Water Resources Management (2018) 32:4681–4706
https://doi.org/10.1007/s11269-018-2078-2

Reservoir Operation by a New Evolutionary Algorithm: Kidney Algorithm

Mohammad Ehteram[1] · **Hojat Karami**[1] · **Sayed Farhad Mousavi**[1] · **Saaed Farzin**[1] · **Alcigeimes B. Celeste**[2] · **Ahmad-El Shafie**[3]

Received: 17 January 2018 / Accepted: 7 August 2018 /
Published online: 13 August 2018

Abstract

This article shows an application of a new algorithm, called kidney algorithm, for reservoir operation which employs three different operators, namely filtration, secretion, and excretion that lead to faster convergence and more accurate solutions. The kidney algorithm (KA) was used for generating the optimal operation of a reservoir namely; Aydoghmoush dam in eastern Azerbaijan province in Iran whose purpose was to decrease irrigation deficit downstream of the dam. Results from the algorithm were compared with those by other evolutionary algorithms, including bat (BA), genetic (GA), particle swarm (PSO), shark (SA), and weed algorithms (WA). The results showed that the kidney algorithm provided the best performance against the other evolutionary algorithms. For example, the computational time for the KA was 3 s, 2 s, 4 s, 6 s and 3 s less than BA, SA, GA PSA and WA, respectively. Also, the objective function for the optimization problem was the minimization of the irrigation deficits and its value for the KA was 55%, 28%, 52%, 44 and 54% less than GA, SA, WA, BA and PSA, respectively. Also, the different performance indexes showed the superiority of the KA compared to the other algorithms. For example, the root mean square error for the KA was 74%, 61%, 68%, 33 and 54% less than GA, SA, WA, BA and PSA, respectively. Different multi criteria decision models were used to select the best models. The results showed that the KA achieved the first rank for the optimization problem and thus, it shows a high potential to be applied for different problems in the field of water resources management.

Keywords Water resource management · Reservoir operation · Kidney algorithm · Optimization Algorithms

✉ Mohammad Ehteram
mohammdehteram@semnan.ac.ir

[1] Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan, Iran

[2] Department of Civil Engineering, Federal University of Sergipe, São Cristóvão, Brazil

[3] Civil Engineering Department, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia

1 Introduction

Water resources are getting scarce these days and decision makers are challenged to utilize them in the best possible way. The water stored in reservoirs behind dams is an important resource which can be utilized during critical or drought periods having in mind that storage volumes can be significantly decreased due to high evaporation losses, particularly in arid regions (Ehteram et al. 2018a; Johnson et al. 1993).

1.1 Background

Nowadays, reservoirs are often operated optimally with the aid of mathematical models such that water is released to meet demands to the greatest extent possible (Zhang et al. 2014, Ehteram et al. 2018b). Different mathematical models, such as meta-heuristic or evolutionary algorithms, have been developed for reservoir operation. These algorithms are able to determine optimal strategies for water release (Schardong and Simonovic, 2015; Bozorg-Hadad et al. 2016).

Akbari-Alashti et al. (2014) used genetic programming for optimizing reservoir operation and showed that it matched the downstream demand better than did the genetic algorithm. Based on genetic algorithms, Bolouri-Yazdeli et al. (2014) found that reservoir operation with higher order nonlinear rule curves supplied demands better than did lower order nonlinear curves. Results from an application of bat algorithm (BA) to reservoir operation showed that water demand supplies were less vulnerable (Bozorg-Hadad et al. 2014) but the algorithm may get trapped in local optima. Ahmadi et al. (2014) extracted adoption and non-adoption rule curves for the operation of a reservoir with a multi-objective genetic algorithm and showed that the adoption rule curve supplied demands well but more computation time was needed. Applying the weed optimization algorithm (WOA) to a multi reservoir system, Asgari et al. (2015) showed that it supplied demands with more resiliency and reliability and required less function evaluations than did genetic and particle swarm algorithms.

Biography-based optimization algorithm (BBOA) was used by Hadad et al. (2015) for reservoir operation with the aim to increase power generation in the downstream power plant. Results showed that, compared to other evolutionary algorithms, BBOA yielded solutions closer to the global solution. Using a genetic algorithm, Tayebian et al. (2016) employed different polices for reservoir operation and showed that the binary standard operating policy better met demands.

Moeini et al. (2017) used a constrained gravitational search algorithm for reservoir operation with the aim to decrease hydropower deficit and found it better than the honey bee mating optimization algorithm (HBMOA) and discrete dynamic programming. Using improved artificial bee colony optimization algorithm (IABCO), a multi-reservoir system with power plants was operated for increasing power generation in the downstream power plants. It was found that IABCO supplied hydropower demand with more reliability than did genetic algorithm (GA) and particle swarm optimization (PSOA) (Choong et al. 2017).

Ming et al. (2017) compared the search space reduction method (SSRM) with the cuckoo algorithm for reservoir operation in China and found that SSRM reached the global solution and was capable of solving the reservoir system problem with many constraints. Applying the shark algorithm for reservoir operation in Iran, Ehteram et al. (2017) showed that it decreased water deficit during critical periods more reliably than did other algorithms. Using the krill algorithm for reservoir operation with the aim to increase power generation in the downstream power plant, Ehteram et al. (2017) showed that the algorithm provided average solutions close to the global solution.

1.2 Problem Statement, Motivation and Novelty

The literature review in the previous section shows that the optimal operation of water resources based on new evolutionary algorithms has gained a lot of importance in recent years. The present study addresses the new evolutionary kidney algorithm for the optimal operation of a reservoir with the aim of decreasing irrigation deficits and the results will show that the KA algorithm leads to less computational time and to better objective function values when compared to other evolutionary algorithms. The study will demonstrate the better performance of the KA based on different multicriteria decision models and different indexes such as the volumetric reliability index, root mean square error and others. Thus, the main contribution of this paper is to introduce a mathematical model based on a new evolutionary algorithm (KA) for reservoir operation with irrigation aims. Also, the present article attempts to modify previous literature reviews for the optimal operation of water resources because past articles targeted at selecting the best models or methods based on some limited indexes such objective function, number of function evaluations or computational time while the present study considers the selection of the best method based on different multicretira decision models in order to achieve more comprehensive outcomes.

Reservoir operation entails a nonlinear non-convex objective function and requires a robust algorithm that can lead to the global solution and can extract rule curves for supplying demands, such as irrigation, hydropower and others. Optimal reservoir operation takes on an added significance in periods of drought and water scarcity. This requirement can be met with the use evolutionary algorithms. The kidney algorithm (KA) has been introduced as a successful optimization algorithm suitable for different engineering applications versus the other algorithms (Jaddi et al. 2017). The method was used by Jaddi et al. (2017) for benchmark mathematical functions but, to the authors' knowledge, the KA has not been used to generate optimal operation rules for dam and reservoir water systems application. The benchmark mathematical functions were evaluated by the kidney algorithm and the method had the good results.

The other reasons for the selection of this algorithm were that it overcame problems of other algorithms, such as trapping in the local solutions, slow convergence, and lack of balance between exploration and exploitation. The kidney algorithm has the filtration operator which causes faster convergence. In fact, this operator identifies the best solution and helps expedite convergence. Also, a balance between exploration and exploitation is generated by filtration so that the algorithm searches the regions which have the solutions with better quality. The solutions with bad quality which may trap the algorithm in local solutions are removed by the secretion operator.

Formulating this algorithm for reservoir operation for the first time in water resource management is the innovation of this paper as well as comparing it to bat, shark, weed, particle swarm and genetic algorithms. Also, some multi-criteria decision indexes, such as complex proportional assessment (COPRAS), technique for order preference by similarity to ideal solution (TOPSIS), modified technique for order preference by similarity to ideal solution (MTOPSIS), and weighted aggregates sum product assessment (WASPAS) were used for evaluation of different algorithms. In addition, the Borda social selection index based on all multi-criteria decision models was used to determine the best method for reservoir operation.

1.3 Objective

The main objective of this study is to investigate the potential of generating reservoir operation rules based on the kidney algorithm in order to minimize the irrigation water deficits. In addition, to examine the kidney algorithm performance over the most recent meta-heuristic algorithms, a comprehensive comparative analysis is carried out.

2 Material Methods

2.1 Kidney Algorithm

Kidneys are situated in the urinary system in the biological structure of a body. They remove waste and excess water from blood and have components, called nephrons, which have capillaries such as glomerular tubes that filter the fluids available in the kidneys. Kidneys have small tubes, known as tubules, which undertake two main functions of reabsorption and secretion. Reabsorption is a process which returns solutes from tubules to the blood, whereas secretion transforms solutes from the blood to the tubules which are then discharged by the urinary system. Different processes occurring in kidneys are defined below (Jaddi et al. 2017):

1- Filtration: transformation of solutes from the blood to tubules.
2- Reabsorption: process by which non-waste solutes move from tubules to the blood.
3- Secretion: process by which waste materials move from the blood to tubules.
4- Excretion: process by which all the waste from the above three processes are discharged from the body.

There are similarities between the kidney algorithm and other meta-heuristic algorithms. For example, an initial population is defined for this algorithm. The solutes are considered as initial population or candidate solutions. Then, new solutions are generated, based on the movement of current solutions to the best solutions. At the next level, the filtration operator is applied to solutions and the solutions are divided into two groups. The useful solutes are transferred to the filtered blood (FB) and the harmful solutes are considered as waste (W). The algorithm gives another opportunity to the solutes which are considered as waste so that these solutes can be returned to the FB, if they can satisfy the rate of filtration. If these solutes cannot use this opportunity, they are excreted and a random solute fills the place of the excreted solution. Also, if the solute is transferred to the FB, the solute quality should be compared to the available worst solute in the FB. If the quality of transferred solute to the FB is worse than the worst solute, the transferred solute is excreted or else if the quality of the worst solute is less than the transferred solute to the FB, the worst solute is excreted. All levels of the algorithm are detailed in the following sections.

2.1.1 Movement Solutes for Generation of New Positions

The movement of solutes for finding the best position with better quality results in new solutes at each iteration. Indeed, trying solutes for increasing their quality gives rise to new solutes. The generation of new solutes can be expressed as:

$$S_{i+1} = S_i + rand(S_{best} - S_i) \tag{1}$$

where S_i: the *i*-th solute, S_{best}: the best solute, rand: the random number, and S_{i+1}: the new solute. Equation 1 increases the population diversity which causes faster convergence.

2.1.2 Filtration

The filtration rate is an operator for identifying the worst solute and the best solute and can be computed as:

$$fr = \alpha \times \frac{\sum_{i=1}^{p} f(x_i)}{p} \tag{2}$$

where, α: a constant value between 0 and 1, p: the population size, $f(x_i)$: the objective function, *fr*: the filtration rate and x_i: decision variables.

The average of all objective function values is used for the computation of the filtration rate. If α equals zero, it means that there is no filtration and if αequals one, the rate of filtration equals the average of the objective function values. This process filtrates the solutes with the best quality from the solutes with the bad quality and thus improves the exploration process.

2.1.3 Reabsorption

When a solute is considered as waste, the reabsorption again gives an opportunity to the solute to join to the FB. This event happens if the new solute, based on Eq. 1, can satisfy the rate of filtration. This process improves the exploitation process.

2.1.4 Secretion

When a solution is transferred to the FB, its quality should be compared with the worst solute. If the quality of this solute is worse than the worst available quality in the FB, it is secreted, or else the worst solute will be secreted.

2.1.5 Excretion

When a solution cannot be given an opportunity again for joining the FB, it will be excreted and a random solute fills the place of the removed solute. Figure 1 shows the details of the kidney algorithm.

To summarize, the following levels are considered for the kidney algorithm:

1- First, an initial population is considered for the algorithm and the solutes are considered as the initial solution.
2- The objective function is computed for each solute.
3- The rate of filtration is computed for each solute and the better solute is transformed to the filtered blood and other solutes are considered as waste.
4- Another chance is given to the waste and, if it uses this chance, it would join the FB. Otherwise, it is excreted and the random solute fills the excreted solute.

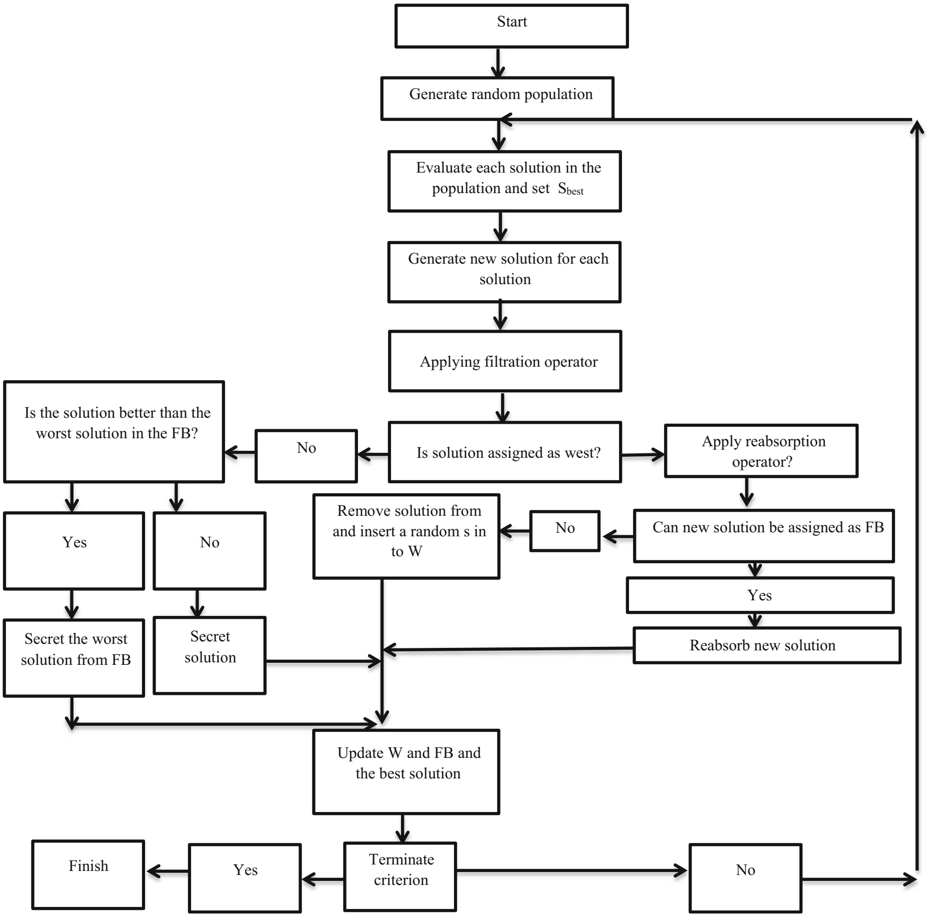

Start
Generate random population
Evaluate each solution in the population and set S_{best}
Generate new solution for each solution
Applying filtration operator
Is solution assigned as west?
No
Is the solution better than the worst solution in the FB?
Apply reabsorption operator?
Yes
No
Remove solution from and insert a random s in to W
No
Can new solution be assigned as FB
Yes
Secret the worst solution from FB
Secret solution
Reabsorb new solution
Update W and FB and the best solution
Finish
Yes
Terminate criterion
No

Fig. 1 Kidney algorithm

5- The transformed solute in the FB is compared with the worst solute in the FB and if it has bad quality compared to the worst solute, the solute will be secreted or else the worst solute will be secreted.
6- The solutes in the FB are sorted for the selection of the best solute.
7- The FB and W join each other to generate the new population.

2.2 Other Algorithms

2.2.1 Bat Algorithm

The bat algorithm acts based on the echolocation ability that allows the bats to identify the obstacle from food based on pulses generated from bats and their return from the surroundings. In fact, they generate loud sounds and then receive the returned sounds from the surroundings.

The following assumptions are considered for the bat algorithm (Bozorg-Hadad et al. 2014, Karami et al. 2018):

1- All bats apply the echolocation ability for the identification of prey from food.
2- The bats fly randomly with velocity v_l, loudness A_0, wavelength λ, and frequency $f_{\min}$ at potion y_l.
3- The loudness changes from A_0 to $A_{\min}$.

The frequency rate is between $f_{\min}$ and $f_{\max}$ and the wavelength changes between $\lambda_{\min}$ and $\lambda_{\max}$. Also, the pulsation rate (r_l) is between 0 and 1. The position, frequency, and velocity are updated as:

$$f_l = f_{\min} + (f_{\max} - f_{\min}) \times \beta \tag{3}$$

$$v_l(t) = \left[y_l^{t-1} - Y_*\right] \times f_l, t = 1, \dots, T \tag{4}$$

$$y_l(t) = y_l(t-1) + v_l(t), t = 1, .., T \tag{5}$$

where, $y_l(t-1)$: the position at time t-1, β: the random value between 0, 1, Y_*: the global best position, and T: the total period of assessment.

Also, a random walk is used for local search as:

$$y(t) = y(t-1) + \varepsilon A(t) \tag{6}$$

where ε: the random number between −1 and 1, and A (t): the average loudness of bats' sound.

The loudness and pulsation rate should be updated for each iteration. When bats find preys, the loudness decreases and the pulsation rate increases. Also, the pulsation rate is expressed as:

$$r_l^{t+1} = r_l^0[1 - \exp(-\gamma t)]A_l^{t+1} = \alpha A_l^t, \alpha \,\text{and}\, \gamma \,\text{constants} \tag{7}$$

Figure 2 shows different levels of the bat algorithm.

2.2.2 Weed Algorithm

Weeds are harmful for the development of agriculture and they adapt easily with their environment. Characteristics such as completion, growth, and reproduction, are seen among weeds. The weed algorithm has the following features (Asgari et al. 2015):

1- There are limited numbers of seeds for the extension of weeds.
2- Seeds can convert to weeds and weeds can produce seeds, based on their quality again.
3- The weed production continues until their number reaches the maximum. Also, there is a competition among weeds and the weeds with bad quality will be eliminated from the weed colony.

The following assumptions can be considered for the weed algorithm:

1- An initial population is considered for the algorithm (P_{initial}) and it is distributed randomly in the solution space. In fact, each weed can be considered as a solution and its location can be considered as a decision variable.

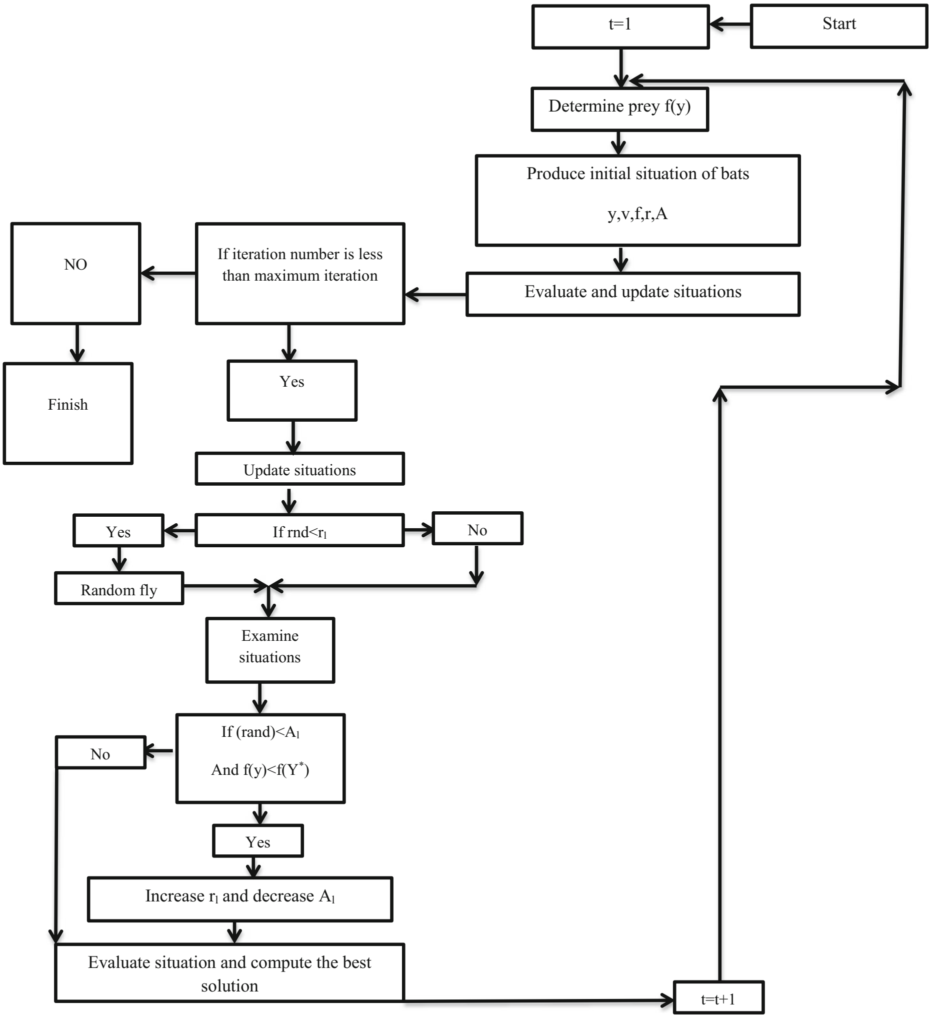

Start
t=1
Determine prey f(y)
Produce initial situation of bats
y,v,f,r,A
Evaluate and update situations
If iteration number is less than maximum iteration
NO
Finish
Yes
Update situations
If rnd<r_l
Yes
No
Random fly
Examine situations
If (rand)<A_l
And f(y)<f(Y*)
No
Yes
Increase r_l and decrease A_l
Evaluate situation and compute the best solution
t=t+1

Fig. 2 Bat algorithm

2- Plants or weeds can produce seeds based on their quality. The members with bad quality in the evolutionary algorithm are eliminated but they can have important information which is useful during the algorithm operation. Thus, reproduction in the weed algorithm gives another chance to the member with bad quality to survive from the bad condition and grow in the better environment.

3- The weeds are distributed randomly in the search space, based on the normal distribution. The standard deviation of the weed distribution is based on the following equation and it starts from the maximum value to reach a minimum value:

$$\sigma_{iter} = \frac{(iter_{\max}-iter)^n}{(iter_{\max})^n}\left(\sigma_{initial}-\sigma_{final}\right) + \sigma_{final} \tag{8}$$

where σ_{iter}: the standard deviation for current iteration, $\sigma_{initial}$: the initial standard deviation, and σ_{final}: the final standard deviation.

4- When the number of weeds reaches $P_{\max}$, the seeds with bad quality should be removed. Figure 3 shows the weed algorithm.

2.2.3 Shark Algorithm

The olfactory system in sharks permits them to identify preys in the surroundings. In fact, smell receptors help them receive odor from the surroundings. The following assumptions are considered in the shark algorithm (Ehteram et al. 2017):

1- It is assumed that the fishes are shark prey and they are injured in the sea. The fish movement is slow because it is injured and, as a result, it is assumed that the location of prey has been fixed.
2- The fish body produces blood, because it is injured and, as a result, the smell receptor in the shark receives the blood odor from the injured fish.
3- It is assumed that there is one blood source around sharks which produces the blood.

First, the initial population $\left[X_1^1, X_2^1, .., X_{NP}^1\right]$ (*NP*: population size) is considered for sharks. In fact, each *X* is considered as shark position and each position has some decision variables which can be expressed as

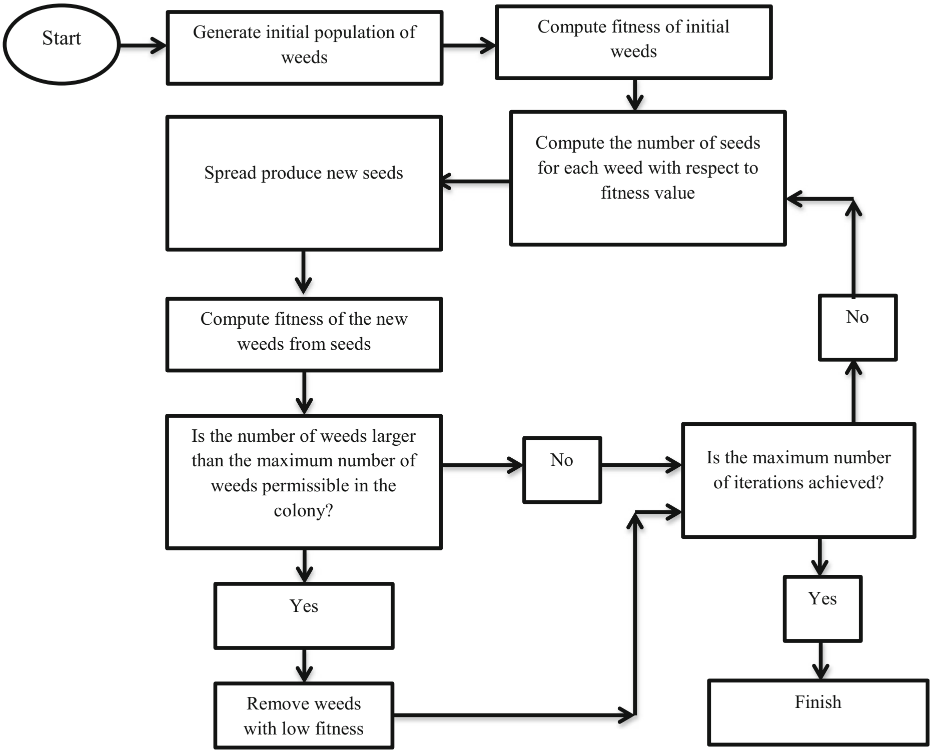

Start
Generate initial population of weeds
Compute fitness of initial weeds
Compute the number of seeds for each weed with respect to fitness value
Spread produce new seeds
Compute fitness of the new weeds from seeds
Is the number of weeds larger than the maximum number of weeds permissible in the colony?
No
Is the maximum number of iterations achieved?
No
Yes
Yes
Remove weeds with low fitness
Finish

Fig. 3 Weed algorithm

$$X_i^l = \left[x_{i,1}^l, x_{i,l}^2, \ldots, x_{i,ND}^l\right] \tag{9}$$

where, $x_{i,j}{}^{1}$: the j-th decision variable.

Each shark has the spatial velocity in the movement with $\left[V_1^1, V_2^1, .., V_{NP}^1\right]$ being the velocity vector. Each vector includes $\left[v_{i,1}^1, v_{i,2}^1, ..\mathrm{v}_{i,ND}^1\right]$, ND being the number of decision variables. When the odor concentration increases, the velocity for shark increases and thus the gradient of the objective function is equal to velocity as

$$V_i^k = \eta_k.R_1.\nabla(OF)\big|_{X_i^k}, i = 1, .., NP, \mathrm{k} = 1, .., \mathrm{k}_{\max} \tag{10}$$

where OF: the objective function, $\mathrm{k}_{\max}$: the number of sharks moving forward, η_k: a random value between 0 and 1, and R_1: a random number between 0 and 1.

The shark movement has inertia and its velocity is limited so that the velocity equation is rewritten as:

$$v_{i,j}^k = \eta_k.R_1.\frac{\partial F}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k R_2 v_{i,j}^{k-1} \tag{11}$$

where α_k: the momentum rate, and R_2: a random value between 0 and 1.

The maximum and minimum velocities for the shark are, respectively, 80 km/h and 20 km/h. Thus, a velocity limiter β_k is considered for velocity such that:

$$\left|v_{i,j}^k\right| = \min\left[\left|\eta_k.R_1.\frac{\sigma(OF)}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k.R_2.v_{i,j}^{k-1}\right|, \left|\beta_k.v_{i,j}^{k-1}\right|\right] \tag{12}$$

Then, the shark position is updated as:

$$Y_i^{k+1} = X_i^k + V_i^k \Delta t_k \tag{13}$$

in which Δt_k: time interval for each stage of movement.

Sharks have an important movement which is called rotational movement. This causes the shark algorithm to exit from local optima (Fig. 4):

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3.Y_i^{k+1} \tag{14}$$

where R_3: random number, m: number of points in the local search around Y_i^{k+1}, and $Z_i^{k+1,m}$: obtained position based on the rotational movement of sharks.

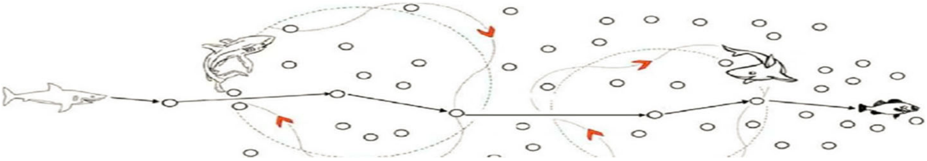


Fig. 4 Rotational movement

Finally, for maximization, the final solution is computed as:

$$X_i^{k+1} = \arg\left[\max\left(OF\left(Y_i^{k+1}\right), OF\left(Z_i^{k+1,1}\right), \ldots, OF\left(Z_i^{k+1,M}\right)\right)\right] \tag{15}$$

Figure 5 shows the shark algorithm.

2.2.4 Particle Swarm Algorithm

The particle swarm optimization algorithm acts based on velocity and position, so that a dimensional vector $X_i = (x_{i,\ 1}, x_{i2}, .., x_{iD})^T$ shows the particle position and another vector $V_i = (v_{i,\ 1}, \mathrm{v}_{i2}, .., \mathrm{v}_{iD})^T$ shows the particle velocity (Delice et al. 2017). Also, the best previously observed position of particles is shown with $P_i = (p_{i,\ 1}, \mathrm{p}_{i2}, .., \mathrm{p}_{iD})^T$. The g index shows the

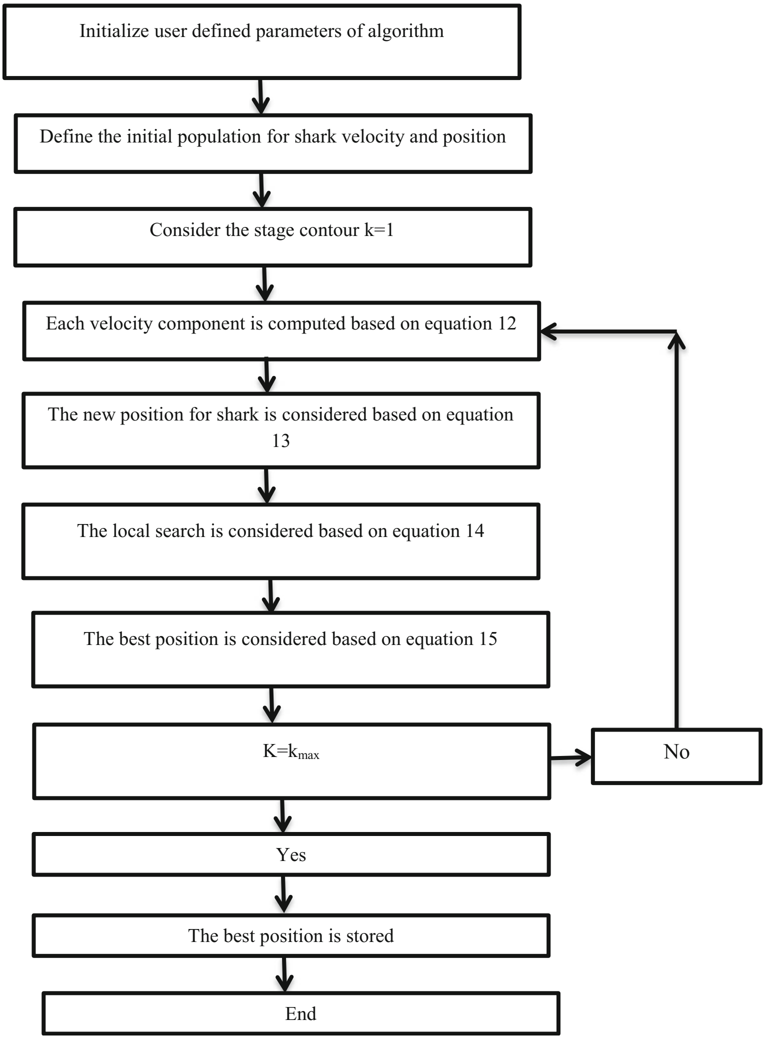

Initialize user defined parameters of algorithm
Define the initial population for shark velocity and position
Consider the stage contour k=1
Each velocity component is computed based on equation 12
The new position for shark is considered based on equation 13
The local search is considered based on equation 14
The best position is considered based on equation 15
K=k$_{max}$
No
Yes
The best position is stored
End

Fig. 5 Shark algorithm

global solution as:

$$v_{id}^{n+1} = \chi\left[wv_{id}^{n} + \frac{c_1 r_1^n\left(p_{id}^n - x_{id}^n\right)}{\Delta t} + c_2 r_2^n \frac{\left(p_{gd}^n - x_{gd}^n\right)}{\Delta t}\right] \tag{16}$$

$$x_{id}^{n+1} = x_{id}^{n} + \Delta t v_{id}^{n+1} \tag{17}$$

where χ: constriction coefficient; w: inertia weight; c_1 and c_2: acceleration coefficients; r_1, r_2: random values; n: time index; and Δt: time interval.

First, the initial velocity and position are defined. Then, the objective function is computed for each particle. The global and local leaders are computed, and velocity and position are updated, based on Eqs. 16 and 17. Finally, the termination criterion is checked.

2.2.5 Genetic algorithm

The real coded genetic algorithm is used frequently for optimization. Elitism together with crossover and mutation operators are used in this algorithm (Arikoglu 2017). The elitist strategy attempts to keep the best members in each generation. Thus, the chance of finding of the global solution increases. At the crossover level, two children $Y^1 = \left(y_1^1, y_2^1, .., y_n^1\right)$ and $Y^2 = \left(y_1^2, y_2^2, .., y_n^2\right)$ are considered, and $X^1 = \left(x_1^1, \mathrm{x}_2^1, .., \mathrm{x}_n^1\right)$ and $X^2 = \left(x_1^2, \mathrm{x}_2^2, .., \mathrm{x}_n^2\right)$ are considered as parents. First, a random number (u) is applied to generate the γ crossover parameter as a polynomial probability distribution:

$$\gamma = \begin{bmatrix} (\alpha u)^{\frac{1}{\eta_c+1}} \leftarrow if\,(u) \le \dfrac{1}{\alpha} \\ \left(\dfrac{1}{2-\alpha u}\right)^{\frac{1}{\eta_c+1}} \leftarrow otherwise \end{bmatrix} \tag{18}$$

with $\alpha = 2 - \beta^{-(\eta_c+1)}$and $\beta$is computed by:

$$\beta = 1 + \frac{2}{x_i^2 - x_i^1}\min\left[\left(x_i^1 - x_i^l\right), \left(x_i^u - x_i^2\right)\right] \tag{19}$$

where x_i^u: upper bound of the decision variables, and x_i^l: lower bound of the of the decision variables. Also, x_i^1 and x_i^2 are first and second parents.

Children are generated as:

$$y_i^1 = 0.50\left[\left(x_i^1 + x_i^2\right) - \gamma\left|x_i^2 - x_i^1\right|\right] \tag{20}$$

$$y_i^2 = 0.50\left[\left(x_i^1 + x_i^2\right) + \gamma\left|x_i^2 - x_i^1\right|\right] \tag{21}$$

The mutation operator is applied to chromosomes for changing some genes. Thus, the mutated chromosome is expressed as:

$$x_i^{'} = \begin{bmatrix} x_i + \Delta\left(t, x_i^u - x_i\right) \\ x_i - \Delta\left(t, x_i - x_i^l\right) \end{bmatrix} \tag{22}$$

The function $\Delta(t, y)$ is computed as:

$$\Delta(t,y) = y\left(1 - r^{\left(1-\frac{t}{t_{\max}}\right)^b}\right) \tag{23}$$

where r: random number, and $t_{\max}$: maximum number of iterations.

2.3 Case study

The Aydoghmoush dam is a rock fill dam located in eastern Azarbayejan province, Iran. The purpose of the dam is to supply water for irrigation demands. The average annual inflow for this reservoir is 228×10^3 m^3/h and the maximum and minimum storages are, respectively, 145.7×10^3 and 8.9×10^3 m^3. A period of 10 years (1991–2000) was considered for the study. Figure 6 shows the Aydoghmoush location and Fig. 7 shows the inflow to the reservoir.

The objective function for reservoir operation was considered as the average of square relative differences between allocated volumes and irrigation demands:

$$Minimize(Def) = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{D_t - R_t}{D_{\max}}\right)^2 \tag{24}$$

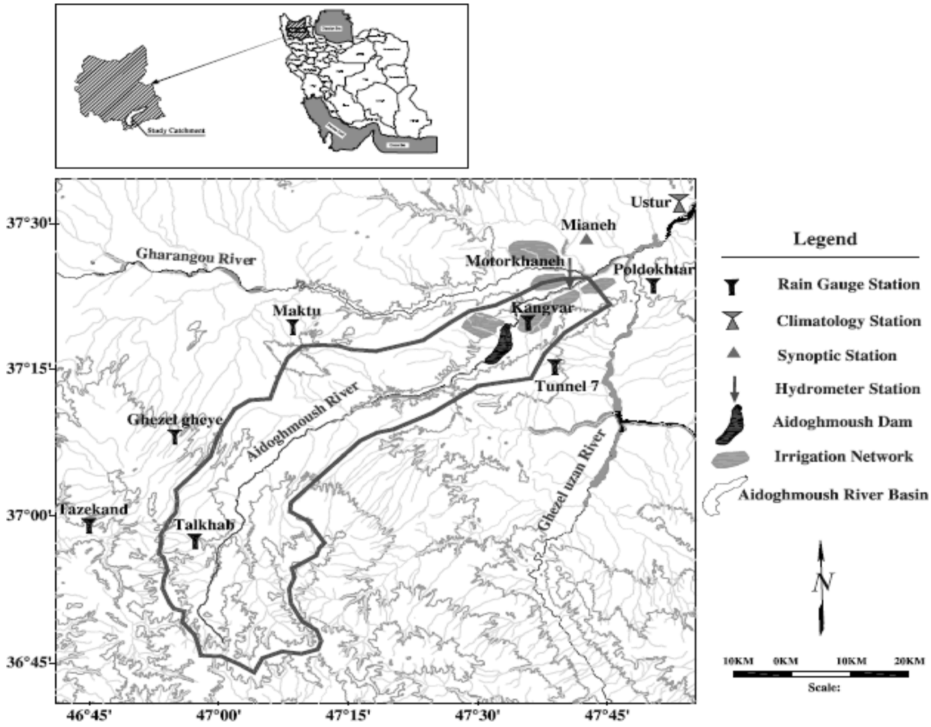


Fig. 6 Aydoghmoush dam

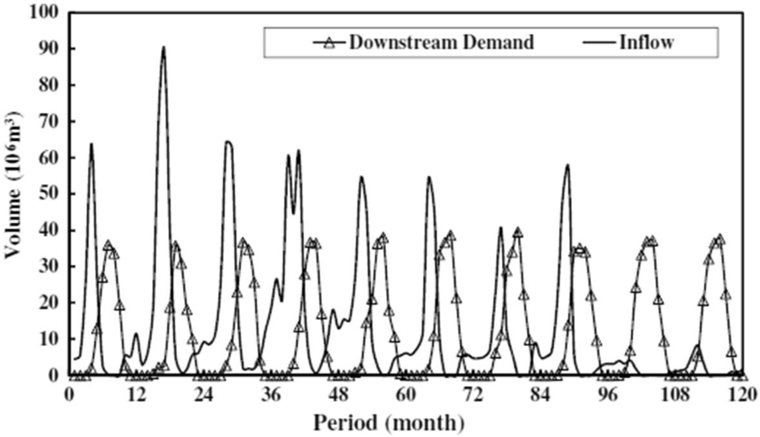


Fig. 7 Inflow to Aydoghmoush dam

$$S_{t+1}=S_t+Q_t-R_t-sp_t-\left(\frac{E_t\times A_t}{1,000}\right),t=1,..,T \tag{25}$$

$$S_{\min}\le S_t\le S_{\max} \tag{26}$$

$$0\le R_t\le D_{\max} \tag{27}$$

$$sp_t=\begin{bmatrix} S_t+Q_t-\left(\frac{E_t\times A_t}{1,000}\right)-S_{\max}\leftarrow if\left(S_t+Q_t-\left(\frac{E_t\times A_t}{1,000}\right)\right)\ge S_{\max} \\ 0\leftarrow otherwise \end{bmatrix} \tag{28}$$

where *Def*: objective function based on deficit, D_t: irrigation demand during the operation periods, R_t: released water during the operation period, $D_{\max}$: maximum irrigation demand in the entire operation time period, S_{t+1}: storage at time $t+1$, Q_t: inflow to reservoir, sp_t: overflow, T: time interval of operation, and A: surface area of reservoir.

Equations 26 and 27 should be satisfied and penalty functions were considered for more assurance:

$$p_1=K\left[\frac{|S_{\min}-S_t|}{S_{\max}-S_{\min}}\right]^2+L \tag{29}$$

$$P_2=C\left[\frac{R_t-D_{\max}}{D_{\max}}\right]^2+G \tag{30}$$

$$P_3=Z\left(\frac{|R_t|}{D_t}\right)+N \tag{31}$$

where p: penalty function, and K, L, C, G, Z and N are the penalty constants. The penalty functions have been reported by Ashofteh et al. (2012) and the value of their coefficients have been computed based on sensibility analysis. The data collection for this study, used for evaluating the proposed model, is set for the period between 1991 and 2000. The main reason is that one of the major objective of the study is to compare the performance of the proposed model versus models that have been applied previously for the same case study. As long as those models were examined during this period, it was decided to examine the proposed model during the same time period (Bozorg-Hadad et al. 2014, 2016).

2.3.1 Investigation of reservoir performance by different indexes

The reliability index depicts the percentage of water supply as (Ashofteh et al. 2012):

$$\alpha=\frac{N_{t=1}^{T}\left(D_t\leq R_t\right)}{T} \tag{32}$$

where α: reliability index, and $N_{t=1}^{T}\left(D_t\leq R_t\right)$: number of periods of water supply. A high value for this index means a higher ability of the system in meeting the target demand during most of the time.

Another performance measure is the vulnerability index. This index shows the ratio of magnitude of the total failure to the total demand and was computed as (Ashofteh et al. 2012):

$$\upsilon=\frac{\sum_{t=1}^{T}\left(D_t-R_t|D_t>R_t\right)}{\sum_{t=1}^{T}D_t} \tag{33}$$

where υ: vulnerability index and $\sum_{t=1}^{T}\left(D_t-R_t|D_t>R_t\right)$: cumulative deficit during the operation period. A low value of vulnerability means that, when there is a failure, its magnitude is not so intense, i.e., a fraction of demand (although not the full target) can still me met.

The resiliency index was considered as:

$$\beta=\frac{N_{t=1}^{T}\left(D_{t+1}\leq R_{t+1}|D_t>R_t\right)}{N_{t=1}^{T}\left(D_t>R_t\right)} \tag{34}$$

where β: resiliency index, and $N_{t=1}^{T}\left(D_{t+1}\leq R_{t+1}|D_t>R_t\right)$: number of system successes after a failure.

Also, the root mean square (RMSE) between demand and value assigned to irrigation was computed as:

$$RMSE=\sqrt{\sum_{t=1}^{T}\frac{\left(D_t-R_t\right)^2}{T}} \tag{35}$$

The mean absolute error (MAE) for allocated water and demand was computed as:

$$MAE = \frac{\sum_{t=1}^{T} |D_t - R_t|}{T} \tag{36}$$

The Nash-Sutcliffe efficiency (NSE) was computed as:

$$NSE = 1 - \frac{\sum_{t=1}^{T} (D_t - R_t)^2}{\sum_{t=1}^{T} \left(D_t - \overline{D}_t\right)^2} \tag{37}$$

$\overline{D}_t$: average demand.

Additionally, the formulation of the kidney algorithm for solving the reservoir operation was based on the following steps:

1- Water release values were considered as decision variables for simplifying the solution of the problem. Thus, release values were defined as the initial population.
2- Initially, the constraints for the problem (Eqs. 26 and 27 and continuity equation) were checked and if they were not satisfied, penalty functions were considered.
3- The objective function for each member of the population was computed.
4- The rate of filtration based on the computed objective function was computed.
5- In this level, filtration, secretion and excretion were applied to these solutes based on the kidney algorithm.
6- If the convergence criterion was satisfied, the algorithm was terminated or else another loop would start from level 2.

2.4 Multi-criteria decision

In this study, some methods are based on different indexes so one of the methods should be selected as the best strategy for reservoir operation. Thus, some multi-criteria decision models were employed. First, a decision variable matrix was considered which included the computed value of each index for each method and each variable was denoted by x_{ij}, i being the i-th method and j the j-th index. Then, these variables were normalized as:

$$\tilde{x}_{ij} = \frac{\left(x_{ij} - x_j^{-}\right)}{x_j^{+} - x_j^{-}} \tag{38}$$

$$\tilde{x}_{ij} = \frac{\left(x_j^{-} - x_{ij}\right)}{\left(x_j^{-} - x_j^{+}\right)} \tag{39}$$

where, x^{-}: lowest value of index j^{th}, and x^{+}: highest value of index j^{th}.

2.4.1 Complex proportional assessment (COPRAS)

In COPRAS, maximizing and minimizing evaluation criteria are computed separately within the computational process. The following steps were considered for using the model (Viteikiene and Zavadskas 2007):

1- The computed values of different indexes in the decision variable matrix based on Eq. 36 and 37 were normalized.

2- The total weight of the evaluation criteria was computed as:

$$S_{+i} = \sum_{j=1}^{k} \left(\tilde{x}_{ij} \times q_j \right) \tag{40}$$

$$S_{-i} = \sum_{j=k+1}^{m} \left(\tilde{x}_{ij} \times q_{ij} \right) \tag{41}$$

where k = number of beneficial criteria, m: number of non-beneficial criteria, q: assigned weight, S_{+i} and S_{-i}: are the sum of the maximizing and minimizing the criteria, respectively.

3- The relative weight Q_i was computed as:

$$Q_i = S_{+i} + \frac{\sum_{i=1}^{n} S_{-i}}{S_{-i} \sum_{i=1}^{n} \left(\frac{1}{S_{-i}} \right)} \tag{42}$$

4- The rank N_i of each method was computed as:

$$N_i = \frac{Q_i}{\max(Q_i)} \times 100 \tag{43}$$

2.4.2 Technique for order preference by similarity to ideal solution (TOPSIS)

This method acted based on the proximity of each solution to the positive ideal solution (D_i^+) and the distance from the negative ideal solution (D_i^-):

$$D_i^+ = \sqrt{\sum_{j=1}^{m} \left(x_{ij} - x_j^+ \right)^2} \tag{44}$$

$$D_i^- = \sqrt{\sum_{j=1}^{m} \left(x_{ij} - x_j^- \right)^2} \tag{45}$$

$$C_i = \frac{D_i^-}{D_i^- + D_i^+} \tag{46}$$

The value of C_i was arranged based on the descending arrangement for determining the rank of each method. Also, D_i^+ is the distance from the ideal solution, D_i^- is the distance from a negative-ideal solution and $C_i$is the similarity ratio.

2.4.3 Modified technique for order preference by similarity to ideal solution (MTOPSIS)

Ren et al. (2007) suggested the following equation instead of computing the C_i value:

$$R_i = \sqrt{\left(\left(D_i^+ - \min\left(D_i^+\right)\right)^2 + \left(D_i^- - \max\left(D_i^-\right)\right)^2\right)} \tag{47}$$

Each method with higher value for R_i has a higher rank and this parameter is known as the similarity ratio.

2.4.4 Weighted aggregates sum for product assessment (WASPAS)

Zavadskas et al. (2012) suggested the model based on weighted sum mode and weighted product mode:

$$A_i = \lambda \sum_{j=1}^{m} \left(x_{ij} \times q_j\right) + (1-\lambda) \prod_{j=1}^{m} \left(x_{ij} \times q_j\right) \tag{48}$$

where the value of λ was equal to 1 (Zavadskas et al. 2012) and each method with higher value for A_i is better ranked.

2.4.5 Borda method

When different multi-criteria decision models are used, they may assign different ranks to each method. Thus, it was important to select the best method, based on the computed ranks of different models for each method. First, each method based on its rank in each multi-criteria decision model had an initial score. This score for each method equaled the difference between the total number of methods with assigned rank number and that method by a multi-criteria decision model. For example, there were $n = 6$ methods in a study. If the rank of one method was 1 in a multi-criteria decision model, its score would be 5 (or n - 1), the method with rank 2 would score 4 (or n - 2), and the worst method would have 0 (or n - n) score. Then, different scores of each method in different multi-criteria decision models were summed with each other to determine the best method.

3 Results and discussion

Table 1 shows the sensitivity analysis for all algorithms. The best population size for the kidney algorithm was 50 and α = 0.50. The population size for shark algorithm was 50 with $M = 30$ and α= 0.6. The population size for the bat algorithm was 50 and the maximum frequency and loudness were, respectively, 5 and 0.60. Other parameters can be seen in Table 1. For example, the crossover and mutation probabilities for the genetic algorithm were 0.50 and 0.60, respectively.

Table 2 shows ten random results for all the investigated algorithms. The value of objective function for 10 runs of KA was 0.789 and it decreased the objective function by 44%, 28%, 55%, 54 and 52% compared to BA, SA, GA, PSA and WA, respectively. Also, the coefficient of variation for KA was 26, i.e. 2.5, 36, 2.6 and 25 (ratios) smaller than BA, SA, GA, PSA and WA, respectively. Thus, the results for KA based on one program run were reliable, since the coefficient of variation was small for this algorithm. Also, the LINGO software, an engineering optimization algorithm, was used for evaluating the different algorithms (Lingo.8. 1997) because the solution of LINGO based on nonlinear programming method can be close to the global solution. The average KA solution was 0.99 of the one found by LINGO. This value resulted based on the ratio of 0.785 (average

Table 1 Sensitivity analysis for different algorithms a: kidney algorithm, b: shark algorithm, c: bat algorithm, d: particle swarm algorithm, e: genetic algorithm and f: weed algorithm

a					
Population size (solutes)	Objective function	α	Objective function		
10	1.211	0.30	1.312		
30	1.010	0.50	1.010		
50	0.789	0.50	0.787		
70	1.040	0.70	0.954		
b					
Population size	Objective function	α_k	Objective function	M	Objective function
10	1.544	0.20	1.766	10	1.545
30	1.322	0.40	1.544	20	1.344
50	1.112	0.60	1.111	30	1.111
70	1.243	0.80	1.244	40	1.232
c					
Population size	Objective function	f_{max} (hz)	Objective function	Maximum loudness (Db)	Objective function
10	1.765	1	1.654	0.2	1.752
30	1.644	3	1.523	0.40	1.612
50	1.422	5	1.422	0.6	1.422
70	1.234	7	1.544	0.80	1.651
d					
Population size	Objective function	$c_1 = c_2$	Objective function	Inertia weight	Objective function
10	2.244	1.6	2.212	0.30	2.211
30	1.987	1.8	1.912	0.50	1.987
50	1.765	2.0	1.764	0.70	1.762
70	1.654	2.1	1.825	0.90	1.878
e					
Population size	Objective function	Crossover probability	Objective function	Mutation probability	Objective function
10	2.565	0.30	2.455	0.20	2.342
30	2.122	0.50	1.787	0.40	2.112
50	1.787	0.70	1.812	0.60	1.787
70	2.111	0.90	1.911	0.80	1.911
f					
Maximum population size	Objective function	S_{max}	Objective function	$\sigma_{initial}$	Objective function
10	1.765	1	1.812	1	1.622
30	1.655	3	1.654	3	1.645
50	1.787	5	1.822	5	1.754
70	1.911	7	1.913	7	1.911

Table 2 10 random results for different algorithms

Run	Kidney algorithm (KA)	Bat algorithm (BA)	Shark algorithm (SA)	Genetic algorithm (GA)	Particle swarm algorithm (PSA)	Weed algorithm (WA)
1	0.789	1.424	1.121	1.787	1.777	1.666
2	0.787	1.434	1.111	1.954	1.765	1.655
3	0.789	1.422	1.111	1.787	1.765	1.655
4	0.789	1.422	1.111	1.789	1.765	1.655
5	0.789	1.422	1.111	1.787	1.765	1.655
6	0.789	1.422	1.114	1.787	1.765	1.655
7	0.789	1.422	1.111	1.787	1.765	1.655
8	0.789	1.422	1.111	1.787	1.765	1.655
9	0.789	1.422	1.111	1.787	1.765	1.655
10	0.789	1.422	1.111	1.787	1.765	1.655
best	0.787	1.422	1.111	1.787	1.765	1.655
worst	0.789	1.434	1.121	1.954	1.777	1.666
average	0.787	1.422	1.111	1.787	1.765	1.655
Variation coefficient	0.0008	0.021	0.002	0.029	0.0021	0.02
Computational time (s)	34	37	36	38	40	37
Global solution	0.785					

solution for KA) to 0.789 (LINGO solution). Of course, the computed solution based on LINGO and nonlinear programming due to the high match with the global solution was considered as the global solution and all algorithms were compared with this solution. This ratio was 0.55, 0.70, 0.43, 0.44 and 0.47 for BA, SA, GA, PSA and WA, respectively. The computational time for the KA was 34 s and it was 3 s, 2 s, 4 s, 6 s and 3 s less than those for BA, SA, GA, PSA and WA, respectively. Thus, it was observed that:

1- KA provided the least value for the objective function
2- KA provided the least value for the coefficient of variation
3- KA provided an average solution close to the global solution
4- GA provided the worst or most value for the objective function

Table 3 shows results of performance indexes for all evaluated algorithms. The least value of RMSE was for the KA (2.12 × 10^6 m^3) and the RMSE for the KA was 74%, 61%, 68%, 33 and 54% less than those for GA, SA, WA, BA and PSA, respectively. In fact, the released water volume for the KA had good match with the demands. The greatest value of the reliability index was also for KA (95%). The reliability index for the KA was 50%, 18%, 41%, 8 and 42% higher that those for GA, SA, WA, BA and PSA, respectively. It means that the released water volume based on KA could supply the demand better than the other methods. BA provided the best value of resiliency (46%) whereas the resiliency for KA was 44%. The ability of the system to recover from failures (critical periods), i.e., the resiliency for the BA was 2% more than that of KA and meaning that the system based on BA can recover itself faster than if it is based on other evolutionary algorithms. Also, KA yielded better objective function than the other methods and SA had the least value for MAE (1.45 × 10^6 m^3). The MAE based on KA was 48%, 34%, 35 and 48% less than those from GA,

Table 3 Computed indexes based on different methods

Algorithm	RMSE ($\times10^6m^3$)	Reliability	Vulnerability	Resiliency	Objective function	MAE ($\times10^6m^3$)	NSE
GA	8.24	45%	20%	18%	1.789	4.12	0.49
KA	2.12	95%	14%	44%	0.789	2.12	0.89
SA	5.45	77%	27%	32%	1.111	1.45	0.76
WA	6.72	54%	18%	28%	1.655	3.24	0.65
BA	3.21	87%	12%	46%	1.422	3.26	0.55
PSA	4.55	53%	19%	16%	1.765	4.12	0.44
	Normalized decision matrix						
GA	0.25	0.47	0.60	0.39	0.44	0.35	0.55
KA	1	1	0.85	0.95	1	0.68	1
SA	0.38	0.81	0.44	0.69	0.710	1	0.85
WA	0.31	0.56	0.66	0.60	0.476	0.44	0.73
BA	0.66	0.91	1	1	0.554	0.43	0.61
PSA	0.46	0.55	0.63	0.34	0.447	0.35	0.49

WA, BA, and PSA, respectively. KA, with a value of 0.89, had the best NSE index. Thus, it can be seen that different methods provided different conditions, based on the different indexes used. High reliability or low vulnerability is important but the best method should also satisfy other indexes. Although the kidney algorithm had more favorable indexes, the multi-criteria decision helped to make a better decision for model selection. Garousi-Nejad et al. (2016) studied this dam based on the firefly algorithm (FFA). The average value for the FFA was 3.60 while the average value from the present study and KA was 0.789 and signifying that the KA performs better for minimizing the objective function. Also, the standard deviation for the KA was 0.0008 while the parameter for the FFA was 0.06. The number of functional evaluation for the KA from the present study (Fig. 9) was 10,000 while it was 20,000 for the FFA. Thus, the Kidney algorithm can optimize the objective function with higher quality and faster than FFA.

Table 4 shows different ranks for the methods based on the multi-criteria decision models. For example, COPRAS suggested a rank of 1 for BA and of 2 for KA, while TOPSIS suggested a rank of 1 for KA and of 3 for BA. Such conditions were seen for different methods in different multi-criteria decision models. It is obvious that relying on COPRAS and TOPSIS to attain a solid conclusion on the best model performance is unclear and undistinguishable, therefore, there is a need to find out more proper method for evaluating the model. In this context, a model able to consider different indexes and multi-criteria decisions could be more effective to judge the performance achieved from the optimization models in more adequate procedure and satisfactory routine in order to implement proper comparison process. In fact, a major weakness of the previous literature is related to the evaluating of different methods based on limited indexes such

Table 4 Computed scores of different methods based on multi criteria decision models

Algorithm	COPRAS	TOPSIS	M-TOPSIS	WASPAS
GA	5	6	6	6
KA	2	1	1	2
BA	1	3	2	1
SA	3	2	3	3
WA	4	4	4	5
PSA	6	5	5	4

Table 5 Computed scores based on Borda index and final rank for different methods

Algorithm	COPRAS (score)	TOPSIS (score)	M-TOPSIS (score)	WASPAS (score)	Total score	Final rank
GA	1	0	0	0	1	6
KA	4	5	5	4	18	1
BA	5	3	4	5	17	2
SA	3	4	3	4	14	3
WA	2	2	2	1	7	4
PSA	0	1	1	1	3	5

as objective function, computational time or coefficient of variation while the present article gives a good insight for the water engineers on how to conduct accurate evaluation of the different evolutionary algorithms.

The usefulness of Borda method is that its aptitude to provide particular score based the examined index for each algorithm and therefore, the summation of all scores based all indexes for particular algorithm is the final score. Subsequently, sorting the attained score for each algorithm is the basic value for comparison among the examined algorithms. The comparison procedure is simply carried out as the higher value of the score reflects the higher rank order of the algorithm. Table 5 shows ranks calculated for the different methods and it is seen that KA with a score of 18 was the best among the other algorithms. On the other hand, it could be observed that the second best is the BA and worst is GA.

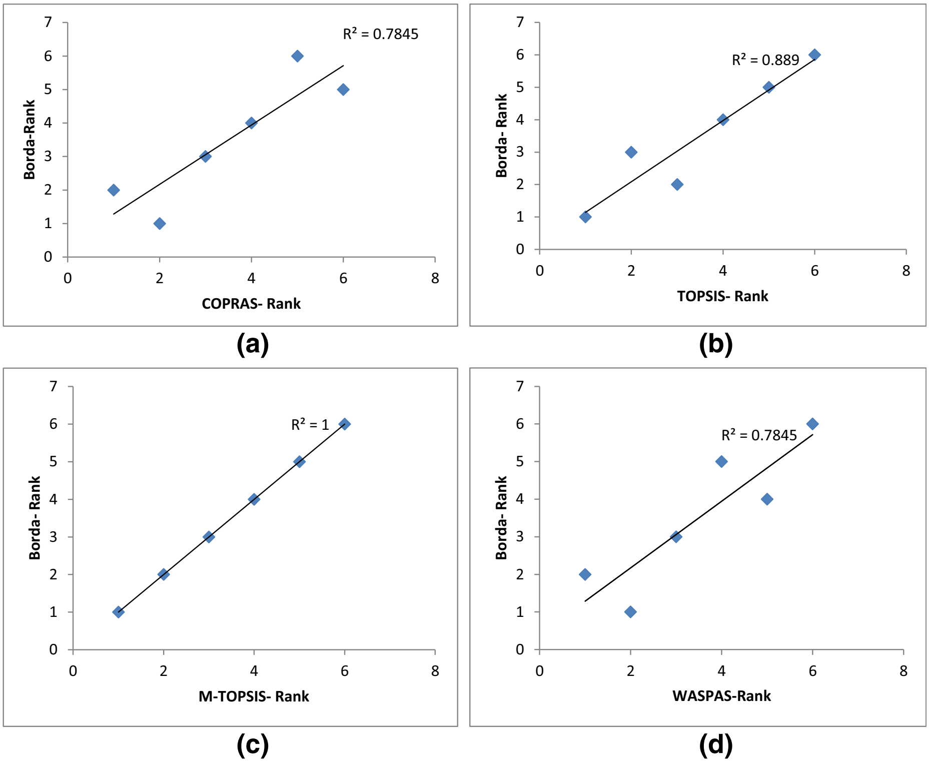


Fig. 8 Correlation of predicated ranks between Borda and **a** COPRAS; **b** TOPSIS; **c** M-TOPSIS; and **d** WASPAS

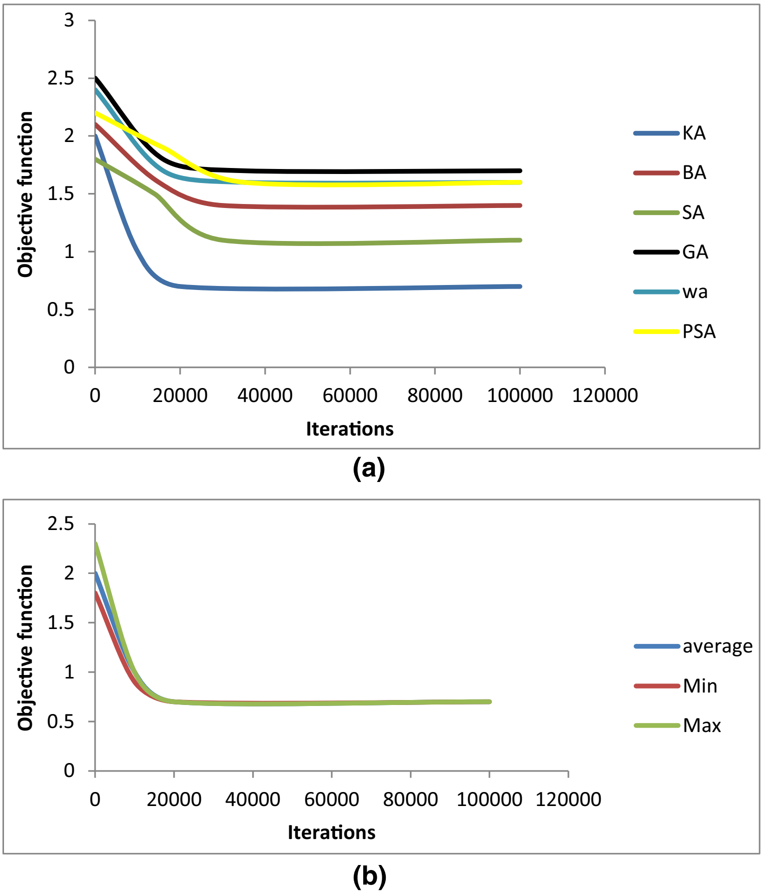
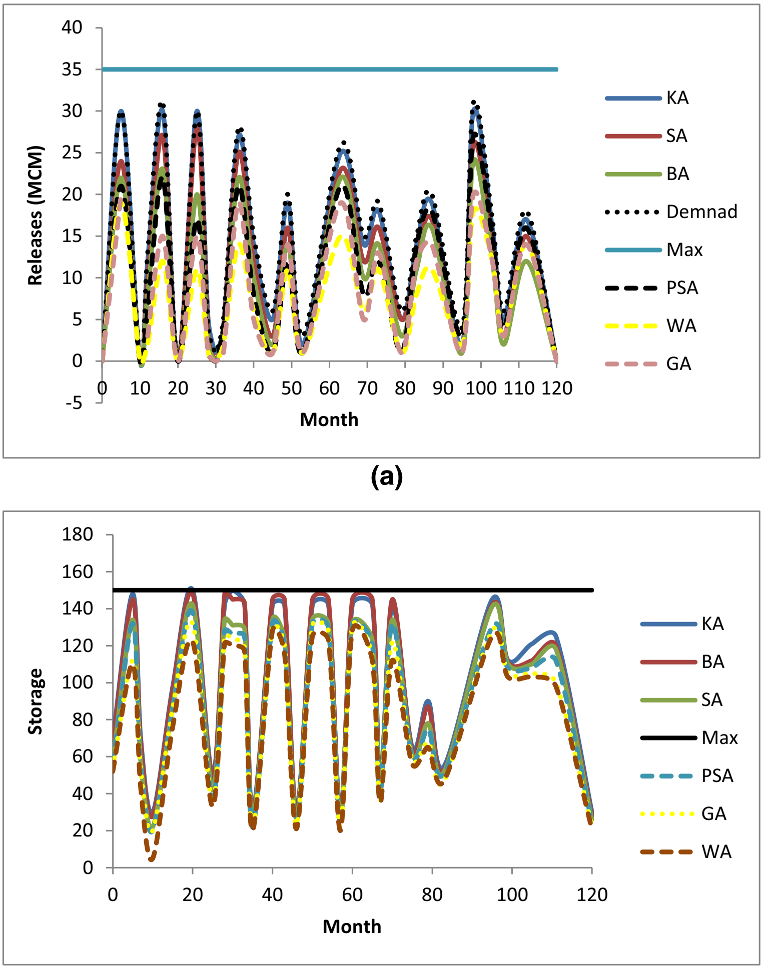


Fig. 9 **a** Convergence of the different algorithms; **b** Convergence of kidney algorithm

Figure 8 shows the correlation of the predicated ranks as presented in Table 5 between the Borda and the other different models. It could be depicted that M-TOPSIS entirely completely matched with the decision index attained using Borda model. On the other hand, both COPRAS and WASPAS models had the lowest correlation value with the Borda index. Thus, the findings from the current study show that the kidney algorithm can be a good selection for optimization dam and reservoir operation problems. Objective function value and computational time as well as different indexes such as RMSE or MAE showed good results when compared to other algorithms often used in the literature like WA, GA, etc. Another aspect was the use of different multi-criteria decision models, which have not comprehensively appeared in past studies for solving similar problems.

Figure 9 shows the convergence of the different algorithms and indicates that the kidney algorithm converged earlier than the other evolutionary algorithms. Also, the maximum and minimum solutions had converged to each other well (see Fig. 9b) revealing the high performance of kidney algorithm. In fact, while utilizing certain optimization algorithm, in case that the mean, average and maximum convergence procedure to the optimal solutions have the identical convergence pattern and same



(a)

(b)

Fig. 10 **a** Release (MCM) and **b** storage (MCM)

end value of the objective function, it indirectly means that the used optimization algorithm experienced the fastest convergence rate achieving the optimal solution. In addition, as shown in Fig. 9b, it could be observed that three convergence procedure are almost identical when examine the KA, the small value of the variation shows the superiority of KA compared to the other algorithms.

Figure 10 shows the water release and storage for the three best methods. It is clear that the water release and storage were within the permissible bounds so they did not violate the maximum water release and reservoir storage. Also, the average water release and storage were 12.74 MCM and 98.89 MCM for the kidney algorithm, respectively. The water release for KA was 15% more than the one for WA, which provided the least release. The less released water volume for the other algorithms (BA, PSA, SA and WA) increased the value of RMSE and vulnerability and decreased the volumetric reliability and resiliency indexes because a less percentage of demands were supplied by the algorithms. The practical aspects of the present study show that engineers should consider different factors for the managing water resources. The application of one mathematical model based on some limited indexes may not guarantee good results.

4 Conclusions

Water resource management is an important research field for engineers. The optimal operation of reservoirs is one of the most studied topics in the water resource management field. The current study dealt with irrigation reservoir operation optimization by means of the new evolutionary KA algorithm. KA is based on advanced operators such as filtration and reabsorption in order to search for the best solutions. The results showed that average objective function values and computational time for KA were all less than those found by GA, SA, WA, BA and PSA. Previous research considered limited indexes for the selection of the best models while this article considered different multi-criteria decision models to give a rank based on different indexes. Although, the KA had the highest rank based traditional performance indices, more comprehensive evaluation method has been applied to help in identifying the selection of the best performance model. In this context, Borda method was applied and showed that the KA had the best performance achieving the first rank among other models. In fact, the current study showed that KA outperformed the other algorithms and overcome their drawbacks to generate optimal operation rule for reservoir system and decision making aspects. Future researches may consider the reservoir operation based on the KA and coupled with reservoir inflow uncertainties and evaporation losses especially for reservoir systems in arid region to obtain more comprehensive optimal release policy from the reservoir system,

Acknowledgements The authors would like to appreciate so much the financial support received from the University of Malaya Research Grant (UMRG) coded RP025A-18SUS University of Malaya, Malaysia.

Compliance with Ethical Standards

Conflict of Interest The authors claim that there is no conflict of interest.

References

Ahmadi M, Haddad OB, Mariño MA (2014) Extraction of flexible multi-objective real-time reservoir operation rules. Water Resour Manag 28(1):131–147

Akbari-Alashti H, Bozorg Haddad O, Fallah-Mehdipour E, Marino MA (2014) Multi-reservoir real-time operation rules: A new genetic programming approach. In Proceedings of the Institution of Civil Engineers-Water Management (Vol. 167, No. 10, pp. 561–576). Thomas Telford Ltd.

Arikoglu A (2017) Multi-objective optimal design of hybrid viscoelastic/composite sandwich beams by using the generalized differential quadrature method and the non-dominated sorting genetic algorithm II. Struct Multidiscip Optim 56(4):885–901

Asgari HR, Bozorg Haddad O, Pazoki M, Loáiciga HA (2015) Weed optimization algorithm for optimal reservoir operation. J Irrig Drain Eng 142(2):04015055

Ashofteh PS, Haddad OB, Mariño A, M. (2012) Climate change impact on reservoir performance indexes in agricultural water supply. J Irrig Drain Eng 139(2):85–97

Bolouri-Yazdeli Y, Haddad OB, Fallah-Mehdipour E, Mariño MA (2014) Evaluation of real-time operation rules in reservoir systems operation. Water Resour Manag 28(3):715–729

Bozorg-Haddad O, Karimirad I, Seifollahi-Aghmiuni S, Loáiciga HA (2014) Development and application of the bat algorithm for optimizing the operation of reservoir systems. J Water Resour Plan Manag 141(8):04014097

Bozorg-Haddad O, Azarnivand A, Hosseini-Moghari SM, Loáiciga HA (2016) Development of a comparative multiple criteria framework for ranking Pareto optimal solutions of a multiobjective reservoir operation problem. J Irrig Drain Eng 142(7):04016019

Choong SM, El-Shafie A, Mohtar WW (2017) Optimisation of Multiple Hydropower Reservoir Operation Using Artificial Bee Colony Algorithm. Water Resour Manag 31(4):1397–1411

Delice Y, Aydoğan EK, Özcan U, İlkay MS (2017) A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. J Intell Manuf 28(1):23–36

Ehteram M, Karami H, Mousavi SF, El-Shafie A, Amini Z (2017) Optimizing dam and reservoirs operation based model utilizing shark algorithm approach. Knowl-Based Syst 122:26–38

Ehteram M, Mousavi SF, Karami H, Farzin S, Singh VP, Chau KW, El-Shafie A (2018a) Reservoir operation based on evolutionary algorithms and multi-criteria decision-making under climate change and uncertainty. J Hydroinf 20(2):332–355

Ehteram M, Karami H, Farzin S (2018b) Reducing Irrigation Deficiencies Based Optimizing Model for Multi-Reservoir Systems Utilizing Spider Monkey Algorithm. Water Resour Manag 32(7):2315–2334

Garousi-Nejad I, Bozorg-Haddad O, Loáiciga HA, Mariño MA (2016) Application of the firefly algorithm to optimal operation of reservoirs with the purpose of irrigation supply and hydropower production. J Irrig Drain Eng 142(10):04016041

Haddad OB, Hosseini-Moghari SM, Loáiciga HA (2015) Biogeography-based optimization algorithm for optimal operation of reservoir systems. J Water Resour Plan Manag 142(1):04015034

Jaddi NS, Alvankarian J, Abdullah S (2017) Kidney-inspired algorithm for optimization problems. Commun Nonlinear Sci Numer Simul 42:358–369

Johnson EG, Kathman AD, Hochmuth DH, Cook AL, Brown DR, Delaney WF (1993) Advantages of genetic algorithm optimization methods in diffractive optic design. In *Diffractive and Miniaturized Optics: A Critical Review* (Vol. 10271, p. 1027105). International Society for Optics and Photonics

Karami H, Ehteram M, Mousavi SF, Farzin S, Kisi O, El-Shafie A (2018) Optimization of energy management and conversion in the water systems based on evolutionary algorithms. Neural Computing and Applications, 1–14

Lingo 8.0 [Computer software] (1997) Lingo Systems, Chicago

Ming B, Liu P, Bai T, Tang R, Feng M (2017) Improving optimization efficiency for reservoir operation using a search space reduction method. Water Resour Manag 31(4):1173–1190

Moeini R, Soltani-nezhad M, Daei M (2017) Constrained gravitational search algorithm for large scale reservoir operation optimization problem. Eng Appl Artif Intell 62:222–233

Ren L, Zhang Y, Wang Y, Sun Z (2007) Comparative analysis of a novel M-TOPSIS method and TOPSIS. Applied Mathematics Research eXpress

Schardong A, Simonovic SP (2015) Coupled self-adaptive multiobjective differential evolution and network flow algorithm approach for optimal reservoir operation. J Water Resour Plan Manag 141(10):04015015

Tayebiyan A, Ali TAM, Ghazali AH, Malek MA (2016) Optimization of exclusive release policies for hydropower reservoir operation by using genetic algorithm. Water Resour Manag 30(3):1203–1216

Viteikiene M, Zavadskas EK (2007) Evaluating the sustainability of Vilnius city residential areas. J Civ Eng Manag 13(2):149–155

Zavadskas EK, Turskis Z, Antucheviciene J, Zakarevicius A (2012) Optimization of weighted aggregated sum product assessment. Elektronika ir elektrotechnika 122(6):3–6

Zhang Z, Jiang Y, Zhang S, Geng S, Wang H, Sang G (2014) An adaptive particle swarm optimization algorithm for reservoir operation optimization. Appl Soft Comput 18:167–177