Journal of Water Supply: Research and Technology—AQUA | 67.1 | 2018

Evaluation of contemporary evolutionary algorithms for optimization in reservoir operation and water supply

Mohammad Ehteram, Hojat Karami, Sayed-Farhad Mousavi,
Saeed Farzin and Ozgur Kisi

ABSTRACT

This study evaluates three contemporary evolutionary algorithms, namely, shark, genetic, and particle swarm algorithms, for optimization in reservoir operation and water supply. The Klang gate dam in Malaysia is selected as the case study to optimize reservoir operation. The key objective of this study is the minimization of water deficits based on demands and released water. The global solution of the problem is computed based on software Lingo and the average solution of the shark algorithm is able to attain 99% of global solution. As well, the shark algorithm can furnish demand values at a faster convergence rate than both genetic and particle swarm algorithms. The reliability index and resiliency index, as useful indices in water resource management, are used and the values of these indices have the highest percent for the shark algorithm, indicating its superiority over other evolutionary algorithms.

Key words | genetic algorithm, particle swarm algorithm, shark algorithm, water resource management

Mohammad Ehteram
Hojat Karami (corresponding author)
Sayed-Farhad Mousavi
Saeed Farzin
Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering,
Semnan University,
Semnan,
Iran
E-mail: *hkarami@semnan.ac.ir*

Ozgur Kisi
School of Natural Sciences and Engineering,
Ilia State University,
Tiblisi,
Georgia

NOTATION

C	penalty coefficient
D	demand
f_{si}	residency index
F_{si}	the number of periods of happened failure
I	inflow
k	the maximum number of shark movement toward the position
$loss$	loss value for reservoir operation
m	number of points of local search
MG	million gallons
P_i	best position of particle
R	released water
S	storage
S_{max}	maximum storage
S_{min}	minimum storage
V	velocity
w_f	criteria's weight
X	decision variable
Z	the new shark position after local search
η_k	constant value
R_1	a random number between 0 and 1
Of	objective function
γ	random number
α_k	inertia coefficient
α_V	volumetric reliability
β_k	limiter velocity
χ	constriction coefficient
λ_v	vulnerability index
γ_i	residency index
Δt	time interval

doi: 10.2166/aqua.2017.109

INTRODUCTION

Increasing drought periods and limiting resources cause many difficulties for decision-makers in managing resources. Water scarcity in different regions of the world has emerged to become a serious problem (Bozorg Haddad *et al.* 2008a, 2011). Construction of large dams can be one choice for water supply, but it may not be the best approach in term of water demands. This is because the construction of large dams entails high costs and, as such, cannot be the first choice. In fact, the best method for providing water and energy supply should be an economical one (Oliveira & Loucks 1997; Fallah-Mehdipour *et al.* 2013a; Kumar *et al.* 2016). Water resource management is a science which studies the resource operation based on economical approaches and optimization. It is important to have the highest efficiency for resources operation. The problems of water resource management can be defined in the framework of an optimization problem (Ngoc *et al.* 2013). The aim of different projects can be defined based on the objective function and, in fact, the effective variable for the problems can be defined based on decision variables. As well, the limitations of projects or problems can be defined based on different constraints. Water resource management attempts to provide comprehensive programming for regulating different resources such as water and energy (Fallah-Mehdipour *et al.* 2011, 2013b). This science uses different mathematical models for reservoir and power plant operations to determine the amount and time of water release for downstream consumers (Ostadrahimi *et al.* 2012; Shirzad & Tabesh 2016). Linear, nonlinear, and dynamic programming methods are used as mathematical models for the problems of reservoir and power plant operation (Kumar & Reddy 2006). Yet these methods have their inherent limitations, which result in difficulties in defining new methods for their replacement. For example, if a certain problem has a nonlinear objective function, linear programming cannot be used as the solving tool and, also, for problems with many decision variables, dynamic programming method requires high computational time. Thus, researchers attempt to find more powerful methods than the conventional methods such as nonlinear programming method or dynamic programming method. Evolutionary algorithms or meta-heuristic algorithms are new approaches for solving complex problems in water resource management and in reservoir and power plant operation. Tospornsampan *et al.* (2005) used a simulated annealing method for reservoir operation to supply irrigation deficits and the results were compared with those by genetic algorithm (GA). Results indicated that a simulated annealing method could furnish demands better than a GA. Jothiprakash *et al.* (2011) generated rule curves for reservoir operation based on GA and results showed that the GA method had the higher reliability index than nonlinear programming method for water supply. Jalali *et al.* (2007) used ant colony optimization method for multi-reservoir operation. The aim of the study was the maximization of the generated power from power plants. Results indicated that ant colony optimization method could generate more power for downstream consumers. Bozorg Haddad *et al.* (2008b) studied reservoir operation based on honey bee mating optimization method. Results showed that the method had faster convergence speed and more satisfactory downstream demands than the GA. Its higher power production and faster convergence indicated that the diploid method demonstrated effective performance in hydropower operation. Ghimire & Reddy (2013) used particle swarm algorithm for the generation of rule curves for hydropower operation. Results indicated that the percentage of power deficit was decreased with particle swarm algorithm. Bozorg Haddad *et al.* (2015) used bat algorithm for the optimization of hydropower and results showed the annual power production was more than its counterparts by genetic and particle swarm algorithms. Also, Bozorg Haddad *et al.* (2015) used the water cycle algorithm for reservoir operation. Results indicated that the water cycle algorithm could supply water demands with less vulnerability index, yet these new meta-heuristic methods demonstrated high performance for water supply in water resource management problems.

In this study, the shark algorithm is used as one of the meta-heuristic and solving tools for reservoir optimization. The shark algorithm was used for benchmark functions and complex problems, and results showed the shark algorithm had high potential for applications in engineering and optimization problems (Ehteram *et al.* 2017a). The shark algorithm has been employed for the maximization of power production in a power plant. Results showed that the power production based on the shark algorithm was higher than conventional meta-heuristic algorithms (Ehteram *et al.* 2017b).

The Klang gate dam is used as a case study for irrigation demands supply. The objective function for this problem is the minimization of irrigation deficit for downstream consumers. As well, three kinds of inflow are considered in this problem to understand the supply routes of water demands based on reservoir operation and meta-heuristic algorithm. The GA and particle swarm algorithm, being successful algorithms in water resource management, are employed as benchmarks for comparison with results by the shark algorithm. In addition, different indices such as reliability, vulnerability, and resiliency index as widespread indices in water resource management, are used to evaluate the performances of different methods.

SHARK ALGORITHM

Sharks have powerful olfaction so that they can find their prey based on this ability. Sharks move to the odor resource and when the concentration of odor is greater, the shark moves faster because it has found the food resource. The shark algorithm acts based on the following assumptions:

1. Fish in the water are considered as prey for sharks. Each fish is considered to have an injured body so that it produces blood from itself. Thus, the movement of the fish is slow and it is approximately fixed.
2. The blood produced from the fish is distributed based on a regular form. The water movement and tides do not have any effect on blood distribution. The shark follows the odor particle and particles which are close to injured fish have more odor concentration. Thus, the shark can find prey based on odor particle.
3. A blood resource is generated by a fish and the shark attempts to find this blood resource as its prey.

First, initial populations of sharks are considered and the initial positions of sharks are shown based on a vector. Equation (1) shows the solution candidates based on the sharks' positions while Equation (2) shows each solution candidate has some decision variables.

$$[X_1^1, X_2^1, \ldots, X_{NP}^1] \leftarrow NP = population(size) \quad (1)$$

$$X_i^1 = [x_{i,1}^1, x_{i,2}^1, \ldots, x_{i,ND}^1] \quad (2)$$

where each $x_{i,j}^1$ shows the jth decision variable for one solution candidate and ND is the number of decision variables in the search space of a problem.

Each shark has specific velocities for its movement, which are considered as another solution candidate. Each velocity, like position, has ND decision variables.

$$[V_1^1, V_2^1, \ldots, V_{NP}^1] \quad (3)$$

$$V_i^1 = [v_{i,1}^1, v_{i,2}^1, \ldots, v_{i,ND}^1] \quad (4)$$

An important point is related to the movement direction of sharks. The movement direction of the shark corresponds to the detected odor intensity. The shark moves towards higher odor intensity and the shark velocity will be higher when it detects higher odor intensity. Thus, this phenonmenon can be shown based on a mathematical equation. Equation (5) shows the variation of the objective function versus shark velocity as the shark velocity changes with variation of odor intensity.

$$V_i^k = \eta_k . R_1 . (\nabla Of)|_{X_i^k} \leftarrow i = 1, \ldots, NP; \quad k = 1, \ldots, k_{\max} \quad (5)$$

where Of is the objective function, R_1 is a random number between 0 and 1, η_k is a constant between 0 and 1, and k_{max} is the maximum number of shark movements toward the position.

The velocity can be rewritten based on the following equation:

$$V_{i,j}^k = \eta_k R_1 \left(\frac{\partial Of}{\partial x_j} \right) |_{X_{i,j}^k} \quad (6)$$

However, another term should be added to the previous equation because the shark movement has inertia and the inertia term should be considered for Equation (6).

$$v_{i,j}^k = \eta_k . R_1 . \left(\frac{\partial (Of)}{\partial x_j} \right) |_{x_{i,j}^k} + \alpha_k R_2 v_{i,j}^{k-1} \quad (7)$$

where α_k is known as the inertia coefficient and a larger value of this parameter denotes that each velocity is dependent more on the previous velocity; and R_2 is the momentum coefficient for the shark algorithm.

The momentum coefficient causes increased search diversity and the chance of attaining global solution increases with the consideration of this term.

The shark velocity has a limitation and thus a limiter should be added to the equation so that the velocity is not more than the permissible bound. Equation (8) shows the term β as the limiter velocity.

$$\left|v_{i,j}^k\right| = \min\left[\left|\eta_k.R_1.\frac{\partial(of)}{\partial x_j}\right|_{x_{i,j}^k} + \alpha_k.R_2 v_{i,j}^{k-1}|, \left|\beta_k.v_{i,j}^{k-1}\right|\right] \tag{8}$$

where β_k is limiter velocity.

The shark position can be updated based on the following equation:

$$Y_i^{k+1} = X_i^k + V_i^k \Delta t_k \tag{9}$$

where Y_i^{k+1} is the new shark position, X_i^k is the previous shark position, and Δt_k is the time interval.

Sharks have a specific movement which is known as rotational shark (Figure 1). The rotational movement is considered as local search based on the following equation:

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3 Y_i^{k+1} \tag{10}$$

where $Z_i^{k+1,m}$ is the new shark position after local search, m is the number of points of local search, and R_3 is a coefficient between -1 and 1.

Finally, shark selects the best position among several positions based on the following equation. For example, the aim of the problem is the minimization of the objective function:

$$X_i^{k+1} = argmin\{OF(Y_i^{k+1}), OF(Z_i^{k+1,1}), \ldots, OF(Z_i^{k+1,M})\} \tag{11}$$

Figure 2 shows the flowchart of the shark algorithm.

GENETIC ALGORITHM

The GA is a well-known algorithm in engineering and optimization files. Real coded GA is used for continuous variables and has three important steps: elitist strategy, crossover, and mutation. The elitist strategy keeps the best members of the population for increasing convergence chance. Crossover operator combines two chromosomes with one another. Mutation operator leads to the increase of the population diversity.

Elitist operator

Elitist operator keeps the best members to be considered directly in the next generation and thus, through this operator, the convergence chance is increased. Crossover and mutation operators are then used for generating new populations.

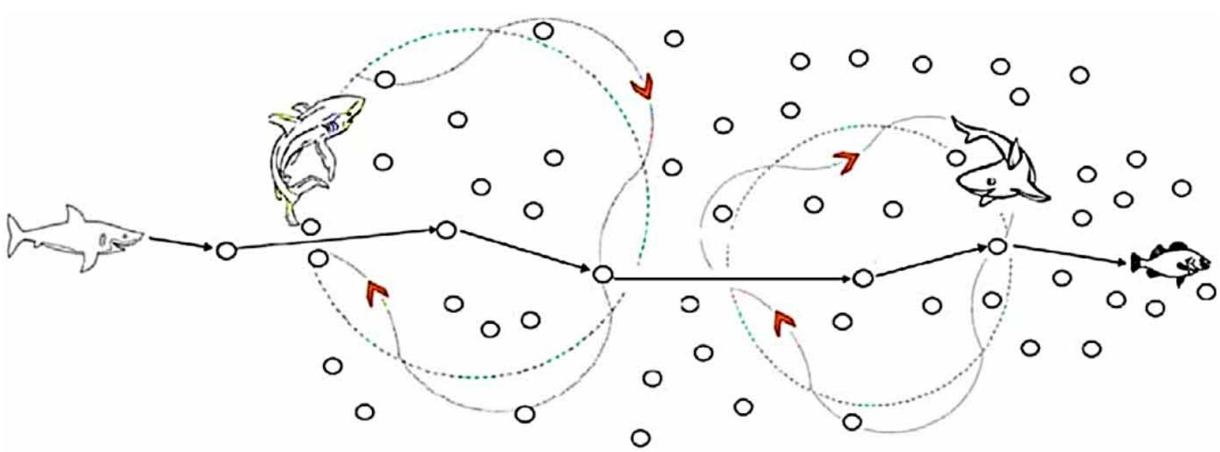

Figure 1 | Rotational movement as local search in the shark algorithm.

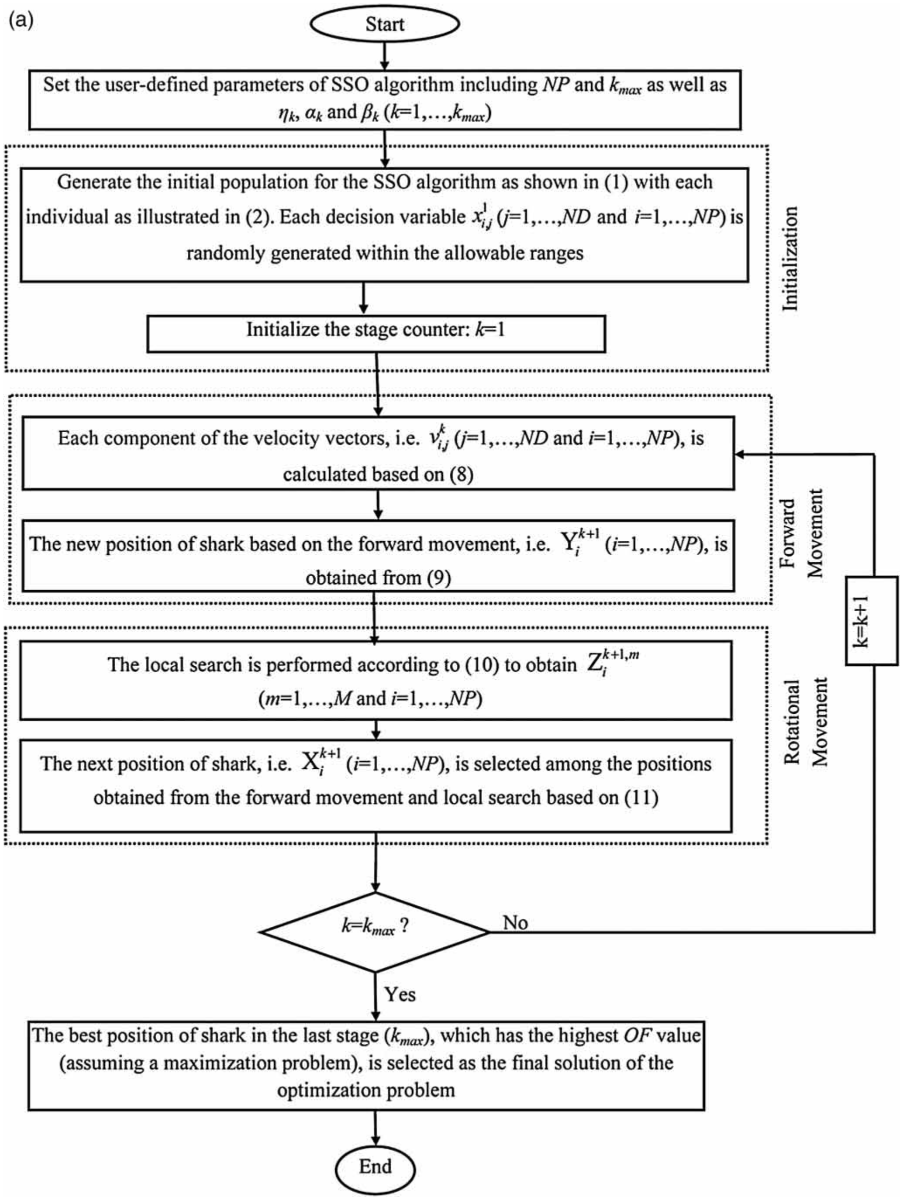

Figure 2 | (a) Shark algorithm method and (b) Klang gate dam. (*Continued*.)

Crossover operator

A random number (parameter u) is first considered and parameter γ is generated based on the following equation:

$$\gamma = \begin{bmatrix} (\alpha u)^{\frac{1}{\eta_c + 1}} \leftarrow if\left(u \leq \frac{1}{\alpha}\right) \\ \left(\frac{1}{2 - \alpha u}\right)^{\frac{1}{\eta_c + 1}} \leftarrow otherwise \end{bmatrix} \tag{12}$$

where $\alpha = 2 - \beta^{-(\eta_c+1)}$ and β is computed based on the following equation:

$$\beta = 1 + \frac{2}{x_i^2 - x_i^1} \min\left[(x_i^{(1)} - x_i^l).(x_i^u - x_i^2)\right] \tag{13}$$

where x denote the parents of children chromosomes; parameter η is the distribution index and a larger value of this parameter denotes that the children chromosomes are closer to the parent chromosomes (global solutions); x_i^l

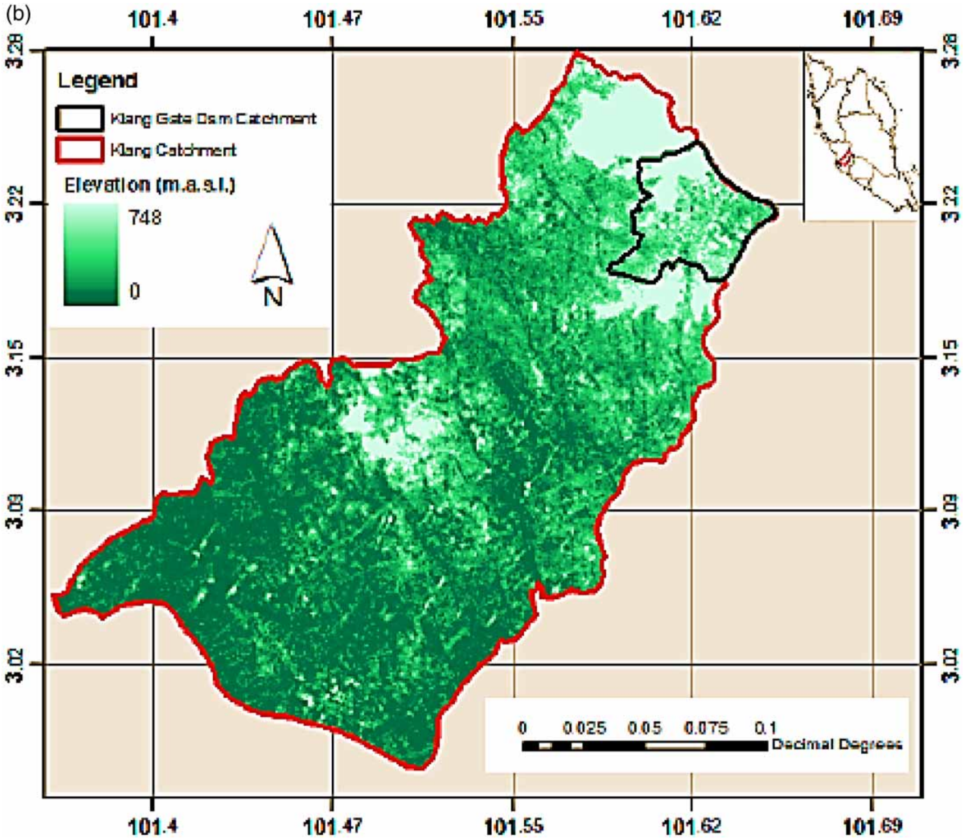

Figure 2 | Continued.

and x_i^u are the upper and lower limit of decision variables. The children candidates are considered based on the following equation:

$$y_i^1 = 0.5[(x_i^{(1)} + x_i^{(2)}) - \gamma\left|x_i^{(2)} - x_i^{(1)}\right|] \tag{14}$$

$$y_i^2 = 0.5[(x_i^1 + x_i^2) + \gamma|x_i^2 - x_i^1|] \tag{15}$$

where y are the children chromosomes.

Mutation operator

A mutation operator selects a numbers of genes for mutation and the new chromosomes are generated. First, one of the parent vectors $(X = x_1, x_2, \ldots, x\mathrm{n})$ is selected and then mutation is considered based on the following equation:

$$x_i' = \begin{bmatrix} x_i + \Delta(t, x_i^u - x_i) \\ x_i - \Delta(t, x_i - x_i^l) \end{bmatrix} \tag{16}$$

and $\Delta(t, y)$ is computed based on the following equation:

$$\Delta(t, y) = y\left(1 - r^{\left(1-\frac{t}{t_{\max}}\right)^b}\right) \tag{17}$$

where t is the maximum iteration.

PARTICLE SWARM ALGORITHM

The particle swarm algorithm is mainly based on particle velocity and position. If the search space of the problem is a D dimensional space, the positions and velocities of particles are shown as vectors $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ and as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$, respectively. The best position of particle at iteration i is shown as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^t$. As well, index g indicates the best particles in a swarm in all iterations. The velocities and positions are updated based on the following equations:

$$v_{id}^{n+1} = \chi\left[wv_{id}^n + c_1 r_1^n \frac{(p_{id}^n - x_{id}^n)}{\Delta t} + c_2 r_2^n \left(\frac{p_{gd}^n - x_{id}^n}{\Delta t}\right)\right] \quad (18)$$
$$x_{id}^{n+1} = x_{id}^n + \Delta t v_{id}^{n+1}$$

where q is the inertia coefficient, c is the acceleration coefficient, r is a random value between 0 and 1, and χ is the constriction coefficient.

For the particle swarm algorithm χ equals 1.

The initial population is considered and initial velocities and positions are generated. Then, the objective function is computed for each particle and the local solution and global solution are selected in any iteration and all iterations, respectively. Velocities and positions are then updated and the algorithm continues to satisfy the specified criteria.

CASE STUDY

The Klang gate dam was constructed in 1958. An important duty of the dam is related to the water supply for people living in the downstream of the dam (Figure 2). The monthly inflow to this dam can be divided into three groups, namely, high inflow, low inflow, and medium inflow. The storage of the reservoir is divided into ten groups from the analysis of 200 months. The objective function for this case study is the minimization of square differences between demand and released water. The following equation shows the objective function:

$$\min\left(f(x)\right) = \sum_{i=1}^{12} (D_t - x_t)^2 \quad (19)$$

where D is the demand and x is the released water.

There are certain constraints for reservoir operation as follows.

Continuity equation:

$$S_{t+1} = S_t + I_t - R_t - loss_t \quad (20)$$

where S_{t+1} is the storage at time t + 1, S_t is the storage at time t, I_t is the inflow at time t, R_t is the released water and loss is the volume of evaporation from the dam.

Storage constraint: The storage should be considered between upper and lower bounds based on the following equation:

$$1648.67(MG) \le S_t \le 6194(MG) \quad (21)$$

where 1,648.67 (MG) is the minimum storage and 6,194 (MG) is the maximum storage.

Release constraint:

$$868(MG) \le R_t \le 1379.5(MG) \quad (22)$$

where 868 (MG) is the minimum release and 1,379.5 (MG) is the maximum release.

Table 1 shows different kinds of inflow to the reservoir. If the constraints are not satisfied, the following penalty functions will be added to the objective function:

$$penalty\ 1 = \begin{bmatrix} 0 \leftarrow if(S_t > S_{\min}) \\ C_1(S_{\min} - S_t)^2 \leftarrow if(S_t < S_{\min}) \end{bmatrix} \quad (23)$$

$$penalty\ 2 = \begin{bmatrix} 0 \leftarrow if(S_t) < S_{\max} \\ C_2(S_t - S_{\max})^2 \leftarrow (S_t) > S_{\max} \end{bmatrix} \quad (24)$$

Coefficient C is considered the penalty coefficient.

Different indies are used for evaluating and comparing the performance of different methods:

Volumetric reliability index: This index explains the ratio of released water and demand so that a higher percent

Table 1 | Monthly average inflow to Klang gate dam (1995–2004) ((MG)

Months	High inflow	Medium inflow	Low inflow	Demand
January	1,506.89	760.85	123.12	1,298.64
February	1,901.08	1,024.49	259.34	1,083.09
March	2,831.70	1,646.31	923.34	1,152.45
April	2,919.74	1,959.92	764.88	1,173.11
May	2,974.20	1,768.32	938.31	1,198.73
June	2,825.20	1,355.22	447.95	1,271.73
July	2,717.32	1,618.95	645.61	1,258.14
August	2,948.26	1,644.53	816.78	1,260.41
September	3,368.12	1,859.86	631.15	1,160.45
October	3,545.83	2,316.25	654.35	1,204.14
November	3,838.47	2,342.89	1,021.79	1,213.09
December	2,699.30	1,455.70	340.69	1,290.59

for this index is more suitable, indicating that demands have been supplied well.

$$\alpha_V = \frac{\sum_{i=1}^{N}\sum_{i=1}^{T} R_{i,t}}{\sum_{i=1}^{N}\sum_{i=1}^{T} D_{i,t}} \tag{25}$$

where R is the released water and D is the demand value.

Vulnerability index: This index shows the ratio of the maximum intensity of happened failure in the system during the operation period.

$$\lambda_v = Max_{t=1}^{T}\left(\frac{D_t - R_t}{D_t}\right) \tag{26}$$

where λ is the vulnerability index and T is the number of operation periods.

Resiliency index: The resiliency index shows the model's efficiency to recover a failure. A higher percent for this index is more suitable and the following equation shows the resiliency index:

$$\gamma_i = \frac{f_{si}}{F_i} \tag{27}$$

where γ_i is the residency index, f_{si} is the number of series of happened failure, F_i is the number of periods of happened failure.

Evaluation of the meta-heuristic methods

A hybrid model based on the weighted sum model and the weighted product model has been defined for the comparison of the different evolutionary methods used in solving the optimization problem of the reservoir. If ϕ_{ef} is considered to be the value of one index for various evolutionary methods, it is possible that there will be two equations that normalize the obtained value of each index for each method. The first equation is related to the indices for which their high percentage is suitable for the reservoir system and the second equation is related to the indices for which their low percentage is suitable for the reservoir system:

$$\bar{\phi}_{ef} = \frac{\phi_{ef}}{\max_e \phi_{ef}} \tag{28}$$

$$\bar{\phi}_{ef} = \frac{\phi_{ef}}{\min_e \phi_{ef}} \tag{29}$$

Next, we calculate the weighted aggregated sum and product of the assessment weight and the normalized decision-making parameters for the summation λ_1^e and the multiplication λ_2^e parts as follows:

$$\lambda_1^e = \sum_{f=1}^{nc} \bar{\phi}_e w_e \tag{30}$$

$$\lambda_2^e = \prod_{f=1}^{n_c} (\bar{\phi}_{ef})^{w_f} \tag{31}$$

where nc is the number of indices, w_e and w_f are the criteria's weight.

Evaluation of the algorithms is performed based on the following equation:

$$\lambda = \kappa(\lambda_1^e) + (1-\kappa)\lambda_2^e \tag{32}$$

where κ is the fraction of the weighted sum model and the weighted product model. In this research paper, the interval of [0, 1], in an increment of 0.1, is considered for κ. Each algorithm which has a higher value of κ is then selected as the best algorithm. Then, different algorithms are compared based on the pairwise contests, and finally, the rank of each algorithm is determined, and the best algorithm is screened selectively to solve the optimization problem.

RESULTS AND DISCUSSION

Table 2 shows the sensibility analysis for reservoir operation. The population size for shark algorithm, based on the best value of the objective function, is 30 and the values of M, α are 20 and 0.76, respectively. For particle swarm algorithm, the population size, inertial weight, and acceleration coefficient are 50, 0.78, and 2, respectively. For GA, the population size, crossover probability, mutation probability, and mutation rate are 30, 0.76, 0.87, and 0.001, respectively. The objective function is computed based on the best parameter values in each method. The Lingo software is an optimization and engineering software tailored especially for water resource management. The linear programming method of this software is used for the computation of a global solution and the evaluation of average solutions of different meta-heuristic algorithms based on a global solution. The global solution for this problem based on Lingo

Table 2 | Sensibility analysis for (a) shark algorithm, (b) particle swarm algorithm, and (c) genetic algorithm

(a)

Population size	Objective function	M	Objective function	α	Objective function
10	1.345	10	1.298	0.56	1.289
30	1.247	20	1.247	0.66	1.254
50	1.312	30	1.310	0.76	1.247
70	1.298	40	1.312	0.86	1.314

(b)

Population size	Objective function	$c_1 = c_2$	Objective function	w	Objective function
10	1.871	1.8	1.612	0.58	1.512
30	1.657	1.9	1.545	0.68	1.505
50	1.451	2.0	1.457	0.78	1.457
70	1.520	2.1	1.567	0.88	1.545

(c)

Population size	Objective function	Crossover probability	Objective function	Mutation probability	Objective function
10	1.698	0.66	1.598	0.66	1.612
30	1.544	0.76	1.544	0.76	1.544
50	1.672	0.86	1.587	0.86	1.589
70	1.712	0.96	1.589	0.96	1.611

software is 1.245. Table 3 shows ten random results for different meta-heuristic algorithms. The average solution for shark algorithm is 1.247, which denotes 99% of the global solution. The average solutions for genetic and particle swarm algorithms are 1.544 and 1.457, denoting 0.80 and 0.85 of the global solution, respectively. The variation coefficient for shark algorithm is 10 and 15 times smaller than the corresponding values for the particle swarm algorithm and GA, respectively. Figure 3 shows the pattern of water release for January in the entire operation period. The volume of released water is based on the average value of January in the operation period. Three kinds of inflow are considered for each month and ten groups are considered for reservoir storage. The first objective was to produce a curve that can show the optimum releases for an inflow defined group. The objective of these techniques was to generate a release policy that decreases the water deficit as far as possible, keeping the release domains for every

Table 3 | Results of ten random runs for reservoir operation

Run	Shark algorithm	Particle swarm algorithm	Genetic algorithm
1	1.247	1.457	1.545
2	1.246	1.449	1.545
3	1.247	1.449	1.544
4	1.247	1.457	1.544
5	1.247	1.457	1.544
6	1.247	1.457	1.544
7	1.247	1.457	1.544
8	1.247	1.457	1.544
9	1.247	1.457	1.544
10	1.247	1.457	1.544
Average	1.247	1.457	1.544
Variation coefficient	0.0002	0.002	0.003
Global solution		1.245	

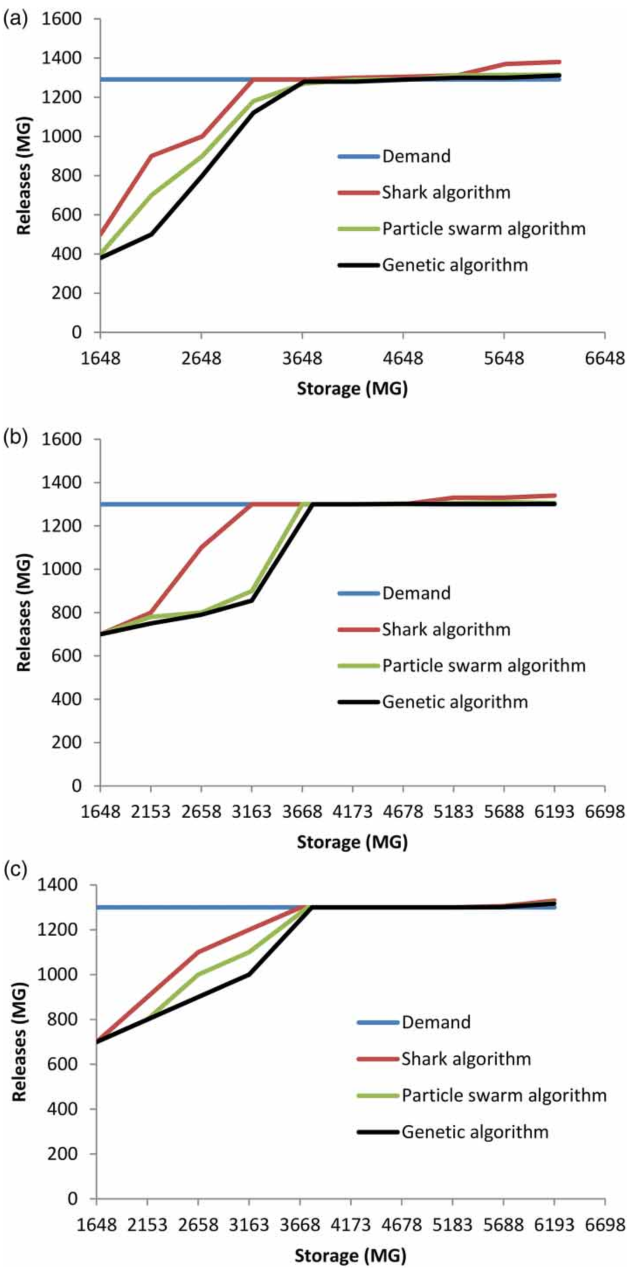

Figure 3 | Released water for (a) high inflow (January), (b) medium inflow (January), and (c) low inflow (January).

month and safe storage volume. Thus, here, the storage condition was also provided with release options to obtain a clear picture of the reservoir system. Figure 3 shows calibrated curves for the month of January and this month was selected because the demand for this month has the most value based on Table 1. Shark algorithm meets demands for high inflow when the storage is 2,153.71 MG while the particle swarm and GAs meet demands when storage values are 2,658.94 and 2,678.87 MG for January, respectively. Shark algorithm method meets the demand for medium inflow when the storage is 3,163.78 MG (Figure 3). Yet particle swarm and GAs meet demands when reservoir storages are 3,668.2 MG and 3,724.12 MG for January, respectively. Shark algorithm meets demands for low inflow when the storage is 3,678.94 MG while genetic and particle swarm algorithms meet demand when the storage values are 3,787.33 MG and 3,711.21 MG for January, respectively. Thus, it can be seen that the demands are supplied with the shark algorithm earlier than those with the GA and particle swarm algorithm. Figure 3 shows that the shark algorithm release options are reached and follow the demand line more quickly and closely than other algorithms and capacity above demand will be reserved for next year.

Another point is related to the convergence behaviors of different algorithms. Figure 4 shows that shark algorithm can converge earlier than the GA and particle swarm algorithm. As such, the quality and velocity of convergence for the shark algorithm are better than those for particle swarm and GAs. To find the closest curve to a demand for a specific inflow category, the root mean square error (RMSE) is used and the RMSE for every month is given in Table 4 for medium inflow (medium inflow is selected because no violation of storage consistency has occurred). Shark optimization algorithm has the lowest error in this case study. Table 4 shows that total RMSE for the shark algorithm is 42 MG and the corresponding indexes for particle swarm and GAs are 53 MG and 63 MG, respectively. Thus, the shark algorithm decreases RMSE compared to

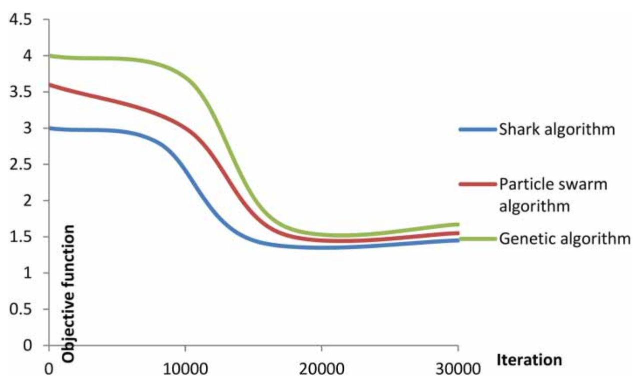

Figure 4 | Convergence behavior for different algorithms.

Table 4 | Computed RMSE for different methods (1994–2003, medium inflow)

Month	Shark algorithm (MG)	Particle swarm algorithm (MG)	Genetic algorithm (MG)
January	8.78	9.11	10.22
February	6.54	7.49	8.14
March	5.56	6.12	7.23
April	5.45	6.78	7.14
May	7.87	8.11	8.44
June	6.54	7.12	8.11
July	3.44	3.89	4.01
August	1.28	1.78	1.86
September	1.18	1.48	1.78
October	0.98	1.01	1.11
November	0.87	0.98	1.04
December	0.86	0.89	0.99
Total	49.35	54.68	60.07

the particle swarm algorithm and GA by about 20 and 33%, respectively.

Table 5 shows the value of different indices for different meta-heuristic algorithms. The shark algorithm has a higher value of reliability index (namely, 97%) compared with the GA. In other words, the shark algorithm can supply demands with higher reliability index after the particle swarm algorithm. The resiliency index for the shark algorithm is 44% and the corresponding indexes for GA and particle swarm algorithm are 29 and 35%, respectively. As such, the system based on the shark algorithm can exit from a failure faster than other methods. The vulnerability index for the shark algorithm is 14% while the corresponding indexes for the GA and particle swarm algorithm are 24 and 21%, respectively. These indexes have been computed for a period of 10 years (1994–2003). Figure 5 indicates

Table 5 | Evaluation and comparison of different algorithms

Method	Reliability index	Vulnerability index	Resiliency index
Shark algorithm	97%	14%	44%
Particle swarm algorithm	99%	21%	39%
Genetic algorithm	85%	24%	25%

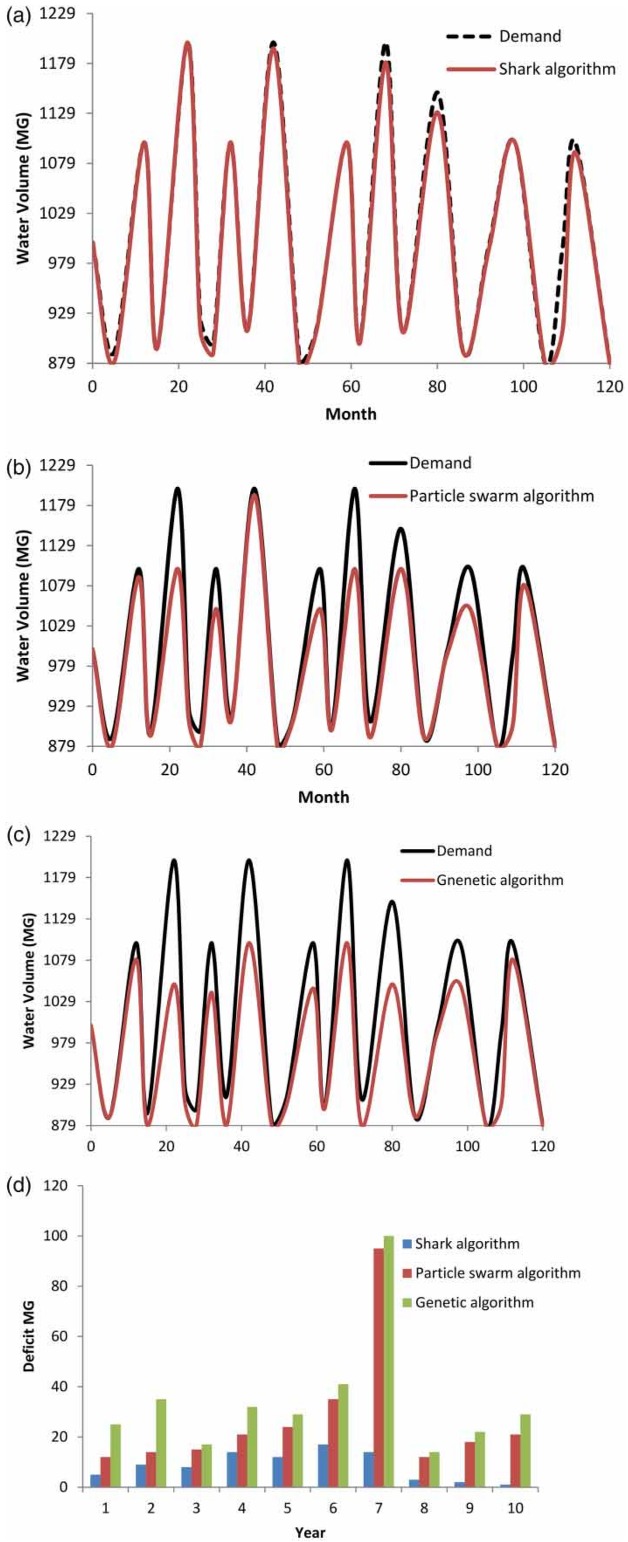

Figure 5 | Release and demand analysis using historical inflow: (a) shark algorithm, (b) particle swarm algorithm, (c) genetic algorithm, (d) annual deficit (1994–2003).

that the shark algorithm corresponds to demand values better than the cases of GA and particle swarm algorithm. Thus, the shark algorithm has high potential in reservoir operation. Also, Figure 5(d) shows the average annual deficit of the reservoir by the applied algorithms. According to Figure 5(d), the severity of deficits computed by the shark algorithm is lower than other algorithms. The maximum deficits in the reservoir according to the shark algorithm, particle swarm algorithm, and GA were 17.12, 96.24, and 100 GB. The shark algorithm provided the lowest deficit during this study, indicating the greater performance of this algorithm for minimizing deficit.

In general, all of these optimization results indicate that the best algorithm should not be selected based only on one index or on one kind of objective function. Also, the system used in this study is a single reservoir, which could limit the performance and versatility of the method when a different study location or different flow conditions are considered. Therefore, it can be postulated that the selection of the best method is likely to be more complex for multi-reservoir systems. As a result, we conclude that an effective tool should be found by means of comparing different evolutionary methods such that a multi-criteria decision model is determined and applied for making better decisions in the optimization of reservoir operations.

Next, we checked the normalized decision-making parameters (λ^1 and λ^2 for different methods) in Table 6 where the weight of all indices is considered based on the same value.

Table 7(a) shows the values of κ for different algorithms tested in the present study. Eleven initial values are considered for κ and, then, the values for different methods are computed and, finally, the rank of each method is determined based on the overall κ value. Evidently, the shark algorithm has the highest rank based on all of the κ values.

To further explore the accuracy of the shark algorithm, Table 7(b) presents a pair contest based on Copeland's procedure for the 11 values of κ. In accordance with Copeland's procedure, the present results demonstrate the superiority of the shark algorithm over the genetic and particle swarm algorithms, which can be confirmed by the final ranks assigned in Table 7(c).

The objective value of the cost function and the average CPU run times for each algorithm (based on a PC with i5 CPU GHz/4B, ram/500 GB HDD) is presented in Table 7(d). CPU run time of the shark algorithm (50.6 s) was the least of all the algorithms.

CONCLUSIONS

In this paper, a newly developed meta-heuristic evolutionary algorithm (known as the shark algorithm) is proposed for the optimization of the Klang gate dam's reservoir operation. The analysis presented revealed that the operation policies and results which have been obtained using the shark algorithm approach (compared to the GA and the improved particle swarm optimization algorithm (IPSOA)

Table 6 | (a) Normalized decision matrix and (b) value of λ^1 and λ^2 for different methods

(a)

Algorithms	Reliability index	Vulnerability index	Resiliency index	RMSE
Shark algorithm	0.97	1	1	1
Particle swarm algorithm	1	0.66	0.88	0.90
Genetic algorithm	0.87	0.58	0.56	0.81

(b)

Part of decision-making index	Shark algorithm	Genetic algorithm	Particle swarm algorithm
λ^1	0.9925	0.705	0.8612
λ^2	0.9999	0.691	0.8541

Table 7 | (a) Performance of each evolutionary method for different values of κ, (b) pairwise contests based on Copeland's procedure, (c) final ranks based on Copeland's procedure, and (d) computational time

(a)

Fraction	λ_{shark}	$\lambda_{genetic}$	$\lambda_{particle}$
$\kappa = 0$	0.9925	0.8541	0.7076
$\kappa = 0.1$	0.9932	0.8548	0.7118
$\kappa = 0.2$	0.9939	0.8555	0.7160
$\kappa = 0.3$	0.9947	0.8562	0.72032
$\kappa = 0.4$	0.9954	0.8569	0.7245
$\kappa = 0.5$	0.9962	0.8576	0.7288
$\kappa = 0.6$	0.9969	0.8583	0.7330
$\kappa = 0.7$	0.9976	0.8590	0.7372
$\kappa = 0.8$	0.9984	0.8597	0.7415
$\kappa = 0.9$	0.9991	0.8604	0.7457
$\kappa = 1.0$	0.9999	0.8612	0.75
Rank	1	2	3

(b)

Contender A	Contender B	Number of victories for A	Number of victories for B	Winner
Shark algorithm	Particle swarm algorithm	11	0	Shark algorithm
Particle swarm	Genetic algorithm	11	0	Particle swarm algorithm

(c)

Method	Rank
Shark algorithm	1
Particle swarm algorithm	2
Genetic algorithm	3

(d)

Method	CPU (time: s)
Shark algorithm	50.6
Particle swarm algorithm	60.1
Genetic algorithm	80.5

approaches) could lead to better water release plans that are likely to help minimize the mismatch existent with the water users' demands. Evaluation of the shark algorithm performance has been carried out for a historical period and the results show that this approach carries a greater robustness compared to the other methods for all inflow scenarios. Furthermore, the results show that the shark algorithm approach provided a closer solution to the global solution and was ranked in first position based on the multi-criteria decision model, followed by lower ranks assigned to the IPSOA and GA approaches. In addition, the shark algorithm approach outperformed the IPSOA and GA approaches in attaining a greater resilience index and a lower vulnerability index for all inflow scenarios relevant to the reservoir. In a situation where the low inflow scenario could be unsatisfactory in terms of meeting the water demands pattern in an adequate manner, the shark algorithm approach for water management was found to perform better than the other two optimization algorithms. This shows that the shark algorithm approach is considerably robust, especially in terms of challenging water management situations (e.g., low inflow scenarios).

REFERENCES

Bozorg Haddad, O., Afshar, A. & Mariño, M. A. 2008a Honey-bee mating optimization (HBMO) algorithm in deriving optimal operation rules for reservoirs. *J. Hydroinform.* **10** (3), 257–264.

Bozorg Haddad, O., Adams, B. J. & Mariño, M. A. 2008b Optimum rehabilitation strategy of water distribution systems using the HBMO algorithm. *J. Water Supply: Res. Technol.-AQUA* **57** (5), 327–350.

Bozorg Haddad, O., Moradi-Jalal, M. & Mariño, M. A. 2011 Design-operation optimisation of run-of-river power plants. *Proc. Inst. Civ. Eng. Water Manage.* **164** (9), 463–475.

Bozorg Haddad, O., Moravej, M. & Lociga, H. A. 2015 Application of the water cycle algorithm to the optimal operation of reservoir systems. *J. Irrig. Drain. Eng.* **141** (5), 04014064. 10.1061/(ASCE)IR.1943-4774.0000832.

Ehteram, M., Karami, H., Mousavi, S.-F., EL-Shafie, A. & Amini, Z. 2017a Optimizing dam an reservoirs operation based model utilizing shark algorithm approach. *Knowledge-based System* **122**, 26–38.

Ehteram, M., Fallah, M., Karami, H., Mousavi, S.-F., Emami, M., EL-Shafie, A. & Faarzin, S. 2017b Optimization of chain reservoirs' operation with a new approach in artificial intelligence. *Water Resour. Manage.* **31** (7), 2085–2014.

Fallah-Mehdipour, E., Haddad, O. B. & Mariño, M. A. 2011 MOPSO algorithm and its application in multi-purpose multi-reservoir operation. *J. Hydroinform.* **13** (4), 794–811.

Fallah-Mehdipour, E., Bozorg Haddad, O. & Mariño, M. A. 2013a Developing reservoir operational decision rule by genetic programming. *J. Hydroinform.* **15** (1), 103–119.

Fallah-Mehdipour, E., Bozorg Haddad, O. & Mariño, M. A. 2013b Extraction of multicrop planning rules in a reservoir system: application of evolutionary algorithms. *J. Irrig. Drain. Eng.* **139** (6), 490–498. 10.1061/(ASCE) IR.1943-4774.0000572.

Ghimire, B. & Reddy, M. 2013 Optimal reservoir for hydropower production using particle swarm optimization and sustainability analysis of hydropower. *ISH J. Hydraul. Eng.* **19** (3), 196–210.

Jalali, M. R., Afshar, A. & Mariño, M. A. 2007 Multi-reservoir operation and adaptive pheromone re-initiated ant colony optimization algorithm. *Int. J. Civ. Eng.* **5** (4), 284–301.

Jothiprakash, V., Shanthi, G. & Arunkumar, R. 2011 Development of operational policy for a multi-reservoir system in India using genetic algorithm. *Water Resour. Manage.* **25** (10), 2405–2423.

Kumar, D. N. & Reddy, M. J. 2006 Ant colony optimization for multi-purpose reservoir operation. *Water Resour. Manage.* **20** (6), 879–898.

Kumar, P., Dhanya, T. & Chahar, B. 2016 A PSO approach for optimum design of dynamic inversion controller in water distribution systems. *J. Water Supply: Res. Technol.-AQUA* **65** (7), 580–591.

Ngoc, T. A., Hiramatsu, K. & Harada, M. 2013 Optimization the rule curves of multi-use reservoir operation using a genetic algorithm with a penalty strategy. *Paddy Water Environ.* **12** (1), 125–137.

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

Ostadrahimi, L., Mariño, M. A. & Afshar, A. 2012 Multi-reservoir operation rules: multi-swarm (PSO)-based optimization approach. *Water Resour. Manage.* **26** (2), 407–427.

Shirzad, A. & Tabesh, M. 2016 New indices for reliability assessment of water distribution networks. *J. Water Supply: Res. Technol.-AQUA* **65** (5), 384–395.

Tospornsampan, J., Kita, I., Ishii, M. & Kitamura, Y. 2005 Optimization of a multiple reservoir system using a simulated annealing – a case study in the Mae Klong system, Thailand. *Paddy Water Environ.* **3** (3), 137–147.

First received 12 June 2017; accepted in revised form 27 September 2017. Available online 15 November 2017