Optimal hydropower operation of multi-reservoir systems: hybrid cellular automata-simulated annealing approach

Mohamad Azizipour, Ali Sattari, Mohammad Hadi Afshar, Erfan Goharian and Samuel Sandoval Solis

ABSTRACT

Hydropower operation of multi-reservoir systems is very difficult to solve mostly due to their nonlinear, nonconvex and large-scale nature. While conventional methods are long known to be incapable of solving these types of problems, evolutionary algorithms are shown to successfully handle the complexity of these problems at the expense of very large computational cost, particularly when population-based methods are used. A novel hybrid cellular automata-simulated annealing (CA-SA) method is proposed in this study which avoids the shortcomings of the existing conventional and evolutionary methods for the optimal hydropower operation of multi-reservoir systems. The start and the end instances of time at each operation period is considered as the CA cells with the reservoir storages at these instances are taken as the cell state which leads to a cell neighborhood defined by the two adjacent periods. The local updating rule of the proposed CA is derived by projecting the objective function and the constraints of the original problem on the cell neighborhoods represented by an optimization sub-problem with the number of decision variables equal to the number of reservoirs in the system. These sub-problems are subsequently solved by a modified simulated annealing approach to finding the updated values of the cell states. Once all the cells are covered, the cell states are updated and the process is iterated until the convergence is achieved. The proposed method is first used for hydropower operation of two well-known benchmark problems, namely the well-known four- and ten-reservoir problems. The results are compared with the existing results obtained from cellular automata. Genetic algorithm and particle swarm optimization indicating that the proposed method is much more efficient than existing algorithms. The proposed method is then applied for long-term hydropower operation of a real-world three-reservoir system in the USA, and the results are presented and compared with the existing results.

Key words | cellular automata, genetic algorithm, hydropower, multi-reservoir, particle swarm optimization, simulated annealing

HIGHLIGHTS

- An innovative hybrid model is developed to optimally operate multi-reservoir hydropower systems.
- The proposed method is insensitive to the initial guess, for finding an optimal solution.
- The efficiency of simulated annealing is modified.
- The results are compared with the existing results in the literature.

INTRODUCTION

Environmental issues associated with fossil fuel energy production have led to increased production of clean and renewable energy resources (Azizipour & Afshar 2018). As a result, hydropower has become one of the most important resources to produce electricity all around the world. With increasing energy demand and decreasing water supply (Babovic 1996), optimal hydropower operation of multi-reservoir systems is becoming more important and more complicated than before due to many practical limitations faced in their operations. Hydropower multi-reservoir operation is a large-scale, nonconvex and highly nonlinear problem, the solution of which has great importance because of its social and economic benefits. Therefore, it is meaningful to search for new algorithms which can efficiently and effectively solve large-scale real-world reservoir operation problems (Ming et al. 2015).

In the last decades, various optimization techniques have been devised and used for optimal hydropower operation of reservoirs in different forms. Traditional optimization methods, such as linear programming (LP) (Ellis & ReVelle 1988; Yoo 2009), nonlinear programming (NLP) (Zambelli et al. 2009; Arunkumar & Jothiprakash 2012; Shenc et al. 2014) and dynamic programming (DP) (Allen & Bridgeman 1986; Labadie 2004; Yurtal et al. 2005; Maceira & Damázio 2006; Li et al. 2012; Zhao et al. 2012; Zhang et al. 2013) have been extensively and successfully used to solve different forms of reservoir operation problems. These methods, however, suffer from some drawbacks in particular when large-scale multi-reservoir operation problem is to be solved. Using LPs, for solving such problems requires some sort of linearization which affects the optimality of the final solution. NLP methods often need some information about derivatives of the objective function and/or constraints which are not available for discontinuous functions. Furthermore, they are well known to be trapped in local optima, especially for nonconvex optimization problems. DP originally being a discrete optimization method requires discretization when solving continuous problems, such as reservoir operation problem considered here, affecting the accuracy of final solution. More importantly, the computational effort of DP grows exponentially with the size of the problem, known as ‘curse of dimensionality’, which limits the application of DP for large-scale real-world reservoir operation problems.

To overcome the shortcomings of common traditional optimization methods, evolutionary algorithms (EAs) have been widely used to solve water resources management problems including hydropower reservoir operation problems. EAs use some strategies, mostly stochastic, to avoid local optima without requiring any information about derivatives of the objective function and/or constraints. Among EAs, genetic algorithm (GA) is the first applied to various forms of reservoir operation problems (Oliveira & Loucks 1997; Wardlaw & Sharif 1999; Chen 2003; Cinar et al. 2010; Tayebiyan et al. 2016). Ant colony optimization (ACO) (Kumar & Reddy 2006; Jalali et al. 2006; Moinei & Afshar 2013), particle swarm optimization (PSO) (Cheng et al. 2009; Afshar 2013; Kiruthiga & Amudha 2016; Zhang et al. 2016), simulated annealing algorithms (Teegavarapu & Simonovic 2002; Tospornsampan et al. 2005; Kangrang et al. 2010), honey-bee mating optimization (HBMO) algorithm (Afshar et al. 2007; Niknam et al. 2011), differential evolution algorithm (Lu et al. 2010; Li & Zuo 2012), biogeography-based optimization algorithm (Bozorg-Haddad et al. 2016), genetic programming (Fallah-Mehdipour et al. 2013; Akbari-Alashti et al. 2014, 2015), water cycle algorithm (Bozorg-Haddad et al. 2015b), bat algorithm (Bozorg-Haddad et al. 2015a), firefly algorithm (Garousi-Nejad et al. 2016a, 2016b) and weed optimization algorithm (WOA) (Asgari et al. 2015; Azizipour et al. 2016) have also been used to solve optimal reservoir operation problems. While EAs are shown to be more effective than conventional methods in locating optimal or near-optimal solutions, they suffer from requiring higher computational effort compared to conventional methods in particular when population-based methods are used to solve the problems.

Hybrid methods in which an evolutionary algorithm is coupled with another optimization algorithm are also proved to be successful in solving complex optimization problems. As the first application of hybrid methods in water resources management, Cai et al. (2001) used a combined GA and LP approach for solving nonlinear water
management models. Reis et al. (2005) employed the same approach for multi-reservoir operation planning. Cheng et al. (2007) presented a Chaos GA developed by hybridizing GA with a chaos optimization algorithm for monthly hydropower operation of a reservoir and compared the results with those of DP and standard GA concluding that CGA was more efficient than DP and standard GA. A hybrid SA-GA algorithm was proposed by Chiu et al. (2007) for optimal reservoir operation, indicating that the hybrid algorithm was superior to the GA alone. An improved GA-SA method was developed for the optimization of multiple reservoirs operation by Li & Wei (2008). Afshar et al. (2010) used a hybrid two-stage GA-LP algorithm for optimal design and operation of a nonlinear, nonconvex and large-scale cyclic storage problem. Mo et al. (2013) solved the short-term hydro generation scheduling problem by a hybrid algorithm based on multi-colony ant system (MACS) and differential evolution (DE). A hybrid GA-NLP approach was developed by Hidalgo et al. (2014) for short-term scheduling of hydropower systems.

Recently, cellular automata (CA) has been used as an efficient and effective optimization algorithm to solve reservoir hydropower operation problems. Afshar & Shahidi (2009) proposed a CA approach for optimal water supply and hydropower operation of a single reservoir and compared the results with those obtained by GA, PSO and ACO, indicating the superiority of the proposed CA to the existing evolutionary algorithms. Afshar (2013) extended the CA of Afshar & Shahidi (2009) to solve multi-reservoir hydropower reservoir operation problems by hybridizing CA with NLP method and concluded that the resulting CA-NLP method was more efficient and effective than GA and PSO for the case examples considered. The CA-NLP method used conventional gradient-based NLP methods in the solution process and, therefore, was theoretically prone to suffer from the common drawback of getting trapped in local optima. This, in fact, was vindicated by the reported results where some of the results produced by the proposed CA-NLP method was inferior to those of GA while being much more efficient than GA.

In this paper, a hybrid cellular automata and simulated annealing algorithm (CA-SA) is proposed for the effective and efficient optimal solution of hydropower operation of multi-reservoir systems fully exploiting the super-efficiency of CA method while removing the shortcomings of the CA-NLP method of Afshar (2013). The concept of CA is employed to break the original large-scale problem into a series of small-scale optimization sub-problems defined on the CA cells and their neighborhood. The objective function and constraints of the original problem are projected on the neighborhood of each cell to derive the local updating rule for CA represented by an optimization sub-problem with the number of decision variables equal to the number of reservoirs in the system. Starting with an arbitrary value for the cell states, the sub-problems are solved on the neighborhood of each cell by a modified simulated annealing approach to find the new values of the cell states which are updated simultaneously at the end of the CA iteration. The process of solving sub-problems at the cell levels and updating the cell states are iterated until convergence is achieved. The proposed method is used for optimal hydropower operation of two well-known problems of four- and ten-reservoir systems, and the results are presented. To evaluate the efficiency and effectiveness of the proposed method, the results are compared with the existing results obtained by CA-NLP and two of the most common evolutionary algorithms, namely GA and PSO reported by Afshar (2013). The results show that the proposed method is both more efficient and effective than existing methods including evolutionary algorithms. A real-world case study of the Savannah River system was then solved and the results presented and compared with those of existing historical data indicating the ability of the proposed method to effectively and efficiently solve real-world operation problems.

**Cellular automata: an overview**

The concept of CA was first developed by Ulam (1960) and Von Neumann (1966), for simulating complex systems. Cellular automata divide the physical domain of the problem into identical elements called cells. Each cell can take a finite set of values named the cell state, which is evolved in time according to a local transition rule or updating rule. The cell states are updated at each time step using an updating rule defined as a function of the current state of cell and its neighbors. The updating process of cell values is continued until defined convergence criteria are met.
Among four main components namely cell, cell state, cell neighborhood and updating rule, the updating rule is the most important component of the CA highly affecting the success of the method. To ensure the success of CA in any problems, updating rule should possess three properties of parallelism, localism and homogeneity (Guo et al. 2007a).

CA was originally developed as a simulation method and since then has been extensively used for simulation of many phenomena, such as gas flow (Frisch et al. 1986), chemical process (Chopard et al. 1998), atmospheric pollution (Marin et al. 2000), traffic flow (Nagel 2002), population dynamics (Droz & Pekalski 2002) and eco-hydraulics (Chen & Mynett 2003; Chen 2004). Efficiency of the CA method has motivated the researchers to use it for solving optimization problems. In the past two decades, CA has been used as an optimization method in different fields of civil engineering including structural engineering (Gholizadeh 2013; Faramarzi & Afshar 2014) and traffic engineering (Maerivoet & Moor 2005; Mallikarjuna & Roa 2009; Ruan et al. 2017).

Application of CA for the solution of water resource engineering problems is rather limited. Keedwell & Khu (2005) used a coupled CA-GA model for the optimal design of water distribution networks. The CA was used for initial seeding of GA leading to significant improvement in GA performance. A hybrid CA and nondominated sorting GA-II (NSGAII) was proposed by Guo et al. (2007b) for the multi-objective design of water distribution networks in which CA was used to produce initial nondominated solutions for NSGAII. For the first time, Guo et al. (2007a) used CA as a stand-alone optimizer for the optimal pipe size design of storm sewer networks with known slopes. The first CA optimization model for solving water supply/hydropower reservoir operation problem was proposed by Afshar & Shahidi (2009). The method was used for single reservoir systems with the results, showing that CA was much more efficient and effective than some of the well-known heuristic search methods. A single-stage (Afshar et al. 2011) and two-stage (Afshar & Rohani 2012) CA method was proposed for the optimal design of sewer networks. Afshar (2013) proposed a CA-NLP method for the optimal operation of hydropower multi-reservoir systems in which NLP methods were used to solve optimization sub-problems resulting from application of CA to the problem. The results obtained by the proposed method was compared to those of GA and PSO, showing that the proposed CA-NLP was more efficient than GA and PSO, specially for large-scale multi-reservoir hydropower operation, while producing comparable results.

**Hydropower multi-reservoir operation**

The main objective of optimal operation of the multi-reservoir system is to maximize the benefits as much as possible by determining the set of optimal releases from each reservoir over operation periods while satisfying all physical and operational constraints (Papageorgiou 1985). In this paper, the goal is to maximize the total benefit from energy production over the operation period which leads to the following objective function:

\[
\text{Maximize } B_{\text{Total}} = \sum_{n=1}^{\text{NR}} \sum_{t=1}^{\text{NT}} b_{nt} E_{nt}
\]

where \( B_{\text{Total}} \) is the total benefit obtained by hydropower operation of NR reservoirs during NT operation periods, \( b_{nt} \) is the benefit function, \( E_{nt} \) is the total energy potential of reservoir \( n \) at period \( t \).

Subject to the following constraints:

1. The mass balance equation for reservoirs:

\[
S_{t+1} = S_t + I_t - MR_t \quad t = 1, 2, \ldots, \text{NT}
\]

in which \( S_t \) is the vector of storage volumes at the beginning of period \( t \), \( R_t \) and \( I_t \) are the vectors of releases from and inflow to the system over period \( t \), respectively, and \( M \) is an \( \text{NR} \times \text{NR} \) matrix describing the connectivity of reservoir network.

2. The maximum and minimum allowable water storage/release in/from each reservoir:

\[
S_{\text{min}}^{n_t} \leq S_{nt} \leq S_{\text{max}}^{n_t} \quad t = 1, 2, \ldots, \text{NT} + 1, \quad n = 1, 2, \ldots, \text{NR}
\]

\[
R_{\text{min}}^{n_t} \leq R_{nt} \leq R_{\text{max}}^{n_t} \quad t = 1, 2, \ldots, \text{NT}, \quad n = 1, 2, \ldots, \text{NR}
\]
wherein $S_{n,t}^{\min}$ and $S_{n,t}^{\max}$ are the minimum and maximum allowable volume of reservoir $n$ in period $t$, respectively; while $R_{n,t}^{\min}$ and $R_{n,t}^{\max}$ are the minimum and maximum allowable releases from $n$th reservoir over period $t$, respectively.

The monthly produced energy by reservoir $n$ in period $t$ is calculated as follows:

$$E_t = \sum_{n=1}^{NR} 2.725 \times e_n R_{n,t} h_{n,t}, \quad t = 1, 2, \ldots, NT \quad (5)$$

where $e_n$ is the efficiency of the $n$th reservoir and $E_t$ is the total produced energy of the system at period $t$, and $h_{n,t}$ is the effective head on the turbine for $n$th reservoir over period $t$ which is calculated as follows:

$$h_{n,t} = \left( \frac{H_{n,t} + H_{n,t+1}}{2} \right) - TW_{n,t}, \quad t = 1, 2, \ldots, NT,$$

$$n = 1, 2, \ldots, NR \quad (6)$$

in which $TW_{n,t}$ is the tailwater elevation of the reservoir $n$ at period $t$, $H_{n,t}$ and $H_{n,t+1}$ are the water elevation of the $n$th reservoir at the beginning and at the end of period $t$, respectively. The water elevation at each reservoir is obtained by an elevation-storage curve defined by

$$H_{n,t} = a_n + b_n S_{n,t} + c_n S_{n,t}^2 + d_n S_{n,t}^3, \quad t = 1, 2, \ldots, NT,$$

$$n = 1, 2, \ldots, NR \quad (7)$$

in which $a_n$, $b_n$, $c_n$ and $d_n$ are constants calculated via fitting Equation (7) to the available data. Similarly, the tailwater elevation of the $n$th reservoir at period $t$ is calculated by

$$TW_{n,t} = u_n + v_n S_{n,t} + x_n S_{n,t}^2 + w_n S_{n,t}^3,$$

$$t = 1, 2, \ldots, NT, \quad n = 1, 2, \ldots, NR \quad (8)$$

in which $u_n$, $v_n$, $x_n$ and $w_n$ are constants calculated via fitting Equation (8) to the available data.

**Proposed hybrid CA-SA algorithm**

Hydropower reservoir operation problems are commonly concerned with how to operate an existing system to produce maximum energy while considering natural and physical limitations. The hydropower reservoir operation problems are categorized as NP-hard problems which are known to be difficult to solve by traditional optimization methods. While evolutionary algorithms are known to overcome the limitations of the traditional optimization methods when solving hydropower reservoir operation problems, their applications are often hindered by increasing computational effort required as the scale of the problem increases (Cai et al. 2001; Reis et al. 2005; Afshar et al. 2010).

Here, a hybrid cellular automata and simulated annealing (CA-SA) method is proposed for the efficient and effective solution of hydropower operation of multi-reservoir systems. The concept of CA is used to recast the original large-scale problem to a series of small-scale optimization sub-problems which are subsequently solved by an SA.

**CA component**

Application of CA to any optimization problem requires that four basic components of the CA, namely cell, cell state, cell neighbors and updating rule, are properly defined. The cells, cell states and the cell neighborhood are so much inter-related that the selection of either of them naturally leads to the definition of other components. In optimization problems, the decision variables are often considered as the cell state. Afshar & Shahidi (2009) showed that proper selection of cell state yields a more efficient and effective CA model. They reported that in reservoir operation problems, the storage volume is a better choice as the cell state of the corresponding CA model. In this study, therefore, the cells are considered as discrete points in time represented by the beginning or the end of each period, so that the corresponding storage volumes can be assumed as the cell states. For an operation problem over NT periods, this leads to NT + 1 cells with corresponding cell states. Cell neighborhoods are naturally considered as the surrounding cells which means that each internal cell has two neighbors, while boundary cells, $t = 1$ and $t = NT + 1$, have only one neighbor.

Definition of updating rule, as the most crucial component of CA, is the final step of formulating the CA approach for the underlying problem. Here, the localized property of the reservoir storage and the underlying
Cells

Figure 1 | Schematic representation of an arbitrary cell t and its neighbors.

operation problem is used for driving the updating rule. Assume an arbitrary cell t with neighboring cells of \( t - 1 \) and \( t + 1 \) shown in Figure 1 representing a one-dimensional CA on the time domain. The local updating rule for the cell \( t \) is defined by the problem of finding the updated value of cell state \( S_t \) such that the benefit of the system operation is maximized, while all other state values are kept constant.

This leads to the following optimization sub-problem with a size equal to the number of reservoirs in the system, as the local updating rule which has to be solved for the cell state \( S_t \):

Maximize \( B_{\text{local}} = \sum_{n=1}^{NR} (b_{n,t-1}E_{n,t-1} + b_{n,t}E_{n,t}) \) \( (9) \)

\( S_{t+1} = S_t + I_t - MR_t \) \( (10) \)

\( S_t = S_{t-1} + I_{t-1} - MR_{t-1} \) \( (11) \)

\( S_{n}^{\text{min}} \leq S_{n,t} \leq S_{n}^{\text{max}}, \quad n = 1, \ldots, NR \) \( (12) \)

\( R_{n,t-1}^{\text{min}} \leq R_{n,t-1} \leq R_{n,t-1}^{\text{max}}, \quad n = 1, \ldots, NR \) \( (13) \)

\( R_{n,t}^{\text{min}} \leq R_{n,t} \leq R_{n,t}^{\text{max}}, \quad n = 1, \ldots, NR \) \( (14) \)

In which \( B_{\text{local}} \) is the local benefit function for cell \( t \). It should be noted that the change of the cell state only affects the energy produced in the neighboring periods \( t - 1 \) and \( t \) because all other cell states are assumed fixed. The mass balance equation is, therefore, only needed to be met for neighboring periods of \( t - 1 \) and \( t \), represented by Equations (10) and (11). The constraints (3) of the original problem need only be enforced for the cell under consideration defined by Equation (12), while Equations (13) and (14) guarantee the satisfaction of constraint (4) of the original problem.

The proposed CA process starts with a set of randomly initial guess for the cell states, reservoir storages at all periods of operation, denoted by \( S_{n}^{\text{old}} \). The updating rule defined by the optimization sub-problem of Equations (9)–(14) is used to find the updated value of the cell state denoted by \( S_{n}^{\text{new}} \) for all periods in turn. Once all cell states are updated, the old cell state values are replaced by the new updated values and the process is continued until convergence is achieved, i.e. no improvement in the cell states are made.

The updating rule of CA represented by Equations (9)–(14) is a small-scale optimization problem with the number of decision variables equal to the number of reservoirs in the system. This problem, however, is a nonlinear, nonconvex optimization problem which may only be solved using proper optimization methods. It is well-known that traditional optimization methods such as NLP may trap in local optima when solving problems of nonconvex nature such as those considered here (Afshar 2013). While evolutionary population-based methods are a natural choice for solving this sub-problem, these methods could undermine the efficiency of the proposed CA-based method since each CA iteration requires the solution of NT optimization sub-problems. An iterative EA such as simulated annealing (SA) seems to be more appropriate and is, therefore, used in this study to solve optimization sub-problems defined by Equations (9)–(14).

SA component

Simulated annealing (SA) as a stochastic search technique was developed by Kirkpatrick et al. (1985). The technique’s name and inspiration is adopted from the annealing process used in metallurgical processes which involves heating and controlled cooling of a material for maximizing its strength and minimizing its brittleness. Heating and cooling the materials affect its temperature and thermodynamic free energy. The rate of cooling dictates the magnitude of energy decrease, with slower cooling producing a bigger decrease in the energy. As an optimization technique, SA finds a particular configuration of the system such that minimizes the energy of the system. In other words, each particle in the material is located such that the thermodynamic free energy is minimized.

SA like any other EAs uses some free parameters including initial temperature, the decrement rate of temperature or cooling mechanism, the number of transitions at each temperature and the final temperature as the stopping criteria of the algorithm. These parameters play an important role in
the performance of SA and should be defined carefully and properly (Li & Wei 2008). The large enough values of initial temperature allow the algorithm to thoroughly explore the search space and increase the probability of finding better solutions. However, a too big initial temperature leads to higher computational time. On the other hand, too small a value of initial temperature limits the algorithm to smaller search space.

At each temperature, particles are allowed to move to new positions such that to achieve the minimum possible energy. The search process of SA can be considered as a Markov chain, whose length is defined by the number of movements allowed at current temperature (Tospornsampan et al. 2005). The new position is accepted or rejected according to an acceptance criterion defined by

\[
\mu < \min \left\{ 1, \exp \left( \frac{e_{m+1} - e_m}{T_k} \right) \right\} \tag{15}
\]

in which \( \mu \) is a uniformly distributed random number between 0 and 1. The new position is accepted if the inequality (15) is satisfied, and rejected otherwise. Here, \( e_{m+1} \) and \( e_m \) are the energy of the particle in consecutive positions and \( T_k \) is the temperature at \( k \)th iteration. The temperature is decreased at each iteration which is marked by a predefined number of movements.

Performance of SA is highly dependent on the cooling rate. SA interprets the cooling as decreasing the acceptance probability of worse solutions while exploring the search space. The higher acceptance probability leads to higher computational time, while the lower acceptance probability decreases the effectiveness of the algorithm. Here, the cooling process is defined as follows:

\[
T_k = \alpha^k T_0 \tag{16}
\]

where \( T_k \) and \( T_0 \) are temperatures at current iteration and initial temperature, respectively, and \( \alpha \) is the cooling rate which ranges between 0.8 and 0.99, as suggested by Kirkpatrick et al. (1983).

In the problem under consideration, each particle represents a solution namely the storage volumes of the reservoirs at all operation periods, while the energy of the system is evaluated by the benefit function. Two options were available to implement the SA. In the first one, a complete cooling cycle is used to solve the sub-problem until freezing is reached. This option, however, was computationally demanding as a complete SA has to be used for all sub-problems faced at each iteration of the CA. To alleviate the computational requirement of the method, an alternative option was used. In this method, the cooling cycle is removed from SA and is embedded into the CA method. In other words, the sub-problems defined by Equations (9)–(14) are solved using a constant temperature at each CA iteration and the temperature is only changed during CA iterations. Following some preliminary runs, the value of cooling rate \( \alpha \) and the initial temperature \( T_0 \) are set to 0.95 and \( 10^5 \), respectively, and the number of movements at each temperature are limited to 10 for computational efficiency. The new points for the next iteration are generated using ‘annealingfast’ function of MATLAB. The solution process starts with setting initial random solutions for all cell states, namely \( S_i^t \) for \( t = 1, 2, ..., NT \), and setting an initial temperature \( T_0 \) for SA. Starting with the first cell \( (t = 1) \), the sub-problem defined by Equations (9)–(14) is solved for each cell in turn by the SA at a constant temperature, considering the current cell state as the SA starting point. Once all sub-problems, equal to the number of time periods (NT), are solved and the value of the corresponding cell states are calculated, the cell states are simultaneously updated and a CA iteration is completed. At the end of each CA iteration, the results are checked for convergence and the algorithm is stopped if converged; otherwise, the temperature is updated using Equation (16) and a new CA iteration is carried out. The proposed iterative process is continued until convergence criteria defined as zero change in the objective function of two consecutive CA iterations is met. The flowchart of CA and SA components of the proposed hybrid CA-SA method for multi-reservoir hydropower operation and the inter-relation between CA and SA components are presented in Figure 2(a) and 2(b), respectively.

**CASE STUDIES**

To test the efficiency and effectiveness of the proposed CA-SA method, some case studies are considered in this section.
The proposed method is first used to solve two well-known benchmark reservoir operation problems namely the four- and ten-reservoir problems which were first introduced to the literature by Chow & Cortes-Rivera (1974) and Murray & Yakowitz (1979), and solved by many researchers to test different methods since their introduction.
The schematic representation of the four-reservoir system as proposed by Chow & Cortez-Rivera (1974) is illustrated in Figure 3. Also, Figure 4 illustrates the ten-reservoir system configuration containing parallel and series connections which were first devised by Murray & Yakowitz (1979).

The problem's objective function is mathematically written as follows:

\[
\text{Maximize } B_{\text{Total}} = \sum_{n=1}^{\text{NR}} \sum_{t=1}^{\text{NT}} b_{n,t} R_{n,t}
\]  

This objective function is the same as the objective function of Equation (1) except for the energy potential.
which has been replaced by the releases $R_{n,t}$ making it a linear problem. All other parameters are the same as those already defined for Equation (1). The operation of the systems is to be carried out over 12 operation periods with the problem parameters given in Murray & Yakowitz (1979).

These problems have been solved by many researchers to test different methods since their introduction. Chow & Cortez-Rivera (1974) introduced and solved the four-reservoir problem using LP. Murray & Yakowitz (1979) developed a discrete dynamic programming (DDP) model to solve these problems. Different methods, such as GA (Wardlaw & Sharif 1999), ACO (Jalali et al. 2007), HBMO (Haddad et al. 2011), imperialist competitive algorithm (ICA) and Cuckoo optimization algorithm (COA) (Hosseini-Moghari et al. 2013), improved bat algorithm (IBA) (Ahmadianfar et al. 2016), krill algorithm (Ehteram et al. 2017), teaching-learning-based optimization (TLBO) and Jaya algorithm (JA) (Kumar & Yadav 2018), hybrid Whale-GA algorithm (HWGA) (Mohammadi et al. 2019), wolf search algorithm (WSA) (Ahmadebrahimpour 2019), weed optimization algorithm particle swarm optimization (WOAPSO) (Asgari et al. 2019), and fully constrained improved artificial bee colony algorithm (FCIABC) (Moelini & Soghrati 2019), were also employed to solve these problems.

Nonlinear four- and ten-reservoir problems

The four- and ten-reservoir problems in their original forms, however, are of the linear type which could not represent the nonlinear nature of the hydropower operation of multi-reservoir systems considered here. Therefore, a modified form of these problems proposed by Afshar (2013) is used here to test the capabilities of the proposed method to handle nonlinear problems. All data regarding the allowable minimum and maximum storages and releases volumes, as well as benefit functions, are the same as those used for the original linear problem reported by Murray & Yakowitz (1979), while the elevation-storage curve data can be found in Afshar (2013).

An important note should be made here. While the original linear problem was defined over 12 months, Afshar (2013) defined and solved the nonlinear version of the problems over longer operation periods. To assess the efficiency and effectiveness of the proposed method for problems of different scales, the nonlinear systems are operated here over three different operation periods of 12, 60 and 240 months with the number of decision variables equal to 44/110, 236/590 and 956/2,390 for four-/ten-reservoir systems, respectively.

Since the proposed algorithm requires an initial solution to start the search process, the four- and ten-reservoir problems are solved 10 times using different randomly generated initial guesses and the maximum,
minimum and average solution are presented to evaluate the sensitivity of the proposed method to the initial solutions.

**Savannah River system**

The ability of the proposed CA-SA method in solving real-world hydropower operation problems is finally evaluated here solving the long-term monthly operation of the Savannah River system. The Savannah River originates from the Blue Ridge Mountains located in the southwest of North Carolina. This river is formed by the convergence of Seneca and Tugaloo Rivers. There are smaller contributing tributaries to the main river, including Chattooga, Little and Broad Rivers. The Savannah River flows along the Georgia and South Carolina border before reaching the Atlantic Ocean.

Savannah River flow is managed by a reservoir network, which mainly includes Hartwell, Richard B. Russell and J. Storm Thurmond Reservoirs, as shown in Figure 5.

![Figure 5](http://iwaponline.com/jh/article-pdf/22/5/1236/762122/jh0221236.pdf)
These reservoirs are primarily operated for flood control purposes by the US Army Corps of Engineers (USACE). The system of reservoirs is well known as the most highly visited nationwide Corps lakes and the system plays a crucial role in the economy and ecosystem of the region. These reservoirs were constructed during the 1950s and 1960s with the main objectives of flood control as well as providing hydropower electricity for the region. All three reservoirs can generate hydroelectric power from peaking power plants which are generally run when there is a high demand for electricity.

The proposed CA-SA method was used for monthly optimal operation of the reservoir system over the period of 1998–2018. The goal is to maximize the total produced energy from the system while meeting the flood control requirements of the system.

RESULTS AND DISCUSSION

In this section, the results of the method obtained for the case studies are presented and compared with the existing results in the literature. The results obtained for the original linear four- and ten-reservoirs systems are first presented followed by the results obtained for the modified nonlinear systems. The solution obtained for the optimal operation of a real-world reservoir system, namely the Savannah River system, located in the USA, are then produced and compared with the existing results. All computational times are reported in terms of the CPU run time on a 1.8 MHz Pentium 4.

Linear four- and ten-reservoir problems

Table 1 summarizes the results obtained by different methods for solving the linear four-reservoir problem along with the results of the proposed CA-SA method. Once again, the same performance is repeated here by the proposed algorithm as described for the four-reservoir problem. The method is seen to achieve the second highest best/worst solution of 1,194.44/1,193.12 emphasizing the effectiveness and insensitivity of the method to the initial random solutions.

An important note should be added here regarding the efficiency of the proposed method compared to that of the JA method. According to Kumar & Yadav (2018), JA required 7,000 and 14,000 iterations using a population size of 50 amounting to 350,000 and 700,000 function evaluations.
evaluations to get final solutions for the four- and ten-reservoir problems, respectively. The proposed method, however, requires 433 and 592 CA iterations to get the final solutions for the four- and ten-reservoir problems, respectively. Since each CA iteration requires the solution of 12 sub-problems and the solution of each sub-problem requires 10 SA movements, the total number of function evaluation required by the proposed method can, therefore, be calculated as 51,960 and 71,160 for four- and ten-reservoir problems, respectively. This clearly shows that the proposed method is much more efficient than JA in finding an optimal solution for the linear problems. The proposed method requires one-seventh and one-tenth of the computational effort required by the JA method to produce solutions for the four- and ten-reservoir problems which are inferior to those of JA by only 0.03 and 0.01%, respectively.

Nonlinear four- and ten-reservoir problems

As pointed out earlier, to consider the nonlinear nature of the real-world hydropower operation problems, the nonlinear version of the four- and ten-reservoir problems, proposed by Afshar (2013), is also solved here and the results are compared with those of Afshar (2013).

Table 3 shows the best, worst and mean solution costs along with the average computational time and the scaled standard deviations (SSD) of the solutions obtained for the nonlinear four-reservoir system. This table emphasizes three very advantageous features of the proposed method. First, the method is nearly insensitive to the initial solution as indicated by the very small SSD of the solutions over different operation periods. Second, the insensitivity level of the solution to the initial guess is completely independent of the problem scale indicated by approximately the same value of the SSD for problems of different scales. Third, the table indicates that the computational effort required by the method to solve problems of different scales only increases linearly with the scale of the problem which emphasizes on the efficiency of the method compared to the existing random-based search method. The computational efficiency of the proposed method will be elaborated upon later when comparing the results with those of the existing methods.

Table 2 | Comparison of reported results for ten-reservoir problem (Muarry & Yakowitz 1979)

<table>
<thead>
<tr>
<th>Study</th>
<th>Method</th>
<th>Best solution</th>
<th>Worst solution</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Murray &amp; Yakowitz (1979)</td>
<td>DDP</td>
<td>1,190.65</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Wardlaw &amp; Sharif (1999)</td>
<td>GA</td>
<td>1,190.25</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Jalali et al. (2007)</td>
<td>ACO</td>
<td>1,192.39</td>
<td>1,180.88</td>
<td>1,185.22</td>
</tr>
<tr>
<td>Haddad et al. (2011)</td>
<td>HBMO</td>
<td>1,156.79</td>
<td>1,139.43</td>
<td>1,148.05</td>
</tr>
<tr>
<td>Ahmadianfar et al. (2016)</td>
<td>IBA</td>
<td>1,193.92</td>
<td>1,191.22</td>
<td>1,192.89</td>
</tr>
<tr>
<td>Ehteram et al. (2017)</td>
<td>GA</td>
<td>1,190.01</td>
<td>1,187.34</td>
<td>1,188.68</td>
</tr>
<tr>
<td></td>
<td>NLP</td>
<td>1,194.44</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Krill</td>
<td>1,190.12</td>
<td>1,189.24</td>
<td>1,189.66</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>1,194.44</td>
<td>1,193.12</td>
<td>1,193.91</td>
</tr>
<tr>
<td>Kumar &amp; Yadav (2018)</td>
<td>TLBO</td>
<td>1,194.44</td>
<td>1,194.26</td>
<td>1,194.37</td>
</tr>
<tr>
<td></td>
<td>JA</td>
<td>1,194.59</td>
<td>1,194.44</td>
<td>1,194.55</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>1,194.44</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Mohammadi et al. (2019)</td>
<td>GA</td>
<td>1,069.52</td>
<td>1,010.24</td>
<td>1,035.27</td>
</tr>
<tr>
<td></td>
<td>NLP</td>
<td>1,100.96</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>1,161.51</td>
<td>1,115.67</td>
<td>1,136.03</td>
</tr>
<tr>
<td>Asgari et al. (2019)</td>
<td>WOA</td>
<td>1,160.30</td>
<td>1,136.63</td>
<td>1,147.93</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>1,156.33</td>
<td>1,142.54</td>
<td>1,151.77</td>
</tr>
<tr>
<td></td>
<td>WOA</td>
<td>1,193.76</td>
<td>1,189.75</td>
<td>1,191.64</td>
</tr>
<tr>
<td>Mocini &amp; Soghrahi (2019)</td>
<td>FCIABC</td>
<td>1,192.02</td>
<td>1,188.23</td>
<td>1,189.97</td>
</tr>
<tr>
<td>Present study</td>
<td>CA-SA</td>
<td>1,194.44</td>
<td>1,193.12</td>
<td>1,193.67</td>
</tr>
</tbody>
</table>

Table 3 | Results of the proposed CA-SA method for hydropower operation of the four-reservoir system

<table>
<thead>
<tr>
<th>Months</th>
<th>Total benefit</th>
<th>Average computational time (s)</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best solution</td>
<td>Worst solution</td>
<td>Mean</td>
</tr>
<tr>
<td>12</td>
<td>3.45 x 10^-04</td>
<td>3.41 x 10^-04</td>
<td>3.45 x 10^-04</td>
</tr>
<tr>
<td>60</td>
<td>1.74 x 10^-05</td>
<td>1.72 x 10^-05</td>
<td>1.73 x 10^-05</td>
</tr>
<tr>
<td>240</td>
<td>7.00 x 10^-05</td>
<td>6.90 x 10^-05</td>
<td>6.97 x 10^-05</td>
</tr>
</tbody>
</table>
Afshar (2013) used a hybrid CA-NLP method to solve the four-reservoir system in its nonlinear form over three operation periods of 12, 60 and 240 months and compared its performance with those of GA and PSO. A population/swarm size of 100, 200 and 300 was used by the GA/PSO for solving the problem over the operation periods of 12, 60 and 240 months, respectively. To ensure convergence of the search, the GA/PSO searches were conducted over 1,500, 3,000 and 6,000 generations/iterations for the operation periods of 12, 60 and 240 months, respectively. A binary tournament selection operator, a standard single-point crossover with a probability of one with the bits averaged at crossover site, and one-bit mutation procedure with a probability of 0.5, in which the value of the mutated decision variable was randomly chosen from the range of one-tenth of mutated decision variable, was employed by the GA to construct the next generation. To guarantee that no useful information is lost during the search process, an elitist strategy was used whereupon the best solution of the last generation was copied to the next generation.

A standard local PSO algorithm was used with a local neighboring size equal to one-tenth of the swarm size, a constriction factor of one, and the cognitive and social parameters equal to 0.5. The inertia weight was linearly varied from 0.95 at the beginning of the search to 0.85 at the end of the search.

Table 4 compares the best solutions obtained by the proposed CA-SA method with those obtained by CA-NLP, GA and PSO along with the average computational time and SSD as reported in Afshar (2013). It is seen that the proposed CA-SA method was able to produce superior solutions to those of the existing methods for all operation periods. Furthermore, it is seen that the proposed CA-SA method was able to produce better solutions than the two evolutionary algorithms, namely GA and PSO, with much less computational efforts which are more evident for large-scale problems. As shown in Table 4, the proposed method produced solutions 3, 6 and 8% better than those obtained by GA for 12, 60 and 240 months of operation with much reduced computational effort. The super-efficiency of the proposed method is clearly seen from Table 4, indicating that the method requires less and less CPU time compared to the GA method as the scale of the problem increases. While the method requires the same CPU time as that of GA for the operation over a 12 month period, its required CPU time reduces to one-third/one-ninth of the CPU time required by the GA for the operations over 60/240 months.

The super-efficiency of the proposed method is mainly due to the novel idea of embedding the cooling cycle of the SA method in the CA iterations. To illustrate this fact, these problems are resolved with a method in which a complete cooling cycle is used for each SA until the freezing condition is achieved. Table 5 illustrates the best, worst and mean solution costs along with the average computational time for 10 runs.

Comparing Table 5 and Table 5 shows that the novel idea of embedding the cooling cycle in the CA iterations increases the efficiency of the proposed hybrid CA-SA method by a factor of 28 compared to using a standard SA to solve the underlying local optimization problems at each CA iteration. A typical variation of the benefit function during CA iterations for the problem of operation over 240 periods is also shown in Figure 6.

Table 4 | Comparison of the results for hydropower operation of the four-reservoir system

<table>
<thead>
<tr>
<th>Months</th>
<th>Method</th>
<th>Best solution</th>
<th>Average CPU time (s)</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>CA-SA</td>
<td>3.45 × 10^{-04}</td>
<td>7.00</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>CA-NLP</td>
<td>3.34 × 10^{-04}</td>
<td>0.34</td>
<td>0.025</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>3.34 × 10^{-04}</td>
<td>6.90</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>3.34 × 10^{-04}</td>
<td>9.60</td>
<td>0.015</td>
</tr>
<tr>
<td>60</td>
<td>CA-SA</td>
<td>1.74 × 10^{-05}</td>
<td>39.00</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>CA-NLP</td>
<td>1.62 × 10^{-05}</td>
<td>12.50</td>
<td>0.016</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>1.64 × 10^{-05}</td>
<td>126.70</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>1.49 × 10^{-05}</td>
<td>181.80</td>
<td>0.009</td>
</tr>
<tr>
<td>240</td>
<td>CA-SA</td>
<td>7.00 × 10^{-05}</td>
<td>165.00</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>CA-NLP</td>
<td>6.59 × 10^{-05}</td>
<td>73.50</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>6.50 × 10^{-05}</td>
<td>1,487.00</td>
<td>0.003</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>5.89 × 10^{-05}</td>
<td>2,208.90</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Table 5 | Results of the hybrid method incorporating complete cooling procedure of SA for solving hydropower operation problems of the four-reservoir system

<table>
<thead>
<tr>
<th>Total benefit</th>
<th>Months</th>
<th>Best solution</th>
<th>Worst solution</th>
<th>Mean</th>
<th>Average computational time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12</td>
<td>3.46 × 10^{-04}</td>
<td>3.44 × 10^{-04}</td>
<td>3.45 × 10^{-04}</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1.77 × 10^{-05}</td>
<td>1.75 × 10^{-05}</td>
<td>1.76 × 10^{-05}</td>
<td>1,236</td>
</tr>
<tr>
<td></td>
<td>240</td>
<td>7.06 × 10^{-05}</td>
<td>7.02 × 10^{-05}</td>
<td>7.04 × 10^{-05}</td>
<td>5,062</td>
</tr>
</tbody>
</table>
Table 6 presents the best, worst and mean solution costs obtained over 10 runs each using different initial random solutions, along with the averaged computational time and the scaled standard deviations (SSD) of the solutions obtained for the nonlinear ten-reservoir system. Figure 7 shows a typical variation of the benefit function.

**Table 6** | Results of the proposed CA-SA method for hydropower operation of the ten-reservoir system

<table>
<thead>
<tr>
<th>Months</th>
<th>Total benefit</th>
<th>Average computational time (s)</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best solution</td>
<td>Worst solution</td>
<td>Mean</td>
</tr>
<tr>
<td>12</td>
<td>$2.71 \times 10^{-05}$</td>
<td>$2.44 \times 10^{-05}$</td>
<td>$2.60 \times 10^{-05}$</td>
</tr>
<tr>
<td>60</td>
<td>$1.36 \times 10^{-06}$</td>
<td>$1.30 \times 10^{-06}$</td>
<td>$1.33 \times 10^{-06}$</td>
</tr>
<tr>
<td>240</td>
<td>$5.33 \times 10^{-06}$</td>
<td>$5.21 \times 10^{-06}$</td>
<td>$5.29 \times 10^{-06}$</td>
</tr>
</tbody>
</table>

**Figure 6** | Variation of the benefit function with CA iterations for the four-reservoir system over 240 months of operation.

**Figure 7** | Variation of the benefit function with CA iterations for the ten-reservoir case over 240 months of operation.
during CA iterations for the problem of 240 months of operation periods. These results vindicate three distinct features of the proposed method already touched upon when presenting the results of the four-reservoir problem. It is seen once again that the method is nearly insensitive to the initial random solution indicated by very small SSD of the final solution for problems of different scales, and more importantly, the computational efficiency of the method in the sense that the computational effort of the method only increases linearly with the scale of the problem, a property which is absent in most of the random base search methods including evolutionary algorithms.

This problem has also been solved by Afshar (2013) using a hybrid CA-NLP method, and two of the most efficient heuristic search methods, namely GA and PSO algorithms using 3,000, 6,000 and 9,000 generations/iterations for 12, 60 and 240 months of operation, respectively. Table 7 compares the best solutions obtained by the proposed CA-SA method with those of CA-NLP, GA and PSO as reported in Afshar (2013) along with the average computational time and SSD of the final solutions.

It is seen from Table 7 that the proposed CA-SA method produces a better solution than GA, as the best performing methods amongst CA-NLP and PSO, for problems of different scales with much reduced computational effort. In particular, the proposed method produces solutions %3, %9 and %30 better than GA with a computational time 4, 13 and 28 times less than those required by the GA. This emphasizes the fact already mentioned that superiority of the proposed method to GA in both efficiency and quality of the final solution amplifies as the scale of the problem increases. And finally, Table 7 clearly shows that the proposed method is the most insensitive to the initial solution amongst other methods which is vindicated by smallest SSD of the final solutions, an important property when the method is to be used for finding engineering solutions to real-world operation problems.

The most interesting advantage of the proposed hybrid CA-SA method is due to its low computational requirement compared to existing methods. Except for the CA-NLP method of Afshar (2013) which is more efficient than the proposed CA-SA method, the proposed method is shown to be able to locate the near-optimal solution in much less computational time compared to existing methods such as GA and PSO. The computational requirement of the method is seen to increase only linearly with the size of the problem. This property can be compared with the semi-exponential increase of the computational time with the problem size for population-based evolutionary algorithms such as GA and PSO.

### Savannah River system

Monthly generated energy from the system of reservoirs obtained using the proposed method along with the historical produced energy are compared in Figure 8. Figure 9 compares the monthly average produced energy using the proposed CA-SA method with those of historical energy production by each reservoir in the Savannah River system. It is quite clear that the proposed CA-SA outperforms the historical energy production in most of the periods. From April to August, when the inflow to the system is naturally decreasing, the CA-SA method tends to release more water to the downstream of reservoirs; such that the system is able to produce more hydroelectric energy. In this case, the first reservoir upstream, Hartwell, produces less energy than historical energy production, while the whole system generates more energy.

The average monthly energy production by each of the reservoirs and the system is compared with the historical energy productions in Table 8, indicating that the CA-SA suggests operation %21 more economical than the historical operation for the period of 1998–2018.
The historical energy duration curve of the system is compared with the one proposed by the CA-SA method in Figure 10 which is fully consistent with the results presented in Table 8.

**Figure 10** shows the variation of average monthly storage volumes of the reservoirs for both CA-SA and historical operation along with the top of conservation levels. It is clearly seen that the operation policy suggested by the CA-SA keeps the water level in all reservoirs below the top of conservation level during the operation periods, while the historical operation information violates the top of conservation level at some periods of operations.

### Table 8 | Average monthly produced energy for reservoirs by CA-SA versus historical data

<table>
<thead>
<tr>
<th>Reservoir</th>
<th>Average monthly produced energy (CA-SA)</th>
<th>Average monthly produced energy (Historical)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hartwell</td>
<td>33,880</td>
<td>32,072</td>
</tr>
<tr>
<td>Russell</td>
<td>57,524</td>
<td>45,456</td>
</tr>
<tr>
<td>Thurmond</td>
<td>57,437</td>
<td>45,788</td>
</tr>
<tr>
<td>System’s average monthly produced energy</td>
<td>148,842</td>
<td>123,317</td>
</tr>
</tbody>
</table>

The historical energy duration curve of the system is compared with the one proposed by the CA-SA method in Figure 10 which is fully consistent with the results presented in Table 8.

**CONCLUSION**

In this study, a hybrid cellular automata and simulated annealing method, namely CA-SA, was proposed for
optimal hydropower operation of multi-reservoir systems. The concept of CA was employed to break the original large-scale problem into a series of small-scale optimization sub-problems, which were then solved by simulated annealing. The proposed method was first employed for the optimal operation of two well-known benchmark problems of four- and ten-reservoir systems over different operation periods of 12, 60 and 240 months. The results of the proposed method were compared with those of GA, PSO and CA-NLP from the literature, indicating that the proposed method was much more efficient and effective than GA, PSO and CA-NLP for the problems considered. The method was able to produce solutions up to 33% better than GA with much reduced computational effort. The proposed method was then used to solve long-term hydropower operation of a real-world three-reservoir system in the USA and the results were compared with those of historical data indicating the superiority of the proposed method by producing 21% more energy than a historical one.

Figure 10 | Energy duration curve.

Figure 11 | Average monthly storage variation in three reservoirs of the system, and top of conservation.
REFERENCES


Chow, V. T. & Cortez-Rivera, G. 1974 Application of DDDP in Water Resources Planning, Department of Civil Engineering. University of Illinois at Urbana-Champaign, IL, USA.


Teegavarapu, R. S. & Simonovic, S. P. 2002 Optimal operation of reservoir systems using simulated annealing. Water


First received 29 September 2019; accepted in revised form 7 May 2020. Available online 25 June 2020