Multi-objective reservoir operation of the Ukai reservoir system using improved Jaya algorithm

Vijendra Kumar a,*, and S. M. Yadavb

a Civil Engineering Department, G H Raisoni College of Engineering and Management, Pune, Maharashtra, India
b Civil Engineering Department, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India

*Corresponding author. E-mails: vijendra.kumar@raisoni.net, vij10000@gmail.com

ABSTRACT

This paper introduces an effective and reliable approach based on multi population approach, namely self-adaptive multi-population Jaya algorithm (SAMP-JA), to extract multi-purpose reservoir operation policies. The current research focused on two goals: minimizing irrigation deficits and maximizing hydropower generation. Three different models were formulated. The results are compared with ordinary Jaya algorithm (JA), particle swarm optimization (PSO), and Invasive weed optimization (IWO) algorithm. In Model-1, the minimum irrigation deficit was obtained by SAMP-JA and JA as 305092.99 (Mm³). SAMP-JA was better than JA, PSO and IWO in terms of convergence. In Model-2, the maximum hydropower generation was achieved by SAMP-JA, JA and PSO as 1723.50 MWh. While comparing the average hydropower generation, SAMP-JA performed better than JA and IWO. In terms of convergence, SAMP-JA was found to be better than PSO. In Model-3, self-adaptive multi-population multi objective Jaya algorithm (SAMP-MOJA) was better than multi objective particle swarm optimization (MOPSO) and multi objective Jaya algorithm (MOJA) in terms of maximum hydropower generation, and MOPSO was better than SAMP-MOJA and MOJA in terms of minimum irrigation deficiency. While comparing convergence, SAMP-MOJA was found to be better than MOPSO and MOJA. Overall, SAMP-JA was found to be outperforming than JA, POS and IWO.

Key words: hydropower generation, irrigation, optimization, self-adaptive, water Resources

HIGHLIGHTS

- Advanced optimization algorithm free from internal parameters.
- Introduces an effective and reliable approach based on multi population approach.
- An improved Jaya algorithm was used for single and multi-objective optimization models for reservoir operation.
- Both hydropower benefit and irrigation deficiencies are considered.
- Proposed algorithms can generate greater hydropower with lower inflow than the actual inflow.

1. INTRODUCTION

In the case of reservoir operation problems, decisions about releases and storage over a period must be taken with regard to variability in inflows and demands in mind for the best possible system performance (Kumar & Reddy 2007). The traditional optimization techniques used in reservoir operation include linear programming (LP), non-linear programming (NLP), and dynamic programming (DP). While these techniques have been used widely in the past, there have been a few restrictions (Hossain & El-shafie 2013). When operating a multi-purpose reservoir, the objectives are more complex than when using a single-function reservoir, with several problems often including inadequate inflows and greater demands. A model should be built as close to reality as possible to ensure the best possible performance of such a reservoir system. In this process, the model is capable of solving nonlinearity and no convexity problems in its field. LP can solve only a linear problem, NLP is complicated and DP fails if there is an increase in the problem scale (Ming et al. 2015). Furthermore, traditional optimization techniques were found to be trapped in local optimal solutions and difficult to solve multi-objective, non-distinctive, non-convex and discontinuous functionalities (Reddy & Kumar 2006).
To overcome the limitation of traditional methods of optimization, various optimization techniques such as heuristic, meta-heuristic and evolutionary algorithms have been developed. With rising computing power, it becomes very popular in solving complex optimization problems. It also easy the handling of the nonlinear and nonconvex relationships of the model developed. These techniques have been used by various researchers, for example ant colony optimization (ACO) (Kumar & Reddy 2006), genetic algorithm (GA) (Ngoc et al. 2014), honey bee mating optimization (HBMO) (Afshar et al. 2010), particle swarm optimization (Afshar 2012), artificial bee colony (ABC) (Ahmad et al. 2016), cuckoo optimization algorithm (COA) (Hosseini-Moghari et al. 2015), firefly algorithm (FA) (Garousi-Nejad et al. 2016), harmony search (Bashiri-Atrabi et al. 2015), weed optimization algorithm (WOA) (Asgari et al. 2016), and differential evolution (DE) (Ahmadianfar et al. 2016). While existing techniques have their own advantages, there are few shortcomings.

It was observed in the literature that most evolutionary and swarm-based algorithms required the tuning of parameters. There are two types of parameters, i.e. common control parameters and algorithm-specific parameters (Rao et al. 2011). The common control parameters are the number of iterations and the population size which are present in all algorithms. The number of iterations corresponds to the number of times algorithms run to find the best solution, whilst the population is the number of particles assigned to solve the problem. The algorithm specific parameters are the internal parameter. The algorithm-specific parameters are different for each algorithm which needs to be tuned before executing the algorithm. Improper adjustment of these parameters can lead to overall performance of the algorithm, chances of being stuck in the local optimal solution and increased computational time and costs.

For example, GA, required tuning of mutation probability, crossover probability and operation selection. PSO needs inertia weight, cognitive and social parameters. DE needed the tuning of crossover parameters and scaling factor. Harmony search (HS) required tuning of memory size and pitch setting. ACB needs tuning of parameters such as scout, onlooker and employed bees. Bat algorithm (BA) necessary parameters such as wavelength and coefficient of emission to be tuned. Other biological algorithms used in reservoir operations such as HBMO, WOA, COA, improved bat algorithm (IBA), shark algorithm, krill herd algorithm (KHA), imperialest competitive algorithm (ICA), ant colony optimization (ACO) also needed their own algorithm-specific parameters to be tuned.

Jaya Algorithm (JA) has been developed quite recently (Rao 2016), in order to overcome this limitation of algorithm-specific parameters, which does not require algorithm-specific parameters to be tuned up and thus reduces the user’s burden. It works by finding the best solution and avoiding the worst. JA was applied to different optimization fields such as; plate-fin heat exchanger problems (Rao & Saroj 2017), traffic light scheduling problem (Gao et al. 2017), facial emotion recognition (Wang et al. 2018), photovoltaic system maximum power point tracking (Huang et al. 2018). Early users of JA over water resources problems. Kumar & Yadav (2018) used JA to optimize net benefits for the multi-reservoir operating system and found JA to be outperforming. In order to maximize the net benefits, (Varade & Patel 2018) used JA to search for an optimal crop pattern. The findings show that JA was better than PSO. The improved JA, i.e. elitist JA, has been used by (Kumar & Yadav 2019b) to find the optimal crops and elitist JA has been found superior. Kumar & Yadav (2020) compared JA with TLBO, PSO and DE, and found that JA outperformed to optimized water releases in the Ukai reservoir.

In view of the success of JA, this paper seeks to improve the diversity of the population in the search space by dividing the population into sub-populations. The idea is to split the search space into different subgroups. The effectiveness of the algorithm is dependent on the selection of sub-population numbers. This problem was overcome by self-adaptive method. The self-adaptive method helps to effectively monitor the problem and modify the number of individual due to changes of the problem’s landscape. That is, the number of sub-populations has increased or decreased (Venkata Rao & Saroj 2017; Kumar & Yadav 2019a). Using this method, i.e., self-adapting multi-population Jaya algorithm (SAMP-JA) is introduced to extract operating policies for reservoir systems. SAMP-JA is applied to an existing reservoir network, namely the Ukai reservoir system in the state of Gujarat, India, to demonstrate its practical utility. The objectives of the study are to minimize irrigation deficits and maximize the generation of hydropower. Both aims are in conflict with one another. More water is required to satisfy irrigation requirements to reduce the irrigation deficit. At the same time, a high storage in the reservoir is required to produce more energy to increase hydropower output. The motives of the present study are, to run the three different reservoir operation models i.e. (a) Model-1: minimization of irrigation deficits, (b) Model-2: maximization of hydropower generation, (c) Model-3: multi-objective model by combinations of Model-1 and Model-2. The results are compared with ordinary Jaya algorithm (JA), particle swarm optimization (PSO), and Invasive weed optimization (IWO) algorithm. The results of the study would give three different alternatives based on a single or multi objective purpose of the dam operation. That can be used to make decisions and provide the opportunity to operate the dam in the desired alternative to satisfy the objectives.
2. MATERIALS AND METHODS

2.1. Self-adaptive multi-population Jaya algorithm (SAMP-JA)

JA works on the principle of getting closer to a better solution by avoiding failure. The steps for the SAMP-JA operation are as follows.

**Step 1:** Decide the population size and the number of iterations.

**Step 2:** Generate the initial solutions between the variable’s upper and lower limits.

**Step 3:** Divide the population into m groups (i.e., the number of sub-populations initially considered as \( m = 2 \)). \( m \) is updated (bigger or smaller depending on the objective function value) at each iteration.

**Step 4:** The next step is to identify the best and worst solution in the population list. If \( Y_{j,k,i} \) is the \( j^{th} \) variable (i.e. \( j = 1, 2, \ldots, D \), where \( D \) is the number of design variables), for the candidate \( k^{th} \) (i.e., population size, \( k = 1, 2, \ldots, n \)), where \( n \) is the number of candidate solutions during the \( i^{th} \) iteration.

**Step 5:** Each subpopulation uses Equation (1) to modify the solution.

\[
Y_{\text{new},j,k,i} = Y_{j,k,i} + r_{1,j} (Y_{j,\text{best},i} - |Y_{j,k,i}|) - r_{2,j} (Y_{j,\text{worst},i} - |Y_{j,k,i}|)
\]

where, \( Y_{j,\text{best},i} \) and \( Y_{j,\text{worst},i} \) are the best and worst solution, and \( Y_{\text{new},j,k,i} \) is the newly updated solution. The term \( r_{2,j} (Y_{j,\text{worst},i} - |Y_{j,k,i}|) \) helps the algorithm to prevent the worst solution, and the term \( r_{1,j} (Y_{j,\text{best},i} - |Y_{j,k,i}|) \) helps the algorithm to move towards the best solution. \( r_{1,j} \) and \( r_{2,j} \) are the two random numbers between \([0,1]\).

**Step 6:** The objective function obtained by \( Y_{j,k,i} \) i.e., \( O(\text{best-before}) \) and \( Y_{\text{new},j,k,i} \) i.e. \( O(\text{best-after}) \) is compared. The modified solution will only be accepted if it is better than the old solution, then \( m \) will be increased by 1, i.e., \( m = m + 1 \), which will help to increase the exploration of search space. Else \( m \) is decreased by 1, i.e., \( m = m - 1 \), it helps to exploit the search space, here, \( m > 1 \).

**Step 7:** This is completing an iteration. The cycle stops when the maximum number of generations is reached; otherwise, it repeats itself. To maintain diversity, duplicate solutions are replaced by the newly obtained solution. The SAMP-JA flowchart is presented in Figure 1.

The details about the JA can be obtained from (Rao 2016). The details about the PSO can be obtained from (Eberhart & Kennedy 1995). The details about the IWO can be obtained from (Mehrabian & Lucas 2006).

2.2. Multi-objective optimization

Multi-objective optimization contributes to achieving the best compromise solution between different goals, i.e., minimizing or maximizing while meeting all constraints. In this research, a priori method was employed to convert the multi-objective into a single objective function. It is accomplished by assigning an appropriate weight to each objective based on preferential multi-objective optimization (Rao et al. 2019). However, if the objective is simply added by an appropriate weight, a higher functional value will dominate the objective. To avoid this, objective functions are normalized.

For example, if the minimization of the same type is specified by \( f_1(y) \) and \( f_2(y) \), the combined function is written as Equation (2) (Rao et al. 2019).

\[
\min f(y) = w_1 \times \left( \frac{f_1(y)}{f_1} \right) + w_2 \times \left( \frac{f_2(y)}{f_2} \right)
\]

where, \( f(y) \) is a combination function, \( f_1 \) and \( f_2 \) are objective function minimum values when \( f_1(y) \) and \( f_2(y) \) are run independently.

If the \( f_1(y) \) and \( f_2(y) \) objective functions are not of the same type. Say \( f_1(y) \) is a minimization, but \( f_2(y) \) is a maximization, so the Equation (2) shall be modified according to Equation (3) (Rao et al. 2019).

\[
\min f(y) = w_1 \times \left( \frac{f_1(y)}{f_1} \right) - w_2 \times \left( \frac{f_2(y)}{f_2} \right)
\]

where, \( f_2 \) is the maximum value when operating independently, without considering \( f_1(y) \). \( w_1 \) and \( w_2 \) are weights assigned to the \( f_1(y) \) and \( f_2(y) \) objective functions, respectively. All weights must have a sum equal to 1.
Figure 1 | Flowchart of SAMP Jaya algorithm.
3. STUDY AREA DESCRIPTION

Ukai dam, built across the Tapi River in 1972, is situated at 21° 14' 55.52" N and 73° 35' 21.84" E. Tapi River is the west-flowing interstate river in India, which flows through major Maharashtra regions, part of Madhya Pradesh and Gujarat. The total length of the Tapi River is 724 km and has a catchment area of 65,145 km². The Ukai dam has a maximum storage capacity of 8,480.18 Mm³ and a gross storage capacity of 7,414.29 Mm³ at full reservoir level. It is a multi-functional reservoir serving as the main sources of irrigation, domestic demand, power generation and partial flood management. Figure 2 displays an index map of the study area. Figure 2(a) and 2(b) show, respectively, the map of India with Tapi basin and catchment area. There are three canals which distribute irrigation water. The first canal diverts straight from the dam, i.e. Ukai left the main canal at the bank (ULBMC). The other two canals are diverted from the Kakrapar weir, which is 29 kilometers downstream of the Ukai dam. i.e. Kakrapar left bank main canal (KLBMC) and Kakrapar right bank main canal (KRBMC). Later, Ukai right bank main canal (URBMC) split from KRBMC. Figure 2(c) shows the Cultivable Command Area (CCA) of Canals. The releases

![Index map of Study area](http://iwaponline.com/ws/article-pdf/doi/10.2166/ws.2021.374/5385877/12021374.pdf)
of dams are used in generating power. The Ukai reservoir is comprised of two power plants, one on the main dam with an installed capacity of 300 MW since 1974. Another ULBMC mini station has been operating since 1988 with an installed capacity of 5 MW. In 1995 on the Rivers Tapi, near Rander Surat, Singanpor Weir cum causeway was built. The limited downstream released from the Ukai dam is performed to meet the domestic demands, industrial demands and water quality requirements of the region. The surplus water from the weir goes into the Arabian Sea.

Table 1 presents silent features for the Ukai basin. The necessary data was compiled from the Ukai left bank division and the Surat irrigation circle. The data for the analysis were monthly inflow from the reservoir (1972–2016), monthly inflow from the Ukai reservoir (1972–2016), monthly storage capacity from the Ukai reservoir (1972–2016), monthly discharge through the powerhouse (1976–2016), irrigation demand for all CCAs, canal releases, generated power (1976–2016), monthly evaporation rates (1972–2016), reservoir areas (1972–2016), domestic demands, industrial demands and water quality demands (1990–2016).

4. MATHEMATICAL MODEL FORMULATION

The research aims to reduce irrigation deficits and maximize the production of hydropower. These are contradictory objectives and are presented as follows.

4.1. Minimize the sum of the square deficits for irrigation demands annually

Equation (4) was used to minimize the sum of the square deficits for irrigation demands annually.

\[
\text{Minimize } SQDV = f_1(y) = \sum_{t=1}^{12} (D_{ULBMC,t} - IR_{ULBMC,t})^2 + \sum_{t=1}^{12} (D_{KLBMC,t} - IR_{KLBMC,t})^2 + \sum_{t=1}^{12} (D_{KRBMC,t} - IR_{KRBMC,t})^2
\]

where, \( SQDV \) is the sum of irrigation demand and release square deviations. \( D_{ULBMC,t}, D_{KLBMC,t} \) and \( D_{KRBMC,t} \) are the irrigation demands for the Ukai left bank main canal (ULBMC), Kakrapar left bank main canal (KLBMC) and Kakrapar right bank main canal (KRBMC), respectively in period \( t = 1, 2, \ldots, 12 \), in Mm³. \( IR_{ULBMC,t}, IR_{KLBMC,t} \) and \( IR_{KRBMC,t} \) are the irrigation released for the Ukai left bank main canal, Kakrapar left bank main canal and Kakrapar right bank main canal, in period \( t = 1, 2, \ldots, 12 \), in Mm³, respectively.

Table 1 | Silent features of the basin

| Features                  | Readings                                      |
|----------------------------|-----------------------------------------------|
| Longitude                  | 72°33' to 78°17'E                             |
| Latitude                   | 20°9'N to 22°N                                |
| Average Rainfall           |  820.07 mm                                    |
| Highest Elevation          |  1,556 m                                      |
| Area                       |  65,145 Km²                                   |
| States in the basin        | Maharashtra, Madhya Pradesh, and Gujarat       |
| Highest dam                | Ukai                                          |
| Number of irrigation projects | Major -13, Medium -68                        |
| No of flood forecasting sites | 3                                           |
| No of sub-basins           | 3                                             |
| No of watersheds           | 100                                           |
| No of villages             | 9,443                                         |
| Catchment area             | 62,225 Km²                                    |
| Top of Dam                 | 111.252 m                                     |
| Type of spillway           | Radial                                        |
| Road width on the spillway | 6.706 m                                       |
4.2. Maximize the annual hydropower generation

The average flow of 1 m$^3$/s at a height of 1 m generates 9,810 watts of power (Vedula & Mujumdar 2006). Power $p$ is generally obtained according to Equations (5) and (6).

\[ p = \rho * g * q_t * H_t \]  
\[ p = 9,810 * q_t * H_t \]

where, $p$ is power in watts, $\rho$ is water density (1,000 kg/m$^3$), $g$ is acceleration due to gravity (9.81 m/s$^2$), $q_t$ is discharge in m$^3$/s and $H_t$ is the head in m.

The total energy production kilowatt-hour (kWh) in period $t$ can be checked as follows in Equations (7) and (8).

\[ P = \frac{9,810 * 10^6 * R_t * H_t}{3,600} \]  
\[ P = 2,725 * R_t * H_t \]

where, $P$ is the total annual power generation in kWh, $R_t$ is the total release in Mm$^3$ and $H_t$ is the head in m (Vedula & Mujumdar 2006).

The Equation (8) is modified as Equation (9) to maximize the supply of hydropower for the Ukai dam.

Maximize $P = f_2(y) = \sum_{t=1}^{12} 2725 * \eta^*(R_{1,t} + IR_{KLMC,t} + IR_{KRMBC,t}) * H_{1,t} + \sum_{t=1}^{12} 2725 * \eta^*(IRULBC,t) * H_{2,t}$

where, $P$ is the total annual power generation in kilowatt-hour (kWh), $\eta$ is the overall power plant efficiency, $R_{1,t}$ is the river bed turbine release in Mm$^3$ in period $t$, $H_{1,t}$ and $H_{2,t}$ are the net head on the river bed turbines and Ukai left bank main canal during period $t$. The objectives are subjective to the following constraints:

4.2.1. Continuity constraints

The relationship between inflow and outflow is expressed using continuity constraints; this is mathematically given by Equation (10).

\[ S_{t+1} = S_t + I_t - (R_{1,t} + IR_{KLMC,t} + IR_{KRMBC,t} + IR_{ULBC,t}) - E_t - Ovf_t \]

where, $S_t$ is the active reservoir storage in Mm$^3$ at the beginning of period $t$, $S_{t+1}$ is the reservoir storage in Mm$^3$ for the period $t$, $I_t$ is the Ukai reservoir inflow in Mm$^3$ during the period $t$, $E_t$ is the evaporation losses in Mm$^3$ during period $t$, $Ovf_t$ is the reservoir overflow in Mm$^3$ during period $t$.

4.2.2. Evaporation constraints

In most models the losses are equal to the algebraic difference between precipitation and evaporation on the surface of the reservoir. Other losses, such as leakage from the bottom of the reservoir, are considered relatively small compared with other factors. The evaporation losses are calculated as in Equations (11) and (12).

\[ E_t = \frac{E_{et} * A_t}{1,000} \]  
\[ A_t = \frac{A_t + A_{t+1}}{2} \]

where, $E_t$ is the reservoir surface evaporation losses in Mm$^3$ during period $t$, $E_{et}$ is the reservoir evaporation in mm during the period $t$, $A_t$ and $A_{t+1}$ are the reservoir areas at the beginning of the period $t$ and $t+1$, in $10^6$ m$^2$, respectively.
4.2.3. Storage constraints
Storage of the reservoir should be less than or equal to the reservoir’s maximum storage capacity, and greater than or equal to the minimum storage capacity as in Equation (13).

\[
S_{\min} \leq S_t \leq S_{\max}
\]  

(13)

where, \( S_{\min} \) and \( S_{\max} \) are the minimum and maximum storage capacity as per the rule curve during period \( t \) in \( Mm^3 \).

4.2.4. Maximum power production constraints
Power production at any time period \( t = 1, 2 \ldots 12 \) must be less than or equal to the maximum power production capacity according to Equations (14) and (15).

\[
2,725 \times \eta \times (R_{1,t} + IR_{KLBMCM,t} + IR_{KRBMCM,t}) \times H_{1,t} \leq P_{1,t,\max}
\]  

(14)

\[
2,725 \times \eta \times (IR_{ULBMCM,t}) \times H_{2,t} \leq P_{2,t,\max}
\]  

(15)

where, \( P_{1,t,\max} \) and \( P_{2,t,\max} \) are the maximum power produced by river bed turbines and Ukai left bank turbines during the period \( t \) in kWh respectively. \( H_t \) is defined using Equation (16) with the head and storage relation.

\[
H_t = c_1 + c_2 \times S_t + c_3 \times S_t^2 + c_4 \times S_t^3 + c_5 \times S_t^4
\]  

(16)

where \( c_1, c_2, c_3, c_4 \) and \( c_5 \) are the constant coefficients of the storage height equation.

4.2.5. Canal capacity constraints
The carrying capacity of the canal must be less than or equal to the maximum carrying capacity of the canal at any time period \( t = 1, 2 \ldots 12 \). The releases into the canal from the dam are expressed in Equations (17)–(19).

\[
IR_{ULBMCM,t} \leq C_{ULBMCM,t,\max}
\]  

(17)

\[
IR_{KLBMCM,t} \leq C_{KLBMCM,t,\max}
\]  

(18)

\[
IR_{KURBMCM,t} \leq C_{KRBMC,t,\max}
\]  

(19)

where, \( C_{ULBMCM,t,\max}, C_{KLBMCM,t,\max} \) and \( C_{KRBMC,t,\max} \) are the maximum canal carrying capacity in ULBMC, KLBMC and KRBMC, respectively in the period \( t = 1, 2, \ldots, 12 \), in \( Mm^3 \).

4.2.6. Overflow constraints
There should be a condition of overflow so that additional water can spill off. If there is no overflow condition, the models will result in spilling water, even if the storage capacity is lower. This is given by Equation (20).

\[
O_{\text{ovf}} \geq S_t + I_t - (R_{1,t} + IR_{KLBMCM,t} + IR_{KRBMC,t} + IR_{ULBMCM,t}) - E_{t} - S_{\max}
\]  

(20)

where, \( O_{\text{ovf}} > 0 \), \( S_{\max} \) is the maximum storage capacity observed in \( Mm^3 \).

4.2.7. Irrigation demands
The irrigation release should be greater than or equal to the minimum irrigation requirements, and less than or equal to the maximum irrigation requirements according to Equations (21)–(23).

\[
D_{ULBMCM,t,\min} \leq IR_{ULBMCM,t} \leq D_{ULBMCM,t,\max}
\]  

(21)

\[
D_{KLBMCM,t,\min} \leq IR_{KLBMCM,t} \leq D_{KLBMCM,t,\max}
\]  

(22)

\[
D_{KRBMC,t,\min} \leq IR_{KRBMC,t} \leq D_{KRBMC,t,\max}
\]  

(23)
where, $D_{ULBMC_{i},min}$ and $D_{ULBMC_{i},max}$ represent the minimum and maximum ULBMC irrigation demands for period $t$. $D_{KLBMC_{i},min}$ and $D_{KLBMC_{i},max}$ represent the minimum and maximum KLBMC irrigation demands for period $t$. $D_{KRBMC_{i},min}$ and $D_{KRBMC_{i},max}$ represent the minimum and maximum KRBMC irrigation demands for period $t$.

4.2.8. Water quality requirements

The environmental flow requirements at any time period $t = 1, 2 \ldots 12$ must be greater than or equal to the minimum downstream release of the river as shown in Equation (24).

$$R_{i,t} \geq DR_{min,t}$$

(24)

where, $DR_{min,t}$ is the minimum downstream river release to satisfy domestic, industrial and aquatic quality requirements in $Mm^3$ for the period $t$.

5. MODEL APPLICATION AND PARAMETERS

Single-objective and multi-objective models are developed with the intention of minimizing the water deficit and maximizing hydropower generation. As stated earlier, a total of three models were developed. The models are (a) Model-1: minimization of irrigation deficits, (b) Model-2: maximization of hydropower generation, (c) Model-3: multi-objective model by combinations of Model-1 and Model-2. All algorithms have been coded by software MATLAB R2014b, and plots and statistical parameters have been developed with the Origin-Pro 8.5 software. Via a system type of 64-bit, system configuration of Intel® core™ i7-7700, @3.60 GHz, with an installed ram of 8.00 GB.

The performances of algorithms in 10 independent runs were checked through different combinations of common control parameters such as population size (e.g., 30, 50, 75 and 100) and number of iterations 10,000. Best results were found in all models in population size 30 for JA and SAMP-JA and for POS and IWO for population size 50. The internal parameters for the PSO, i.e., the cognitive parameter ($c_1$) and social parameter ($c_2$) were taken as 1.5 and 2.0, respectively and the inertia weight $\omega(t)$ as 1. The internal parameters for IWO were the minimum and maximum number of seeds, i.e., $Seed_{min}$ and $Seed_{max}$ were held as 0 and 2, the non-linear modulation index ($a$) was taken as 2, the initial standard deviation (SD) ($\sigma_{initial}$) and final SD ($\sigma_{final}$) were taken as 0.6 and 0.001. No inner parameters were available for SAMP-JA and JA. The static penalty is applied to the reservoir in order to meet the storage constraint of the reservoir. The penalty is $p_{(n,t)}$, and the penalty functions are expressed in Equations (25) and (26).

If $p_{(n,t)} == 0$, then penalty function = 0

Else if $p_{(n,t)} \neq 0$, then penalty function $= g(n) * abs (p_{(n,t)})^2$

(25) (26)

The penalty was used in the context of the main objective. Penalty function depends on the violation of the different constraints and penalty parameter is the figure of penalty applied. The modified objective is written according to Equations (27)–(29). A positive sign represents the addition of this violation from the objective function minimization problem and a negative sign represents the subtraction of this violation from the objective function maximization problem.

The model-1 is modified as Equation (27).

$$\min f(y) = f_1(y) + g(n) * abs (p_{(n,t)})^2$$

(27)

The model-2 is modified as Equation (28).

$$\max f(y) = f_2(y) - g(n) * abs (p_{(n,t)})^2$$

(28)

The model-3 is modified as Equation (29).

$$\min f(y) = w_1 * \left( \frac{f_1(y)}{f_1} \right) - w_2 * \left( \frac{f_2(y)}{f_2} \right) + g(n) * abs (p_{(n,t)})^2$$

(29)

The next section presents the results and discussion.
6. RESULTS AND DISCUSSION

The dependable annual inflows for Ukai reservoir were calculated from the annual probability distribution. The 75% dependable inflow was computed using the Weibull formula based on available historical monthly reservoir inflow data (Subramanya 2013). Inflow at any percentage probability ($P_p$) represents the flow magnitude in an average year that can be expected to be equal to or exceed ($P_p$) percent of the time and is termed as $P_p$% dependable inflow $I_l$ (Subramanya 2013). Forty-two years (1972-2016) of available historical inflow data were used to generate 75% dependable inflow (Vasan & Raju 2009; Jothiprakash & Arunkumar 2013). Frequency analysis is obtained to determine the relationship between the magnitude of the rainfall and its probability of exceedance as shown in Equation (30).

\[ P_p = \frac{M}{N + 1} \]  

(30)

where, $N$ is the number of a record year, $M$ is the number of orders in the descending order of magnitude, $P_p$ is the probability of an event. Figure 3 shows the probability of exceedance vs inflow in $Mm^3$. It shows that as the probability of exceedance increases the inflow decreases, while when the probability of exceedance decreases the inflow increases.

6.1. Model-1: minimization of irrigation deficits

6.1.1. Analysis of irrigation deficiencies

Irrigation deficiencies have been calculated as the sum of irrigation release square deficiencies ($Mm^3)^2$. Table 2 displays JA, SAMP-JA, PSO, and IWO's 10 different run objective function value. Comparing the best optimal solution, SAMP-JA and JA achieved a better minimum solution of 305,092.99 ($Mm^3)^2$ compared to PSO and IWO with an optimal minimum solution of 305,093.02 and 14,702,579.28 ($Mm^3)^2$, respectively. When comparing the mean optimal solution, SAMP-JA performed better with a mean optimal solution of 305,093.00 ($Mm^3)^2$ compared to JA, PSO and IWO. When comparing the standard deviation (SD) it was found that the least SD was obtained by SAMP-JA, then by JA, PSO and IWO. Compared to SAMP-JA, JA and PSO, the IWO progressed very slowly. The IWO's near optimum solution has been accomplished with 1,00,000 iteration numbers, i.e. 312,045.28 ($Mm^3)^2$.

![Figure 3](image-url) | Probability of exceedance vs total inflow.
6.1.2. Convergence plot by different algorithm

Figure 4 shows the convergence rates of the different algorithms for model-1 up to 10,000 iterations. It was noted that the SAMP-JA, JA and PSO convergence rates were similar and better than IWO. It has been observed that SAMP-JA converges faster than JA and PSO to achieve a global solution. Figure 5 demonstrates the convergence rate of IWO for 100,000 iterations. It was noticed that around 100,000 iterations IWO was able to obtain near optimal solution, i.e., 312,045.28 (Mm³)².

6.1.3. Statistical efficient parameters

The following statistical efficiency parameters were used to determine the efficiency.

| Sr No. | JA       | SAMP-JA  | PSO      | IWO      |
|--------|----------|----------|----------|----------|
| 1      | 305,093.45 | 305,093.00 | 305,339.40 | 20,876,940.77 |
| 2      | 305,096.71 | 305,093.01 | 305,093.40 | 104,195,436.33 |
| 3      | 305,092.99 | 305,092.99 | 305,093.02 | 15,408,678.12  |
| 4      | 305,093.45 | 305,092.99 | 305,095.67 | 123,926,179.62 |
| 5      | 305,093.45 | 305,092.99 | 305,093.02 | 20,776,615.12  |
| 6      | 305,096.71 | 305,093.04 | 305,109.74 | 128,256,770.98 |
| 7      | 305,092.99 | 305,095.67 | 305,093.07 | 18,788,140.81  |
| 8      | 305,108.70 | 305,093.03 | 305,093.16 | 18,345,618.49  |
| 9      | 305,096.71 | 305,092.99 | 305,109.74 | 105,125,699.50 |
| 10     | 305,093.15 | 305,093.00 | 305,109.74 | 128,256,770.98 |
| Best   | 305,092.99 | 305,092.99 | 305,093.02 | 14,702,579.28  |
| Worst  | 305,108.70 | 305,093.03 | 305,121.60 | 57,040,265.90  |
| Mean   | 305,095.83 | 305,093.00 | 305,121.60 | 57,040,265.90  |
| Standard Deviation | 4.81 | 0.02 | 76.82 | 50,761,252.88 |

**Table 2 | Optimal solution obtained by different algorithm**
6.1.3.1. Root mean square error (RMSE). This index calculates the square root mean error between water demand and releases as shown in Equation (31).

$$\text{absolute RMSE} = \sqrt{\frac{1}{t} \sum_{t=1}^{T} (D_t - R_t)^2}$$  \hspace{1cm} (31)$$

where, $D_t$ is the demand for water and $R_t$ is the released water during $t = 1, 2, \ldots, 12$, in $Mm^3$. $RMSE = 0$, suggest good agreement between observed and simulation (modelled) values.

6.1.3.2. Mean absolute error (MAE). This index calculates the absolute mean error. It is the absolute difference between water demand and water release according to Equation (32). The MAE may differ between 0 and $\infty$. $MAE=0$, suggest good agreement between observed and simulation (modelled) values.

$$MAE = \frac{1}{t} \sum_{t=1}^{T} |D_t - R_t|$$  \hspace{1cm} (32)$$

The results of water demand and water release for irrigation are shown in Table 3 based on different error indexes for the best solution for various algorithms. The root mean square error (RMSE) calculated for SAMP-JA, JA, and PSO is equal to 154.05 $Mm^3$ and IWO is 208.84 $Mm^3$. SAMP-JA, JA and PSO had a comparatively small mean absolute error (MAE) compare to IWO.

| Error Indexes                        | SAMP-JA | JA    | IWO   | PSO   |
|--------------------------------------|---------|-------|-------|-------|
| Root mean square error (RMSE) in $Mm^3$ | 154.05  | 154.05| 208.84| 154.05|
| Mean absolute error (MAE) in $Mm^3$  | 266.33  | 266.33| 280.45| 266.33|
6.1.4. Performance evaluation measures

The following assessment indicators are used.

6.1.4.1. Reliability index. This index shows the relationship between released and demand water, and a high value is anticipated (Ehteram et al. 2018) as shown in Equation (33).

\[
\alpha_v = \frac{\sum_{i=1}^{t} R_i}{\sum_{i=1}^{t} D_i}
\]  

(33)

where, \( \alpha_v \) is the reliability index. The indicator’s higher percentage is desirable.

6.1.4.2. Vulnerability index. This index is related to the system’s failure intensity and it is preferred to have a lower percentage (Ehteram et al. 2018) as shown in Equation (34).

\[
\lambda = \max\left(\frac{D_i - R_i}{D_i}\right)
\]

(34)

where, \( \lambda \) is the vulnerability index.

Table 4 shows the results of different algorithm of water releases and water demand for irrigation based on performance indicators for the best solution. The greater reliability index percentages for SAMP-JA, JA and PSO indicate better irrigation water supply and optimal reservoir operation compared to IWO. Also, the lower SAMP-JA, JA and PSO vulnerability index values also indicate that the severity of the failure is lower than IWO. Thus, the operation of the reservoir using SAMP-JA, JA and PSO can better prevent water shortages than the IWO.

6.1.5. Optimal reservoir operation policy

Figure 6 displays the resulting storage strategy for the reservoir obtained from different algorithms for Model-1 along with actual average storage for the Ukai reservoir. The storage curve of various algorithms is found to follow a similar trend except for IWO. Filling of the reservoir happens between July to October during the monsoon season. The maximum capacity is reached in the month of October. Around November to June the water level of the dam goes down to supply for irrigation and other uses during the non-monsoon season. The levels of SAMP-JA and JA storage were nearly identical but for all the months MOJA was slightly lower in storage. For the months of October to February, the actual average storage was higher, and other months were similar or lower. The PSO storage levels among all algorithms were the lowest. Water level of the IWO was high for the month of April, after the low month of March; this is not possible in real cases during the non-monsoon season. The lower side of the storage by all the algorithms shows that the reservoir attempts to operate for irrigation as only priority at 75% dependable inflow.

Later the best optimal reservoir storage strategy was compared with storage data for 2008 and 2009 where the actual inflow and the 75% dependable inflow were close. Figure 7 shows the SAMP-JA Storage Policy with Actual Storage of 2008 and 2009. The actual data of 2008 were almost the same for the months of October to January, high for February to April, and low for June to September compared with the developed model. The actual 2009 data were similar in terms of trends but the model developed was on the higher side.

Figure 8 shows the comparative plot of actual average demand and different algorithms for the release of water from the Ukai dam via the Kakrapar weir to the Kakrapar left bank main canal (KLBMC). For all the months, the optimum releases of SAMP-JA, JA and PSO have been almost the same. And with the actual average demand for the month of June to January it

Table 4 | Results of different algorithm based on performance indicators

| Performance Index (%) | SAMP-JA | JA  | PSO  | IWO  |
|------------------------|---------|-----|------|------|
| Reliability index      | 48.01   | 48.01 | 48.01 | 45.26|
| Vulnerability index    | 98.03   | 98.03 | 98.03 | 99.70|
was about the same or smaller, and the average releases were more for other months. All the algorithms resulted in releases per month according to the irrigation requirement. When comparing the IWO, the release was found to be very high for the few months and very low for a few months. That indicates the distribution of the release water is not even. Figure 9 shows the comparative plot of actual average demand and different algorithms for the release of water from the Ukai dam via the Kakrapar weir to the Kakrapar right bank main canal (KRBMC). Again, for this canal the optimum releases of SAMP-JA, JA and PSO were approximately the same for all the months. But the releases were lower than the actual average demand. Again, for this canal, the IWO, the release was found to be very high for the few months and very low for a few months. That indicates the distribution of the release water is not even. The SAMP-JA, JA, and PSO were better at minimizing irrigation and demand.
deficits than the IWO algorithm. Figure 10 shows the comparative plot of actual average demand and various algorithms for the release of water from the Ukai dam to the Ukai left bank main canal (ULBMC) considering the 75% dependable inflow. Again, for this canal the optimum releases of SAMP-JA, JA and PSO were approximately the same for all the months. But the releases were lower than the actual average demand. The result of IWO, the release was found to be very high for the few months and very low for a few months. Thus, the operation of the reservoir using SAMP-JA, JA and PSO can better prevent water shortages than the IWO.

6.2. Model-2: maximization of hydropower generation

6.2.1. Analysis of hydropower generation

In order to maximize the production of hydropower, a high level of storage in the reservoir is needed to generate more energy. Table 5 shows the 10 different run objective function values of JA, SAMP-JA, PSO, and IWO. Comparing the best optimal
solution, SAMP-JA, JA and PSO achieved a better hydropower generation of 1723.50 M kWh compared to IWO with an optimum hydropower generation of 1566.52 M kWh. When comparing the mean optimal solution, SAMP-JA and PSO performed better with the average generation of hydropower compared to JA and IWO. While comparing the standard deviation (SD) it was observed that the least SD was obtained from SAMP-JA and PSO. The IWO progressed very slowly compared to SAMP-JA, JA and PSO. When the IWO algorithm is run for 100,000 iterations, the near-optimal solution has been achieved as 1645.15 M kWh.

| Sr No. | JA (M kWh) | SAMP-JA (M kWh) | PSO (M kWh) | IWO (M kWh) |
|--------|------------|-----------------|-------------|-------------|
| 1      | 1723.50    | 1723.50         | 1723.50     | 1566.52     |
| 2      | 1711.14    | 1723.50         | 1723.50     | 1534.72     |
| 3      | 1723.50    | 1723.50         | 1723.50     | 1514.60     |
| 4      | 1723.50    | 1723.50         | 1723.50     | 1499.64     |
| 5      | 1723.50    | 1723.50         | 1723.50     | 1498.39     |
| 6      | 1723.50    | 1723.50         | 1723.50     | 1490.39     |
| 7      | 1723.50    | 1723.50         | 1723.50     | 1565.73     |
| 8      | 1723.50    | 1723.50         | 1723.50     | 1525.65     |
| 9      | 1719.08    | 1723.50         | 1723.50     | 1490.39     |
| 10     | 1723.50    | 1723.50         | 1723.50     | 1566.52     |
| Best   | 1723.50    | 1723.50         | 1723.50     | 1525.65     |
| Worst  | 1711.14    | 1723.50         | 1723.50     | 1490.39     |
| Mean   | 1721.62    | 1723.50         | 1723.50     | 1522.91     |
| Standard Deviation | 3.96 | 0.00 | 0.00 | 26.54 |
6.2.2. Convergence plot by different algorithm
The convergence rates of the Model-2 for various algorithms up to 10,000 iterations are shown in Figure 11. It was noted that to obtain the optimum solution, SAMP-JA convergence rate was faster than POS and JA. The rate of IWO convergence was very slow. The IWO convergence rate for 100,000 iterations is shown in Figure 12. It was found that around 100,000 iterations IWO was able to obtain near optimal solution, i.e., 1645.15 $M\ kWh$.

6.2.3. Optimal reservoir operation policy
Figure 13 shows the resulting storage strategy for the reservoir obtained from different algorithms for Model-2 along with actual average storage for the Ukai reservoir. Here the operation’s primary focus is hydropower. It is found that a similar trend follows the storage curve of different algorithms except the IWO. Since hydropower was only a priority, the storage of the reservoir was kept to a maximum so that maximum hydropower could be achieved. The filling of the reservoir occurs during the monsoon season from July to October. The maximum capacity is reached in the month of October. The model developed by SAMP-JA, JA and PSO and the actual average storage have followed similar trends, but the algorithms where on the higher side. So, the hydropower can be maximized.

6.3. Model-3: multi-objective model by combinations of Model-1 and Model-2
In this research, a priori method was used to convert the multi-objective into a single objective function. Minimizing irrigation deficits is considered as ($w_1$) and maximizing the generation of hydropower as ($w_2$), the weights was taken as $w_1 = 50, w_2 = 50$ i.e., equal weighting has been given. Model-3 analysis was performed using SAMP-JA, JA and PSO. Since IWO output in Model 1 and 2 was slow to achieve the optimal solution, and more iteration was required, it wasn’t selected for Model-3. Table 6 shows the best optimum solution obtained from 10 independent runs. It was found that the maximum power generation was achieved by SAMP-MOJA and that the minimum irrigation deficiency was achieved by MOPSO. Figure 14 show the convergence rates for Model-3 up to 10,000 iterations for the various algorithms. It was found that SAMP-JA and JA were similar and better than PSO convergences. Figure 15 shows the hydropower plot generated by algorithms and the actual data from the Ukai reservoir. The blue column indicates the inflow, and the hydropower generated by the orange curve. It can be observed that all algorithms can generate greater hydropower with lower inflow. In 2004, similar generation of hydropower was observed with higher inflow.
Figure 12 | Convergence of the IWO algorithm for Model-2.

Figure 13 | Reservoir storage policy obtained by different algorithms along with actual average storage.

Table 6 | Optimal solution obtained by different algorithms for Model-3

| Algorithm   | Maximum Power Generation (M kWh) | Minimum Irrigation Deficiency (Mm$^3$)$^2$ |
|-------------|----------------------------------|---------------------------------------------|
| SAMP-MOJA   | 460.85                           | 535,063.99                                  |
| MOJA        | 458.15                           | 614,848.66                                  |
| MOPSO       | 432.82                           | 496,956.73                                  |
6.3.1. Optimal reservoir operation policy

Figure 16 shows the resulting reservoir storage policy obtained from different multi objective algorithms for Model-3 along with actual average storage for the Ukai reservoir. It is noted that the storage curve of different algorithms, along with actual data, is following a similar trend. The filling of the reservoir occurs during the monsoon season from July to October. The maximum capacity is reached in the month of October. Around November to June, the water level of the dam declined during the non-monsoon season to irrigation supplies and other uses. The actual average storage for the months October to December was comparable with SAMP-MOJA, and the rest of the month was on the higher side. MOJA and MOPSO had less capacity than the storage level SAMP-MOJA had. The best optimal reservoir storage policy was later compared with the
storage data of 2008 and 2009, where the actual inflow and the 75 percent dependable inflow were similar. Figure 17 shows the SAMP-MOJA Storage Policy with Actual Storage of 2008 and 2009. The actual data were close in patterns for 2008 and 2009, but the model developed was on the higher side. The greater side of the storage would help to maximize the production of hydropower.

Figure 18 shows the comparative plot of actual average demand and different algorithms for the release of water from the Ukai dam via the Kakrapar weir to the Kakrapar left bank main canal (KLBMC). MOPSO’s optimum releases were divided more evenly compared to the SAMP-MOJA and MOJA, except the month of January. It is also observed in the monsoon months, the algorithms attempted to release less water and more release in the non-monsoon months, taking into account

Figure 16 | Reservoir storage policy obtained by different algorithms along with actual average storage for Model-3.

Figure 17 | Storage policy for the reservoir with actual storage for 2008 and 2009 for Model-3.
the 75 percent dependable inflow. Figure 19 shows the comparative plot of actual average demand and different algorithms for the release of water from the Ukai dam via the Kakrapar weir to the Kakrapar right bank main canal (KRBMC). In the monsoon months, the SAMP-MOJA attempted to release less water, and more release in the non-monsoon months. Compared with SAMP-MOJA and MOJA, MOPSO tried to maintain uniform water release. Figure 20 shows the comparative plot of actual average demand and various algorithms for the release of water from the Ukai dam to the Ukai left bank main canal (ULBMC) considering the 75% dependable inflow. MOPSO's optimum releases were better and more uniform than the MOJA and SAMP-MOJA. For Model-3, MOPOS was better than the SAMP-MOJA and MOJA to minimize the irrigation and demand deficits. But the SAMP-MOJA storage level was higher than the MOPOS, so more releases could be possible.
7. CONCLUSIONS

In this study, the operation of the reservoir is solved by using 75% dependable inflow. The three different models of reservoir operation i.e. (a) Model-1: minimization of irrigation deficits, (b) Model-2: maximization of hydropower generation, (c) Model-3: multi-objective model by combinations of Model-1 and Model-2. The self-adaptive multi-population Jaya algorithm (SAMP-JA) is used to solve the models. The results are compared with Jaya algorithm (JA), particle swarm optimization (PSO), and Invasive weed optimization (IWO) algorithm. In Model-1, the minimum irrigation deficit was obtained by SAMP-JA and JA as 305092.99 (Mm$^3$)$^2$. SAMP-JA was better than JA, PSO and IWO in terms of convergence. In Model-2, the maximum hydropower generation was achieved by SAMP-JA, JA and PSO as 1723.50 M kWh. While comparing the average hydropower generation SAMP-JA and PSO performed better than JA and IWO. In terms of convergence, SAMP-JA was better than PSO. In Model-3, SAMP-MOJA was better than MOPSO and MOJA in terms of maximum hydropower generation, and MOPSO was better than SAMP-MOJA and MOJA in terms of minimum irrigation deficiency. While comparing convergence, SAMP-MOJA was found to be better than MOPSO and MOJA. SAMP-JA was found to be outperforming JA, POS and IWO.

The results of the study provide three different alternatives priorities based on a single or multi-objective reservoir operation. It has been observed that, if irrigation is the only priority for the reservoir, it needs higher release throughout to fulfil the irrigation requirements. While this is reversed when the focus is for hydropower, it tends to keep the storage head at a high level throughout the year to generate more electricity. When operating the reservoir, these two goals conflict with each other. The compromise multi-objective reservoir operation was achieved when there was an equal priority for irrigation and hydropower. The study provides the potential for three alternative priorities for both objectives and their respective operational policies to run the dam in order to meet the desired priority. The developed irrigation model will help to avoid water shortages in the supply of irrigation water, taking into account the 75% dependable inflow. It was also observed that algorithms can generate greater hydropower with lower inflow than the actual inflow.

ACKNOWLEDGEMENTS

The authors are grateful to the editor and the reviewers for their useful comments and suggestions.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.
REFERENCES

Afshar, A., Shafii, M. & Haddad, O. B. 2010 Optimizing multi-reservoir operation rules: an improved HBMO approach. *Journal of Hydroinformatics* 13 (1), 121.

Afshar, M. H. 2012 Large scale reservoir operation by constrained particle swarm optimization algorithms. *Journal of Hydro-Environment Research* 6 (1), 75–87.

Ahmad, A., Razali, S. F. M., Mohamed, Z. S. & El-shafie, A. 2016 The application of artificial bee colony and gravitational search algorithm in reservoir optimization. *Water Resources Management* 30 (7), 2497–2516.

Ahmadianfar, I., Adib, A. & Salarijazi, M. 2016 Optimizing multireservoir operation: hybrid of bat algorithm and differential evolution. *Journal of Water Resources Planning and Management* 142 (2), 05015010.

Asgari, H.-R., Bozorg Haddad, O., Pazoki, M. & Loaiciga, H. A. 2016 Weed optimization algorithm for optimal reservoir operation. *Journal of Irrigation and Drainage Engineering* 142 (2), 04015055.

Bashiri-Atrabi, H., Qaderi, K., Rheinheimer, D. E. & Sharifi, E. 2015 Application of harmony search algorithm to reservoir operation optimization. *Water Resources Management* 29 (15), 5729–5748.

Eberhart, R. & Kennedy, J. 1995 A new optimizer using particle swarm theory. MHS’95. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. pp. 39–43.

Ehteram, M., Karami, H. & Farzin, S. 2018 Reducing irrigation deficiencies based optimizing model for multi-reservoir systems utilizing spider monkey algorithm. *Water Resources Management* 32 (7), 2315–2334.

Gao, K., Zhang, Y., Sadollah, A., Lentzakis, A. & Su, R. 2017 Jaya, harmony search and water cycle algorithms for solving large-scale real-life urban traffic light scheduling problem. *Swarm and Evolutionary Computation* 37 (February), 58–72.

Garousi-Nejad, I., Bozorg-Haddad, O. & Loaiciga, H. A. 2016 Modified firefly algorithm for solving multireservoir operation in continuous and discrete domains. *Journal of Water Resources Planning and Management* 142 (9), 04016029.

Hossain, M. S. & El-shafie, A. 2013 Intelligent systems in optimizing reservoir operation policy: a review. *Water Resources Management* 27 (9), 3587–3547.

Hosseini-Moghari, S. M., Morovati, R., Moghadas, M. & Araghinejad, S. 2015 Optimum operation of reservoir using two evolutionary algorithms: imperialist competitive algorithm (ICA) and cuckoo optimization algorithm (COA). *Water Resources Management*, 3749–3769.

Huang, C., Wang, L., Yeung, R. S. C., Zhang, Z., Chung, H. S. H. & Bensoussan, A. 2018 A prediction model-guided Jaya algorithm for the PV system maximum power point tracking. *IEEE Transactions on Sustainable Energy* 9 (1), 45–55.

Jothiprakash, V. & Arunkumar, R. 2013 Optimization of hydropower reservoir using evolutionary algorithms coupled with chaos. *Water Resources Management* 27 (7), 1963–1979.

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

Kumar, V. & Yadav, S. M. 2018 Optimization of reservoir operation with a new approach in evolutionary computation using TLBO algorithm and Jaya algorithm. *Water Resources Management* 32 (13), 4375–4391.

Kumar, V. & Yadav, S. M. 2019a Self-adaptive multi-population based Jaya algorithm to optimize the cropping pattern under a constraint environment. *Journal of Hydroinformatics*, 1–18.

Kumar, V. & Yadav, S. M. 2019b Optimization of cropping patterns using elitist-Jaya and elitist-TLBO algorithms. *Water Resources Management* 33 (5), 1817–1833.

Kumar, V. & Yadav, S. M. 2020 In: *Advanced Engineering Optimization Through Intelligent Techniques*, Vol. 949 (Venkata Rao, R. & Taler, J., eds). Springer, Singapore.

Mehrhabian, A. R. & Lucas, C. 2006 A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1 (4), 355–366.

Ming, B., Chang, J. x., Huang, Q., Wang, Y.-m. & Huang, S.-z. 2015 Optimal operation of multi-reservoir system based-on cuckoo search algorithm. *Water Resources Management* 29 (15), 5671–5687.

Nagesh Kumar, D. & Janga Reddy, M. 2007 Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management* 133 (3), 192–201.

Ngoc, T. A., Hiramatsu, K. & Harada, M. 2014 Optimizing the rule curves of multi-use reservoir operation using a genetic algorithm with a penalty strategy. *Paddy and Water Environment* 12 (1), 125–137.

Rao, R. V., Savsani, V. J. & Vakharia, D. P. 2011 Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design* 43 (3), 305–315.

Rao, R. V., Saroj, A., Oclon, P., Taler, J. & Lakshmi, J. 2019 A posteriori multiobjective self-adaptive multipopulation Jaya algorithm for optimization of thermal devices and cycles. *IEEE Access* 7, 4115–4134.

Reddy, M. J. & Kumar, D. N. 2006 Optimal reservoir operation using multi-objective evolutionary algorithm. *Water Resources Management* 20 (6), 861–878.

Subramanya, K. 2013 Engineering hydrology. *Tata McGraw-Hill Edu*, 45–60.

Varade, S. & Patel, J. N. 2018 Determination of optimum cropping pattern using advanced optimization algorithms. *Journal of Hydrologic Engineering* 23 (6), 05018010.
Vasan, A. & Raju, K. S. 2009 Comparative analysis of simulated annealing, simulated quenching and genetic algorithms for optimal reservoir operation. Applied Soft Computing Journal 9 (1), 274–281.

Vedula, S. & Mujumdar, P. P. 2006 Water Resource Systems Modelling Techniques and Analysis. Available from: http://www.cabdirect.org/abstracts/19812609957.html.

Venkata Rao, R. 2016 Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations 7 (1), 19–34.

Venkata Rao, R. & Saroj, A. 2017 A self-adaptive multi-population based Jaya algorithm for engineering optimization. Swarm and Evolutionary Computation 37 (April), 1–26.

Wang, S. H., Phillips, P., Dong, Z. C. & Zhang, Y. D. 2018 Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm. Neurocomputing 272, 668–676.

First received 6 September 2021; accepted in revised form 18 October 2021. Available online 28 October 2021.