

# Long-Term Joint Operation of Cascade Reservoirs Using Enhanced Progressive Optimality Algorithm and Dynamic Programming Hybrid Approach

Jia Chen (✉ [cj0370@126.com](mailto:cj0370@126.com))

Zhuji Municipal Water Conservancy Bureau

---

## Research Article

**Keywords:** Cascade, Reservoir, Hydropower station, Optimal operation, Dynamic programming, Progressive optimality algorithm

**Posted Date:** March 11th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-307334/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Long-term Joint Operation of Cascade Reservoirs Using Enhanced  
Progressive Optimality Algorithm and Dynamic Programming Hybrid  
Approach**

Jia Chen <sup>1</sup>

<sup>1</sup> Zhuji Municipal Water Conservancy Bureau, Shaoxing 311800, China

\*Corresponding author: Jia Chen

E-mail: cj0370@126.com

**Abstract:** Dynamic programming (DP) is one of the most classical methods adopted for reservoir operation. It reduces the computational efforts of complex high-dimensional problems by piecewise dimensionality reduction and provides the global optimums of the problems, but it suffers the “curse of dimensionality”. Progressive optimality algorithm (POA) has been used repeatedly in reservoir operation studies during last decades because it alleviates the “curse of dimensionality” of DP and has good convergence and extensive applicability. Nonetheless, POA encounters two difficulties in multi-reservoir operation applications. One is the transfer interrupt problem that makes the search procedure hard to achieve free allocation of water between two nonadjacent stages, and the latter is the dimensionality problem that leads to a low convergence rate. In order to overcome these deficiencies, this paper makes some enhancements to POA and proposes a hybrid approach combining the enhanced POA and DP (EPOA-DP) for long-term operation of cascade reservoir systems. In EPOA-DP, EPOA is employed

to improve the quality of the solutions and DP is used to reduce the computational effort of the two-stage problem solution. The proposed approach was tested using a real world four-reservoir cascade system and a ten-reservoir benchmark test example, and the results demonstrate that it outperforms POA both in computational time and quality of the solution.

**Key words:** Cascade, Reservoir, Hydropower station, Optimal operation, Dynamic programming, Progressive optimality algorithm

## **1 Introduction**

Joint operation of cascade reservoirs is an important way to improve water and hydropower utilization efficiency on basin scale. As hydraulic connections and compensatory relationships between the various reservoirs are very complicated in a cascade system, joint operation process of cascade reservoirs is a complex high-dimensional constrained optimization problem which is hard to get the globally optimal solution. Numerous classical methods, such as linear programming (Lee et al 2008; Kang et al. 2017), nonlinear programming (Catalao et al. 2010; Jothiprakash and Arunkumar 2014), dynamic programming (DP) (Ji et al. 2014; Rani et al. 2020; Zhao et al. 2017) and DP's variants (Opan 2011; Feng et al. 2017; Cheng et al. 2012) and intelligent algorithms (Jahandideh-Tehrani et al. 2020; Shaikh 2020), have been applied to joint operation of cascade reservoirs. In contrast to other methods, DP and its variants have been used more extensively in engineering practice because they can easily handle non-linear as well as non-differentiable factors and are able to produce satisfactory solutions in most cases. DP is one of the most classical and popular methods employed for reservoir operation problems. It produces the globally optimal solutions for a given magnitude of discretization, but suffers the “curse of dimensionality” in that the memory and computational requirements grow exponentially with the number of reservoirs. The dimensionality problem make DP inapplicable to multi-reservoir operation.

In order to alleviate the “curse of dimensionality”, a sequence of improvements have been made to DP and different variants have been developed. Larson and Korsak (1970) proposed the dynamic programming successive approximations technique (DPSA) which reduced the number of reservoirs to optimize with respect to at a time for DP by decomposing a multi-reservoir problem into a sequence of

single reservoir ones. Heidari et al. (1971) provided the discrete differential dynamic programming approach (DDDP) whose most attractive feature is that it lessens the number of states for DP by iteratively searching in constantly changing corridors. Howson and Sancho (1975) presented the progressive optimality algorithm. It is, in contrast to DP, characterized by the fact that it decreases the number of stages to focus on at a time by transforming a multi-stage problem into a sequence of two-stage ones. Bai et al. (2015) and Zhang et al. (2016) applied hybrid approaches combining POA and DPSA to multi-reservoir operation, which inherited the advantages of both methods. Cheng et al. (2014) and Ma et al. (2020) solved multi-reservoir operation problems using parallel computing techniques, which lessened the execution time of the search procedure. Feng et al. (2020a) applied the Latin hypercube sampling technique to optimization of a cascade reservoir system, which reduced the computational burden of DP.

As an improved version of DP, POA decreases the computational effort required by DP via reducing time dimensions. Since POA optimizes only with respect to two stages at a time, the computational efficiency is improved significantly in contrast to DP. Howson and Sancho (1975) proved that POA produces the globally optimal solution when the objective function of the problem is convex. For these reasons, it has been widely used in multi-reservoir operation studies. Nevertheless, it encounters two difficulties in multi-reservoir operation applications: the transfer interrupt problem and the dimensionality problem (Feng et al. 2020b). These limit its performance in multi-reservoir operation problems.

In this paper, some enhancements are made to POA and a hybrid approach combining the enhanced POA and DP (EPOA-DP) is developed to improve the

computational efficiency of the search procedure as well as the quality of the solution for long-term operation of cascade reservoir systems. A real world four-reservoir cascade system and a benchmark ten-reservoir text example are used to verify the feasibility and validity of the proposed approach. Due to our focus on improving the performance of the search procedure, the stochasticity of inflows is not considered in the model for simplifying calculations.

## 2 Mathematical Model for Cascade Reservoir Operation Problems

### 2.1 Objective Function

The object for long-term operation of cascade reservoir system is to maximize the benefits from the system over the entire operating period. Assume that a cascade system consists of  $M$  reservoirs that are numbered 1, 2, ...,  $M$  sequentially from upstream to downstream and the operating period consists of  $T$  same time periods. Thus the objective function can be written as

$$\max B = \sum_{i=1}^M \sum_{t=1}^T b_i(V_{i,t-1}, R_{i,t}, I_{i,t}) \quad (1)$$

where  $B$  denotes the total benefits from the cascade system over the entire operating period;  $V_{i,t-1}$  denotes the storage in reservoir  $i$  at time period  $t-1$ ;  $R_{i,t}$  denotes the release from reservoir  $i$  at time period  $t$ ;  $I_{i,t}$  denotes the inflow to reservoir  $i$  at time period  $t$ ;  $b_i(V_{i,t-1}, R_{i,t}, I_{i,t})$  denotes the benefit function for reservoir  $i$  at time period  $t$ .

Usually,  $R_{i,t}$  is a function of  $V_{i,t-1}$ ,  $V_{i,t}$  and  $I_{i,t}$ . Thus Eq. 1 can be written as

$$\max B = \sum_{i=1}^M \sum_{t=1}^T g_i(V_{i,t-1}, V_{i,t}, I_{i,t}) \quad (2)$$

where  $g_i(V_{i,t-1}, V_{i,t}, I_{i,t})$  denotes the benefit function for reservoir  $i$  at time period  $t$ .

### 2.2 Constraints

(1) Water balance equation:

$$\begin{cases} V_{i,t} = V_{i,t-1} + I_{i,t} - L_{i,t} - R_{i,t} \\ I_{i,t} = Q_{i,t} + R_{i-1,t} \end{cases} \quad (3)$$

(2) Reservoir storage limits:

$$\underline{V}_{i,t} \leq V_{i,t} \leq \overline{V}_{i,t} \quad (4)$$

(3) Reservoir release limits:

$$\underline{R}_{i,t} \leq R_{i,t} \leq \overline{R}_{i,t} \quad (5)$$

(4) Boundary conditions:

$$V_{i,0} = V_i^{\text{ini}}, V_{i,T} = V_i^{\text{fin}} \quad (6)$$

where  $L_{i,t}$  denotes the evaporation loss of reservoir  $i$  at time period  $t$ ;  $Q_{i,t}$  denotes the interval inflow to reservoir  $i$  at time period  $t$ ;  $R_{0,t} = 0$ ;  $\underline{V}_{i,t}$  and  $\overline{V}_{i,t}$  denote the minimum and maximum permitted storages for reservoir  $i$  at time period  $t$ ;  $\underline{R}_{i,t}$  and  $\overline{R}_{i,t}$  denote the minimum and maximum permitted releases for reservoir  $i$  at time period  $t$ ;  $V_i^{\text{ini}}$  and  $V_i^{\text{fin}}$  denote the initial and final storages in reservoir  $i$ , and are usually given at the beginning of the operating period.

### 3 EPOA-DP Hybrid Approach

#### 3.1 Rationale of Prototypical POA

In order to alleviate the “curse of dimensionality” of DP, Howson and Sancho (1975) presented a principle of progressive optimality based on Bellman’s principle of optimality, which states that:

“The optimal path has the property that each pair of decision sets is optimal in relation to its initial and terminal values.”

Based on the above theory, POA was developed by Howson and Sancho (1975) so that the multi-stage problem of reservoir operation was decomposed into a sequence

of two-stage subproblems. The most essential feature of POA is that the procedure advances toward the globally optimum by successively and iteratively solving the two-stage problems which can be written as

$$\max_{V_{1,t}, V_{2,t}, \dots, V_{M,t}} F = \left\{ \sum_{i=1}^M [g_i(V_{i,t-1}, V_{i,t}, I_{i,t}) + g_i(V_{i,t}, V_{i,t+1}, I_{i,t+1})] \right\} \quad (7)$$

where  $F$  denotes the benefits from the system over time periods  $t$  and  $t+1$ ;  $V_{i,t-1}$ 's and  $V_{i,t+1}$ 's are known fixed values;  $V_{i,t}$ 's are variables to optimize;  $I_{i,t}$ 's and  $I_{i,t+1}$ 's are determined by  $V_{i,t-1}$ 's,  $V_{i,t}$ 's and  $V_{i,t+1}$ 's through Eq. 3.

An iteration is a complete sweep of the two-stage problem solution from  $t=1$  to  $T-1$ . POA always takes the computational results of the current two-stage problem as the initial conditions of the next one. After several iterations, the accuracy requirement will be satisfied and the algorithm will locate a solution. The implementation details are introduced step by step as follows:

Step1: Initialize the state vector  $\mathbf{s}_i = (V_{i,1}, V_{i,2}, \dots, V_{i,T})$  for  $i=1$  to  $M$ .

Step2: Fix the state set  $\{V_{i,j} | i=1, \dots, M; j=1, \dots, t-1, t+1, \dots, T\}$  and optimize the one  $\{V_{i,t} | i=1, \dots, M\}$  until the maximization in Eq. 7 achieved. After that,  $\mathbf{s}_i$ 's are updated with new state values of the reservoirs. This process is executed in order from  $t=1$  to  $T-1$ .

Step3: Iteratively execute Step2. When the difference between the state vectors at two adjacent iterations satisfies the specified accuracy requirement, stop the iteration.

### 3.2 Difficulties Encountered by Prototypical POA

#### 3.2.1 Transfer Interrupt Problem

POA improves the quality of solutions by redistributing water between a pair of stages. For certain form of objective functions such as convex functions, POA can achieve free allocation of water between arbitrary two stages (especially two nonadjacent stages) by successively solving several two-stage problems. Howson and Sancho (1975) proved that POA converges to the globally optimal solution when the objective function of the problem is convex. However, for some other forms of objective functions, POA suffers the transfer interrupt problem and cannot ensure the global convergence. Here an empirical analysis is carried out to illustrate this belief.

Let's focus on a single reservoir three-stage optimization problem. It is presumed that  $V_0$  and  $V_3$ , the initial and final storages in the reservoir, are both equal to  $\underline{V}_t$ , the storage lower limit of the reservoir, which is taken to be 5 units;  $Q_1$ ,  $Q_2$  and  $Q_3$ , the inflows to the reservoir, are taken to be the same value of 1 unit;  $\overline{V}_t$ , the storage upper limit of the reservoir, is 10 units;  $\underline{R}_t$  and  $\overline{R}_t$ , the release lower and upper limits of the reservoir are taken to be 0 and 5 units, respectively and the objective function is  $\max (2R_1+R_2+3R_3)$ . This completes the specification of the problem.

Apparently, the optimal release policy for this problem is (0, 0, 3) and the optimal objective function value is 9. Now give POA an initial policy (1, 1, 1) and the corresponding objective function value is 6. Starting at the initial policy, POA will finally locate the policy (2, 0, 1) and the corresponding objective function value is 7. Obviously, the convergence policy is not the optimal one. The search process of POA can be briefly described as follows:

Starting at the initial policy (1, 1, 1), POA first transferred 1 unit of water from stage 2 to stage 1 by solving the two-stage problem in connection with stages 1 and 2 so that the policy (2, 0, 1) produced. After that the water transfer is interrupted

because any amount of water transferred from one stage to its neighbor stage will lead to a decrease of the objective function value. As a result, POA converged to the policy (2, 0, 1) instead of the optimal one.

Comparing POA's convergence policy (2, 0, 1) to the optimal policy (0, 0, 3), it is not difficult to find that in order to produce the optimal policy POA just needs to move 2 units of water from stage 1 to stage 3 on the basis of the convergence policy. But this cannot be achieved due to the transfer interrupt problem.

This case demonstrates that using POA cannot achieve free allocation of water between two nonadjacent stages in some cases.

### *3.2.2 Dimensionality Problem in Two-stage Problem Solution*

POA gets the optimal state combination of the various reservoirs via evaluating the objective function values corresponding to every state combination in the two-stage problem solution. We might as well take time period  $t$  as an example to analyze the computational complexity of POA for the two-stage problem solution. For clarity, assume that arbitrary reservoir  $i$  ( $i=1, 2, \dots, M$ ) has  $N$  discrete states at time period  $t$  and let  ${}^nV_{i,t}$ , where  $n=1, 2, \dots, N$ , denotes the  $n$ th discrete state. Then a complete state of the cascade reservoir system at time period  $t$  can be represented as  $({}^{n_1}V_{1,t}, {}^{n_2}V_{2,t}, \dots, {}^{n_M}V_{M,t})$  where  $n_1, n_2, \dots, n_M=1, 2, 3, \dots, N$ . Considering different discrete states for every reservoir, altogether  $N^M$  combinations are required to evaluate in order to find the optimal one. Therefore, the computational complexity of POA for the two-stage problem solution is  $O(N^M)$ . It can be found that POA suffers serious dimensionality problem with increasing the number of reservoirs.

## **3.3 EPOA-DP**

### *3.3.1 Enhancements to Prototypical POA (EPOA)*

Let's focus on the two-stage problem for cascade reservoir systems shown in Eq. 7.

Taking cascade reservoirs 1, 2, ...,  $i$  as a whole for water balance calculation, we have

$$W_i = R_{i,t} + R_{i,t+1} = \sum_{j=1}^i (V_{j,t-1} - V_{j,t+1} + Q_{j,t} + Q_{j,t+1} - L_{j,t} - L_{j,t+1}) \quad (8)$$

where  $W_i$  denotes the sum of releases from reservoir  $i$  at time periods  $t$  and  $t+1$ . As  $Q_{j,t}$ ,  $Q_{j,t+1}$ ,  $L_{j,t}$ ,  $L_{j,t+1}$ ,  $V_{j,t-1}$  and  $V_{j,t+1}$  are known,  $W_i$  can be calculated. That is, the sum of  $R_{i,t}$  and  $R_{i,t+1}$  keeps constant for any reservoir  $i$  ( $i=1, 2, \dots, M$ ) in the two-stage problem solution of POA. Thus Eq. 7 can be written as the following equivalent form

$$\max_{R_{1,t}, R_{2,t}, \dots, R_{M,t}} F = \left\{ \sum_{i=1}^M [b_i(V_{i,t-1}, R_{i,t}, I_{i,t}) + b_i(V_{i,t}, W_i - R_{i,t}, I_{i,t+1})] \right\} \quad (9)$$

where  $V_{i,t-1}$ 's and  $W_i$ 's are known;  $R_{i,t}$ 's are variables to optimize;  $V_{i,t}$ 's,  $I_{i,t}$ 's and  $I_{i,t+1}$ 's are determined by  $V_{i,t-1}$ 's and  $R_{i,t}$ 's through Eq. 3.

Eq. 9 shows that the essence of POA solving two-stage problems of cascade reservoir systems is to achieve the optimal allocation of  $W_i$  between time periods  $t$  and  $t+1$ . That is, POA always optimizes the values of  $R_{i,t}$ 's and  $R_{i,t+1}$ 's while fixing those of  $R_{i,1}$ 's, ...,  $R_{i,t-1}$ 's,  $R_{i,t+2}$ 's, ...,  $R_{i,T}$ 's at a time in order to achieve the maximization in Eq. 9.

In general, we extend the above principle to arbitrary two time periods such as  $t_1$  and  $t_2$ ,  $t_2=t_1+k$ ,  $t_1=1, 2, \dots, T-1$ ,  $k=1, 2, \dots, T-t_1$  (i.e. let the procedure optimize the values of  $R_{i,t_1}$ 's and  $R_{i,t_2}$ 's while fixing those of  $R_{i,1}$ 's, ...,  $R_{i,t_1-1}$ 's,  $R_{i,t_1+1}$ 's, ...,  $R_{i,t_2-1}$ 's,  $R_{i,t_2+1}$ 's, ...,  $R_{i,T}$ 's in the two-stage problem solution). Thus we have

$$\begin{aligned}
& \max_{R_{i,t_1}, \mathbf{L}, R_{M,t_1}} E = \sum_{i=1}^M [b_i(V_{i,t_1-1}, R_{i,t_1}, I_{i,t_1}) + \sum_{j=t_1+1}^{t_2-1} b_i(V_{i,j-1}, R_{i,j}, I_{i,j}) + b_i(V_{i,t_2-1}, W_i^* - R_{i,t_1}, I_{i,t_2})] \\
& \text{subject to } W_i^* = R_{i,t_1} + R_{i,t_2} = \begin{cases} \sum_{j=1}^i [V_{j,t_1-1} - V_{j,t_2} + \sum_{t=t_1}^{t_2} (Q_{j,t} - L_{j,t})] & \text{if } t_2 - t_1 = 1 \\ \sum_{j=1}^i [V_{j,t_1-1} - V_{j,t_2} + \sum_{t=t_1}^{t_2} (Q_{j,t} - L_{j,t})] - \sum_{t=t_1+1}^{t_2-1} R_{i,t} & \text{if } t_2 - t_1 > 1 \end{cases} \quad (10)
\end{aligned}$$

where  $E$  denotes the benefits from the system over time periods  $t_1 \sim t_2$ ;  $V_{i,t_1-1}$ 's,  $V_{i,t_2}$ 's,  $R_{i,j}$ 's and  $W_i^*$ 's are known;  $R_{i,t_1}$ 's are variables to optimize;  $V_{i,j-1}$ 's,  $V_{i,t_2-1}$ 's,  $I_{i,t_1}$ 's,  $I_{i,j}$ 's and  $I_{i,t_2}$ 's are determined by  $V_{i,t_1-1}$ 's,  $V_{i,t_2}$ 's,  $R_{i,t_1}$ 's and  $R_{i,j}$ 's through Eq. 3.

Thus free allocation of water between two nonadjacent stages is reachable by solving the problem shown in Eq. 10.

### 3.3.2 DP Model for Subproblem Solution of EPOA

In the prototypical POA, reservoir storage  $V_{i,t}$  is taken as the state variable in the two-stage problem solution. Thus the “curse of dimensionality” emerges because of a large number of redundant calculations. By analyzing the characteristics of cascade reservoir systems, it is not hard to find that if we take reservoir release  $R_{i,t}$  as the state variable in the two-stage problem solution of POA, the dimensionality problem can be avoided and the computational efficiency can be improved. Thus a DP model for the sub-problem solution of EPOA is developed as follows.

#### (1) Stage variable

Let the sequence number of cascade reservoir, namely  $i$ , be the stage variable. As EPOA optimizes the values of  $R_{i,t_1}$ 's for  $M$  reservoirs in solving the sub-problem shown in Eq. 10, the total number of stages is  $M$ .

#### (2) State variable

The reservoir release  $R_{i,t}$  is used as the state variable. It is found that the optimal

path after stage  $i$ , namely  $(R_{i+1,t_1}^*, R_{i+2,t_1}^*, \dots, R_{M,t_1}^*)$ , is determined by  $R_{i,t_1}$ , the reservoir state at stage  $i$ , and has nothing to do with the path to  $R_{i,t_1}$  before stage  $i$ , namely  $(R_{1,t_1}, R_{2,t_1}, \dots, R_{i-1,t_1})$ . That is, the selected state variable satisfies the non-aftereffect property of Markov chain.

Thus a complete state for reservoir  $i$  can be written as  $(R_{i,t_1}, R_{i,t_1+1}, \dots, R_{i,t_2-1}, W_i^* - R_{i,t_1})$ . As  $R_{i,t_1+1}, \dots, R_{i,t_2-1}$  are taken to be some fixed values, for clarity, let  $(R_{i,t_1}, W_i^* - R_{i,t_1})$  briefly express the state of reservoir  $i$ . Now we undertake a discretization of the state interval  $[R_{i,t_1}, W_i^* - R_{i,t_1}]$  and constrain  $R_{i,t_1}$  to a set of  $N$  values. Thus the number of states for a stage is  $N$ .

### (3) Decision variable

Simultaneously,  $R_{i,t}$  is taken as the decision variable.

### (4) State transfer equation

According to Eq. 3, the state transfer equation can be written as

$$R_{i,t} = R_{i-1,t} + Q_{i,t} - L_{i,t} + V_{i,t-1} - V_{i,t} \quad (11)$$

### (5) Recurrence equation

We have the recurrence equation as follows

$$\begin{cases} f_i^*(R_{i,t_1}) = \max \left\{ \sum_{t=t_1}^{t_2} b_i(V_{i,t-1}, R_{i,t}, I_{i,t}) + f_{i-1}^*(R_{i-1,t_1}) \right\} \\ f_0^*(R_{0,t_1}) = 0 \end{cases} \quad (12)$$

### (6) Constraints

The Constraints are the same as Eqs. 4~6.

The detailed process for the sub-problem solution of EPOA is shown in Fig. 1.

Figure 1 shows that DP transforms the  $M$ -dimensional sub-problem of EPOA into a  $M$ -stage 1-dimensional DP problem. As  $N^2$  water balance calculations are required

for a stage, the computational complexity of EPOA-DP for the sub-problem solution is  $O(MN^2)$ .

### 3.3.3 Flow Diagram of EPOA-DP

To start the algorithm, initial values are assigned to each parameter as well as variable and an accuracy or precision limit which is the basis of the stopping rule is defined. Generally, the initial values of variables are random generated or gained by experience. After this initialization, the algorithm runs as illustrated in Fig. 2.

## 3.4 Extension of EPOA-DP for Mixed Multi-reservoir Systems

EPOA-DP applies to cascade reservoir systems and cannot be used directly for a general mixed multi-reservoir system. It is found that any mixed multi-reservoir system can be decomposed into several cascade reservoir systems and any operation problem of cascade reservoir system can be solved using EPOA-DP. Hence a hybrid approach combining EPOA-DP and the successive approximations technique (EPOA-DP-SA) is developed for mixed multi-reservoir systems. For simplicity, we take the mixed multi-reservoir system shown in Fig. 3 as an example to briefly introduce the EPOA-DP-SA method. The system consists of 3 reservoirs where reservoirs 1 and 2 are first connected in parallel and then in series with reservoir 3. The mixed system can be decomposed into two cascade subsystems, namely subsystems 1-3 and 2-3. The implementation details of EPOA-DP-SA are introduced step by step as follows:

Step1: Initialize the storage and release sets of reservoirs, i.e.,  $\{V_{i,t}|i=1, 2, 3 \text{ and } t=1, \dots, T\}$  and  $\{R_{i,t}|i=1, 2, 3 \text{ and } t=1, \dots, T\}$ .

Step2: Optimize the release trajectory of subsystem 1-3 using EPOA-DP while

fixing  $\{V_{2,t}|t=1,\dots, T\}$  and  $\{R_{2,t}|t=1,\dots, T\}$ , the storage and release sets of reservoir 2.

Step3: Optimize the release trajectory of subsystem 2-3 using EPOA-DP while fixing  $\{V_{1,t}|t=1,\dots, T\}$  and  $\{R_{1,t}|t=1,\dots, T\}$ , the storage and release sets of reservoir 1.

Step4: Iteratively execute Step2 and Step3. When the difference of storage trajectories at two adjacent iterations satisfies the specified accuracy requirement, stop the iteration.

In general, the implementation details of EPOA-DP-SA for a mixed multi-reservoir system that contains  $m$  cascade subsystems are as follows:

Step1: Initialize the storage and release sets of reservoirs, i.e.,  $\{V_{i,t}|i=1,\dots, M \text{ and } t=1,\dots, T\}$  and  $\{R_{i,t}|i=1,\dots, M \text{ and } t=1,\dots, T\}$ .

Step2: Optimize the release trajectory of subsystem 1 using EPOA-DP while fixing the ones of other reservoirs.

Step3: Optimize the release trajectory of subsystem 2 using EPOA-DP while fixing the ones of other reservoirs.

**N**

Step $m$ : Optimize the release trajectory of subsystem  $m$  using EPOA-DP while fixing the ones of other reservoirs.

Step $m+1$ : Iteratively execute Step2~Step $m$ . When the difference of storage trajectories at two adjacent iterations satisfies the specified accuracy requirement, stop the iteration.

## 4. Case Study

### 4.1 Application to Four-reservoir Problem

A cascade system consists of four reservoirs (reservoirs A, B, C and D) was taken as an illustrative example to test the performance of EPOA-DP. The configuration of the system is shown in Fig. 4 and the characteristic parameters for the various reservoirs are listed in Table 1. The operating period is one year consisting of 12 time periods (months). The object is to maximize the annual power generation from the system as

$$\max \sum_{i=1}^4 \sum_{t=1}^{12} 9.81 \cdot \eta_i \cdot q_{i,t} \cdot H_{i,t} \quad (13)$$

where  $\eta_i$  is the energy conversion efficiency of reservoir  $i$ ;  $q_{i,t}$  is the flows through the turbines for reservoir  $i$  at time period  $t$ ;  $H_{i,t}$  is the water head of reservoir  $i$  at time period  $t$ .

The constraints include the water balance equation, the upper and lower limits of reservoir storage as well as release, the boundary conditions and the upper and lower limits of power output which can be written as

$$\underline{P}_{i,t} \leq P_{i,t} \leq \overline{P}_{i,t} \quad (14)$$

where  $P_{i,t}$  is the power output of reservoir  $i$  at time period  $t$ ;  $\overline{P}_{i,t}$  and  $\underline{P}_{i,t}$  are the upper and lower limits of power output for reservoir  $i$  at time period  $t$ .

Other constraints are the same as Eqs. 3~6. A simulating compute was conducted during 10 scenarios (years) using POA, EPOA and EPOA-DP, independently. The initial trajectory was produced by adjusting the releases so that the storages in each reservoir did not change. The objective function value corresponding to the initial trajectory was  $70.883 (10^8 \text{ kW} \cdot \text{h})$ . The procedures were performed on a Pentium 2.0 MHZ PC and the optimization indexes derived from each procedure for  $N=41$  are given in Table 2.

As shown in Table 2, EPOA-DP and EPOA produced near-equal power generations of the cascade system during 10 years and the absolute values of relative deviations were less than 0.09% (the minor deviations, we believe, were caused by discretization). However, the execution time of EPOA-DP for each year was much less than that of EPOA, about 0.12% of the latter on average. These suggest that the DP model can distinctly lighten the calculation burden while ensuring the overall convergence in the sub-problem solution of EPOA. Compared to POA, the power generation derived from EPOA-DP for each year had a certain increase but the corresponding execution time had a remarkably decrease. The average growth of the power generations was about 0.75% and the biggest and second-biggest growths were 2.37% and 2.00%, respectively appearing in the 5th and 2th years. The execution time of EPOA-DP for each year was much less than that of POA, about 0.60% of the latter on average. These indicate that EPOA-DP can effectively improve the computational efficiency as well as the quality of the solutions of POA. The return, as a function of the number of iterations, for the 5th year has been plotted in Fig. 5 and the corresponding execution time, as a function of  $N$ , has been plotted in Fig. 6.

As shown in Fig. 6, the time curve of EPOA-DP is below those of EPOA and POA with a negligible rising trend. On the contrary, the time curve of EPOA is above those of the other two methods and ascends sharply. Unsurprisingly, the time curve of POA occupies the center region and goes up relatively rapidly. On further analysis, it was found that the computational complexity of either EPOA or POA grows as  $N^4$  but that of EPOA-DP grows as  $N^2$ . As  $N$  increased further, for example to 201, the execution time of either POA or EPOA was more than 24 hours but that of EPOA-DP was just about 17 seconds on the Pentium 2.0 MHZ PC.

## 4.2 Application to Ten-reservoir Problem

A classical ten-reservoir problem was taken as another example to test the performance of EPOA-DP-SA for mixed multi-reservoir systems. The problem was first formulated and solved by Murray and Yakowitz (1979) and more recently by Wardlaw and Sharif (1999) and Ahmadianfar et al. (2017). This problem is complicated not only in terms of size, but also because of many time-dependent constraints on storage. The ten-reservoir system comprises reservoirs in series and parallel, as shown in Fig. 7 wherein a reservoir may receive supplies from one or more upstream neighbors.

The operating period is one year and the object is to maximize power benefits from the system over 12 months. The objective function can be written as

$$\max \sum_{i=1}^{10} \sum_{t=1}^{12} c_{i,t} \cdot R_{i,t} \quad (15)$$

where  $c_{i,t}$  denotes the benefit coefficient of reservoir  $i$  at time period  $t$ . The values of the  $c_{i,t}$ 's were given by Murray and Yakowitz (1979).

Inflows are defined for each of the upstream reservoirs and initial as well as final storages in the various reservoirs are specified at the beginning of the operating period. In addition, the lower and upper limits of storage and release for each reservoir are not allowed to break. The inflow and constraint parameters are available in the work of Murray and Yakowitz (1979). The water balance equation can be written as

$$V_{i,t} = V_{i,t-1} + Q_{i,t} + \sum_{j \in U_i} R_{j,t} - R_{i,t} \quad (16)$$

where  $U_i$  denotes the set of reservoirs which directly supply reservoir  $i$ .

The problem is beyond the capacity of DP and is difficult with POA and EPOA, but is relatively simple with the simplex method. From the simplex method, it was known that the global optimum for the problem is 1194.44. As shown in Fig. 7, the

system contains 6 cascade subsystems, namely subsystems 1-7-10, 2-4-7-10, 3-4-7-10, 5-7-10, 6-7-10 and 8-9-10. EPOA-DP-SA was employed to solve the problem and POA-SA (Zhang et al. 2016) was used for a comparison purpose. The initial trajectory for either method was taken from the work of Murray and Yakowitz (1979) wherein it was termed “the static release policy” and the objective function value for it was 1080.9836. The procedures were performed on the Pentium 2.0MHZ PC. The return of EPOA-DP-SA was 1194.21, about 99.98% of the global optimum, and the execution time was 29.4 s for  $N=201$ . At the same magnitude of discretization, the return of POA-SA was 1175.45, about 98.41% of the global optimum, and the execution time was 1.1 s.

In contrast to POA-SA, the return of EPOA-DP-SA increased by 1.60% but the execution time only increased by 28.3 s. Although the global optimum was not obtained, the return of EPOA-DP-SA was pretty inspiring. The optimal storage trajectories produced from EPOA-DP-SA and the simplex method are shown in Fig. 8. For reservoirs 1, 2, 6, 7, 8, 9 and 10, a good match in trajectories is obtained. In reservoirs 3, 4 and 5, there are minor shifts in trajectories. Figure 9 displays the returns of EPOA-DP-SA and POA-SA after a given number of iterations.

Overall, given the size of the problem, the results of EPOA-DP-SA are satisfactory.

## 5 Conclusions

As a well-known improved version of dynamic programming, POA suffers the transfer interrupt problem and the dimensionality problem. It fails to provide satisfactory scheduling results within a reasonable execution time and memory use. Thus, to overcome these defects, a hybrid approach EPOA-DP was proposed for optimization operation of cascade reservoirs. First an enhanced version of POA

(EPOA) was developed to achieve free allocation of water between any two stages. Then a DP model was built to deal with the dimensionality problem in the two-stage problem solution.

The test results show the superior performance of the proposed approach. The proposed approach is conceptually simple and easy to implement for large scale cascade reservoir operation problems. It provides an avenue toward overcoming the “curse of dimensionality” for multi-reservoir control.

## **Declarations**

**Funding:** No funding was received for conducting this study.

**Conflict of interest:** No conflict of interest.

**Availability of data and material:** The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

**Ethics approval:** Not applicable.

**Consent to participate:** Not applicable.

**Consent for publication:** Not applicable.

**Authors' contributions:** the whole work was independently accomplished by JC, including conceiving the approach, designing and performing the experiments, analyzing the data and results and writing the paper.

**Compliance with Ethical Standards.**

## **References**

Ahmadianfar I, Samadi-Koucheksaraee A, Bozorg-Haddad O (2017) Extracting optimal policies of hydropower multi-reservoir systems utilizing enhanced differential evolution algorithm. *Water Resour Manag* 31(10):4375–4397

- Bai T, Chang J, Chang F et al (2015) Synergistic gains from the multi-objective optimal operation of cascade reservoirs in the Upper Yellow River basin. *Journal of Hydrology* 523:758–767
- Catalao J, Pousinho H, Mendes V (2010) Scheduling of head-dependent cascaded hydro systems: mixed-integer quadratic programming approach. *Energy Convers Manag* 51(3):524–530
- Cheng C, Shen J, Wu X et al (2012) Short-term hydro scheduling with discrepant objectives using multi-step progressive optimality algorithm. *J Am Water Resour As* 48(3):464–479
- Cheng C, Wang S, Chau KW et al (2014) Parallel discrete differential dynamic programming for multireservoir operation. *Environ Model Softw* 57:152–164
- Feng Z, Niu W, Cheng C et al (2017) Hydropower system operation optimization by discrete differential dynamic programming based on orthogonal experiment design. *Energy* 126:720–732
- Feng Z, Niu W, Jiang Z et al (2020a) Monthly operation optimization of cascade hydropower reservoirs with dynamic programming and Latin hypercube sampling for dimensionality reduction. *Water Resour Manag* 34(6):2029–2041
- Feng Z, Niu W, Zhou J et al (2020b) Linking nelder–mead simplex direct search method into two-stage progressive optimality algorithm for optimal operation of cascade hydropower reservoirs. *J Water Plan Manag ASCE*. doi:10.1061/(ASCE)WR.1943-5452.0001194
- Heidari M, Chow V, Kokotovic Pet al (1971) Discrete differential dynamic programming approach to water resources systems optimization. *Water Resour Res*

7(2):273–282.

Howson H, Sancho N (1973) A two-stage algorithm for sequential decisions problems. *Infor II* (2):163–176

Howson H, Sancho N (1975) A new algorithm for the solution of multi-state dynamic programming problems. *Math Programming* 8(1):104-116

Jahandideh-Tehrani M, Bozorg-Haddad O, Loaiciga HA (2020) Application of particle swarm optimization to water management: an introduction and overview. *Environ Monit Assess.* doi:10.1007/s10661-020-8228-z

Ji C, Jiang Z, Sun P et al (2014) Research and application of multidimensional dynamic programming in cascade reservoirs based on multilayer nested structure. *J Water Resour Plan Manag.* doi:10.1061/(ASCE)WR.1943-5452.0000489

Jothiprakash V, Arunkumar R (2014) Multi-reservoir optimization for hydropower production using NLP technique. *KSCE J Civ Eng* 18(1): 344–354

Kang C, Guo M, Wang J (2017) Short-term hydrothermal scheduling using a two-stage linear programming with special ordered sets method. *Water Resour Manag* 31(11):3329-3341

Larson R, Korsak A (1970) A dynamic programming successive approximations technique with convergence proofs. *Automatica* 6(2):245–252

Lee Y, Kim S, Ko I (2008) Multistage stochastic linear programming model for daily coordinated multi-reservoir operation. *J Hydroinform* 10(1):23–41.

Ma Y, Zhong P, Xu B (2020) Multidimensional parallel dynamic programming algorithm based on spark for large-scale hydropower systems. *Water Resour Manag* 34(3):3427–3444

- MURRAY D, YAKOWITZ S (1979) Constrained differential dynamic programming and its application to multireservoir control. *Water Resour Res* 15(5):1017–1027
- Opan M (2011) Real-time optimal operation of multiple reservoirs system. *Teknik Dergi* 22(2):5359–5385
- Rani D, Mourato S, Moreira M (2020) A generalized dynamic programming modelling approach for integrated reservoir operation. *Water Resour Manag* 34(4):1335–1351
- Shaikh, S. A. (2020) Application of artificial neural network for optimal operation of a multi-purpose multi-reservoir system, II: optimal solution and performance evaluation. *Sustain. Water Resour Manag*. doi:10.1007/s40899-020-00423-6
- Wardlaw R, Sharif M (1999) Evaluation of genetic algorithms for optimal reservoir system operation. *J. Water Resour. Plann. Manag* 125 (1): 25–33
- Zhang W, Liu P, Chen X et al (2016) Optimal operation of multi-reservoir systems considering time-lags of flood routing. *Water Resour Manag* 30(2):523–540
- Zhao T, Zhao J, Lei X et al (2017) Improved dynamic programming for reservoir flood control operation. *Water Resour Manag* 31(7):2047–2063

### **Figure Captions**

**Fig. 1** Sketch for sub-problem solution of EPOA by DP

**Fig. 2** Flow diagram of EPOA

**Fig. 3** Sketch for a simple mixed multi-reservoir system

**Fig. 4** Sketch for the four-reservoir cascade system

**Fig. 5** Work progresses of POA, EPOA and EPOA-DP in the 5th year

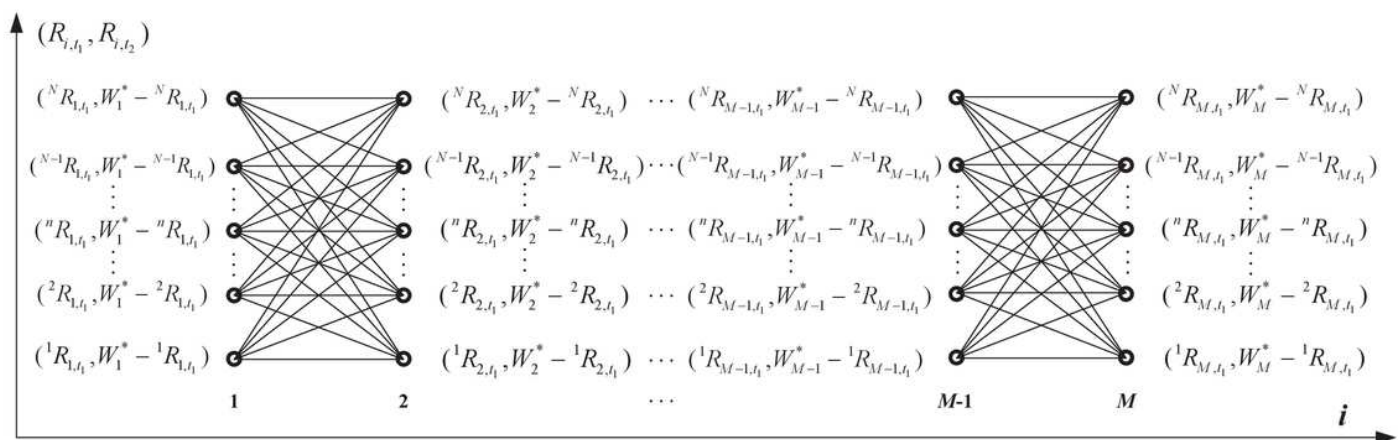
**Fig. 6** Execution times for POA, EPOA and EPOA-DP in the 5th year

**Fig. 7** Sketch for the ten-reservoir system

**Fig. 8** Storage trajectories produced from EPOA-DP-SA and the simplex method

**Fig. 9** Work progresses of EPOA-DP-SA and POA-SA

# Figures



**Figure 1**

Sketch for sub-problem solution of EPOA by DP

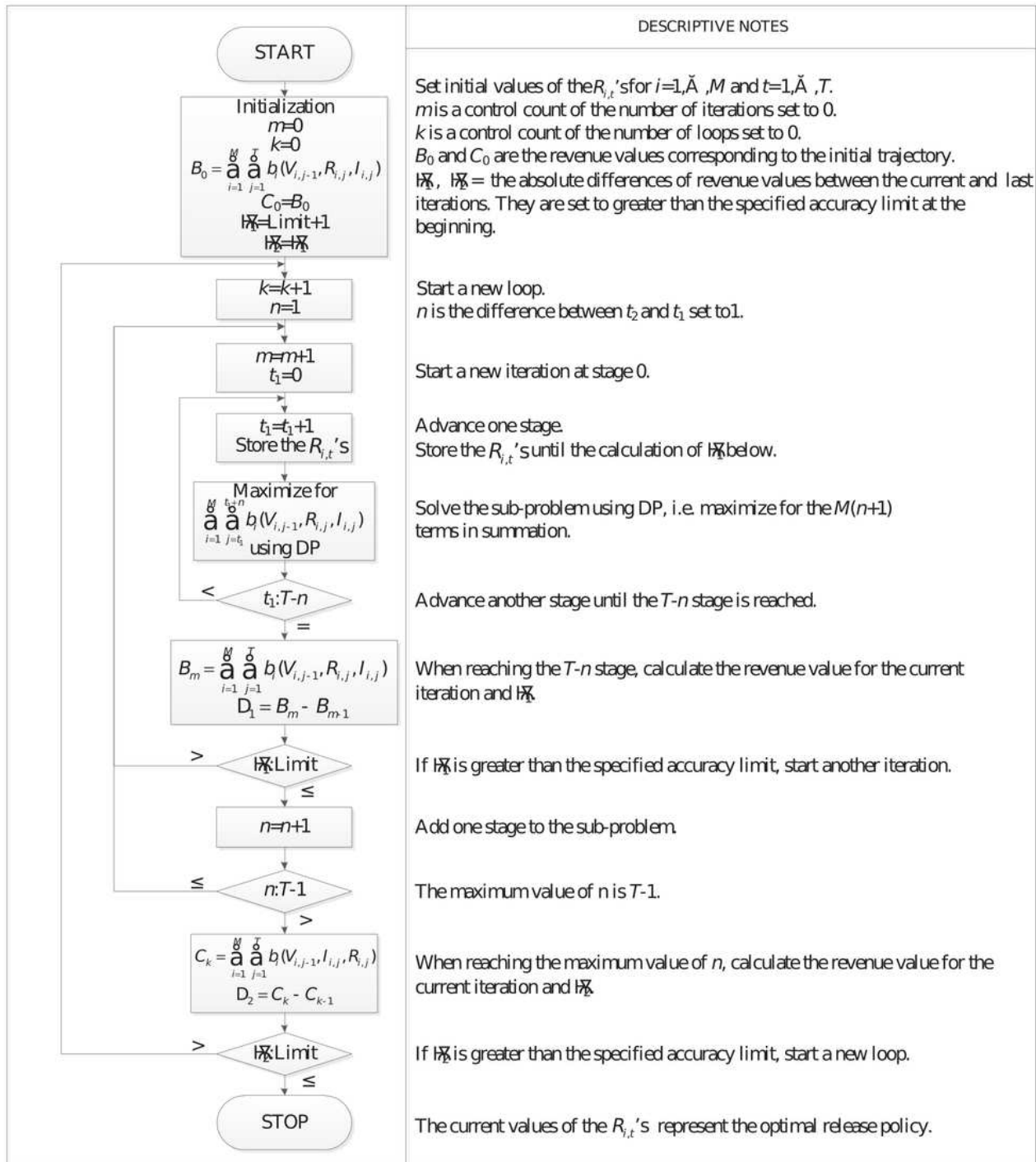
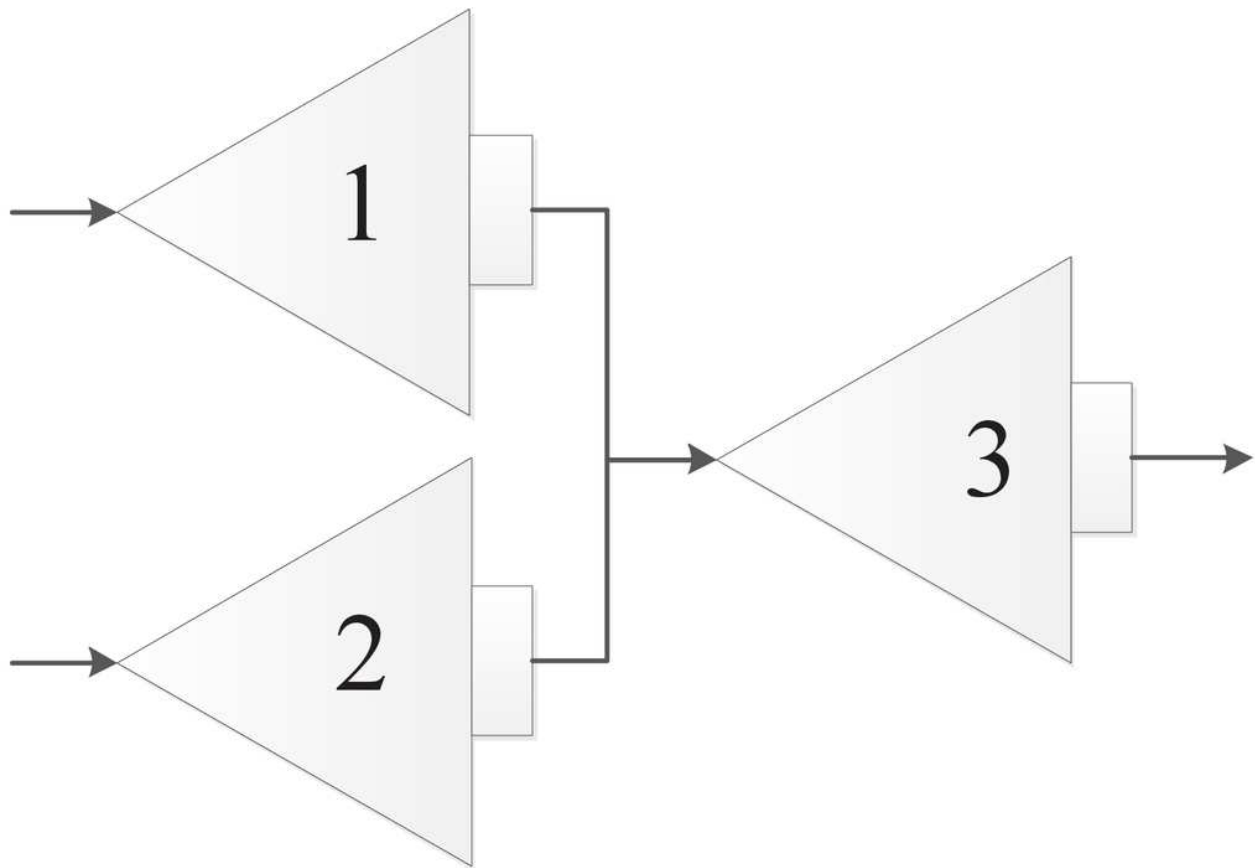


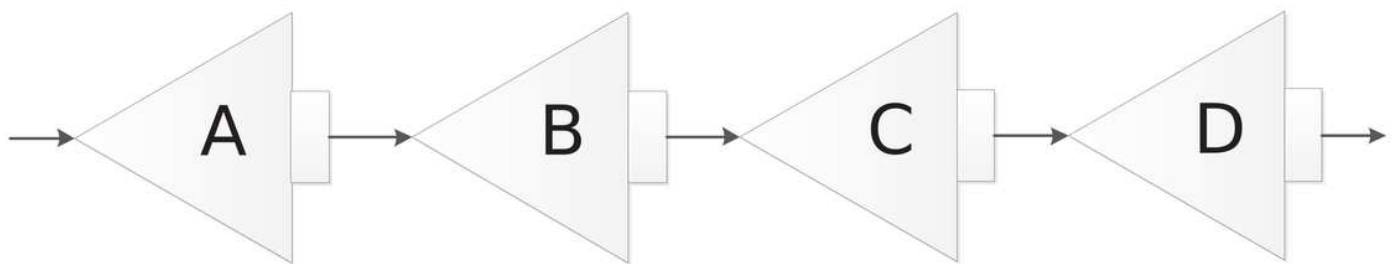
Figure 2

Flow diagram of EPOA



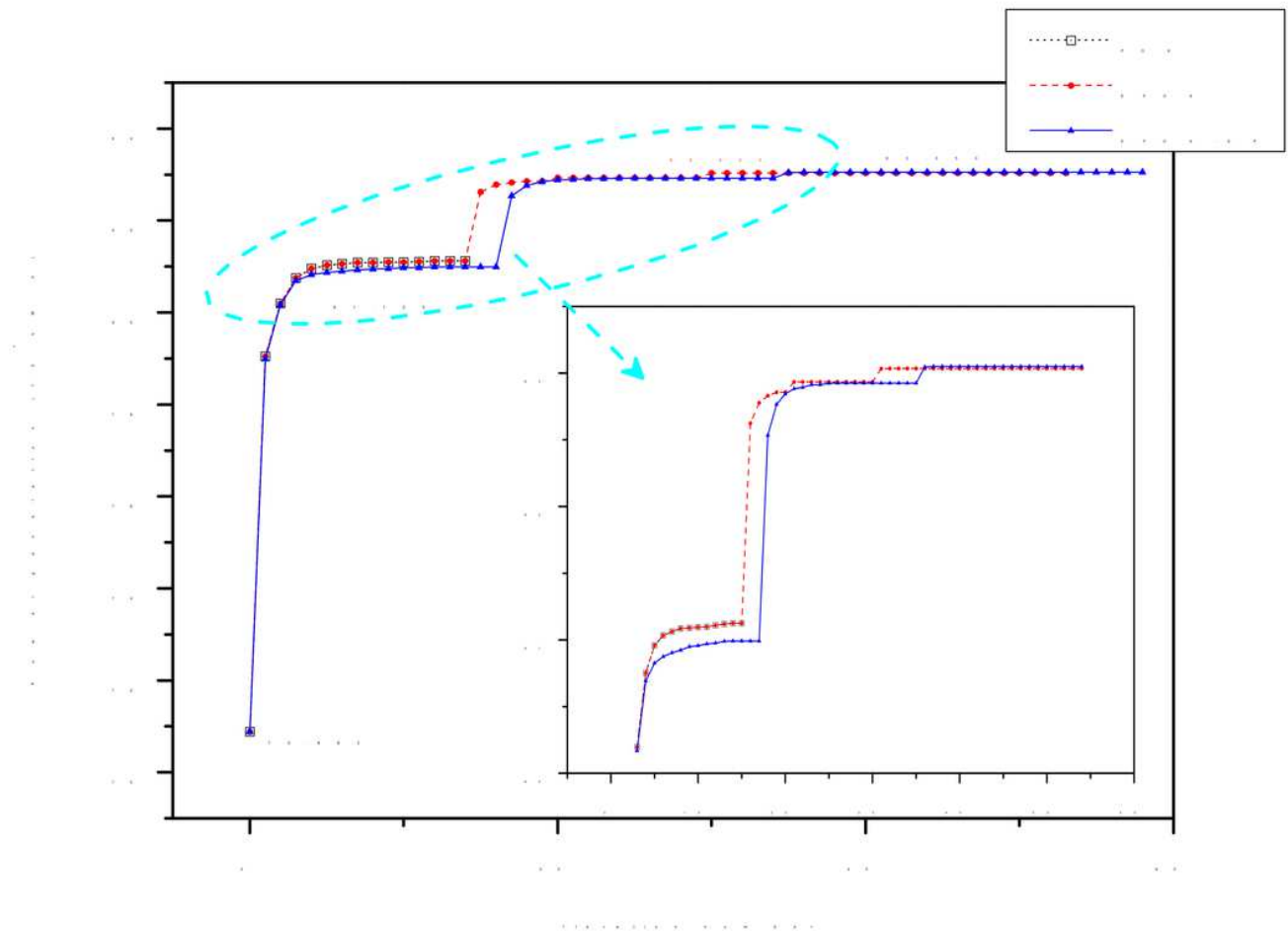
**Figure 3**

Flow diagram of EPOA



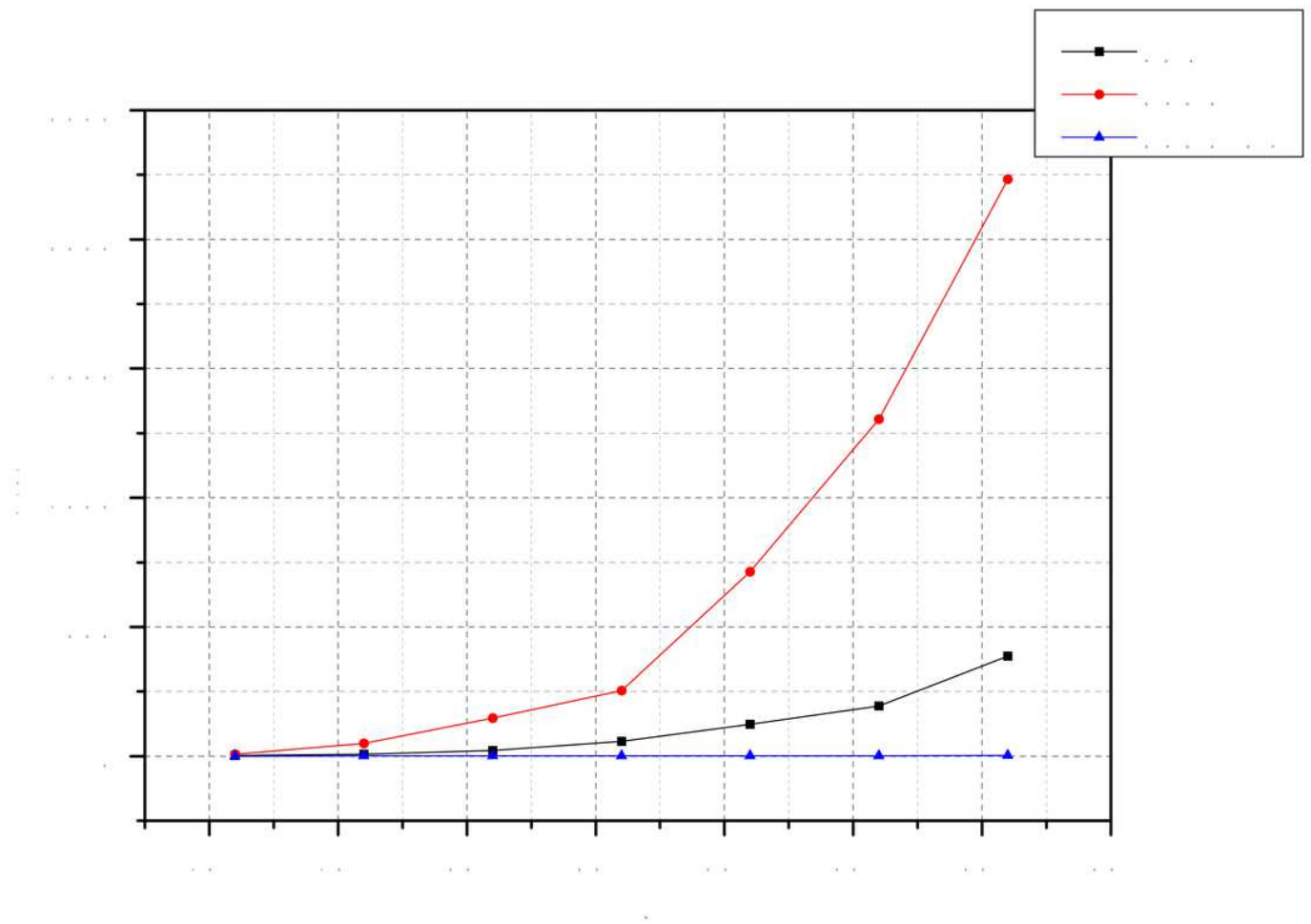
**Figure 4**

Sketch for the four-reservoir cascade system



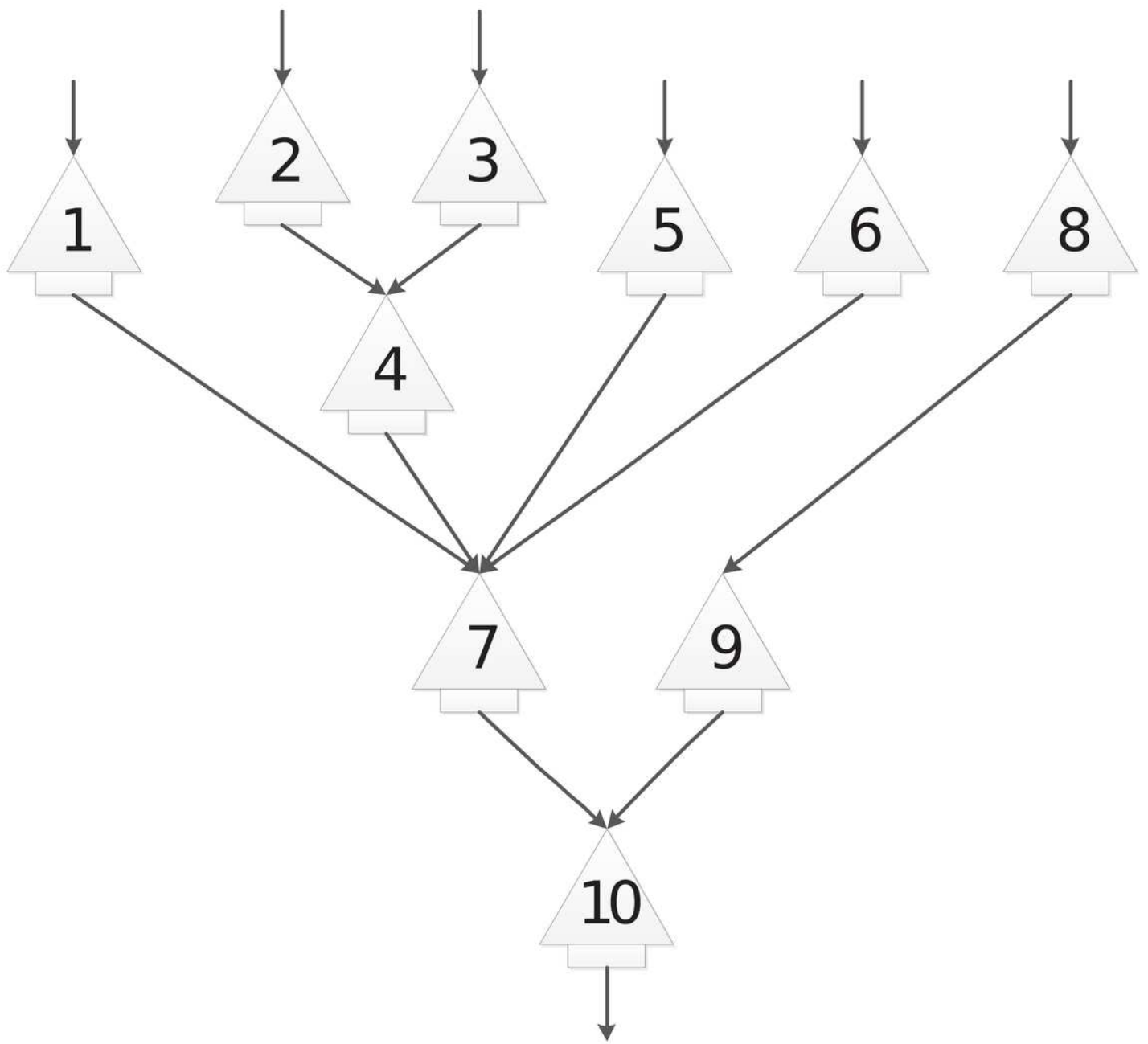
**Figure 5**

Work progresses of POA, EPOA and EPOA-DP in the 5th year



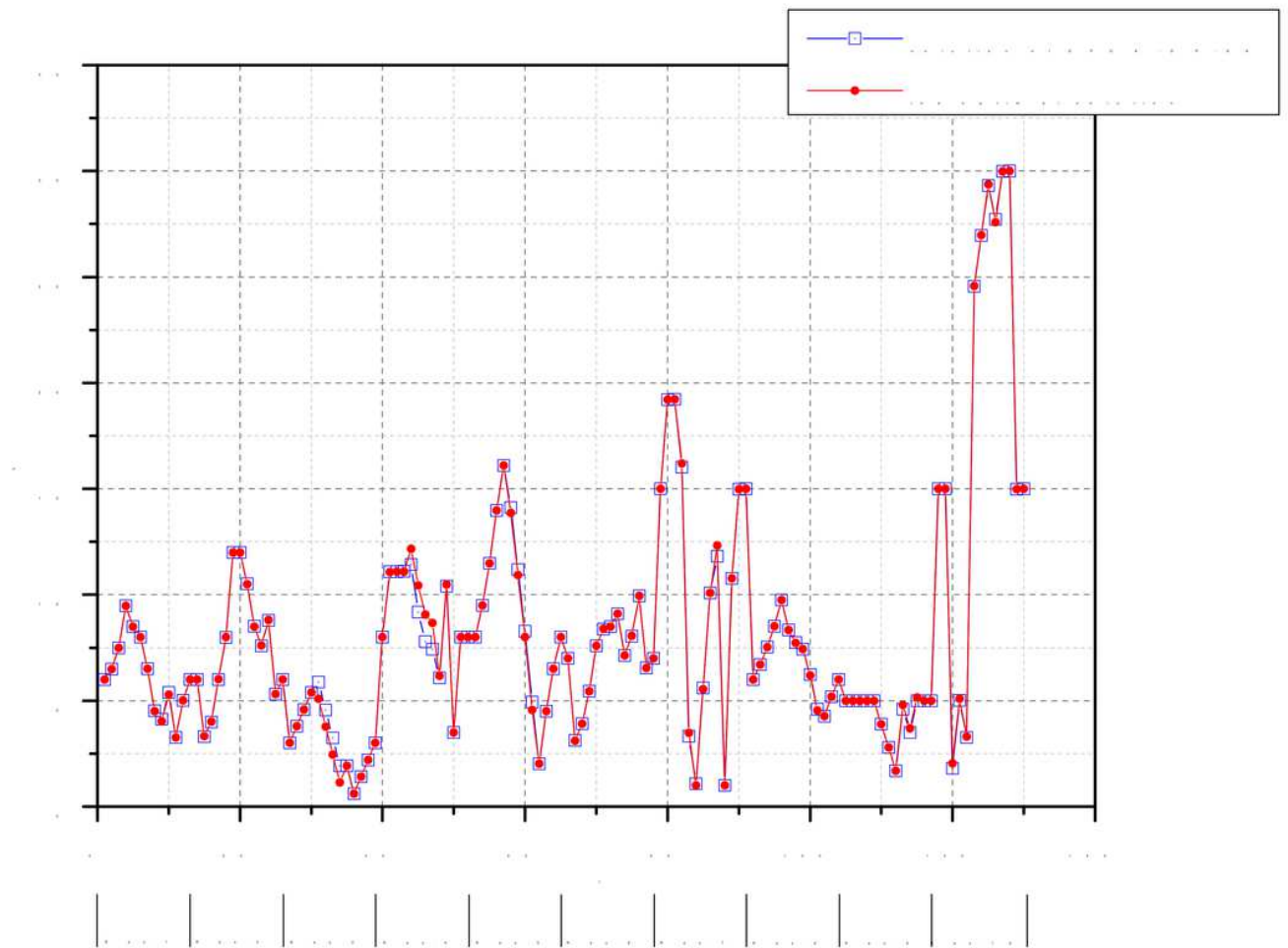
**Figure 6**

Execution times for POA, EPOA and EPOA-DP in the 5th year



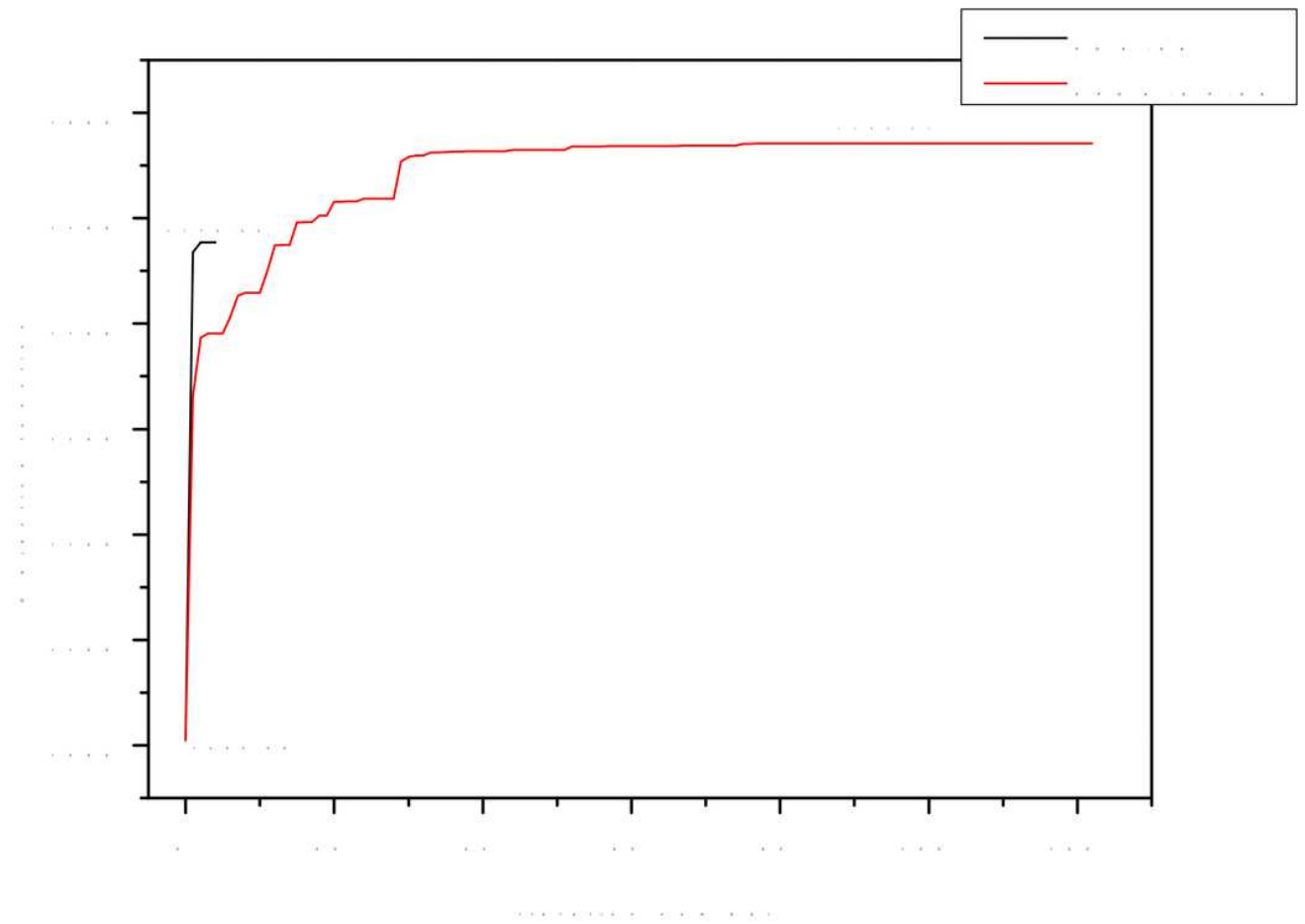
**Figure 7**

Sketch for the ten-reservoir system



**Figure 8**

Storage trajectories produced from EPOA-DP-SA and the simplex method



**Figure 9**

Work progresses of EPOA-DP-SA and POA-SA