density value of the particles in a nondominated solution set is estimated and the algorithm generates grids to divide the search objective space for finding the *Gbest*.

To test and show the MOPSO algorithms in reservoir operation problems, Baltar & Fontane (2006) applied the method of Coello & Lechuga (2002) in single-reservoir operation to find nondominated solutions with four objectives: (1) maximization of the annual firm-water supply; (2) maximization of the annual firm-energy production; (3) minimization of flood risk; and (4) maximization of the overall reliability of the reservoir. Results showed that the found Pareto using the MOPSO algorithm successfully reaches to the Pareto front using the ϵ -constraint method as the optimal Pareto front in two-dimensional problems. Kumar & Reddy (2007) used the PSO algorithm to solve multipurpose, multireservoir operation problems. To handle multiple objectives of the problem, a weighted approach was adopted. The objectives were minimization of the sum of squared deficits for annual irrigation demand and maximization of annual energy production. In this algorithm, the elitist process replaces the worst particle solution by the best solution of the swarm, after performing a mutation mechanism on the best particle (Kumar & Reddy 2007). Thus, the optimal solution is found by a single-objective search process. Reddy & Kumar (2007a) then proposed a multi-objective optimization algorithm based on swarm intelligence using a mutation strategy called elitist mutation (EM). This strategy is then incorporated in the optimization algorithm. To test the proposed combined method, some mathematical benchmark problems were solved and the multi-objective algorithm was then applied to three engineering design problems. Reddy & Kumar (2007b) proposed the MOPSO approach, which is a combination of the EM strategy with

the PSO algorithm and is called EM-MOPSO. Results of the proposed algorithm were compared with those of the nondominated sorting genetic algorithm (NSGA-II) in some test problems. After achieving satisfactory performance in the test problems, EM-MOPSO was then applied to a single-reservoir operation considering three objectives: (1) minimization of irrigation deficit; (2) maximization of hydropower and (3) maximization of the satisfaction level of downstream water quality requirements. Reddy & Kumar (2009) proposed the EM-MOPSO algorithm for water management and compared the results of the proposed method with those of NSGA-II, showing the EM-MOPSO approach as having more capability to find the optimal Pareto front. The EM-MOPSO method was then used in integrated water resource management. There are several other recent studies in the multi-objective reservoir operation which are based on other evolutionary algorithms. Kim et al. (2008) presented a single-reservoir operating rule for a year using a multi-objective genetic algorithm. They generated synthetic inflow data over 100 years using a time series model while linear operational rules were found as the relationships between the inflow, water demand and terminal storage. Chang & Chang (2009) applied NSGA-II to examine the operation of a multi-reservoir system in northern Taiwan. They developed a daily operational simulation model to guide the releases of the reservoir system in shortage events considering two objective functions. Afshar et al. (2009) proposed the nondominated archiving multi-colony ant algorithm for multi-objective optimization. Performance of the proposed algorithm was tested on two well-known mathematical multi-objective benchmark problems and it was then applied for optimization of multiple-objective reservoir operation policies.

This paper is divided into two parts. The first part considers three different methods based on the PSO algorithm, using dominated and nondominated solutions in the search process. To test the applicability of the methods, bi-objective mathematical problems are solved and results are compared using different criteria. A ranking method, which takes into account all criteria, is used to select the best method. In the second part, the nominated method is tested in single and multireservoir operation with different combinations of objectives. To improve the quality and quantity of nondominated solutions, a new technique called warm-up is used. This

technique uses the single-objective PSO (SOPSO) algorithm to find a uniform Pareto front in space.

MULTI-OBJECTIVE PSO (MOPSO) ALGORITHM

Following an overview of the SOPSO method, three multi-objective PSO techniques are discussed.

The PSO algorithm is a swarm intelligence algorithm based on the social behavior of birds (Kennedy & Eberhart 1995). Each bird is represented by a particle and a collection of particles is identified as a swarm. Each particle moves in decision space with two kinds of velocity (*Pbest* and *Gbest*). *Pbest* is the best position in the particle's history and the best previous evaluated position of the swarm is represented as *Gbest*. The PSO algorithm starts by searching in random solutions with a uniform distribution. If the number of decision variables is D , each particle can be represented by a D -dimensional vector. Thus, the current position (X_i) and velocity (V_i) of the i th particle are represented as

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad (1)$$

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iD}). \quad (2)$$

The velocity of the i th particle is calculated by

$$\left[\begin{aligned} v_{id}^{n+1} &= \alpha(w \cdot v_{id}^n + c_1 \cdot r_1^n (Pbest_{id}^n - x_{id}^n) + c_2 \cdot r_2^n (Gbest_{id}^n - x_{id}^n)) \\ \text{for } i &= 1, 2, \dots, N \text{ and } d = 1, 2, \dots, D \end{aligned} \right] \quad (3)$$

in which v_{id}^{n+1} = velocity of the d th dimension of the i th particle in the $(n+1)$ th iteration; α = constriction factor that controls the velocity in each iteration. (The upper values of α enlarge the searching space while lower values reduce the searching space); w = inertia weight parameter which is an important index to converge the swarm and controls the effects of the current velocity (thus, the larger value causes searching in a wide space while the smaller value leads the swarm to local optimal points); v_{id}^n = velocity of the d th dimension of the i th particle in the n th iteration; c_1 = cognitive parameter; c_2 = social parameter (c_1 and c_2 control movement toward *Pbest* and *Gbest*); r_1^n and r_2^n = random numbers uniformly distributed within $[0,1]$;

$Pbest_{id}^n$ = the best position of the d th dimension for the i th particle up to the n th iteration; $Gbest_d^n$ = d th dimension of the best position of the swarm up to the n th iteration.

To control the variation of a particle's velocity, lower and upper bounds for velocity are limited to user-specified values of v_{\min} and v_{\max} :

$$v_{\min} \leq v_{id}^{n+1} \leq v_{\max}. \quad (4)$$

A particle's position is updated using Equation (5):

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1}. \quad (5)$$

This paper presents three different methods to illustrate the MOPSO algorithm.

First method. The structure of this method is similar to that of the VEPSO algorithm of Parsopoulos & Vrahatis (2002). The latter compared results of the VEPSO algorithm with those of VEGA in mathematical test problems and showed that the VEPSO algorithm yields optimal Pareto fronts. In the proposed first method, however, different swarms are employed to find an optimal Pareto front. The number of swarms is the same as the number of objectives. The velocity of each particle is evaluated by the best position of the previous swarm and its best position so far. Thus, exchanging information among swarms in the most recent iteration can lead to Pareto-front optimal points.

A particle's velocity when the number of particles is N is calculated by

$$\left[\begin{array}{l} v_{id}^{n+1} = \alpha(w \cdot v_{id}^n + c_1 \cdot r_1^n (Pbest_{id}^n - x_{id}^n)) + c_2 \cdot r_2^n (Gbest_{j,d}^n - x_{id}^n) \\ \text{for } i = 1, 2, \dots, N \text{ and } d = 1, 2, \dots, D \end{array} \right] \quad (6)$$

in which j is calculated as follows:

$$j = \begin{cases} m & \text{for } sn = 1 \\ sn - 1 & \text{for } sn = 2, \dots, m \end{cases} \quad (7)$$

where sn = swarm number and m = maximum number of swarms.

Figure 1 shows a four-objective problem with four swarms in three sequential iterations. According to Figure 1, the best position of each swarm ($Gbest$) in the formulation is identified as the best position of the previous swarm and the $Pbest$ of each particle is selected from its swarm.

In this method, there is an external archive that stores nondominated solutions. All particles move to the external archive at the end of each iteration. Those particles are then compared to each other and nondominated particles are stored while the others are removed. Thus, the archive size is dynamic and nondominated solutions can be reported in each iteration.

Second method. The structure of this method is similar to the method proposed by Coello & Lechuga (2002). The latter compared results of their method with those of PAES and NSGA-II, which had been previously extracted by Knowles & Corne (2000) and Deb et al. (2000), respectively. The performance of different particles is always compared in terms of their dominance relations. The algorithm starts with random solutions as an initial swarm. The position and velocity equations of this method are similar to the SOPSO algorithm. The main difference between the SOPSO algorithm and the second method is in the selection of $Pbest$ and $Gbest$. In the second method, the best vector of each particle is $Pbest$. Thus, the initial position of particles is identified as $Pbest$ in the initial iteration. In the subsequent iteration, the best position vector is updated by a dominance relation. In each iteration, if the current position dominates the new position, then the current position is selected as $Pbest$. If the new position dominates the current one, then the new position is selected as $Pbest$. If both of them are in the nondominated position, the current position is left as $Pbest$. To determine $Gbest$, nondominated solutions are stored in an external archive. In the roulette-wheel selection method, the probability of choosing one vector/particle is inversely related to its density. Thus, the particle that is located in the region with more densely nondominated points has a lower probability of being selected. Figure 2 shows the transformation of a particle with lower density in a two-dimensional space. This process leads particles to a better distribution of nondominated solutions/Pareto fronts. To gather nondominated solutions, an external archive is used. The mechanism used to store nondominated solutions in the external archive is as in the first method.

Third method. In this method, the process of searching starts with random solutions that have a uniform distribution. All objectives are used to find $Pbest$ and $Gbest$. Cognitive and social parameters (c_1 and c_2) are important in this method

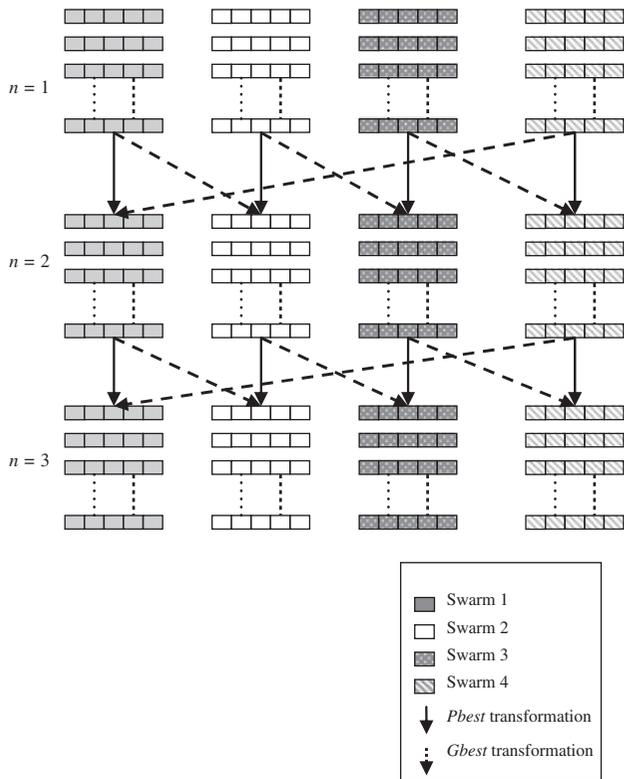


Figure 1 | Transformation of *Pbest* and *Gbest* between different swarms and iterations in the first method.

and the velocity vector is distributed to all objectives. Thus, Equation (3) changes to Equation (8):

$$v_{id}^{n+1} = \alpha \left(w \cdot v_{id}^n + \frac{c_1}{m} \cdot r_1^n \sum_{j=1}^m (Pbest_{j,id}^n - x_{id}^n) + \frac{c_2}{m} \cdot r_2^n \sum_{j=1}^m (Gbest_{j,d}^n - x_{id}^n) \right) \quad (8)$$

in which m = number of objectives; $Pbest_{j,id}$ = the best position of the i th particle in the j th objective; and $Gbest_{j,d}$ = best position of the swarm in the j th objective. Other parameters and relations such as velocities, boundaries and updating of a particle's position are similar to the SOPSO. **Figure 3** shows two kinds of vectors as *Pbest* and *Gbest*, resulting from all *Pbest* and *Gbest* vectors from all of the objectives while the final velocity vector is the summation of these two vectors. Thus, a particle moves from the n th position to the $(n + 1)$ th position by a final vector. In each

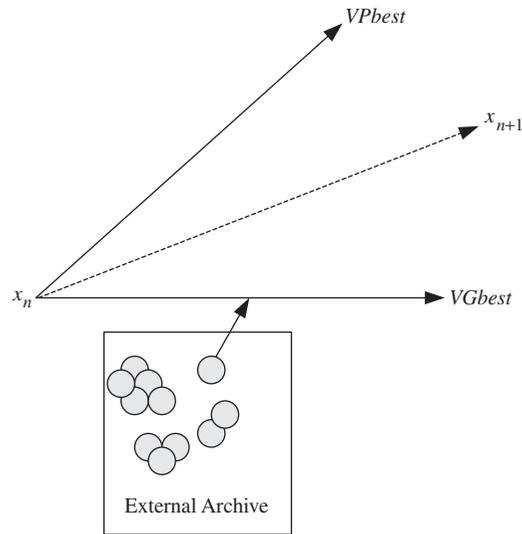


Figure 2 | Transformation of a particle from an external archive as *Gbest* in the second method.

iteration, all particles move to an external archive, employing the same mechanism as in the first and second methods.

RESULTING CHARACTERISTICS

Evolutionary algorithms have a stochastic base in which the searching starts with a random set of solutions. Thus, results

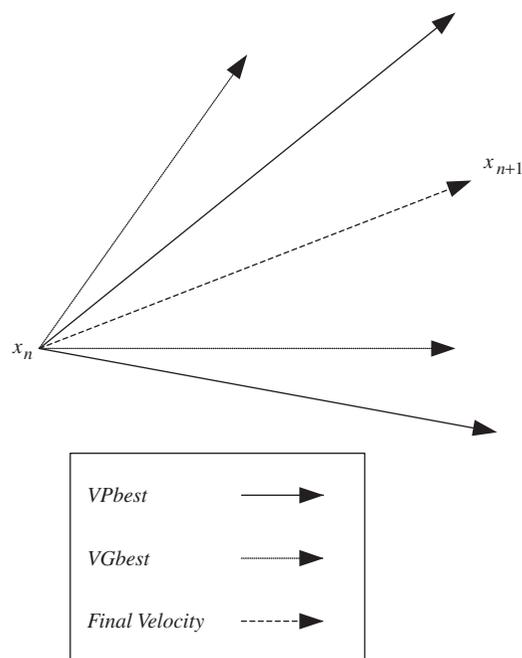


Figure 3 | Effects of two kinds of velocity (*Pbest* and *Gbest*) in the third method.

may be different in several runs. Different results may also occur when different methods or algorithms are used. To compare results of different runs in multi-objective problems, some criteria need to be identified. Those criteria can represent quality and quantity of Pareto-front points. To determine the best multi-objective method, comparison criteria are calculated and a ranking approach is used to select the best criterion.

Generational distance (GD)

This metric was proposed by Veldhuizen (1999) to find the average distance between the resulted nondominated solutions and the optimal Pareto front:

$$GD = \frac{1}{NS} \sqrt{\sum_{i=1}^{NS} d_i^2} \quad (9)$$

where GD = generational distance; NS = number of solutions found; and d_i = Euclidean distance (in the objective space) between each nondominated solution and the nearest member of the optimal Pareto front.

Spacing (S)

This is a metric proposed by Schott (1995) that gives the distribution of the found solutions:

$$S = \sqrt{\frac{1}{NS-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (10)$$

where S = spacing criteria; NS = number of solutions found; \bar{d} = mean of all d_i ; and d_i is calculated by Equation (11):

$$d_i = \text{Min}_j \left(\sum_{k=1}^m |f_k^i(x) - f_k^j(x)| \right) \quad (11)$$

for $i=1,2,\dots, NS$ and $j = 1,2,\dots, NS - 1$

where m = number of objectives and $f_k^j(x)$ = k th objective for the j th solution. Thus, the best value of the spacing is zero, meaning that found solutions are equidistantly spaced.

Number of solutions (NS)

This is a value which presents the number of solutions found. There is no upper bound for NS . That means that the greater the number of solutions, the better the set of nondominated solutions.

EXPERIMENTS

The following experiments are performed to compare the three methods proposed in this paper. All of the methods are stochastic-based and their results are different in each run. Thus, for making a generalized decision on choosing the preferable method, five runs are performed and merged for each method in each problem. In this way, the judgment is more reliable, due to a simultaneous look at the statistical measures of several runs. The resulting characteristics of each Pareto front are calculated and statistical measures are performed for the proposed method in different problems. The parameters of the PSO algorithm in all problems are the same with $c_1 = c_2 = 2$, $\alpha = 1$ and w has a dynamic value which is calculated by Equation (12). This equation was first proposed by Shi & Eberhart (1998a, b) to increase the PSO algorithm capability in searching for the optimal solution:

$$w = w_{\max} - \frac{(w_{\max} - w_{\min}) \times n}{iter_{\max}} \quad (12)$$

in which $w_{\max} = 0.9$; $w_{\min} = 0.4$; n = number of iterations; and $iter_{\max}$ = maximum number of iterations.

Test problem (Deb 1999)

The bi-objective problem proposed by Deb (1999) is a constrained problem with two decision variables:

$$\text{Min. } f(x) = (f_1(x), f_2(x)) \quad (13)$$

$$f_1(x) = x_1 \quad (14)$$

$$f_2(x) = \frac{g(x)}{x_1} \quad (15)$$

$$g(x) = 2 - e^{-\left(\frac{x_2 - 0.2}{0.004}\right)^2} - 0.8e^{-\left(\frac{x_2 - 0.6}{0.4}\right)^2} \quad (16)$$

$$0.1 \leq x_i \leq 1 \text{ and } i = 1, 2. \quad (17)$$

Test problem (Kita et al. 1996)

The constrained problem proposed by Kita et al. (1996) can be defined as

$$\text{Max. } f(x) = (f_1(x), f_2(x)) \quad (18)$$

$$f_1(x) = -x_1^2 + x_2 \quad (19)$$

$$f_2(x) = \frac{1}{2}x_1 + x_2 + 1 \quad (20)$$

$$\frac{1}{6}x_1 + x_2 - \frac{13}{2} \leq 0 \quad (21)$$

$$\frac{1}{2}x_1 + x_2 - \frac{15}{2} \leq 0 \quad (22)$$

$$\frac{5}{x_1} + x_2 - 30 \leq 0 \quad (23)$$

$$0 \leq x_i \leq 7 \text{ and } i = 1, 2. \quad (24)$$

Test problem DTLZ2

Deb et al. (2001) proposed a problem with an unlimited number of objectives and decision variables, defined as

$$\text{Min. } f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad (25)$$

$$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right) \cdots \cdots \cos\left(x_{m-2} \frac{\pi}{2}\right) \cos\left(x_{m-1} \frac{\pi}{2}\right) \quad (26)$$

$$f_2(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right) \cdots \cdots \cos\left(x_{m-2} \frac{\pi}{2}\right) \sin\left(x_{m-1} \frac{\pi}{2}\right) \quad (27)$$

$$f_3(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right) \cdots \cdots \sin\left(x_{m-1} \frac{\pi}{2}\right) \quad (28)$$

⋮

$$f_m(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right) \quad (29)$$

$$g(x) = \sum_{x_i \rightarrow x_m} (x_i - 0.5)^2 \quad 0 \leq x_i \leq 1. \quad (30)$$

In this paper, a bi-objective problem with 12 decision variables is considered.

Test problem (Kursawe 1991)

The bi-objective problem proposed by Kursawe (1991) is a constrained problem which can be defined as

$$\text{Min. } f(x) = (f_1(x), f_2(x)) \quad (31)$$

$$f_1(x) = \sum_{i=1}^{n-1} \left(-10 \exp\left(-0.2 \sqrt{x_i^2 + x_{i+1}^2}\right) \right) \quad (32)$$

$$f_2(x) = \sum_{i=1}^n \left(|x_i|^{0.8} + 5 \sin(x_i)^3 \right) \quad (33)$$

$$-5 \leq x_i \leq 5 \text{ and } i = 1, 2, 3. \quad (34)$$

Constraint handling

There are several approaches proposed in evolutionary algorithms to handle constrained optimization problems (Bozorg Haddad & Mariño 2007). These approaches can be grouped into four major categories (Michalewicz & Schouenauer 1996): (1) methods based on the penalty function; (2) methods based on a search of feasible solutions; (3) methods based on preserving the feasibility of solutions; and (4) hybrid methods.

In this paper, the hybrid method is used to handle constraints. In the existing model, decision variables are found during the searching process of feasible solutions and their values are fixed between upper and lower allowable boundaries. Moreover, a dynamic penalty function, which is a linear relation, is used for other constraints. The penalty function is added to or subtracted from the minimization and maximization objective functions, respectively.

Results of mathematical problems

Results of the three aforementioned methods for five runs with the same function evaluation have been merged and presented in Figure 4. To evaluate and compare all of the methods in the same conditions, the same value for the number of function evaluations (number of particles \times maximum number of iterations) has been considered for all methods in a problem. According to the structure of the three methods, two swarms in the first method and one

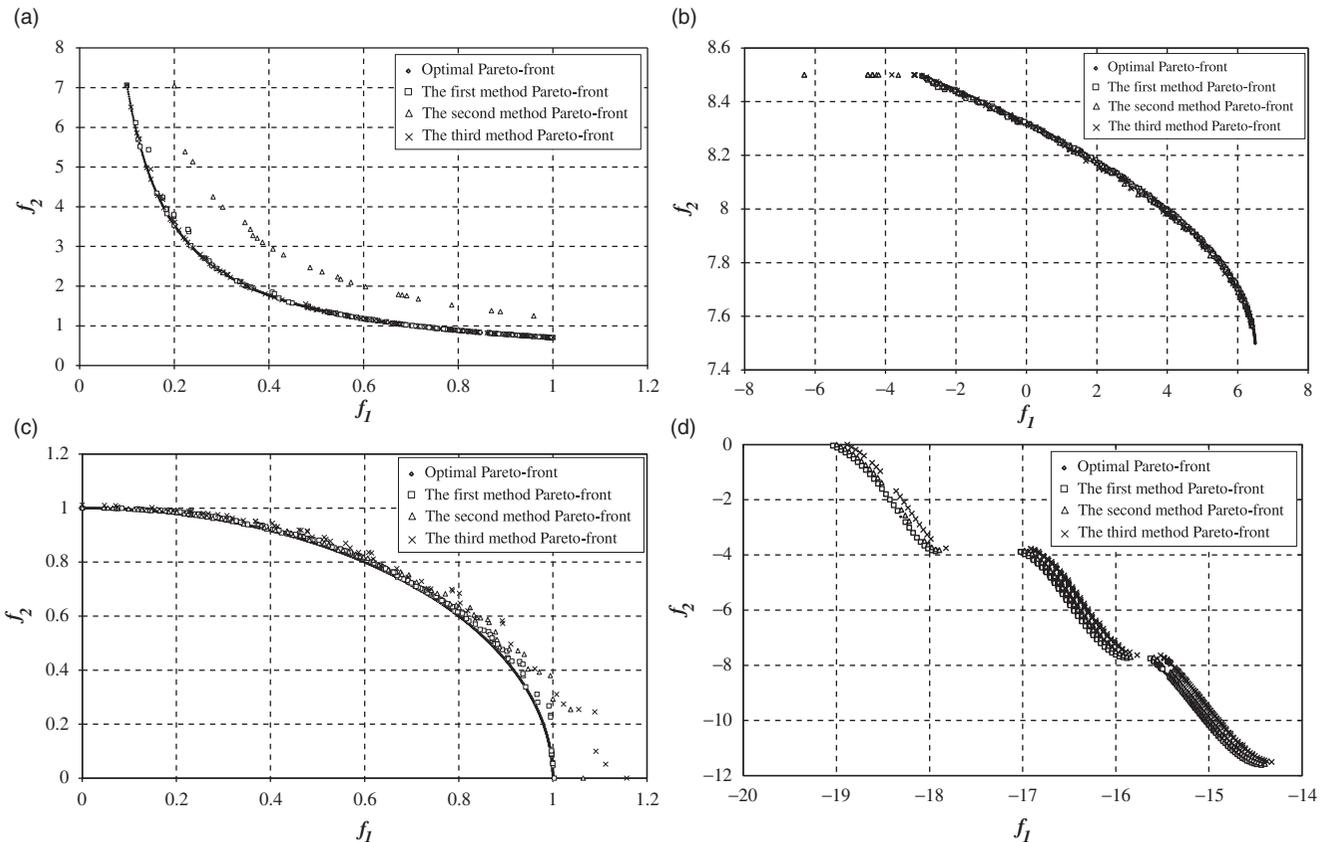


Figure 4 | Resulting Pareto fronts for the: (a) Deb; (b) Kita; (c) DTLZ2; and (d) Kursawe problems.

swarm in the second and third methods are used in the bi-objective problems. Thus, the maximum iteration of the first method should be considered half of the second and third methods in the same function evaluation. The maximum iteration for the second and third methods based on the MOPSO algorithm is 5000, 500, 20 000, and 20 000 for the Deb, Kita, DTLZ2, and Kursawe problems, respectively. The number of maximum iterations for the first method has been considered half of the aforementioned numbers using two swarms for the calculating process.

Three different characteristics of found Pareto fronts (GD , S , and NS) for all the benchmark problems, Deb, Kita, DTLZ2, and Kursawe, have been calculated and presented in Table 1.

The distance of the found Pareto front from the optimal Pareto front for the Deb problem in the second method has been shown in Figure 4. In contrast, all of the characteristics (GD , S and NS) of this Pareto front are in the worst state. Thus, the second method has been eliminated in the following

comparison in the Deb problem. The value of GD in the first method is more than that in the third method. However, the S value is lower than that obtained in the third method and the number of solutions for the first method is more than in the third method.

To select an appropriate method, a multi-criteria decision-making (MCDM) method is needed to rank different methods considering all criteria. The technique for order preference by similarity to ideal solution (TOPSIS) which is capable of ranking different normalized alternatives even with different units has been selected as the MCDM method in this paper. TOPSIS, first proposed by Yoon & Hwang (1995), is based on the idea that the chosen alternative should have the shortest distance from the ideal solution and the furthest distance from the anti-ideal solution. Thus, TOPSIS leads to a function that allows the ranking of alternatives from best to worst. In this paper, multi-objective optimization methods are identified as the decision alternatives and generational distance (GD), spacing (S) and number of solutions (NS) as the

Table 1 | Characteristics of resulting Pareto front for the Deb, Kita, DTLZ2, and Kursawe problems

Problem	Method number	GD	S	NS
Deb	1	4.28E-05	0.1258	134
	2	4.76E-02	0.1158	27
	3	2.16E-05	0.1213	125
Kita	1	7.63E-05	0.0732	179
	2	2.93E-01	0.1055	141
	3	8.20E-03	0.1197	105
DTLZ2	1	7.69E-05	0.0368	200
	2	4.32E-04	0.0658	90
	3	2.00E-03	0.0748	58
Kursawe	1	1.94E-04	0.1119	225
	2	3.96E-03	0.1060	209
	3	1.15E-01	0.0215	220

criteria. In the TOPSIS method, decision and weighting matrices are calculated. The decision matrix is then normalized and multiplied by the weighting matrix in which the sum of each row is equal to 1. In the next step, the ideal and anti-ideal solutions are determined. The Euclidean distances from the ideal and anti-ideal solutions are then calculated for each decision option. The relative closeness of each solution, rating the goodness of each solution, is calculated for all solutions, as follows:

$$D_i^+ = \frac{d_i^-}{d_i^+ + d_i^-} \quad 0 \leq D_i^+ \leq 1 \text{ and } i = 1, 2, \dots, p \quad (35)$$

in which D_i^+ = relative closeness; d_i^- = Euclidean distance from the anti-ideal solution; d_i^+ = Euclidean distance from the ideal solution; and p = number of solutions. Finally, preference orders are ranked. Thus, a solution (option) with the maximum value of D_i^+ has the first rank and other options are sorted in decreasing order.

Results of using TOPSIS are shown in Table 2. The weight of all characteristics was considered the same (1/3) so as to allocate the same degree of importance to all criteria. Results showed that the first method for both Deb and Kursawe problems achieves the maximum value of D_i^+ which presents the maximum distance from the anti-ideal point. Thus, the first method is selected as the nominated method. There is a slight difference in the D_i^+ value of the first and third methods in the Deb problem, which is due to the considerable

difference between the criteria of these methods with those of the second method. Since the criteria of the three methods should be normalized and then be used in TOPSIS, the normalized value of the first and third methods will be located on the top with a negligible difference in D_i^+ and the lowest value of this parameter will be assigned to the second method.

Although the difference between the best and worst values for the characteristics is not considerable, all of the characteristics are in the best value using the first method, even in the Kita and DTLZ2 problems. Thus, the first method is selected and recommended as the optimization tool.

MOPSO methods were coded in the software package Matlab7.0 and run on a PC/Windows XP/256 MB RAM/ 2 GHz computer and the execution time of each problem is less than one minute.

RESERVOIR SYSTEM SIMULATION

Application of the nominated method in the MOPSO algorithm has been considered in a real multi-objective operation problem. Three different objectives with combinations of two and three have been considered. Equations (36)–(38) illustrate the minimization of the total squared deviation of: (1) release from demand; (2) storage from target storage; and (3) generated power from installed capacity as

$$\text{Min. } Z_1 = \sum_{i=1}^N \sum_{t=1}^T \left(\frac{(R_{it}^{Power} + R_{it}^{De} - D_{it})}{D_i^{Max}} \right)^2 \quad (36)$$

$$\text{Min. } Z_2 = \sum_{i=1}^N \sum_{t=1}^T \left(\frac{S_{it} - S_{it}^{target}}{S_i^{Max}} \right)^2 \quad (37)$$

$$\text{Min. } Z_3 = \sum_{i=1}^N \sum_{t=1}^T \left(1 - \frac{P_{it}}{PPC_i} \right) \quad (38)$$

in which: N = number of reservoirs; T = number of operating periods; R_{it}^{Power} = hydropower release of the i th

Table 2 | Comparison between relative closeness to the ideal point for the Deb and Kursawe problem

Problem	1	2	3
Deb	0.999 96	0.000 00	0.991 66
Kursawe	0.566 046	0.551 098	0.439 942

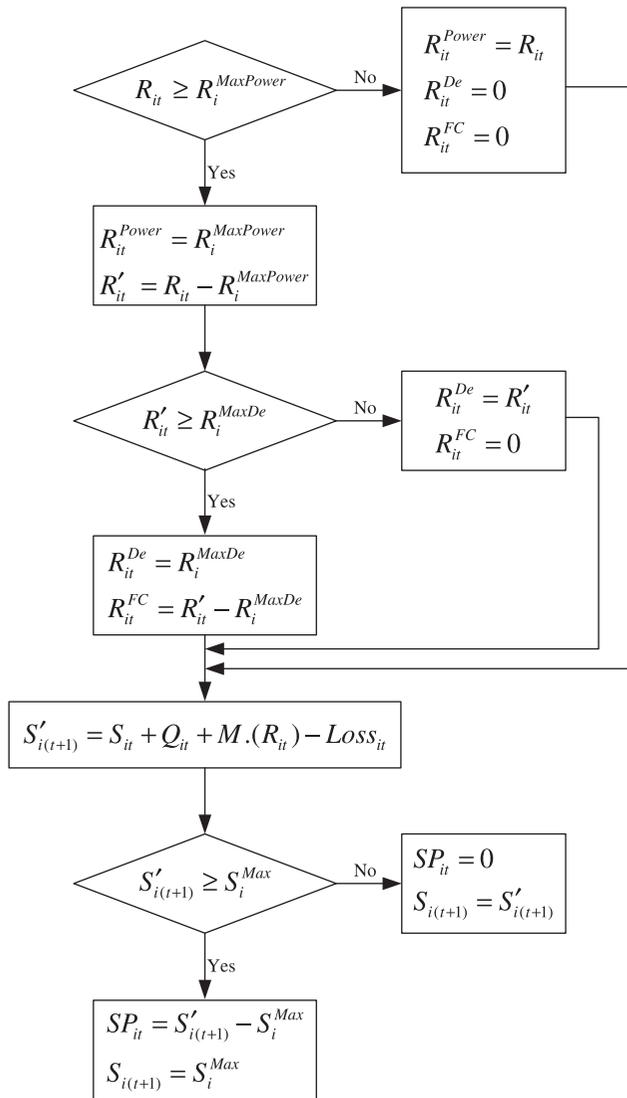


Figure 5 | Relations between different releases in reservoir operation.

reservoir at period t ; R_{it}^{De} = agricultural release of the i th reservoir at period t ; D_{it} = downstream demand of the i th reservoir at period t ; D_i^{Max} = maximum demand of the i th reservoir; S_{it} = storage volume of the i th reservoir at period t ; S_{it}^{target} = targeted storage of the i th reservoir at period t ; S_i^{Max} = maximum storage of the i th reservoir; P_{it} = generated power by the i th reservoir during period t ; and PPC_i = installed capacity of the i th power plant.

In the operational model, the release priority is given to hydropower, then to meeting downstream demand, and finally to flood control and recreational purposes. If the excess water from meeting the aforementioned purposes is more than the reservoir storage, it will be spilled from the

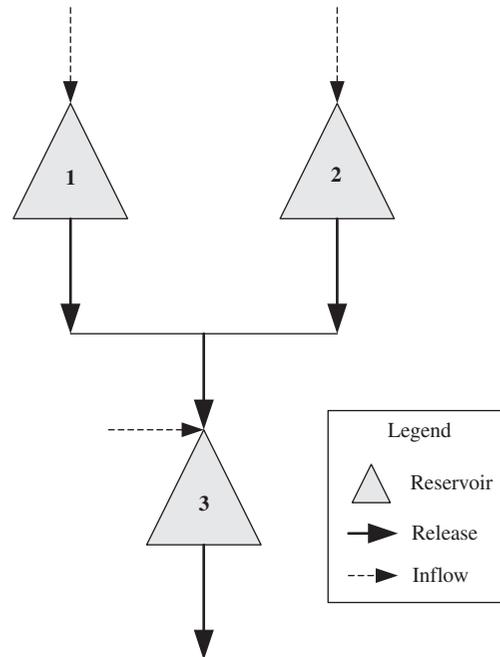


Figure 6 | Schematic of three-reservoir system.

crest. Figure 5 shows the relations between different releases, in which: R_{it} = srelease of the i th reservoir at the period t ; $R_i^{Maxpower}$ = maximum allowable capacity for hydropower release in the i th reservoir; R_i^{MaxDe} = maximum allowable capacity for downstream release in the i th reservoir; R_i^{MaxFc} = maximum allowable capacity for flood control release in the i th reservoir; R_{it}^{Fc} = flood control release of the i th reservoir at period t ; and SP_{it} = volume of spilled water from the i th reservoir at period t .

The model's formulation is constrained by the following relations:

$$P_{it} = Z_4(\gamma, e_i, RP_{it}, PF_i, \overline{H}_{it}, TW_{it}) \tag{39}$$

$$\overline{H}_{it} = (H_{it} + H_{i(t+1)})/2 \tag{40}$$

$$H_{it} = Z_5((S_{it})^3) \tag{41}$$

Table 3 | Characteristics of three-reservoir system

Reservoir number	Active volume (10 ⁶ m ³)	Installed capacity (10 ⁶ W)	Average inflow (10 ⁶ m ³)
1	308.15	290	2013
2	792.34	420	2469
3	748.71	1000	645

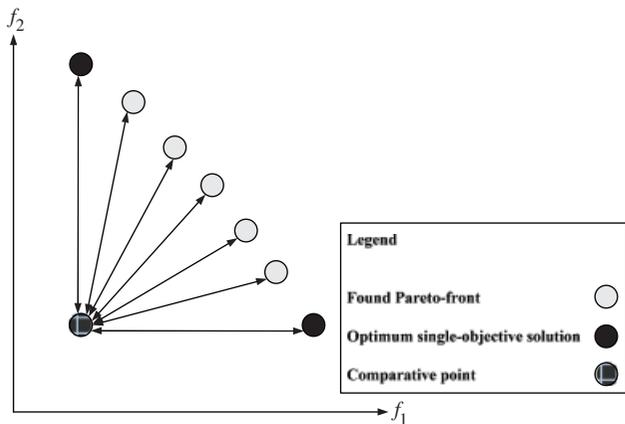


Figure 7 | Comparative point to calculate GD in a bi-objective problem.

$$TW_{it} = Z_6((R_{it}^{Power})^3) \tag{42}$$

$$RPS_{it} = R_{it}^{Power} - RP_{it} \tag{43}$$

$$S_{i(t+1)} = S_{it} + Q_{it} + M_{N \times N}(R_{it} + SP_{it}) - Loss_{it} \tag{44}$$

$$Loss_{it} = Z_7(Ev_{it}, \overline{A}_{it}) \tag{45}$$

$$\overline{A}_{it} = (A_{it} + A_{i(t+1)})/2 \tag{46}$$

$$A_{it} = Z_8(S_{it}) \tag{47}$$

$$R_i^{Min} \leq R_{it} \leq R_i^{Max} \tag{48}$$

$$S_i^{Min} \leq S_{it} \leq S_i^{Max} \tag{49}$$

$$S_{i1} = S_{i(T+1)} \tag{50}$$

in which Z_4 =function of hydropower energy generated; γ =specific weight of water; e_i =efficiency of the i th power plant; RP_{it} =release from power plant for generated power of the i th reservoir at period t ; PF_i =plant factor of the i th power plant; \overline{H}_{it} = average head of the i th reservoir during period t ; TW_{it} =tail water elevation for the i th reservoir at period t ; Z_5 =nonlinear function for transferring storage volume to water elevation; Z_6 =nonlinear function for transferring hydropower release to tail water elevation; H_{it} =water elevation of the i th reservoir at the start of period t ; Q_{it} =inflow to the i th reservoir during period t ; $M_{N \times N}$ =transfer matrix of reservoirs; $Loss_{it}$ =volume of lost water from the i th reservoir at period t ; Z_7 =function for finding volume of lost water considering evaporation rate; Ev_{it} =evaporation depth in the i th reservoir at period t ; \overline{A}_{it} =average surface of the i th reservoir at period t ; A_{it} =water surface of the i th reservoir at the start of period t ; Z_8 =linear function for transferring

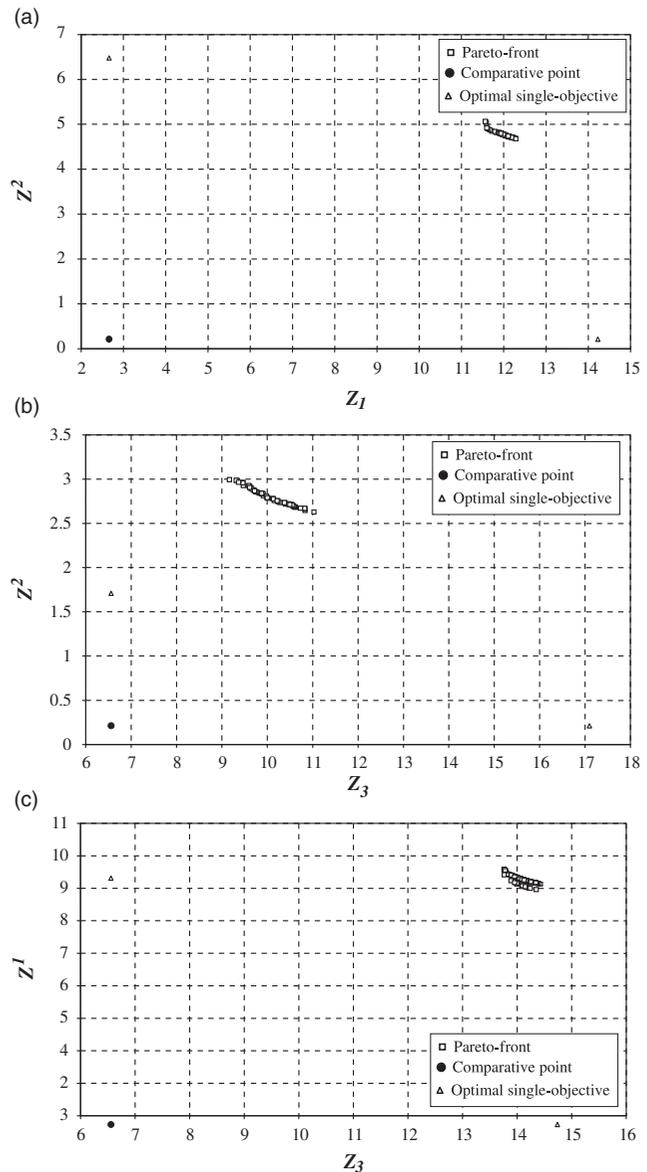


Figure 8 | Pareto front for single-reservoir operation with two objectives: (a) Z_1, Z_2 ; (b) Z_1, Z_3 ; and (c) Z_1, Z_3 .

storage volume to water surface; and S_i^{Min} = minimum storage of the i th reservoir.

To control decision variables within allowable boundaries, monthly releases are found in the search for a feasible solution. Moreover, a dynamic penalty function has been used for monthly reservoir storages:

$$DPF = a(|Min.((S_{it} - S_i^{Min}), (S_i^{Max} - S_{it}), 0)|) + b(|S_{i1} - S_{i(T+1)}|) + c \tag{51}$$

where DPF = dynamic penalty function; and a , b and c = positive constants of the linear relation of the penalty factor. Both Equations (49) and (50) have been considered in Equation (51) and the value of DPF has been added to the objective functions that are presented in Equations (36)–(38).

Algorithm application

To apply the MOPSO algorithm in multireservoir system operation problems, a three-reservoir system has been considered (Figure 6). Table 3 shows the characteristics of reservoirs. At first, optimal operation of a single-reservoir system has been considered and then the nominated method has been used in the multireservoir system. Monthly release and initial storage of each reservoir are decision variables in all problems.

Because there is no predefined Pareto front in unsolved problems, the intersection of the best value of all objectives functions has been considered as the comparative point and distances are measured from it. Figure 7 shows the comparative point of two objectives. Thus, the GD value is calculated from the comparative point.

Single-reservoir operation problem

To apply the nominated method, different combinations of three objectives (Z_1 , Z_2 and Z_3) have been used with the first

Table 4 | Comparison criteria for three different combinations of single-reservoir operation

Objectives	GD	S	NS
Z_1, Z_2	26.0033	0.5684	43
Z_2, Z_3	13.7801	0.4386	56
Z_1, Z_3	44.5006	0.2789	172

method for a five-year period. Thus, the number of decision variables is 61. Resulting Pareto fronts are shown in Figure 8. Results show that all of the points of Pareto fronts have been gathered in the middle regions at a long distance from optimal single objectives. All of the Pareto fronts have been presented after convergence of the algorithm. In fact, the solutions found have a fixed state after considerable iterations. Thus, it is not possible to reach the optimal single-objective using more iterations. On the other hand, there are many feasible points near the center. The Pareto-front characteristics are presented in Table 4. It is noted that the optimal single-objective points have been added to the Pareto front, and then the S characteristic has been calculated. Thus, the gap between Pareto-front points and the optimal single-objective solution has been considered.

To fulfill the existing gaps between Pareto-front and optimal single-objective points, a new technique called warm-up has been applied which uses SOPSO to spread out the Pareto front to the appropriate places with a suitable

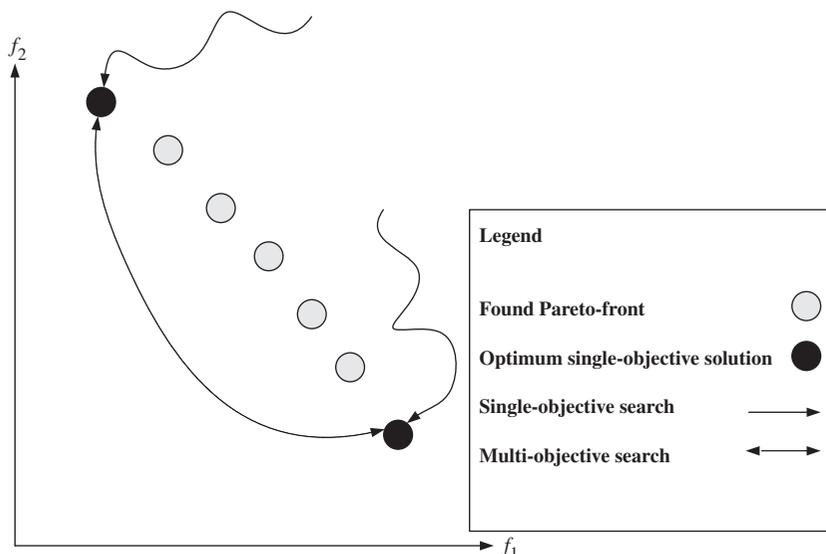


Figure 9 | Searching process using warm-up technique.

distribution of points. This method allocates a swarm to each objective and lets each swarm move towards the near-optimal solutions of their objectives. After completion of the single-objective process, the resulting solutions are used for multi-objective optimization. Figure 9 shows the movement of two swarms in the decision space using the warm-up technique. In fact, the first (best) point of each objective can be found using the warm-up technique and the Pareto-front points can be generated in the optimal/near-optimal points.

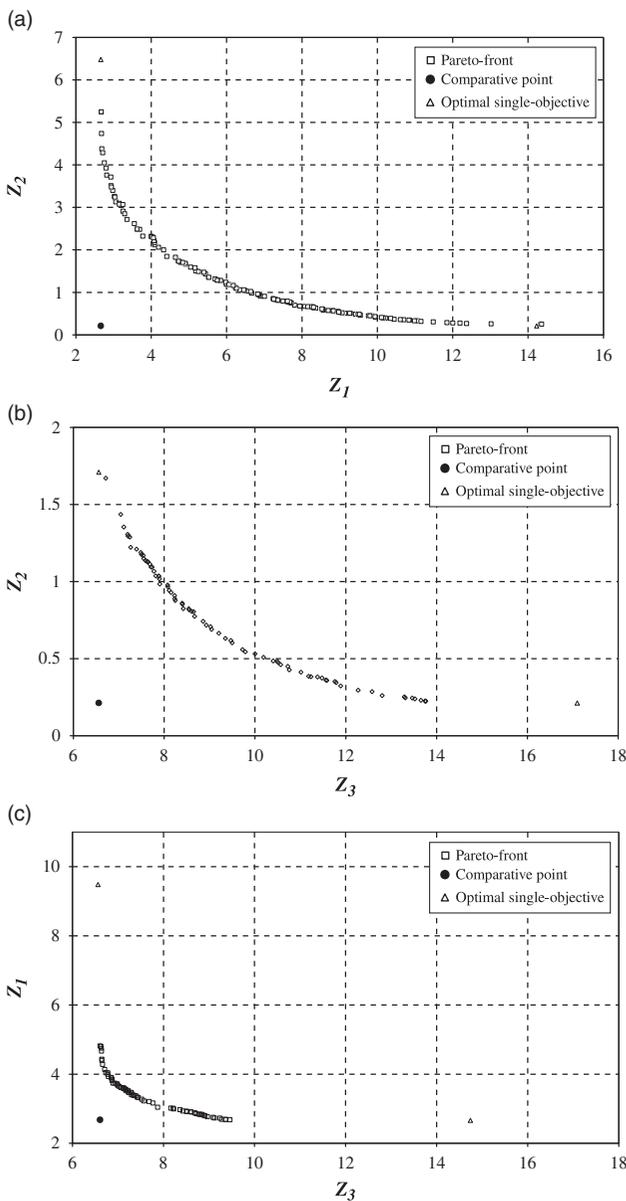


Figure 10 | Pareto front using warm-up technique for single-reservoir operation with (a) Z_1, Z_2 ; (b) Z_2, Z_3 ; and (c) Z_1, Z_3 .

Table 5 | Comparison criteria for three different combinations of single-reservoir operation using warm-up

Objectives	GD	S	NS
Z_1, Z_2	25.4275	0.0598	138
Z_2, Z_3	9.3715	0.2097	78
Z_1, Z_3	34.2373	0.2053	70

To test the warm up technique, three different problems, with the same conditions (the same function evaluation and the same algorithmic parameter) have been considered. The number of particles in all problems is 10 and the number of function evaluations for the Z_1, Z_2 ; Z_2, Z_3 and Z_1, Z_3 are 45 000, 90 000 and 100 000, respectively. Figure 10 shows the optimal Pareto front using the warm-up technique. As is shown, all of the Pareto fronts have an appropriate distribution, from the best value of the first objective to the worst value of it, corresponding to the best value of the second objective. Table 5 shows the comparison criteria of the resulting Pareto front. The value of GD has decreased by 0.5758, 4.4086 and 10.2633 units for the Z_1, Z_2 ; Z_2, Z_3 and Z_1, Z_3 , respectively. Moreover, the S value has been decreased by 0.5086, 0.2289 and 0.0736 units for the Z_1, Z_2 ; Z_2, Z_3 and Z_1, Z_3 , respectively. Thus, the warm-up technique can improve the quality of the results.

Due to the effects of different objectives on each other, the coordinates of the found Pareto front in combinations of two objectives may be different from the coordinates of a projected two-dimensional Pareto front resulted from a three-objective solution. To apply the MOPSO method in a

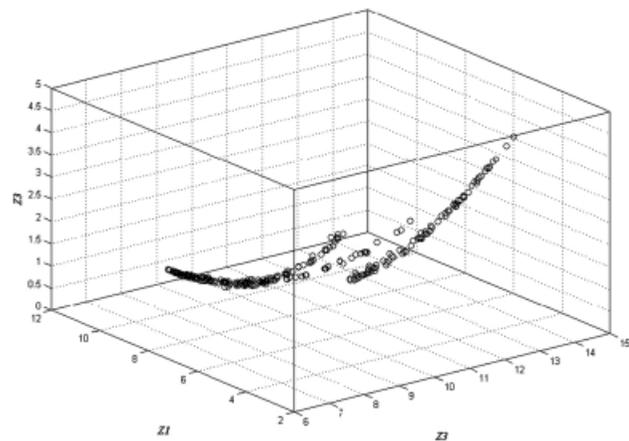


Figure 11 | Optimal Pareto front for the three-objective operation problem of single-reservoir system.

more complex problem, the optimal three-objective operation of a reservoir has been considered. The number of particles in this problem is 10 and the number of function evaluations is 100 000. Figure 11 presents the optimal Pareto front considering the three-objective operation. The comparison criteria of this Pareto front are 32.045, 0.209 and 289 for the *GD*, *S* and *NS*, respectively. Figure 12 presents two-dimensional coordinates of Pareto fronts considering two and three

objectives. It shows that the three-objective problem has not produced the same results as has been achieved by considering different two-objective combinations.

Multireservoir operation problems

To test the efficiency of the MOPSO algorithm using warm-up in more complex and non-convex problems, a

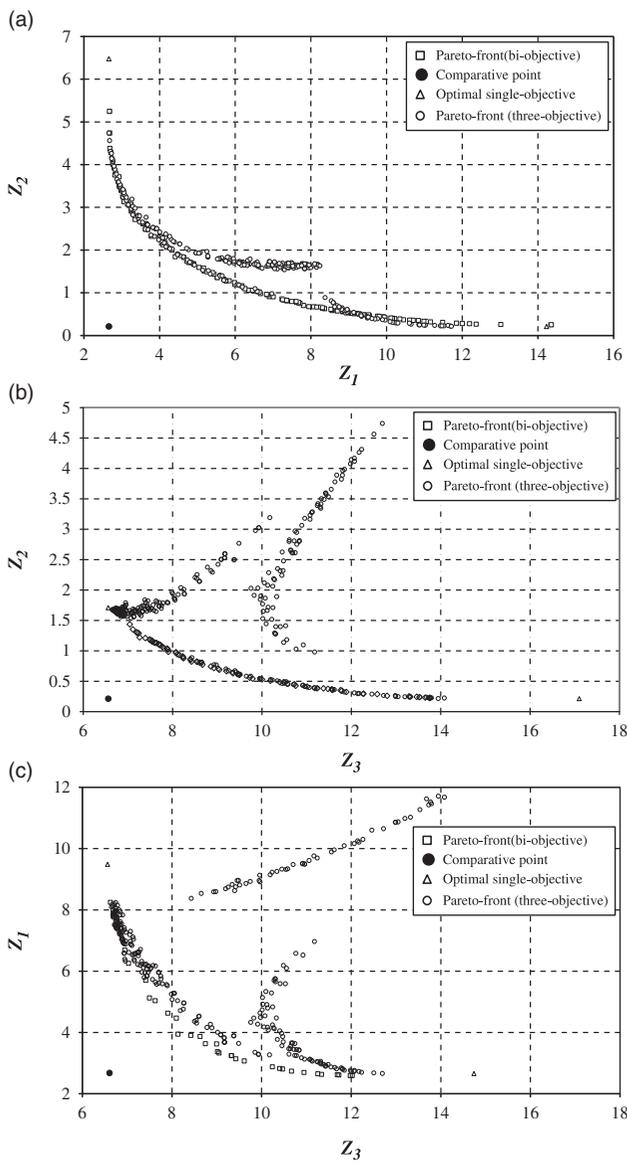


Figure 12 | Two-dimensional coordinates of nondominated solutions for two- and three-objective problems: (a) Z_1, Z_2 ; (b) Z_2, Z_3 and (c) Z_1, Z_3 .

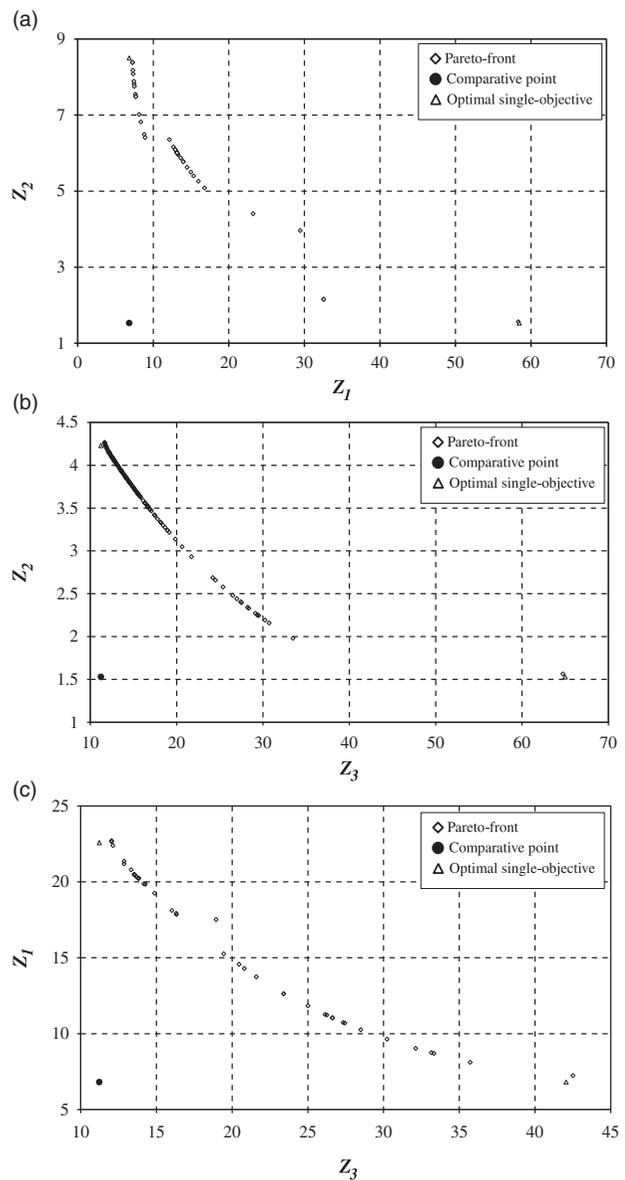


Figure 13 | Pareto front using warm-up technique for multireservoir operation with two objectives: (a) Z_1, Z_2 ; (b) Z_2, Z_3 and (c) Z_1, Z_3 .

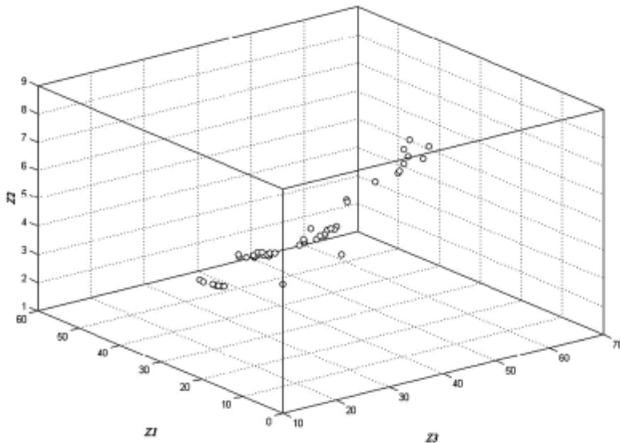


Figure 14 | Pareto front using warm-up technique for multi-reservoir operation with three objectives (Z_1 , Z_2 , and Z_3).

three-reservoir system has been considered (Figure 6). The transfer matrix among reservoirs is

$$M_{3 \times 3} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix}. \tag{52}$$

In this system, the number of decision variables for a five-year operational period is 183. Different combinations of

objectives are selected as the objective functions of multi-reservoir operation problems.

Figures 13 and 14 respectively show nondominated solutions for two- and three-objective operations. Results are presented for all combinations of objectives. The MOPSO algorithm using the warm-up technique is capable of finding the Pareto front with a suitable distribution. As is shown, the number of constraints in the three-reservoir operation model is more than in the single-reservoir system. Thus, the number of feasible solutions for a three-reservoir system with all combinations of objectives has decreased compared to a single-reservoir system. On the other hand, there are different series of monthly releases and reservoir storages for each nondominated solution, which is a combination of objectives. As an example, the monthly releases for the best value of Z_1 and the monthly reservoir storages for the best value of Z_2 in the three-objective operation have been presented in Figures 15 and 16. In these time series, the monthly releases and reservoir storages have the minimum distance from the demand and target storage, respectively.

To demonstrate the capability of the MOPSO algorithm, the best (minimum) value of each objective has been calculated by nonlinear programming (NLP). These values have been normalized to values between zero and one. Thus, one has been assigned the minimum value of each objective

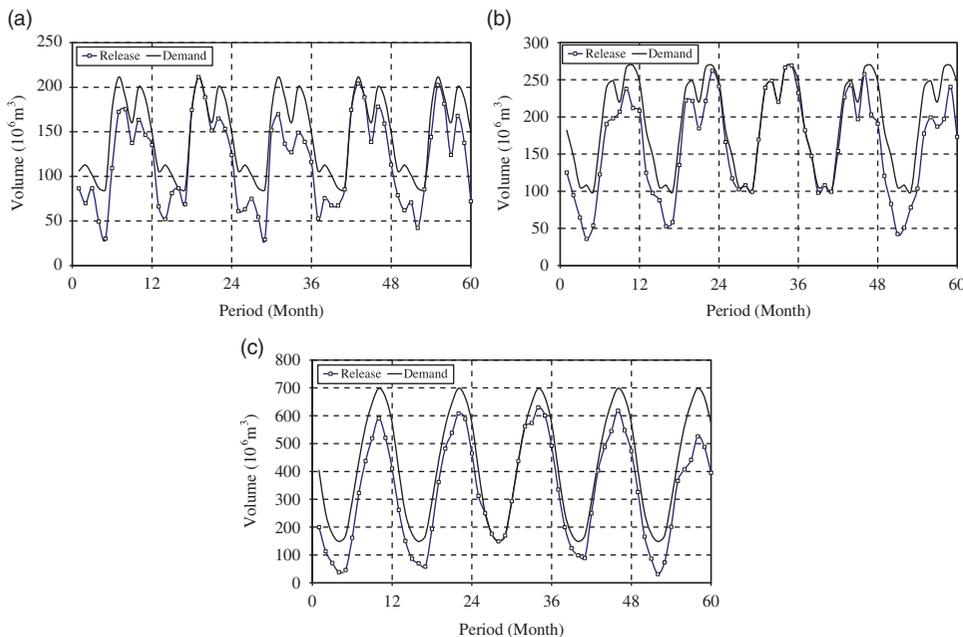


Figure 15 | Monthly releases corresponding to the best value of Z_1 in the three-objective operation for the: (a) first; (b) second; and (c) third reservoir.

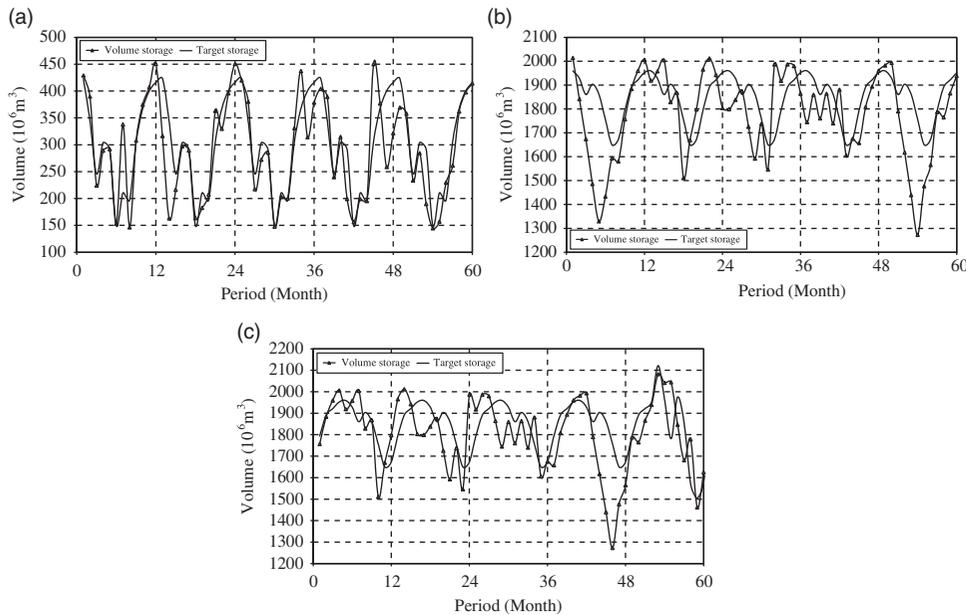


Figure 16 | Monthly reservoir storages corresponding to the best value of Z_2 in the three-objective operation for the: (a) first; (b) second; and (c) third reservoir.

(best one) while the maximum value is zero. Table 6 shows normalized quantities for the nearest point of the Pareto front in each objective function for different problems. Results show that the MOPSO algorithm, coupled with the warm-up technique, can produce nondominated solutions which are more than 90% of single-objective optimal points. Moreover, results show that the premature convergence which occurs in the MOPSO algorithm without warm-up does not happen when using the warm-up technique.

Table 6 | Comparison of the best values of objectives in multireservoir operation problem

Algorithm	Objectives	Z_1	Z_2	Z_3
NLP	Z_1	1	0	0.4266
	Z_2	0	1	0
	Z_3	0.6944	0.5997	1
PSO	Z_1, Z_2	0.9917	0.0167	0.4266
		0.0026	0.9951	0.0046
	Z_1, Z_3	0.6919	0.6063	0.9851
		0.9917	0.0160	0.4182
	Z_2, Z_3	0.6949	0.6074	0.9920
		0.0026	0.9951	0.0046
	Z_1, Z_2, Z_3	0.9956	0.0309	0.4354
		0.0107	0.9969	0.0099
		0.6953	0.6014	0.9921

CONCLUDING REMARKS

This paper presents an introduction and comparison of three methods based on the PSO algorithm. The first and third methods use dominated solutions in the search process while the second method uses nondominated solutions. To test the applicability of the proposed methods, four test problems were considered. General distance, spacing and number of solutions were identified as the comparison criteria and the TOPSIS were used to rank different methods. To compare different methods, weights of each criterion were considered to be the same. Results showed that the first method in all problems can find nondominated solutions with minimum distance from ideal points/solutions and the furthest distance from anti-ideal points/solutions. The nominated method was then presented as the optimization tool in a reservoir system operation problem. A new technique called warm-up was coupled with the MOPSO algorithm to improve the quality and the quantity of Pareto fronts. This technique uses the mechanism of single-objective searching in SOPSO. Three different bi-objective problems in single-reservoir operation were considered. Results showed that the proposed technique can find solutions around optimal solutions with a proper distribution of points. The warm-up technique was then used in more complex and non-convex problems of single and

multireservoir operation with different combinations of objectives. In all problems considered, the MOPSO algorithm using the warm-up technique found an appropriate Pareto front. Solutions achieved by the MOPSO algorithm for each objective were near to the single-objective optimal points obtained from nonlinear programming. In addition, there were many nondominated solutions between those points which can provide a wide range and many choices of points, including several combinations of objectives for managers and decision-makers to choose from. Therefore, the proposed MOPSO algorithm coupled with the warm-up technique can increase the flexibility and efficiency of decisions made for reservoir operation with different priorities and/or situations.

ACKNOWLEDGEMENTS

The authors thank three anonymous reviewers for providing constructive comments that greatly improved this paper.

REFERENCES

- Abido, M. A. 2007 Two-level of nondominated solutions approach to multiobjective particle swarm optimization. In: *Genetic and Evolutionary Computation Conference (GECCO'2007) London, July* (Thierens, D. (Ed.)). pp. 726–733, ACM Press, London.
- Afshar, A., Sharifi, F. & Jalali, M. R. 2009 Applying the non-dominated archiving multi-colony ant algorithm for multi-objective optimization: application to multi-purpose reservoir operation. *Engng. Optimiz.* **41**(4), 313–325.
- Balter, A. M. & Fontane, D. G. 2006 A multiobjective particle swarm optimization model for reservoir operations and planning. In: *Proc. International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, June* pp. 1544–1552.
- Bozorg Haddad, O. & 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.* **3**(9), 235–250.
- Chang, L.-C. & Chang, F.-J. 2009 Multi-objective evolutionary algorithm for operating parallel reservoir system. *J. Hydrol.* **337**(1), 12–20.
- Chuanwen, J. & Bompard, E. 2005 A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment. *Energy Conserv. Mngmnt.* **46**(17), 2689–2696.
- Coello, C. A. & Lechuga, M. S. 2002 MOPSO: A proposal for multiple objective particle swarm optimization. In: *Proc. IEEE Congress on Evolutionary Computation (CEC'2002), May*. IEEE Service Center, Piscataway, NJ, vol 2, pp. 1677–1681.
- Deb, K. 1999 Multi-objective genetic algorithms: problem difficulties and construction of test problem. *Evol. Comput.* **7**(3), 205–230.
- Deb, K., Agrawal, S., Pratab, A. & Meyarivan, T. 2000 A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Proc. Parallel Problem Solving Nature VI Conference*. Springer, Berlin, pp. 849–858.
- Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. 2001 *Scalable Test Problems for Evolutionary Multi-Objective Optimization*. Technical Report No.112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.
- Fieldsend, J. E. & Singh, S. 2002 A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence. In: *Proc. 2002 UK Workshop on Computational Intelligence, Birmingham, September*. Exeter University, UK, pp. 37–44.
- Hu, X. & Eberhart, R. 2002 Multiobjective optimization using dynamic neighborhood particle swarm optimization. In: *Proc. Congress on Evolutionary Computation (CEC'2002), May*. IEEE Service Center, Piscataway, NJ, pp. 1677–1681.
- Izquierdo, J., Montalvo, I., Perez, R. & Fuertes, V. 2008 Design optimization of wastewater collection network by PSO. *J. Comput. Math. Appl.* **56**(3), 777–784.
- Kennedy, J. & Eberhart, R. 1995 Particle swarm optimization. In: *Proc. International Conference on Neural Networks, Perth, Australia*. IEEE, Piscataway, NJ, pp. 1942–1948.
- Kim, T., Heo, J.-H., Bae, D.-H. & Kim, J.-H. 2008 Single-reservoir operating rules for a year using multiobjective genetic algorithm. *J. Hydroinf.* **10**(2), 163–179.
- Kita, H., Yabumoto, Y., Mori, N. & Nishikawa, Y. 1996 Multi-objective optimization by means of the thermodynamical genetic algorithm. In: *Parallel Problem Solving from Nature-PPSN IV*, Lecture Notes in Computer Science Voigt, H.-M., Ebeling, W., Rechenberg, I. & Schwefel, H.-P. (Eds.), Springer-Verlag, Berlin, pp. 504–512.
- Knowles, J. D. & Corne, D. W. 2000 Approximating the nondominated front using the pareto archive evolution strategy. *Evolut. Comput.* **8**(2), 149–172.
- Kumar, D. & Reddy, J. 2007 Multiple reservoir operation using particle swarm optimization. *J. Wat. Res. Plann. Mngmnt.* **133**(3), 192–202.
- Kursawe, F. 1991 A variant of evolution strategies for vector optimization. In: *Parallel Problem Solving from Nature, 1st Workshop, Proceedings* vol 496 (Schwefel, H.-P. & Manner, R. (Eds.)), Lecture Notes in Computer Science, Springer, Berlin, pp. 193–197.
- Matott, L. S., Rabideau, A. J. & Craig, J. R. 2006 Pump-and treat optimization using analytic element method flow models. *Adv. Wat. Res.* **29**(5), 760–775.
- Montalvo, I., Izquierdo, J., Perez, R. & Tung, M. M. 2008 Particle swarm optimization applied to the design of water supply systems. *J. Comput. Math. Appl.* **56**(3), 769–776.
- Michalewicz, Z. & Schouenauer, M. 1996 Evolutionary algorithms for constrained parameter optimization problems. *Evolut. Comput.* **4**, 1–32.
- Mostaghim, S. & Teich, J. 2003 Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: *IEEE Swarm Intelligence Symposium, Indianapolis, IIN, April* IEEE Service Center, Piscataway, NJ, pp. 26–33.

- Mostaghim, S. & Teich, J. 2004 Covering pareto-optimal fronts by subswarms in multi-objective particle swarm optimization. In: *Proc. Congress on Evolutionary Computation (CEC'2004), June*. IEEE Service Center. Portland, OR, pp. 1404–1411.
- Parsopoulos, K. E. & Vrahatis, M. N. 2002 Particle swarm optimization method in multiobjective problems. In: *Proc. 2002 ACM Symposium on Applied Computing, Madraid, Spain, March*. ACM Press, Spain, pp. 603–607.
- Parsopoulos, K. E., Tasoulis, D. K. & Vrahatis, M. N. 2004 Multi-objective optimization using parallel vector evaluated particle swarm optimization. In: *Proc. IASTED International Conference on Artificial Intelligence and Applications (AIA'2004), Innsbruck, Austria, February*. ACTA Press, Austria, pp. 823–828.
- Reddy, M. J. & Kumar, D. N. 2007a [An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design](#). *Engng. Optim.* **39**(1), 49–68.
- Reddy, M. J. & Kumar, D. N. 2007b [Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation](#). *Hydrol. Process.* **21**(21), 2897–2909.
- Reddy, M. J. & Kumar, D. N. 2009 [Performance evaluation of elitist-mutated multi-objective particle swarm optimization for integrated water resources management](#). *J. Hydroinf.* **11**(1), 78–88.
- Schott, J. 1995 *Fault Tolerant Design using Single and Multi-criteria Genetic Algorithm*. Master's thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. Cambridge, MA.
- Shi, Y. & Eberhart, R. C. 1998a Parameter selection in particle swarm optimization. In: *Evolutionary Programming VII* (Porto, V. W., et al (Eds.)). Springer, Berlin, pp. 611–616.
- Shi, Y. & Eberhart, R. C. 1998b A modified particle swarm optimizer. In: *Proc. IEEE International Conference on Evolutionary Computation, May*. IEEE Service Center, Piscataway, NJ, pp. 69–73.
- Suribabu, C. R. & Neelakantan, T. R. 2006 [Design of water distribution networks using particle swarm optimization](#). *Urban Wat. J.* **3**(2), 111–120.
- Veldhuizen, D. 1999 *Multiobjective Evolutionary Algorithms: Classification, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering Air Force Institute of Technology. Dayton, OH.
- Yang, J., Zhou, J., Liu, L. & Li, Y. 2009 [A novel strategy of Pareto-optimal solution searching in multi-objective particle swarm optimization \(MOPSO\)](#). *J. Comput. Math. Appl.* **57**, 1995–2000.
- Yoon, K. P. & Hwang, C. L. 1995 *Multiple Attribute Decision Making: An Introduction*. Sage Publications, Thousand Oaks, CA.

First received 17 December 2009; accepted in revised form 28 June 2010. Available online 14 December 2010