

Article

Computer Modeling for the Operation Optimization of Mula Reservoir, Upper Godavari Basin, India, Using the Jaya Algorithm

Vartika Paliwal ^{1,*}, Aniruddha D. Ghare ^{1,†}, Ashwini B. Mirajkar ^{1,†},
Neeraj Dhanraj Bokde ^{2,*} and Andrés Elías Feijóo Lorenzo ^{3,†}

¹ Department of Civil Engineering, Visvesvaraya National Institute of Technology, Nagpur 440010, India; adghare@civ.vnit.ac.in (A.D.G.); abmirajkar@civ.vnit.ac.in (A.B.M.)

² Department of Engineering—Renewable Energy and Thermodynamics, Aarhus University, 8000 Aarhus, Denmark

³ Departamento de Ingeniería Eléctrica—Universidade de Vigo, Campus de Lagoas-Marcosende, 36310 Vigo, Spain; afeijoo@uvigo.gal

* Correspondence: vartikapaliwal19@gmail.com (V.P.); neerajdhanraj@eng.au.dk (N.D.B.); Tel.: +91-9028415974 (N.D.B.)

† These authors contributed equally to this work.

Received: 5 December 2019; Accepted: 18 December 2019; Published: 20 December 2019



Abstract: In this paper, an application of the Jaya Algorithm (JA) is presented, to develop an operation optimization model for the Mula reservoir, located on the upper Godavari Basin, in India. The mentioned algorithm is a relatively new optimization technique, which is algorithm-specific and parameterless. In JA, there is no need for algorithm-specific parameter tuning, unlike with other heuristic techniques. To test its applicability, the model performance has been compared with that of other models for hypothetical four reservoir system studies available in the literature. Simulations for hypothetical four reservoir system have proven that JA is a better solution for a number of Function Evaluations when compared with the results obtained by means of other evolutionary methods such as Genetic Algorithms, Particle Swarm Optimization, Elitist Mutated Particle Swarm Optimization, and Weed Optimization Algorithm models reported in previous studies. Simulations have been carried out for real time operation of the Mula reservoir, and have revealed its superior performance when comparing the water releases proposed by it and the ones proposed by existing policy. Hence, from the two case studies presented, it can be concluded that the JA has potential in the field of reservoir operation and can be further explored to operation optimization of existing multi-reservoir system, with lower computations.

Keywords: Jaya algorithm; reservoir operation; optimization

1. Introduction

Nowadays, the management of resources has become vital for the sustenance of humankind. Water is among the most important ones for living beings. Freshwater has now become a scarce natural resource due to the effect of climate change and water pollution, and majorly due to the rapid increase of population [1]. The management of water resources has been the supreme prerequisite for today's rapidly developing world, as the conflict between the supply and requirement of water resources has become a major concern. These conflicts can be resolved by developing new water resource projects and increasing the overall management efficiency [1]. The development of a new water resource project to serve human needs is not easy due to environmental issues and because the cost of building new projects increases every year; therefore, the operation optimization of the existing projects is a better

alternative to meet the present and future water requirements. Many researchers have been working towards rational approaches to allot the water very optimally in every essence. Many approaches can be found in the literature, such as can be read in in every essence; vagaries of approach have been commended in the literature with excellent opinions regarding the distribution of resources [2].

The techniques suggested and implemented in reservoir operation studies are chance constrained Linear Programming (LP) by [3], in which multi-purpose reservoir operation optimization was carried out with the aim of maximization of hydropower generation and fulfillment of irrigation requirements according to the reliability level. The nonlinear power production function was linearized and a solution was obtained in the range of the specified tolerance. In a similar way, a multi-purpose reservoir operation optimization was carried out by using Non-Linear Programming (NLP) in [4] for the Koyna dam, Maharashtra, India. Dynamic Programming (DP) [5] was implemented in the integration with fuzzy rules and simulation studies to generate the general operating policies for Dez and Karoon reservoirs in Iran. Ant Colony Optimization (ACO) was implemented for the reservoir operation optimization of the Dez reservoir system, Iran, in [6], and resulted in a global optimal solution although the tuning of parameters was recommended as the model is parameter sensitive. The application of the Bat Algorithm (BA) to the Karoun-4 reservoir system in Iran and to another hypothetical reservoir system showed the merits of the BA over other traditional techniques [7]. The Biogeography Based Optimization algorithm was tested on three mathematical benchmark functions and then applied to a single reservoir and a four reservoir system [8]. Furthermore, the authors in [9] applied a Charged System Search Algorithm (CSSA) to a benchmark function and to the analysis of the Dez reservoir, in Iran, and a comparison was established with results by Genetic Algorithms (GA), Particle Swarm Optimization (PSO), ACO, and gradient based NLP. The results proved superiority and robustness in comparison to other approaches. The Crow Algorithm (CA) proved its potential for reservoir operation optimization studies in [10]. It was implemented in the case of a multi-reservoir system in China for hydropower generation and was found to outperform other metaheuristic approaches by taking less computational time. Differential Evolution (DE) was used by [11] to determine the optimal cropping pattern for the Bisalpur project, Rajasthan, India. Furthermore, the authors in [12] proved the superiority of the Firefly Algorithm (FA) over the Genetic Algorithm in terms of convergence rate of global optima and variance of the result when tested over five mathematical benchmark functions, reservoir system operation optimization with irrigation as a purpose, and similarly with the purpose of hydropower production. Jothiprakash et al. [13] applied the GA to Pechiparai reservoir, Tamil Nadu, India, and derived rule curves on the basis of reliability of the GA model. Considering meta-heuristic optimization methods as powerful and reliable ones, applied a Gradient Evolution Algorithm (GEA) to Khersan-1 and Dez reservoir, Iran, as a novel approach in [14]. Harmony Search (HS), when implemented by [15] to a benchmark problem and to the Narmab reservoir, Iran, resulted in a higher rate of convergence with promising results and was found to be effective in flood management operations of the Narmab reservoir. Honey Bee Mating Optimization (HBMO) provides comparable results to those obtained by means of LP and other well developed approaches [16]. Hybrid Algorithm (HA) of Artificial Fish Swarm Algorithm (AFSA) and Particle Swarm Optimization (PSO) Algorithm were developed and implemented by [17] for the analysis of the Karun-4 hydropower system and concluded that hybrid results are better than AFSA and PSO by overcoming the drawback of these methods. By hybridization, the possibility of AFSA being trapped in a local optimum is decreased and hence it increases its convergence. In the case of PSO, the diversity of its response is increased by hybridization. HA is assessed on the basis of reliability, resilience, and vulnerability indices. For verification, it is tested on few mathematical functions. The JA model was developed by [18] for Jayakwadi-I reservoir, Maharashtra, India, to derive optimal releases and performance indices. A study by [19], applied Particle Swarm Optimization (PSO), Elitist Mutated PSO (EMPSO), and GA to the Bhadra reservoir system, India, which is a multipurpose reservoir. A weighted approach was used for handling the multi-objectives. It was found that Elitist Mutated PSO performed better than GA and standar PSO. Shark Algorithm (SA) was applied by [20] to the complex cases of reservoir

operation optimization and was found to yield good results. Teaching Learning Based Optimization (TLBO) and JA implemented in [21] to the benchmark studies and concluded that both methods provide satisfactory solutions like other ones. The Water Cycle Algorithm (WCA) implemented in [22] to reservoir operation optimization stating that, for solving complex optimization problems, evolutionary optimization approaches are reliable and simple. The authors in [23] introduced the Weed Optimization Algorithm (WOA) to reservoir operation optimization by its application to continuous and discrete four reservoir system problem and compared with the global results from the past studies. The authors in [24] implemented Wolf Search Algorithm (WSA) to a continuous Four Reservoir system and a single reservoir hydropower optimization and found that WSA resulted in being close to a global solution and that it is better than GA for the two scenarios.

A review study has been carried out by [25] to access the reservoir operation optimization and management using different optimization models like LP, NLP, DP, Differential Dynamic Programming (DDP), Discrete Differential Dynamic Programming (DDDP) model, and stochastic models, describing their pros and cons and concluding that it is difficult to state which particular technique would be the best for all real world problems. Furthermore, a review of evolutionary approaches in reservoir operation studies is done by [2] concluding that evolutionary approaches are the most popular options for reservoir operation study due to its improving capability and the advancement of computer technology. In addition to this, a survey on Artificial Intelligence (AI) techniques in reservoir operation is presented in [26], where it was found that an evolutionary algorithm (EA) has a great potential in the field of optimization, but new techniques which proved to be powerful in other domains should be applied to reservoir operations, and concluded that, with an increasing number of methods, it is difficult to state which technique fits better for a specific reservoir system.

Though extensive research has been done to provide appropriate operational policies for reservoir system, there is still scope for improvement in the aspect of accuracy, handling of increasing water demands and reducing operational complexity [26]. There is also a need for a better approach to improve the effectiveness and efficiency of the reservoir operation, which can be attempted by application of recently developed optimization techniques based on their success in dealing with real world problems [2]. In the reservoir operations, an optimization algorithm can be said to be a better one when either it provides better optimal values or it achieves the same value (global) for less effort and computational time [19,22].

One such technique, newly developed, is the Jaya Algorithm, with potential features, such as requiring fewer parameters compared to other population based optimization techniques and needs algorithm specific parameters [27,28]. Furthermore, it is capable to enhance its exploration and exploitation capacities with a lower number of function evaluations (FE) [29]. JA has been used in the present work for the optimization of reservoir operation studies. It is a meta-heuristic optimization technique with a more presumptuous approach to revise the variable value to obtain the optimal solution. JA was developed [28] after the progress and success of the previously developed TLBO algorithm [30] in engineering applications.

Interestingly, JA does not depend on the nature of the objective function, i.e., maximization or minimization. Several other optimization algorithms are built-up particularly for either minimization or maximization problems, which necessitates a change in the sign of the objective function to transform the problem from one case to the other. JA does not need such a modification and can be well applied to both cases. It is also noted that JA can also be easily applied to the complex functions since it uses the objective function value as an information source to upgrade the results. Upgrading the variable in the JA is associated with two random numbers, one of them with the best solution among the population and the other with the worst one. With this, there is a high rate of randomness involved, and, as such, it explores search in uncertain wider areas. Hence, JA can be considered as a stochastic optimization algorithm. The constraints are incorporated by using a penalty function; hence, more constraints do not affect the convergence of the model.

In general, heuristic approaches use a direct search in the problem space and follow the most promising path for the desired solution, hence they bridge the gap between the complexity and completeness of algorithms, which constitute the two basic qualities of search. However, these techniques have various parameters to be tuned when trying to achieve the optimal solution. Even if a single parameter is not appropriate in evolutionary methods, it leads to divergence from the optimal solution. However, in JA, there are no such tunings required, and this offers the advantage of reduced efforts, easy understanding, and application. In the JA technique, the only parameters to be set are population size and generations; thus, it requires comparatively less effort. JA takes less computational time than other algorithms while providing better results in several instances [31].

JA has so far been implemented for the analysis of a wide variety of problems such as tea category identification [32], complex constrained problems describing various benchmark functions [33], heat exchanger problems [34], pathological brain detection [35], forecast model for currency exchange production [29], design optimization of truss [31], and for deriving optimum cropping patterns [36]. The real-time reservoir operation optimization problem is more complicated when compared to a theoretical or hypothetical problem. Hence, powerful optimization methods are required to solve these problems [37]. Rao and Waghmare [33] compared nine algorithms on five test problems, 11 algorithms on four problems and six algorithms on 11 problems and concluded that JA provides better results than the other approaches. These led authors to think of the possibility of application of JA to a reservoir operation problem that is complex in nature [38]. In the present study, a computer model is developed for JA using MATLAB (R2015a) for reservoir operation optimization of the Mula Reservoir, Upper Godavari Basin, India, which is an existing reservoir system, and simulation studies have been carried out to check the model performance in comparison to the existing one. In addition, JA is applied to a four reservoir system problem (Theoretical case) [39] to test its applicability and performance in comparison with other approaches to the basic reservoir operation problem (benchmark problem).

2. Materials and Methods

2.1. Description of the Jaya Algorithm

JA requires only two basic parameters: population size and number of generations. This feature of JA reduces the complexity of tuning of the other parameters (algorithm-specific parameters) as is faced in other evolutionary algorithms. Therefore, the authors in [28,40] described JA as an algorithm-specific, parameter-less algorithm. In this approach, a simulation and evaluation strategy is applied. It means that, in each generation, first, the decision variables and objective function values are evaluated and then constraints are checked. If there is any violation of the constraints, the penalty is applied to the objective function, resulting in a modified function value. From the best and the worst values of an altered function, further processing is carried out. The penalty function is formulated such that the solution containing the violation does not give the best result and is rejected. The termination criterion is the pre-set maximum number of generations after which the model will stop functioning and will provide the results. The way JA operates is illustrated in Figure 1.

A stepwise detailed description of JA is as follows: let ' m ' be the number of a variable, ' b ' the position of a variable ($b = 1$ to m), ' n ' the population size, ' c ' the candidate solution per population ($c = 1$ to n), and ' a ' the number of iterations, for $a = 1$, the value of these variables are initialized using (1) based on the minimum and maximum bounds of the respective variable:

$$X_{b,c} = \min(X_b) + r \times (\max(X_b) - \min(X_b)), \quad (1)$$

where

- $\min(X_b)$ is the minimum value of the variable X_b ,
- r is a random number ($r \in [0, 1]$),
- $\max(X_b)$ is the maximum value of the variable X_b .

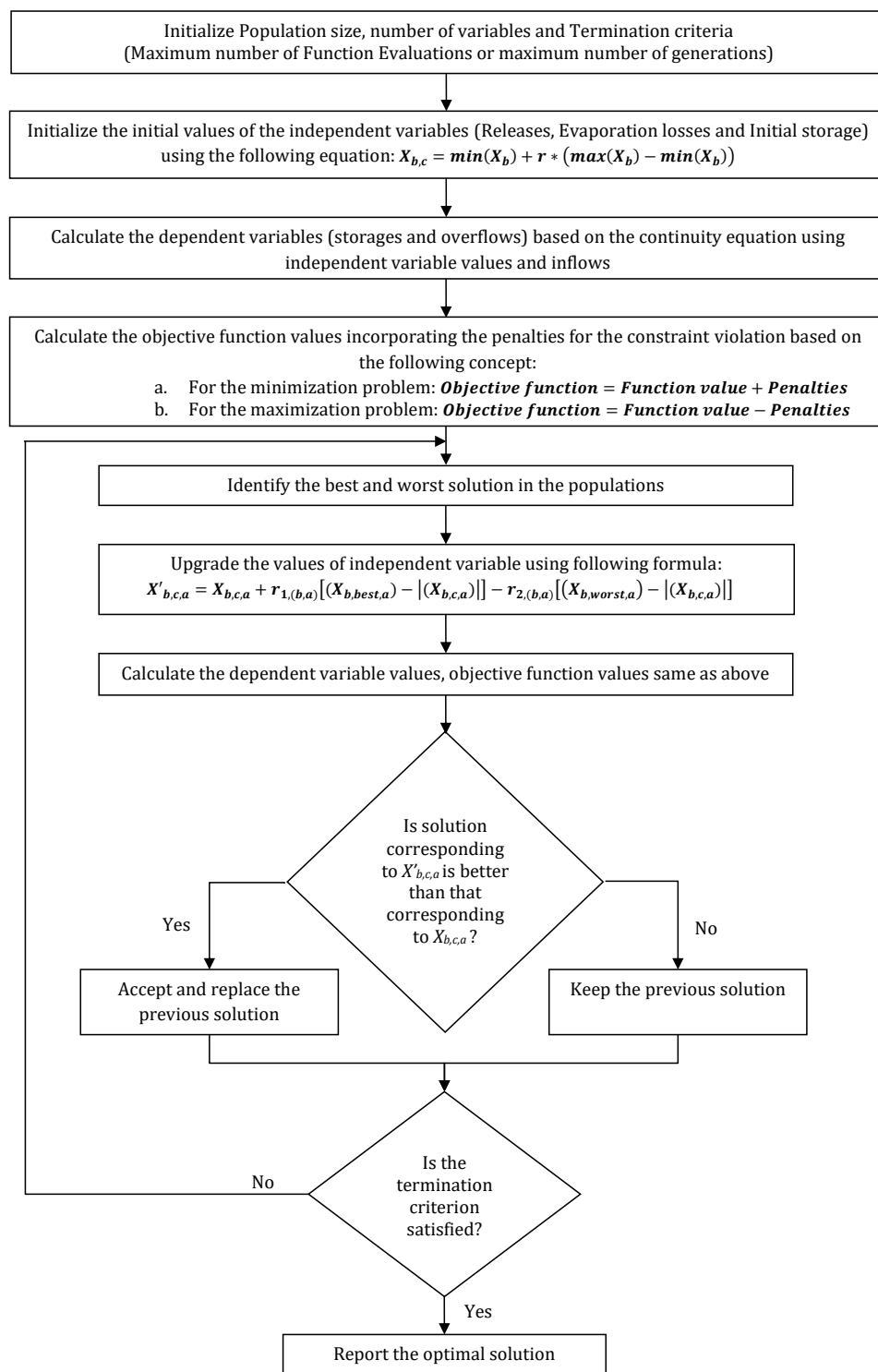


Figure 1. Flowchart of Jaya Algorithm (derived from [28]).

Then, the function value is calculated. A further objective function value is calculated using the function value and penalties for the constraint violation. For the maximization problem, penalties are subtracted and, for the minimization one, penalties are added to the function value to obtain the objective function value according to

1. For the minimization problem: *Objective function* = *function value* + *penalties*,
2. For the maximization problem: *Objective function* = *function values* – *penalties*.

After this, from the objective function values for the ' n ' candidate solutions and for the a^{th} iteration, the best and the worst solutions are selected. For the maximization problem, the solution with the maximum value of the objective function will be the best solution and the one with minimum value will be the worst solution and vice versa. Then, the values of b^{th} variables for c^{th} population and for a^{th} iteration are updated using (2):

$$X'_{b,c,a} = X_{b,c,a} + r_{1,(b,a)}[(X_{b,best,a}) - |X_{b,c,a}|] - r_{2,(b,a)}[(X_{b,worst,a}) - |X_{b,c,a}|], \quad (2)$$

where

- $X'_{b,c,a}$ is the updated value of the variable,
- $X_{b,c,a}$ is the old value of the variable,
- $r_{1,(b,a)}$ and $r_{2,(b,a)}$ are random variable for the b^{th} variable during the a^{th} generation ($r_{1,(b,a)}, r_{2,(b,a)} \in [0, 1]$),
- $X_{b,best,a}$ is the b^{th} variable corresponding to the best candidate solution for a^{th} iteration,
- $X_{b,worst,a}$ is the b^{th} variable corresponding to the worst candidate solution for a^{th} iteration.

The objective function value is calculated for these updated values of the variables. The objective function value corresponding to the old and updated values of the b^{th} variables for the c^{th} population are compared and the best among two is adopted, the worst one being rejected. These best values are treated as an initial set for the next iteration, and the process is repeated until the termination criterion (preset maximum number of iterations) is reached.

The main equation of the working of JA, i.e., the upgrading of variable values is done on the basis of (2) in which two random numbers (r_1 and r_2) are used. Random numbers play a crucial role in upgrading the decision variable, thereby improving the fitness function value. Hence, a random number range can be tested for improvement in convergence. The random number r_1 is associated with the term representing the gap of the present value from the value corresponding to the best solution for particular iteration and similarly the random number r_2 is associated with the term representing the gap of the present value from the value corresponding to the worst solution for particular iteration. In this equation, a positive sign is associated with r_1 and negative sign to r_2 indicating its approach of moving towards the best solution and away from the worst solution, which is like moving towards the victory, the word translated as Jaya in Sanskrit. This is the reason for the name of the algorithm [28]. The way JA works has been illustrated in Figure 1, and has been derived from [28], and updated for the different case studies discussed in the following sub-sections.

2.2. Case Study 1

This section applies JA to a hypothetical Four reservoir system shown in Figure 2 [39], with the aim of checking its potential to derive an operating policy for a multi-reservoir system. The Four reservoir system problem has been investigated by several researchers to test the performance of other algorithms such as GA, PSO, EMPSO [19], and WOA [23].

The system comprises four reservoirs serving the purpose of hydropower generation and irrigation. Hydropower generation is possible through all the reservoirs, while irrigation is possible only by the fourth one. The objective function consists of maximizing the profits obtained from the system during a twelve-hour operating period. The inflow to the reservoir varies every year for real time reservoir operation. It is predetermined for reservoir operation optimization. The inflows to the reservoirs 1 to 4 are 2, 3, 0, and 0 units, respectively, for all time periods. Usually, the releases of a reservoir are bound to be in a range. Minimum releases correspond to minimum water supply to serve the purpose, and maximum releases correspond to the demands of water for that particular period and for a particular purpose.

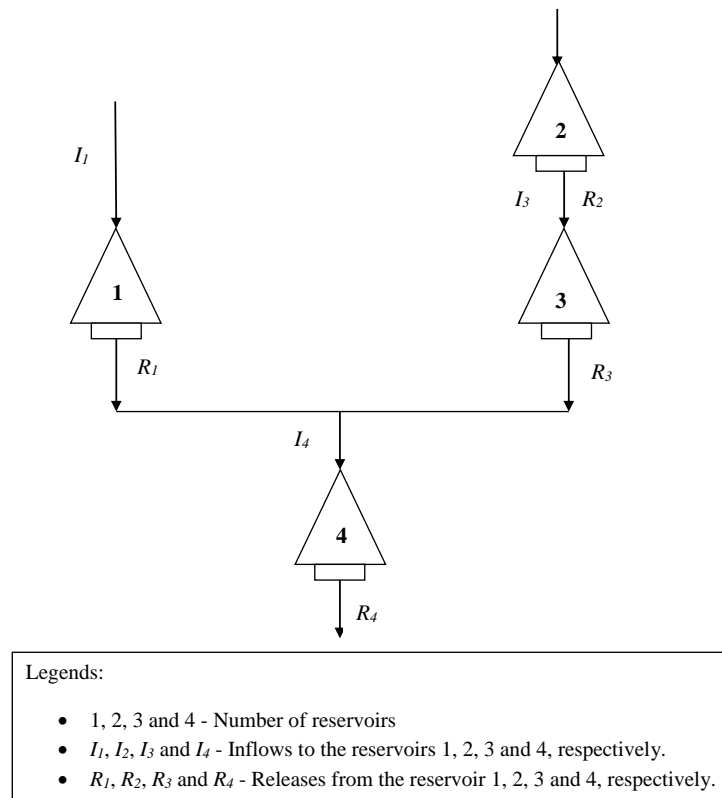


Figure 2. Case Study 1: Four reservoir system [39].

The minimum releases for all reservoirs in the study are 0, and the maximum releases are 3, 4, 4, and 7 units, respectively. The minimum storage is 0 for all reservoirs and the maximum are 10, 10, 10, and 15 units, respectively. Furthermore, the minimum initial storage should be maintained to make the reservoir functional. The end storage for a given time period is the initial storage for the next one. The operational policy of the reservoir must be kept. The initial storage for all reservoirs has the value of 5 units, while it is required to maintain the storage of 5, 5, 5, and 7 units, respectively, at the end of the period. The restriction to the inflows, maximum releases, and storage are pre-defined in the problem [39]. Furthermore, all of the past studies used these constraints and performed comparisons of optimization techniques in the field of reservoir operations in this common platform. The objective function and profit function associated with it can be referred from [39,41] and is discussed hereunder.

The objective function to be maximized (F) is the summation of profit from the hydropower operation from all four reservoirs and the irrigation profit from the fourth reservoir. Mathematically, it can be expressed as in (3):

$$\max F = \sum_{i=1}^4 \sum_{t=1}^{12} b_{i(t)} \cdot R_{i(t)} + \sum_{t=1}^{12} b_{5(t)} \cdot R_{4(t)}, \quad (3)$$

where

- $b_{i(t)}$ is a 4×12 matrix of profit, function associated with all the four reservoirs for hydropower,
- $R_{i(t)}$ represents the releases from the reservoirs $i = 1$ to 4 during the period ' t '. The benefit function associated with the hydropower is as follows:

$$b_{4 \times 12} = \begin{bmatrix} 1.1 & 1 & 1 & 1.2 & 1.8 & 2.5 & 2.2 & 2 & 1.8 & 2.2 & 1.8 & 1.4 \\ 1.4 & 1.1 & 1 & 1 & 1.2 & 1.8 & 2.5 & 2.2 & 2 & 1.8 & 2.2 & 1.8 \\ 1 & 1 & 1.2 & 1.8 & 2.5 & 2.2 & 2 & 1.8 & 2.2 & 1.8 & 1.4 & 1.1 \\ 1 & 1.2 & 1.8 & 2.5 & 2.2 & 2 & 1.8 & 2.2 & 1.8 & 1.4 & 1.1 & 1 \end{bmatrix}. \quad (4)$$

- $b_{5(t)}$ is the benefit associated with the fourth reservoir for irrigation

$$b_{5(t)} = [1.6 \quad 1.7 \quad 1.8 \quad 1.9 \quad 2 \quad 2 \quad 2 \quad 1.9 \quad 1.8 \quad 1.7 \quad 1.6 \quad 1.5]. \quad (5)$$

- $R_{4(t)}$ describes the releases from the fourth reservoir during the period ' t '

The objective function is subject to the continuity and end storage constraints.

The continuity constraints (6) for each reservoir during each operating period ' t ' are mathematically expressed as:

$$S_{i(t+1)} = S_{i(t)} + I_{i(t)} + M \times R_{i(t)}, \quad (6)$$

where

- $S_{i(t+1)}$ is the storage at the beginning of the next time period ' $t + 1$ ' for reservoirs $i = 1$ to 4,
- $S_{i(t)}$ is the storage at the beginning of the time period ' t ' for reservoirs $i = 1$ to 4,
- M is a 4×4 matrix of indices of the reservoir connections

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}. \quad (7)$$

The End Storage constraint (8) is mathematically expressed as:

$$S_{i(13)} \geq d_i, \quad (8)$$

where

- $S_{i(13)}$ is the storage at the beginning of next time period (generally irrigation year) for the i th reservoir,
- d_i represents the target storage at the beginning of the next time period (generally irrigation year),
- $d_i \in \{5, 5, 5, 7\}$ for $i \in \{1, \dots, 4\}$.

The optimization of the Four reservoir system problem is formulated in LP and solved with the help of the LINGO solver. The results obtained from the LP model are used for the evaluation of this approach. In LP, the final storage can be set as a constraint, whereas there is a requirement of a penalty function in case of JA. The penalty function applied is the same as the one applied in other references, and is expressed through (9) and (10):

$$g_i(S_{i(13), d_i}) = 40(S_{i(13)} - d_i)^2 \quad \text{for} \quad (S_{i(13)} - d_i) \leq 0 \quad (9)$$

and

$$g_i(S_{i(13), d_i}) = 0 \quad \text{for} \quad (S_{i(13)} - d_i) > 0, \quad (10)$$

where $g_i(S_{i(13), d_i})$ is the penalty function for the Four reservoir system problem.

Hence, the modified objective function is expressed by (11):

$$\max F = \sum_{i=1}^4 \sum_{t=1}^{12} b_{i(t)} \cdot R_{i(t)} + \sum_{t=1}^{12} b_{5(t)} \cdot R_{4(t)} - \sum_{i=1}^4 g_i(S_{i(13,d_i)}). \quad (11)$$

2.3. Case Study 2

JA has also been applied to an existing reservoir system, the Mula Reservoir of Upper Godavari Basin in India (Figure 3). It is located at Latitude of 19°21'30" and Longitude of 74°34'30", and is 8.04 km upstream of Rahuri, Ahmednagar District, Maharashtra State, India. The project was completed in 1972 on the Mula river, which later meets the Pravara, a major tributary of the Godavari. The catchment area of the project is about 2275 km². It receives an average rainfall of 5080 mm in the hilly region and 508 mm in the Lower Catchment. The gross storage capacity of the reservoir is 735.80 MCM with live storage of 608.45 MCM. The purposes served by this reservoir are irrigation, domestic, and industrial supply. The total area irrigated under this project is 82,920 ha.

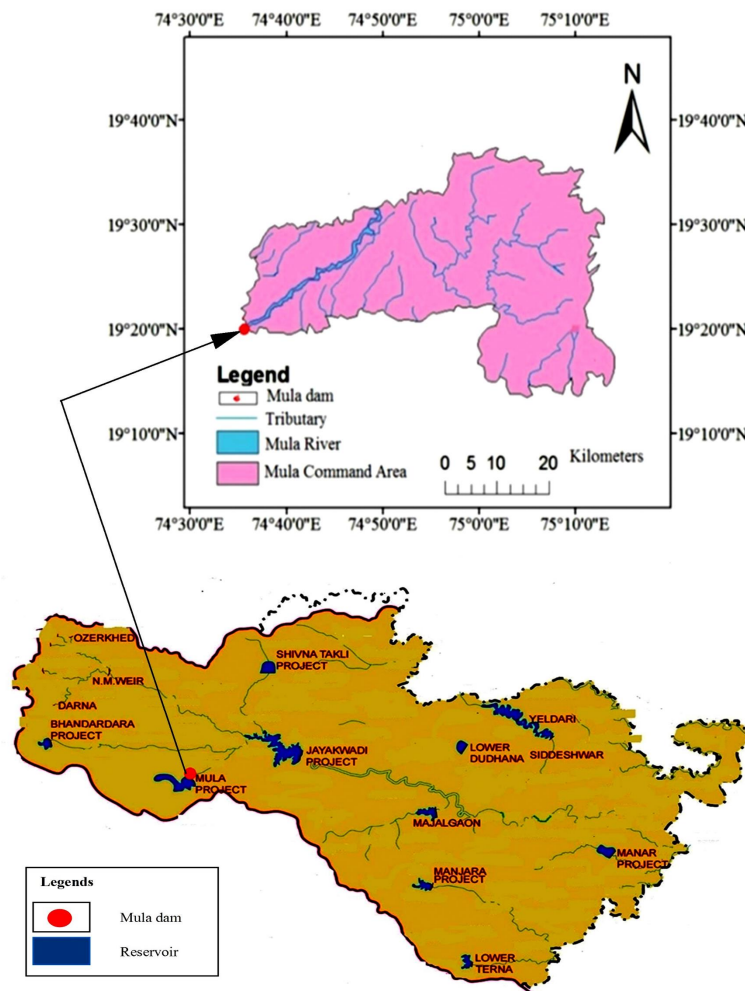


Figure 3. Index map of the Mula Reservoir Project, Upper Godavari Basin, India (Case Study 2).

Daily reservoir data corresponding to 31 years (1984–2015) are collected from the Mula Irrigation, Subdivision Rahuri, Ahmednagar District, Maharashtra, India. Probable inflows are set as input data for the models. The demands to be achieved are set as target releases or demands in the objective function (12) and are calculated considering irrigation, industrial, and urban water requirements

from the downstream side. Target demands are set by combining the information i.e., Net Irrigation Requirement (NIR) for principal crops and actual releases. NIR is used for the months of July to February, and actual releases are used for the months of March to June. This information was collected from the Division Office of Mula Reservoir [42]. In the present study, the target demands are considered the maximum demands. The inflows for December to May are not significant (Figure 4).

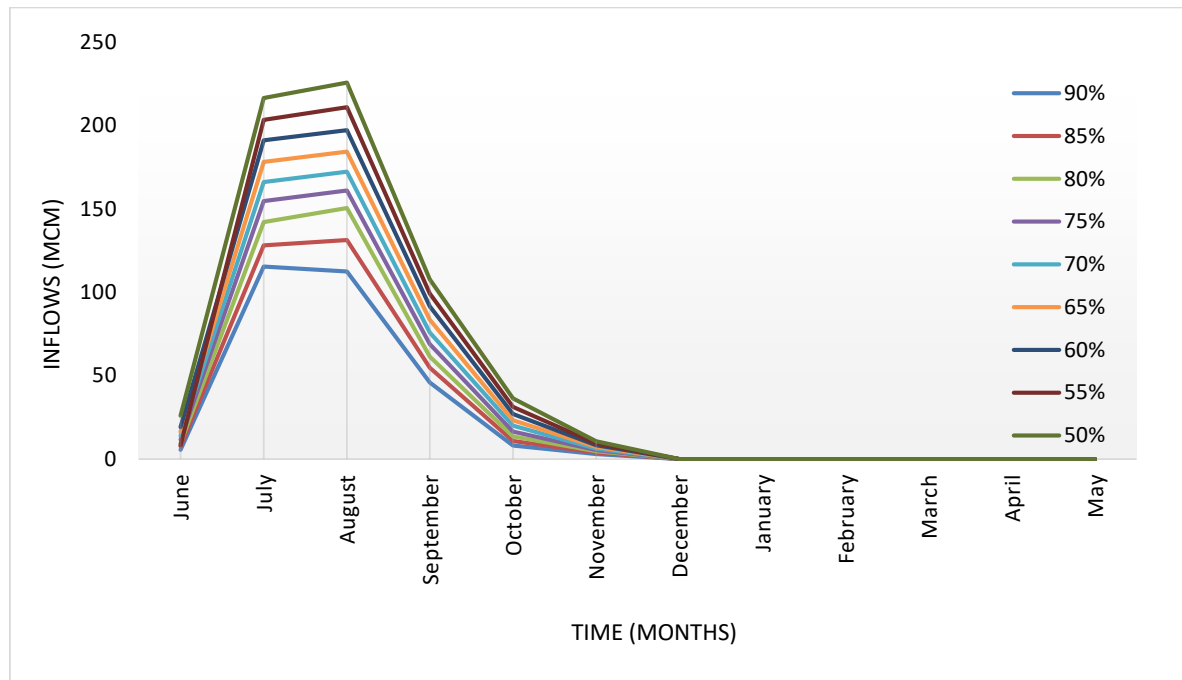


Figure 4. Inflows for different probable condition for Mula reservoir, Upper Godavari basin, India (Case Study 2).

Model Formulation

The objective of the study is the minimization of the sum of squares of deficit of total releases from the demands. Mathematically, it can be expressed as in (12):

$$\min F = \sum_{t=1}^{12} [(R_t - D_t)^2], \quad (12)$$

where

- F is the squared deviation of releases from the target releases,
- t is the time in months ($t \in \{1 = \text{June}, \dots, 12 = \text{May}\}$),
- R_t represents the total releases during period ' t ' in $MCM = R_{(1,t)} + R_{(2,t)} + R_{(3,t)}$,
- $R_{(1,t)}$ is the Left Bank Canal (LBC) releases during period ' t ' in MCM ,
- $R_{(2,t)}$ is the Right Bank Canal (RBC) releases during period ' t ' in MCM ,
- $R_{(3,t)}$ is the industrial and urban releases during period ' t ' in MCM ,
- D_t is the total demand during period ' t ' in MCM .

The objective function is subject to storage continuity, storage limit, canal carrying constraint, constraint for irrigation and industrial demand, overflows constraint, steady state constraint, and non-negativity constraint.

The storage continuity constraint is described as the difference between inflow and outflow and is equal to the change in storage. In this case study, reservoir lift is one of the outflows from the reservoir.

Reservoir lift is the water lifted from the reservoir through pumping. The mathematical expression for continuity constraint for this case is shown in (13):

$$S_{t+1} = S_t + I_t - I'_t - (R_t + E_t + O_t), \quad (13)$$

where

- $S_{(t+1)}$ is the storage of the reservoir at the beginning of time period ' $t + 1$ ' in MCM,
- S_t is the storage of the reservoir at the beginning of time period ' t ' in MCM,
- I_t is the inflow into the reservoir during period ' t ' in MCM,
- I'_t is the reservoir lift (if any) during period ' t ' in MCM,
- E_t is the evaporation loss from the reservoir during period ' t ' in MCM,
- O_t is the overflow from the reservoir during the period ' t ' in MCM.

The storage of a reservoir is not expected to be smaller than its dead storage, and it cannot be larger than the maximum reservoir storage capacity. Thus, the storage limit constraint is expressed by (14):

$$S_{min} \leq S_t \leq S_{max}, \quad (14)$$

where

- S_{min} is the dead pool storage of the reservoir in MCM,
- S_{max} is the reservoir capacity in MCM.

The release through a particular canal should not be larger than its capacity. Hence, the canal carrying capacity constraint is mathematically presented through (15) and (16) for LBC and RBC, respectively,

$$R_{(1,t)} \leq C_{(1,max)} \quad (15)$$

and

$$R_{(2,t)} \leq C_{(2,max)}, \quad (16)$$

where

- $C_{(1,max)}$ is the maximum canal carrying capacity for LBC in MCM,
- $C_{(2,max)}$ is the maximum canal carrying capacity for RBC in MCM.

The releases from the reservoir for purposes like irrigation, domestic, and industrial water supply should neither be smaller than the minimum requirement nor greater than the maximum demand. Hence, a mathematical formulation of the constraint for irrigation and industrial demands is presented in (17):

$$\min D_{(x,t)} \leq R_{(x,t)} \leq \max D_{(x,t)}, \quad (17)$$

where

- $x \in \{1, \dots, 3\}$ and $t \in \{1, \dots, 12\}$,
- $R_{(x,t)}$ is the release from supply ' x ' for time period ' t ' in MCM,
- $D_{(x,t)}$ is the demand for the x^{th} supply for the time period ' t ' in MCM.

Overflow occurs when the maximum storage level is exceeded and, mathematically, the overflow constraint is expressed in (18):

$$O_t \geq S_t + I_t + I'_t - R_t - E_t - S_{max}. \quad (18)$$

The storage level at the end of irrigation year is equal to the storage at the beginning of irrigation year to maintain steady state condition (19) and mathematically it is given as:

$$S_{13} \cong S_1, \quad (19)$$

where

- S_{13} is the storage at the end of irrigation year,
- S_1 is the storage at the beginning of the irrigation year.

All of the variables involved in the model should be positive. Mathematically expressed using (20),

$$I_t, R_t, S_t, E_t, O_t \geq 0. \quad (20)$$

3. Results

The Four reservoir systems is a benchmark problem of reservoir operation optimization that has been used to compare the performance of various strategies proposed in the domain. JA has been first applied to the Four reservoir system (benchmark) to test its applicability to reservoir operation studies. With its success in this benchmark study, it has been further applied to optimize the operation of the Mula reservoir, Upper Godavari Basin, India. The results of the two case studies are explained in the next sections.

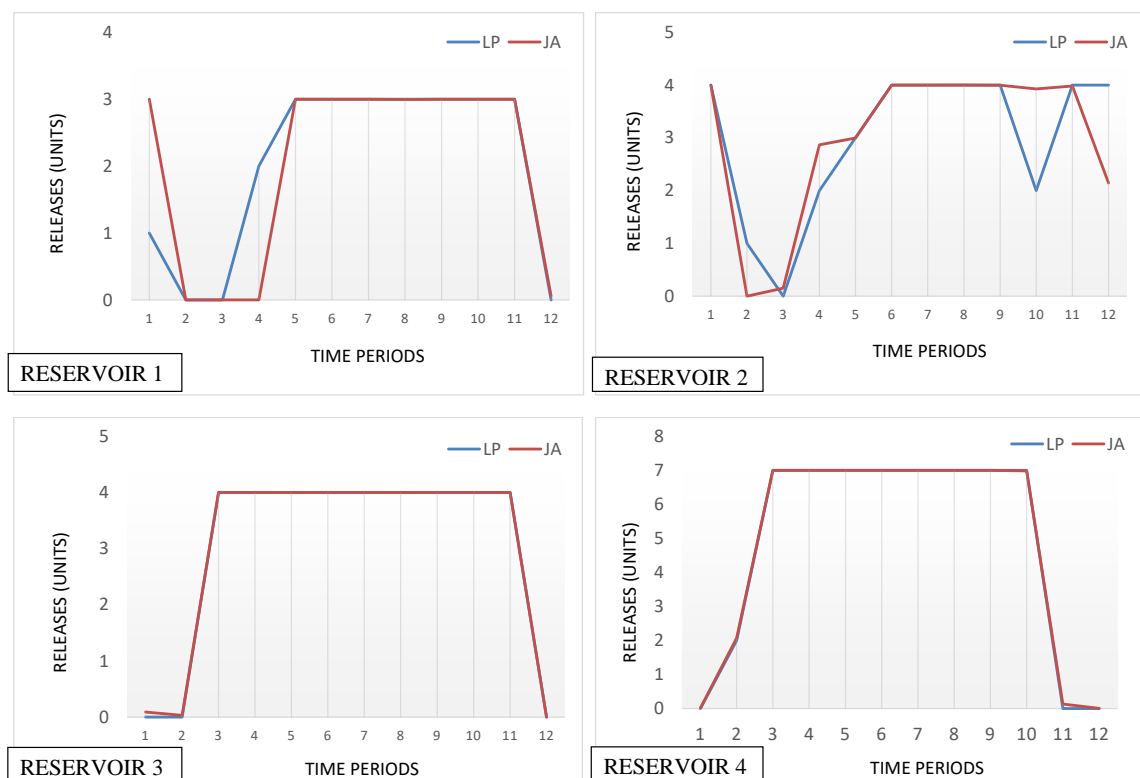
3.1. Hypothetical Four Reservoir System

The JA and LP are applied to this benchmark problem and results obtained are compared with those obtained by means of earlier studies mentioned in the literature. LP is applied to replicate the results mentioned in the literature and to present a comparison graph between the two techniques, with the aim of validating it. When the JA is applied to the benchmark problem, it results in the optimal solution (401.4) for a lower number of FEs when compared with the other evolutionary approaches (Table 1). FEs represent the number of times the objective function value is evaluated and calculated as $FEs = population\ size \times generations$. The optimal solution is obtained at 325,000 FEs for a population size of 150. In most of the cases, the population size plays a significant role in the performance of EAs as it defines the search space. The higher the population size, the greater the number of possible combinations of the variables in the search space. Increasing the population size will also increase the time of computation without any significant improvement in the optimal value. Hence, the population size should be decided with a proper analysis. The population size can be decided on the basis of sensitivity analysis with population sizes of 5, 10, 50, 100, 150, 250, and 500. It has been found that the population size of 150 resulted in the best one, based on the proximity to the objective function value and thus finally the best optimal solution is obtained for a population size of 150. The average of optimal values obtained in 10 runs has been found to be 98.84% at 325,000 FEs of the global solution (401.3) as reported in the literature. The release and storage trajectories obtained from LP and JA models for the Four reservoir system are shown in Figures 5 and 6, respectively. The JA results are not consistent with LP results at all time steps. At some times, JA gives higher releases than LP and vice versa at the remaining time steps, thereby leading to the maximum profit in both aspects. Hence, it can be said that JA is capable of producing optimal results in the reservoir operation optimization problem. JA, when applied [21], produced the optimal value of 401.4 at population size of 50 and 7000 iterations (i.e., $FEs = 50\ population\ size \times 7000\ generations = 350,000$). In the present study, the JA results the same optimal value as that of [21] for lower number of FEs with suitable combination of parameters.

Table 1. Performance of the models applied for hypothetical Four reservoir system problem, i.e., Benchmark problem (Case Study 1).

Source	Model ¹	Best Objective Function Value	Population Size	Function Evaluations Taken
[39]	DPSA	401.30	N.A. ²	N.A.
[41]	DDDP	401.30	N.A.	N.A.
[43]	FDP	399.06	N.A.	N.A.
[19]	GA	401.30	500	2,279,500
	PSO	399.70	500	748,000
	EMPSO	401.30	500	325,400
[23]	WOA	401.30	40	400,000
Present Study	LP	401.30	N.A.	N.A.
	JA	401.40	150	325,000

¹ DPSA—Dynamic Programming with Successive Approximation, DDDP—Discrete Differential Dynamic Programming, FDP—Folded Dynamic Programming, GA— Genetic Algorithm, PSO—Particle Swarm Optimization, EMPSO—Elitist Mutated Particle Swarm Optimization, WOA—Weed Optimization Algorithm, LP—Linear Programming, JA—Jaya Algorithm. ² N.A. stands for ‘Not Applicable’.

**Figure 5.** Release trajectories in Four reservoir system problem (Case Study 1).

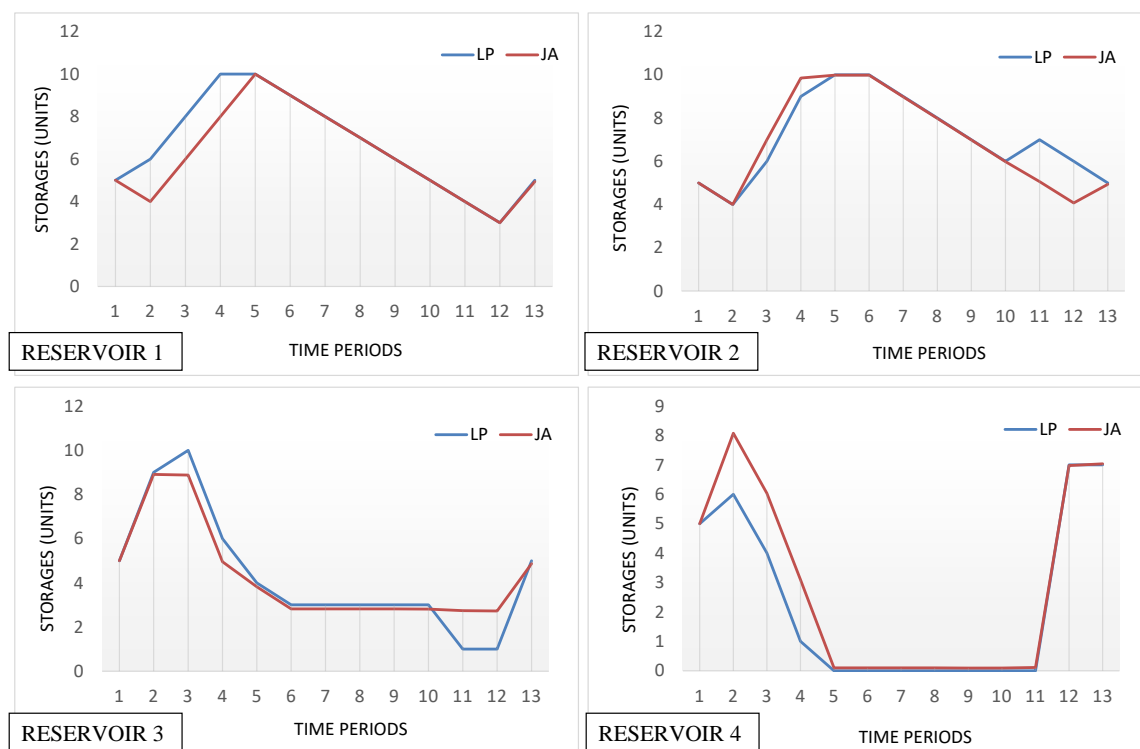


Figure 6. Storage trajectories in Four reservoir system problem (Case Study 1).

The versatility of LP has been proven in the literature as it is the most robust and profoundly used technique over its diverse application in planning, scheduling, resource assignment, and design. However, in contrast, LP is inflexible specifically in the capacitated facility location problem or the simple terms in local search where it is approximated within a constant factor highlighting the local optima. As anticipated, taking into account the success of JA when applied to the benchmark problem, it has also been applied to optimize the operation of the Mula reservoir, Upper Godavari Basin, India.

3.2. Mula Reservoir

In the reservoir operation, the design inflows are generally 75% of dependable inflows. Real-time studies are random; therefore, to incorporate randomness, the operation for the case study is done by using 75% probable inflows. The probable inflows are random than the dependable inflows. JA has been developed with the help of MATLAB for population size of 30 after performing sensitivity analysis for population sizes of 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. The optimal value is achieved for 150,000 FEs; the square root of the objective function achieved is 218.88 MCM. The convergence graph for square root of the objective function value with respect to the number of generation for the JA model for Mula reservoir corresponding to the optimal solution is represented in Figure 7. The model was permitted to run for 5000 generations, and it converges at 2141 generations such as presented through the convergence graph (Figure 7). The rate of improvement decreases as the number of generation increases that can be seen in Figure 7. The model has been further developed for different probable inflows ranging from 50% to 90% with an interval of 5%, considering all the scenarios from an excess of water to a scarce period. The deficit of releases from their target releases for these scenarios is shown in Figures 8 and 9, respectively. From these figures, it can be noted that the deficit increases with the increase in the percentage of probable inflow. It also means that the deficit is increasing with the decrease in inflow, which implies the model has been applied appropriately. Further simulation studies are carried out for nine time periods i.e., 1993–94, 2009–10, 2012–13, 1986–87, 2003–04, 1992–93, 1987–88, 1995–96 and 1995–96 corresponding to 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85%, and 90%

inflows, respectively. Simulation is done for these nine years. Years corresponding to 85% and 90% inflow are the same as the annual inflow among the data of 31 years are closest to the annual 85% and 90% probable inflow corresponding to these years. Real inflows have been selected by annual inflow values among the complete data set nearest to the respective percentage of probable inflow value. As seen in Figure 9, the deficit is the same for 85% and 90% real inflows, since both deficits belong to the same year (1995–96). Allocation of resources for design (75%) real inflow (1992–93) has been simulated and is shown in Figure 10. JA has been observed to result in better allocation in comparison to the existing policy of Mula reservoir (Figure 10). In the years 1992–93, the annual inflow was 391.912 MCM for which utilization of resources by JA and existing policy are 349.30 and 286.39 MCM, respectively, which implies optimal utilization of the resources. Releases obtained from JA are found to be more in comparison to the existing policy for the lean period (March to May), which signifies better allocation through JA. It can be said that JA will be useful for reservoir operation of Mula reservoir as it is utilizing the resources optimally. Hence, JA can be adopted to derive the optimal operation policy for the Mula reservoir for its sustainable use.

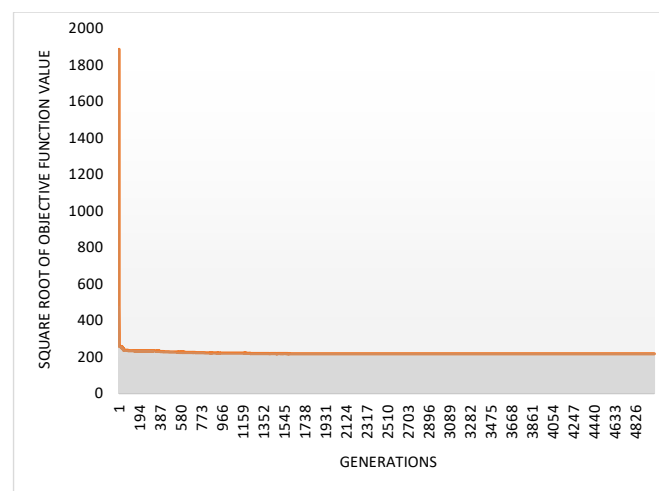


Figure 7. Plot of square root of objective function value with generations.

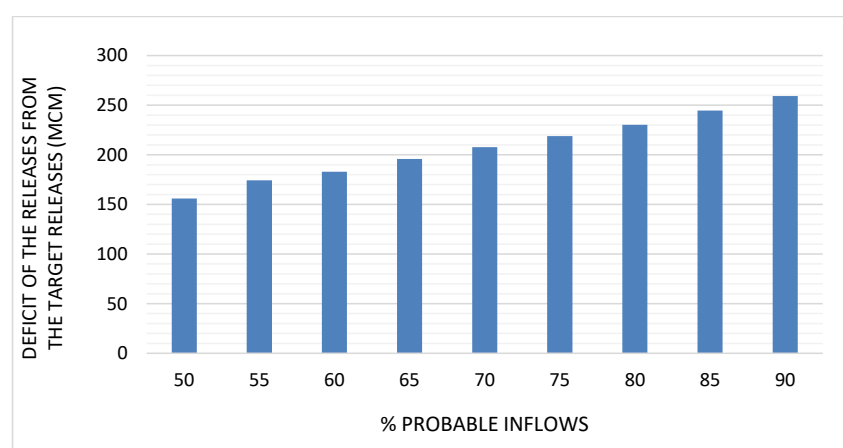


Figure 8. Deficit of releases from the target releases for different percentage probable inflows (Case Study 2).

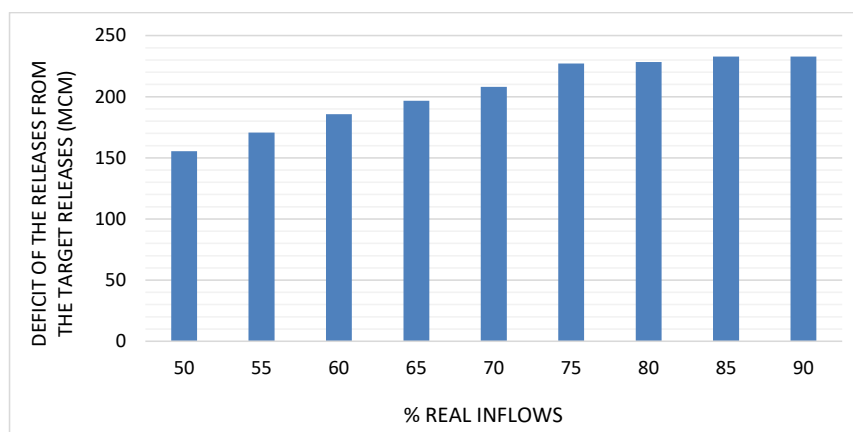


Figure 9. Deficit of releases from the target releases for different percentage real inflows (Case Study 2).

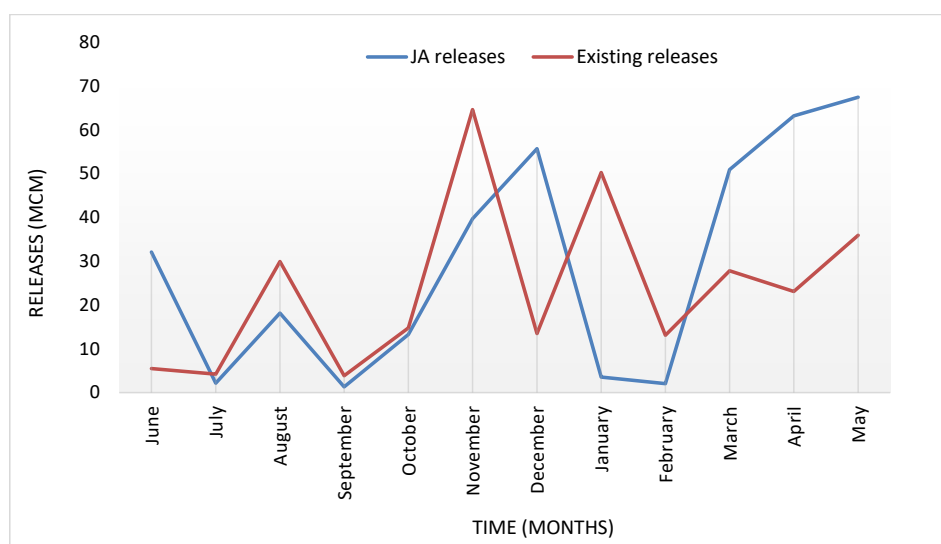


Figure 10. Release trajectories for 75% real inflow (1992–93) for Mula reservoir, Upper Godavari basin, India (Case Study 2).

Furthermore, the computational complexity of JA has been examined in terms of big-O notations with varying number of iterations and compared with the GA algorithm. This comparison has been carried out with the help of the GuessComp tool [44], which estimates the computational complexity empirically. This tool uses the ‘Leave one out-mean square error (LOO-MSE)’ approach to estimate the best fit of complexity using multiple and increasing-sizes samples from provided data [45]. Figures 11 and 12 show the nature of computational complexity, which represents the relation between computation time and the number of iterations. Both methods show the linear complexity in terms of big-O notations. This comparison signifies that both methods behave identically in the perspective of computational complexity.

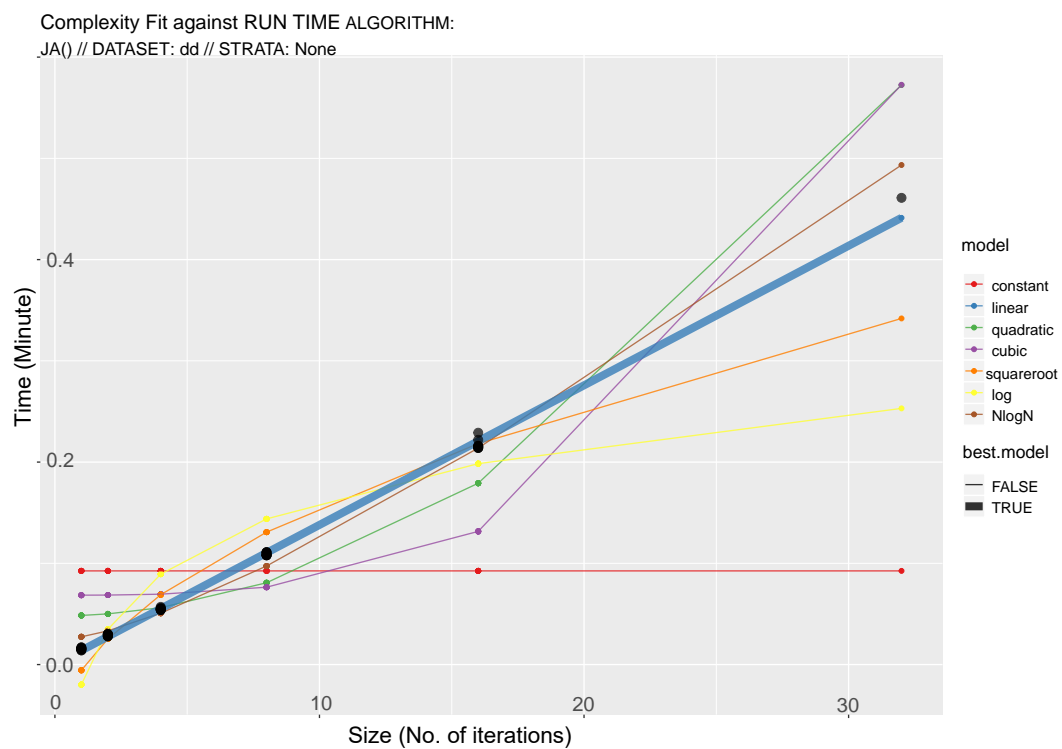


Figure 11. Computational complexity fitted for Jaya algorithm with GuessComp tool [45].

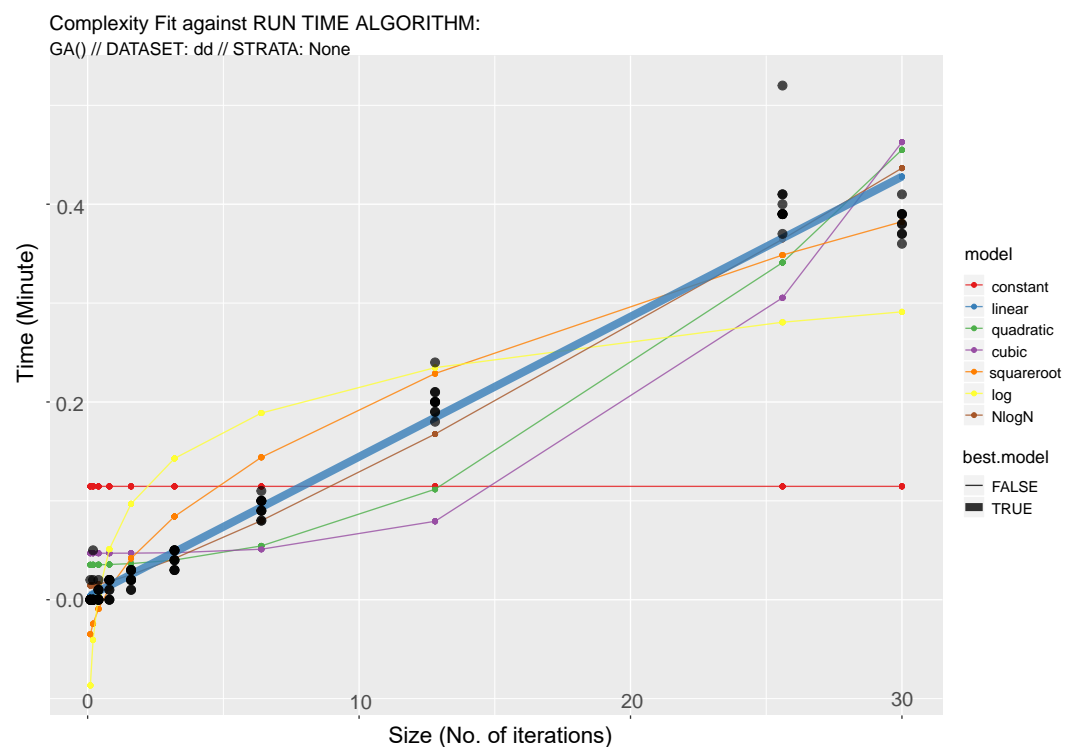


Figure 12. Computational complexity fitted for Genetic algorithm with GuessComp tool [45].

4. Conclusions

In this paper, JA has been implemented and applied to two case studies, with the aim of contributing to the optimal use and sustenance of reservoir systems. It has been checked that, when JA

has been applied to a hypothetical four-reservoir system, it has provided the optimal resource allocation results for the lowest FE values. JA has been successfully applied to the mentioned four-reservoir system problem with faster convergence than other algorithms. With the increasing need for water resources and the requirements associated, reservoir operation optimization is of great importance. In this study, an attempt has been made to have better releases in comparison to the existing policy releases; therefore, JA has been applied to the reservoir operation optimization of Mula Reservoir, an existing installation in the Upper Godavari Basin, India. It was optimized for different probable inflows and was simulated for a scarce to wet period. The releases thus obtained from the simulation corresponding to an average inflow have been compared to those from existing ones. As a result, better releases were obtained with the help of the JA model in terms of maximum utilization of resources and better allocation of the resources in terms of its need during the lean period. Hence, from this study, it can be concluded that JA was successful in achieving better operational releases for the Mula Reservoir. Therefore, it can be summarized that JA can be further explored for other case studies and for the more complex problem of multi-reservoir systems.

Author Contributions: Conceptualization, V.P., A.D.G., and A.B.M.; Methodology, V.P., A.D.G., and A.B.M.; Software, V.P. and A.B.M.; Validation, N.D.B., A.E.F.L., and V.P.; Formal Analysis, V.P.; Investigation, V.P., A.D.G., and A.B.M.; Resources, V.P., A.D.G., and A.B.M.; Data Curation, V.P., A.D.G., and A.B.M.; Writing—Original Draft Preparation, V.P., A.D.G., and A.B.M.; Writing—Review and Editing, N.D.B., A.E.F.L., and V.P.; Visualization, N.D.B. and V.P.; Supervision, A.E.F.L., A.D.G., A.B.M., and N.D.B.; Project Administration, V.P., A.B.M., and A.D.G. All authors have read and agree to the published version of the manuscript.

Funding: This research was funded by Science and Engineering Research Board: ECR/2016/001409.

Acknowledgments: The authors are thankful to the Science and Engineering Research Board (SERB), Department of Science and Technology, Government of India, for providing financial support (ECR/2016/001409). The cooperation extended by Mula Irrigation Sub-division Rahuri, Ahmednagar, Maharashtra, India and Command Area Development Authority (CADA), Nashik, Maharashtra, India, for providing the necessary data for the present study, is also gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
AFSA	Artificial Fish Swarm Algorithm
AI	Artificial Intelligence
CA	Crow Algorithm
CADA	Command Area Development Authority
CSSA	Charged System Search Algorithm
DDDP	Discrete Differential Dynamic Programming
DDP	Differential Dynamic Programming
DE	Differential Evolution
DP	Dynamic Programming
DPSA	Dynamic Programming with Successive Approximation
EA	Evolutionary Algorithm
EMPSO	Elitist Mutated Particle Swarm Optimization
FA	Firefly Algorithm
FDP	Folded Dynamic Programming
FEs	Function Evaluations
GA	Genetic Algorithm
GEA	Gradient Evolution Algorithm
HA	Hybrid Algorithm
HS	Harmony Search
HBMO	Honey Bee Mating Optimization
JA	Jaya Algorithm
Kh.	Kharif

Kh. Hy.	Kharif Hybrid
LBC	Left Bank Canal
LINGO	Language for Interactive General Optimization
LP	Linear Programming
MATLAB	Matrix Laboratory
MAX	Maximization
MCM	Million Cubic Metre
MHLLBC	Mula High Level Left Bank Canal
MHLRBC	Mula High Level Right Bank Canal
MIN	Minimization
MLBC	Mula Left Bank Canal
MRBC	Mula Right Bank Canal
NIR	Net Irrigation Requirement
NLP	Non-Linear Programming
PBC	Pathardi Branch Canal
PSO	Particle Swarm Optimization
Rb.	Rabi
Rb. Hy.	Rabi Hybrid
RBC	Right Bank Canal
SA	Shark Algorithm
TLBO	Teaching Learning Based Optimization
WCA	Water Cycle Algorithm
WOA	Weed Optimization Algorithm
WSA	Wolf Search Algorithm

References

1. Tian, J.; Guo, S.; Liu, D.; Pan, Z.; Hong, X. A Fair Approach for Multi-Objective Water Resources Allocation. *Water Resour. Manag.* **2019**, *33*, 3633–3653. [\[CrossRef\]](#)
2. Ahmad, A.; El-Shafie, A.; Razali, S.F.M.; Mohamad, Z.S. Reservoir optimization in water resources: A review. *Water Resour. Manag.* **2014**, *28*, 3391–3405. [\[CrossRef\]](#)
3. Sreenivasan, K.; Vedula, S. Reservoir operation for hydropower optimization: A chance-constrained approach. *Sadhana* **1996**, *21*, 503–510. [\[CrossRef\]](#)
4. Arunkumar, R.; Jothiprakash, V. Optimal reservoir operation for hydropower generation using nonlinear programming model. *J. Inst. Eng. India Ser. A* **2012**, *93*, 111–120. [\[CrossRef\]](#)
5. Mousavi, S.; Ponnambalam, K.; Karray, F. Reservoir operation using a dynamic programming fuzzy rule—Based approach. *Water Resour. Manag.* **2005**, *19*, 655–672. [\[CrossRef\]](#)
6. Jalali, M.; Afshar, A.; Marino, M. Reservoir operation by ant colony optimization algorithms. *Iran. J. Sci. Technol. Trans. B Eng.* **2006**, *30*, 107–117.
7. Bozorg-Haddad, O.; Karimirad, I.; Seifollahi-Aghmiuni, S.; Loáiciga, H.A. Development and application of the bat algorithm for optimizing the operation of reservoir systems. *J. Water Resour. Plan. Manag.* **2014**, *141*, 04014097. [\[CrossRef\]](#)
8. Haddad, O.B.; Hosseini-Moghari, S.M.; Loáiciga, H.A. Biogeography-based optimization algorithm for optimal operation of reservoir systems. *J. Water Resour. Plan. Manag.* **2015**, *142*, 04015034. [\[CrossRef\]](#)
9. Asadieh, B.; Afshar, A. Optimization of Water-Supply and Hydropower Reservoir Operation Using the Charged System Search Algorithm. *Hydrology* **2019**, *6*, 5. [\[CrossRef\]](#)
10. Ehteram, M.; Binti Koting, S.; Afan, H.A.; Mohd, N.S.; Malek, M.A.; Ahmed, A.N.; El-shafie, A.H.; Onn, C.C.; Lai, S.H.; El-Shafie, A. New evolutionary algorithm for optimizing hydropower generation considering multireservoir systems. *Appl. Sci.* **2019**, *9*, 2280. [\[CrossRef\]](#)
11. Vasan, A.; Raju, K.S. Optimal reservoir operation using differential evolution. In Proceedings of the International Conference on Hydraulic Engineering: Research and Practice (ICON-HERP), Roorkee, India, 26–28 October 2004.

12. Garousi-Nejad, I.; Bozorg-Haddad, O.; Loáiciga, H.A.; Mariño, M.A. Application of the firefly algorithm to optimal operation of reservoirs with the purpose of irrigation supply and hydropower production. *J. Irrig. Drain. Eng.* **2016**, *142*, 04016041. [\[CrossRef\]](#)
13. Jothiprakash, V.; Shanthi, G. Single reservoir operating policies using genetic algorithm. *Water Resour. Manag.* **2006**, *20*, 917–929. [\[CrossRef\]](#)
14. Samadi-koucheksaraee, A.; Ahmadianfar, I.; Bozorg-Haddad, O.; Asghari-pari, S.A. Gradient Evolution Optimization Algorithm to Optimize Reservoir Operation Systems. *Water Resour. Manag.* **2019**, *33*, 603–625. [\[CrossRef\]](#)
15. Bashiri-Atrabi, H.; Qaderi, K.; Rheinheimer, D.E.; Sharifi, E. Application of harmony search algorithm to reservoir operation optimization. *Water Resour. Manag.* **2015**, *29*, 5729–5748. [\[CrossRef\]](#)
16. Afshar, A.; Haddad, O.B.; Mariño, M.A.; Adams, B.J. Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. *J. Frankl. Inst.* **2007**, *344*, 452–462. [\[CrossRef\]](#)
17. Yaseen, Z.M.; Karami, H.; Ehteram, M.; Mohd, N.S.; Mousavi, S.F.; Hin, L.S.; Kisi, O.; Farzin, S.; Kim, S.; El-Shafie, A. Optimization of reservoir operation using new hybrid algorithm. *KSCE J. Civ. Eng.* **2018**, *22*, 4668–4680. [\[CrossRef\]](#)
18. Paliwal, V.; Ghare, A.D.; Mirajkar, A.B. Single-Reservoir operation optimization using Jaya Algorithm for Jayakwadi-1 dam, India. In Proceedings of the E-proceedings of the IAHR World Congress, Kuala Lumpur, Malaysia, 13–18 August 2017; pp. 1–8.
19. Nagesh Kumar, D.; Janga Reddy, M. Multipurpose reservoir operation using particle swarm optimization. *J. Water Resour. Plan. Manag.* **2007**, *133*, 192–201. [\[CrossRef\]](#)
20. Ehteram, M.; Allawi, M.F.; Karami, H.; Mousavi, S.F.; Emami, M.; Ahmed, E.S.; Farzin, S. Optimization of chain-reservoirs' operation with a new approach in artificial intelligence. *Water Resour. Manag.* **2017**, *31*, 2085–2104. [\[CrossRef\]](#)
21. Kumar, V.; Yadav, S. Optimization of reservoir operation with a new approach in evolutionary computation using TLBO algorithm and Jaya algorithm. *Water Resour. Manag.* **2018**, *32*, 4375–4391. [\[CrossRef\]](#)
22. Haddad, O.B.; Moravej, M.; Loáiciga, H.A. Application of the water cycle algorithm to the optimal operation of reservoir systems. *J. Irrig. Drain. Eng.* **2014**, *141*, 04014064. [\[CrossRef\]](#)
23. Asgari, H.R.; Bozorg Haddad, O.; Pazoki, M.; Loáiciga, H.A. Weed optimization algorithm for optimal reservoir operation. *J. Irrig. Drain. Eng.* **2015**, *142*, 04015055. [\[CrossRef\]](#)
24. Ahmadebrahimpour, E. Optimal operation of reservoir systems using the Wolf Search Algorithm (WSA). *Water Supply* **2019**, *19*, 1396–1404. [\[CrossRef\]](#)
25. Labadie, J.W. Optimal operation of multireservoir systems: State-of-the-art review. *J. Water Resour. Plan. Manag.* **2004**, *130*, 93–111. [\[CrossRef\]](#)
26. Hossain, M.S.; El-Shafie, A. Intelligent systems in optimizing reservoir operation policy: A review. *Water Resour. Manag.* **2013**, *27*, 3387–3407. [\[CrossRef\]](#)
27. Rao, R.V.; Patel, V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Sci. Iran.* **2013**, *20*, 710–720. [\[CrossRef\]](#)
28. Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34.
29. Das, S.R.; Mishra, D.; Rout, M. A hybridized ELM-Jaya forecasting model for currency exchange prediction. *J. King Saud Univ. Comput. Inf. Sci.* **2017**. [\[CrossRef\]](#)
30. Rao, R.V.; Savsani, V.J.; Vakharia, D. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput.-Aided Des.* **2011**, *43*, 303–315. [\[CrossRef\]](#)
31. Degertekin, S.; Lamberti, L.; Ugur, I. Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. *Appl. Soft Comput.* **2018**, *70*, 903–928. [\[CrossRef\]](#)
32. Zhang, Y.; Yang, X.; Cattani, C.; Rao, R.; Wang, S.; Phillips, P. Tea category identification using a novel fractional Fourier entropy and Jaya algorithm. *Entropy* **2016**, *18*, 77. [\[CrossRef\]](#)
33. Rao, R.V.; Waghmare, G. A new optimization algorithm for solving complex constrained design optimization problems. *Eng. Optim.* **2017**, *49*, 60–83. [\[CrossRef\]](#)
34. Rao, R.V.; Saroj, A. Economic optimization of shell-and-tube heat exchanger using Jaya algorithm with maintenance consideration. *Appl. Therm. Eng.* **2017**, *116*, 473–487. [\[CrossRef\]](#)

35. Nayak, D.R.; Dash, R.; Majhi, B. Development of pathological brain detection system using Jaya optimized improved extreme learning machine and orthogonal ripplelet-II transform. *Multimed. Tools Appl.* **2018**, *77*, 22705–22733. [\[CrossRef\]](#)
36. Varade, S.; Patel, J.N. Determination of Optimum Cropping Pattern Using Advanced Optimization Algorithms. *J. Hydrol. Eng.* **2018**, *23*, 05018010. [\[CrossRef\]](#)
37. Ahmadianfar, I.; Samadi-Koucheksaraee, A.; Bozorg-Haddad, O. Extracting optimal policies of hydropower multi-reservoir systems utilizing enhanced differential evolution algorithm. *Water Resour. Manag.* **2017**, *31*, 4375–4397. [\[CrossRef\]](#)
38. Jain, S.K. *Introduction to Reservoir Operation*; Technical Report; National Institute of Hydrology: Roorkee, India, 2019.
39. Larson, R.E. *State Increment Dynamic Programming*; American Elsevier Publishing: New York, NY, USA, 1968.
40. Jaya-Algorithm. Available online: <https://sites.google.com/site/jayaalgorithm/home>. (accessed on 23 July 2019).
41. Heidari, M.; Chow, V.T.; Kokotović, P.V.; Meredith, D.D. Discrete differential dynamic programming approach to water resources systems optimization. *Water Resour. Res.* **1971**, *7*, 273–282. [\[CrossRef\]](#)
42. Command Area Development Authority (CADA). *Mula Reservoir Project Mula Project 5th Revised Project Volume 1*; Technical Report; Godavari Marathwada Irrigation Development Corporation: Aurangabad, India, 1999.
43. Kumar, D.N.; Baliarsingh, F. Folded dynamic programming for optimal operation of multireservoir system. *Water Resour. Manag.* **2003**, *17*, 337–353. [\[CrossRef\]](#)
44. Agenis, M.; Bokde, N. GuessCompX: Empirically Estimates Algorithm Complexity; R package version 1.0.3. 2019. Available online: <https://CRAN.R-project.org/package=GuessCompX> (accessed on 20 December 2019).
45. Agenis-Nevers, M.; Bokde, N.D.; Yaseen, Z.M.; Shende, M. GuessCompX: An empirical complexity estimation in R. *arXiv* **2019**, arXiv:1911.01420.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).