Medium and Long-Term Stochastic Optimization of Hybrid Pumped Storage Reservoir via Reinforcement Learning Method

Daniel Eliote Mbanze[1,2], Li Wenwu[1,2], Zhang Xueying[1,2]

1. Hubei Key Laboratory of Cascaded Hydropower Stations Operation & Control, China Three Gorges University, Yichang 443002, China.

2. College of Electrical Engineering & New Energy, China Three Gorges University, Yichang 443002, China.

Correspondent Author: Daniel Eliote Mbanze, College of Electrical Engineering & New Energy, China Three Gorges University, Yichang 443002, China, Tel +8615571773649, Email dmbanze12@gmail.com

DOI: 10.29322/IJSRP.8.11.2018.p8309
http://dx.doi.org/10.29322/IJSRP.8.11.2018.p8309

Abstract

Due to the great advantage in regulating the peak level in wet season and compensating the storage deficit in dry season, the hybrid pumped storage reservoir gets more attention from researchers in the scientific world. In this paper, the mid-long term stochastic optimization model for hybrid pumped storage reservoir is established to obtain the maximum energy generation based on Reinforcement Learning Algorithm. The case study simulates the designed model and decide the operational policies to one hybrid pumped storage reservoir. The results show that by applying the Stochastic Dynamic Programming (SDP) algorithm for solving the reservoir optimization problem, an annual production of **4435** ***GWh*** can be achieved while by applying Reinforcement Learning (Q Learning and SARSA) algorithm, gives an annual production of **4418*GWh*** and **4355** ***GWh***, respectively. The Q Learning and SARSA algorithms have a deviation error of ***0.38%*** and ***1.8%*** respectively, compared with the result obtained by SDP algorithm. The operation of the reservoir with pumping system increase the guaranteed output energy instead of operation without pumped unities. Furthermore, the Simulation results proved that by applying Reinforcement Learning method, the computation time decay **55.7%** compared to SDP demanded computation time.

Key Words: *Reinforcement Learning, hybrid pumped storage unity, Markov Decision Process, Reservoir Operation, Stochastic Optimization.*

Introduction

Hybrid pumped storage unity is a reverse pumping system that can increase the volume stored in the upstream basin, improving the efficiency of energy generation from the hydropower station. Stochastic Dynamic Programming (SDP) is the most widely used and recommended algorithm for solving optimization problems in water resource systems. Its strength compared to others is the ability to assume uncertainties which are the main characteristics of runoff in water resources system. However, many researchers have proved that stochastic dynamic programming suffers from curses of dimensionality. For complex problems such as multi-reservoir system operations, the combination of feasible discrete system states grows exponentially with the number of state variables, ultimately overwhelming even the fastest modern CPU processors and DRAM memory capacities [1-5]. Dynamic programming approach was introduced by Bellman in 1957 to solve optimization problems that involved non-linear multi-stage decision problems. In the reservoir operation problem, the objective function of optimization model seeks to maximize the expected energy production over the entire planning horizon. In Dynamic Programming formulation for reservoir operation problem, time is often considered as stage and the volume of water stored in the reservoir at the beginning of the time periods represents the state of the system. The decisions to be taken at each stage are the quantities of water to be released. Since uncertainty is the main characteristic of runoff in a river basin, it is often inadequate to opt for a deterministic decision model at both planning and operational stages.[6,7] To incorporate the uncertainty of natural stream flows, the Stochastic Dynamic Programming (SDP) model is used. Runoff is assumed as a Markovian process and the state transition probabilities are computed. Reinforcement learning has emerged as an effective approach in solving sequential decision problems by combining concepts from artificial intelligence and operations research.[8,9] A Reinforcement Learning system has a mathematical foundation similar to dynamic programming and Markov Decision Processes. It's main objective is the maximization of the long-term return as conditioned on the state of the system environment and the immediate reward obtained from operational decisions.[10-16] Reinforcement Learning Method based on Q-learning algorithm and fitted Q-iteration algorithm were applied in designing of operation policies in single and multi-reservoir system considering the energy market price variation.[1,2,9] In this study, Reinforcement Learning method based on State-Action-Reward-State-Action (SARSA) algorithm is proposed to design the operation policy in the long term scheduling of hybrid pumped reservoir system. It is aimed that it would maximize the expected annual energy generation. In this article we present and discuss our topic in four main sections. The introduction part discusses the scope of this article and review the existing and published research papers about this amazing field with special application to energy production. Sections 1 and 2 presents the SDP and Reinforcement Learning models for hybrid pumped reservoir system, respectively. In section 3, the simulations and results analysis of reservoir operation problem are presented. Finally, we present the conclusion and important references to this work.

1. SDP Model for Hybrid Pumped Reservoir

Hybrid pumped storage is an arisen generation form in recent years. The hybrid pumped storage power station is composed by adding reversible pumped storage units to conventional hydropower stations. So it is different from conventional hydropower station and pure pumped storage power station. This reservoir includes advantages of both conventional hydropower units and pumped storage units[14]. Hybrid pumped storage power station not only can turn redundant energy under low load to higher price energy under peak load, but also can undertake many significant functions of frequency modulation, phase modulation and emergency reserve. It is also applied to steady cyclic wave and voltage in power system[17]. Due the multiple functions of hybrid pumped storage power station in power system, it is necessary to design the optimal operational policies to make full advantages of the hybrid pumped storage power station[9]. The Markov Decision Process (MDP) Model for the hybrid pumped storage power station can be written as follows:

$$E^* = \max \sum_{t=1}^{T} R_t(s_t, Q_t) = \max \sum_{t=1}^{T} \sum_{s'}^{S} p(s_{t+1} = s' / s_t = s, a_t = a) E_t\left(s_t, q_t, Q_t, Qc_t\right) \tag{1}$$

where:

$$\begin{cases} E_t\left(s_t, q_t, Q_t, Qc_t\right) = k\eta Q_t H_j \Delta T_t - Ec_t(tc_t) \\ R_t(s_t, Q_t) = \sum_{s'}^{S} p(s_{t+1} = s' / s_t = s, a_t = a) * E_t\left(s_t, q_t, Q_t, Qc_t\right) \end{cases} \tag{2}$$

Subjected to:

$$s_{t+1} = s_t + \left[q_t - Q_t - sp_t\right] * \Delta T_t + Qc_t * tc_t \tag{3}$$

$$s_t^{\min} < s_t < s_t^{\max} \tag{4}$$

$$Q^{\min} < Q_t < Q^{\max} \tag{5}$$

$$z_t^{\min} < z_t < z_t^{\max} \tag{6}$$

$$s_t = f(z_t) \tag{7}$$

$$P_t^{\min} < P_t < P_t^{\max} \tag{8}$$

Where E^* is the total expected energy generation over the optimization horizon, $E_t(s_t, q_t, Q_t, Qc_t)$ is generated energy which represents the immediate reward function, Ec_t is the energy demanded by the hybrid storage pumped unit, tc_t is the pumping time in hour, Qc_t represents the amount of water to be pumped in m^3/s during the time tc$_{t.}$, s_t (m^3) is the initial water storage at time t, q_t (m^3/s) is the inflow at time t, $s_t^{\min}$ and $s_t^{\max}$ (m^3) are the minimum and maximum bounds on storage level during time t, Z_t (m) represents the reservoir water level. Q_t represents the water release (m^3/s) in each time t. $Q^{\min}$ and $Q^{\max}$ (m^3/s) denote the minimum and maximum bounds on reservoir release subjected to physical constraints, respectively, $p(s_{t+1}=s'/s_t=s, a_t=a) = p(s,a,s')$ denotes the state transition probability from state $s_t = s$ at time t to state $s_{t+1} = s'$ at time $t+1$ by taking the action $a_t = a$, η is overall hydropower efficiency which is assumed to be 0.9, k is a hydropower parameter which is equal to 8.5, $H_j = Z_{\mathrm{sy}} - Z_{\mathrm{xy}} = (Z_t + Z_{t+1})/2 - Z_{\mathrm{xy}}$ is the water level (m) which is given by the relation curve between up water level Z_{sy} and down water level Z_{xy}, P_t represents the power demand at time t, ΔT_t is the operation time in hours, sp_t is the spilled water in m^3/s due to restrictions on reservoir storage space, turbine release discharge outlet capacity, and downstream channel capacity.

$$sp_t = \begin{cases} Q_t - Q_t^{\max} & if \quad Q_t > Q_t^{\max} \\ 0 & if \quad Q_t \le Q_t^{\max} \end{cases} \tag{9}$$

The constraint (3) refer to the mass balance equation of the reservoir. Restrictions (4), (5), (6) and (9) represent the operating limits of the reservoir basin, turbine and storage level constraints. The storage level is a non linear function defined as a function of reservoir water level which is described by constraint 7. The bounds on power demand in a hydroelectric plant is represented by constraint 8. The energy generation is a non linear function which depends on the turbine release and the height of the water level, equation 2. The Stochastic Dynamic Programming approach has the characteristics of easily representing non-linearity and stochastic aspects of optimization problem with uncertainty. However, this approach requires the discretization of the state space of the problem variables. In the SDP model, the problem is divided into stages (months), and the best decision (release) at each stage is determined according to the state (storage) in which the system is located. The optimization process is based on previous knowledge of future possibilities and their consequence in order to satisfy the Bellman's optimality principle[6,11]. Since the problem is stochastic, the decision at each stage is obtained on the basis of the probability distribution of random variables at the respective stage. In SDP there are two different ways for finding the optimal policy and value functions namely: policy iteration and value iteration[7-9]. In this study we applied value iteration to define the optimal scheduling strategy of reservoir operation problem. If the iterative process of updating the value function is repeated infinite steps, the policy and corresponding value functions will converge to an optimal and steady state points where there's no more improvement in the policy and value functions. The value iteration version of the SDP is initialized as a random value function for all possible states in the last period T+1 and continues by updating these values iteratively with a recursive function. In reservoir operation problem, the runoff at time period t is sampled according to the specific Probability Distribution Function (PDF). The storage is discretized into S intervals from minimum to maximum bounds. The state transition probability is also computed from the observed streamflow series. The recursive Bellman equation for updating the value function is made by employing Equation 10. The optimal solution is found after several iterations where the value function converges to a single value:

$$V_t(s_t,q_t)=\max_{a_t}\left\{r_t\left(s_t,s_{t+1},a_t\right)+\gamma\sum_{s'}p(s_{t+1}=s'/s_t=s,a_t=a,)V_{t+1}(s_{t+1},q_{t+1})\right\} \tag{10}$$

Subjected to the constraints described in equations 3 to 9. In equation 10, the variable a_t is a control decision to be selected from a given set of admissible actions $\left(a_t\in A\right)$ and $\gamma\in[0,1]$ is discount factor.

2. Reinforcement Learning Model

Reinforcement Learning is an optimization tool that is widely applied in solving artificial intelligence control problems. The main objective is to maximize the long term return. In this study, Reinforcement Learning method is applied in solving reservoir scheduling problem. The agent interacts with the environment, selecting actions and receiving rewards from the system environment which is the scalar value to be maximized over time. In the reservoir operation problem, the system environment is stochastic and the agent must make a decision in discrete time horizon. In each time *t=1, 2, 3...T*, the agent is in state s_t and must take the action a_t from a set of available discrete actions. The executed action a_t in state s_t will lead the system to transit to the next state s_{t+1} and the cycle is repeated. The set of all discrete state is known as state space and denoted by S ($s_t\in S$) and the set of discrete actions is denoted by A ($a_t\in A$). In each state transition s_t to s_{t+1}, the agent receive a reward value $r_t\in R_t$. For each time step t, the agent is in a specific state $s_t\in S$, here S is a set of all possible discrete state. The agent select action $a_t\in A$, where A is the set of all possible discrete actions available in state s_t. In the next step, according to the action taken in time t, the agent receive the reward r_{t+1} and now is in state s_{t+1}. The goal of the agent is to maximize the reward of the system. This paper discusses discounted reward, where $\gamma\in[0,1]$ is defined as the discount factor. The discounted reward of policy π is given as:

$$V^\pi=\gamma^0r_0+\gamma^1r_1+\gamma^3r_3+...+\gamma^tr_t=\sum_{t=0}^{\infty}\gamma^tr_t \tag{11}$$

Where r_t is the reward observed in time t. Furthermore $V(s_t)$ is defined as a value function in each state s_t. Under policy π, the value function can be defined as the expected value in long-term scheduling:

$$V^\pi(s_t)=E_\pi\{r_t/s_t=s\}=E_\pi\left[\sum_{k=0}^{\infty}\gamma^kr_{t+k+1}/s_t=s\right] \tag{12}$$

Where the expectation E_π is over a run of policy π starting at state s_t. Another important parameter defined is a state action value function:

$$Q_t(s_t,a_t)=r_t(s_t,s_{t+1},a_t)+\gamma\sum_{s'}p(s_{t+1}=s'/s_t=s,a_t=a)V^\pi(s_t) \tag{13}$$

whose value is the return of initially performing an action a_t at state s_t following the policy π. In the following sections we will represent the state transition probability $p(s_{t+1}=s'/s_t=s,a_t=a)$ as $p(s,a,s')$. Let π^* be an optimal policy which maximizes the value function from any start state s_t. For any policy π and any state s_t we have $V^{\pi^*}(s_t)\geq V^\pi(s_t)$ and the optimal policy in each state s_t is defined as:

$$\pi^*(s_t) = \arg\max_{a_t}(r_t(s_t,a_t) + \gamma\sum_{s'} p(s,a,s')\max_{a_{t+1}}(Q_{t+1}(s_{t+1},a_{t+1}))) \quad \textbf{(14)}$$

In this step, the optimal policy is the only one fixed point of the operator $Q_t(s_t,a_t) = (r_t(s_t,a_t) + \gamma\sum_{s'} p(s,a,s')\max_{a_{t+1}}(Q_{t+1}(s_{t+1},a_{t+1}))$. Thus, it can be said that a policy π is a valid approximation of the optimal policy $\pi *$ if $\left\| V^{\pi^*} - V^{\pi} \right\| \leq \xi$, where ξ is a small value.

2.1. Q Learning Algorithm

Q-Learning is the most popular reinforcement Learning algorithm where the agent learns iteratively the optimal policy $\pi *$ when the model of the system is unknown. The Q learning was developed in 1989 by Chris Watkins' where the temporal-difference and optimal control threads were fully incorporated together.Bellman's equation can iteratively be computed using the optimal action value function or Q function:

$$Q^{\pi}(s_t,a_t) = r\left(s_t,a_t\right) + \gamma V^{\pi}(s_{t+1}) \quad \textbf{(15)}$$

$$V_t(s_t) = \max_{a_t} Q_t(s_t,a_t) \quad \textbf{(16)}$$

The value iteration method is recursively applied for each period, whereby equation **(10),** subjected to equations (3 to 9) is repeatedly solved in each cycle (t =1, 2, 3…T) as backward computation. Q-learning is an online RL algorithm; it only uses its last experience to update its policy. The algorithm starts with an arbitrary Q-function. For each pair (state-action), the Q-function is updated by iteratively computing equation 17:

$$Q_t(s_t,a_t) = Q_t(s_t,a_t) + \alpha\left\{ r_t(s_t,a_t) + \gamma(\max_{a} Q_{t+1}(s_{t+1},a_{t+1}) - Q_t(s_t,a))\right\} \quad \textbf{(17)}$$

Where $\alpha \in [0,1]$ is the learning rate, γ is the discount factor, $r_t(s_t,a_t)$ is the immediate reward by taking action a_t at state s_t , t represents the discrete time, $\max_{a}(Q_t(s_{t+1},a))$ is the Q value which corresponds to the action a with high utility in the future. The update process of equation (17) continues for K episodes, or until optimal policies $\pi^*\left(s_t\right)$ are stationary or at steady state. In this step, $V^{\pi}(s_t)$ is not improving after several episodes in any period t. The state value function and the optimal policy in any period t can be computed by using equations 18 and 19 respectively:

$$V_t(s_t) = \max_{a_t}(Q_t(s_t,a_t) \quad \textbf{(18)}$$

$$\pi^*(s_t) = \arg\max_{a_t}(Q_t(s_t,a_t)) \quad \textbf{(19)}$$

During the update process, the start action a_t used in equation (17) is selected randomly based on specific probability distribution such as e-greedy. The action applied during the iterative process of updating the value function, can be selected based on exploration or exploitation strategies. In this paper we apply e-greedy policy for action selection. ε -greedy policy is a random exploration where the agent can execute the action a_t that returns a high Q-value with probability $1-\varepsilon$ by choose a random action with probability ε . In order to guarantee the convergence of the Q learning, the algorithm must be accurate enough so that each pair of state-action value can be visited several times to update its value function.

http://dx.doi.org/10.29322/IJSRP.8.11.2018.p8309

2.2. SARSA Algorithm

State-Action-Reward-State-Action (SARSA) Algorithm is called Model Dependent or Model Based Reinforcement Learning while Q-Learning is known as Model-free Reinforcement Learning. The major difference between SARSA and Q Learning is that in SARSA algorithm the maximum reward for the next state is not necessarily used for updating the Q-values, instead, a new action a_t, and therefore reward $r_t(s_t, a_t)$ is selected using the same policy that determined the original action. In model-free learning, the agent simply relies on some trial-and-error experience for action selection[10,16]. The SARSA algorithm is based in the following statement: " if the agent is in state s_t, the action a_t is selected according to e-greedy policy, and the system give the reward $r_t(s_t, a_t)$ and then transits to the next state s_{t+1}, after which an action a_{t+1} is selected based on the same policy".[16] On the other hand, the SARSA algorithm is an On-Policy algorithm for Time Difference Learning (the value of being in state s_t is based on the same policy for choosing action a_t as we used to determine the action a_{t+1} out of the next state s_{t+1}). In SARSA algorithm, the update rule performed for each step time is:

$$Q_t(s_t, a_t) = Q_t(s_t, a_t) + \alpha\{r_t(s_t, a_t) + \gamma Q_{t+1}(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)\} \tag{20}$$

Where the value function is compute using $V(s_t) = \max\limits_{a_t} Q_t(s_t, a_t)$. The update process of the value function is done after several transitions and visits of the non-terminal state s_t. If the state s_{t+1} is the terminal state, the initial value of $Q_{t+1}(s_{t+1}, a_{t+1})$ is set to zero. The learning rate is a small step-size parameter that controls the speed of changes in the estimation of the action value function.[1,10] If the step-size parameter is reduced properly over time, reinforcement learning method converges to an optimal value. The iteration starts with higher learning rate which allows fast changes on Q values and then gradually decreases accordingly as time progresses. If every state-action pair is often visited infinitely and the learning rate is decreased over time, the Q values converge to $Q*$ with a probability of one.

$$\alpha = \frac{1}{N^{\Psi}} \tag{21}$$

Where: N is the number of iteration representing the age of the agent, Ψ Parameter $\Psi \in [0.5,1]$

3. Case Study

The proposed methodology has been applied in the design of control policies for reservoir operation problem. Stochastic Dynamic Programming, Q-Learning and SARSA algorithms are simulated using MATLAB programming and compared the results performance. The state of the system is defined as the water level and the action to be taken in each time is the amount of water to be released from the reservoir in m^3/s. We also define the optimal operating time to the hybrid pumping unity. The performed results of each method are analyzed in this section. The reservoir is multipurpose operation and multi-year regulation. The hydropower station has an installed capacity of 1500 MW, divided into five units with equal capacity. The reservoir adds two installed reversible hybrid units of 30 MW to pump the water from downstream to the upstream reservoir. Its normal storage level is 413m with a corresponding storage capacity of $4.967 \times 10^9 m^3$. The dead water level is 380 m which is equivalent to the dead storage capacity of $2.024 \times 10^9 m^3$. The main objective of this study is to maximize the annual energy generation of the hydropower station given some limiting conditions in the operation process.

$$E^* = \max \sum_{t=1}^{T} R_t(s_t, Q_t) = \max \sum_{t=1}^{T} \sum_{s'}^{S} p(s_{t+1} = s' / s_t = s, a_t = a) E_t(s_t, q_t, Q_t, Qc_t) \tag{22}$$

Where:

$$\begin{cases} E_t(s_t, q_t, Q_t, tc) = k\eta Q_t H_j \Delta T_t - Ec_t(tc_t) \\ R_t(s_t, Q_t) = \sum_{s'}^{S} p(s_{t+1} = s' / s_t = s, a_t = a) * E_t(s_t, q_t, Q_t, Qc_t) \end{cases} \quad \textbf{(23)}$$

The objective function in equations 22 and 23 is subjected to constraints described in equations 3 to 9. Where: $E_t(s_t, q_t, Q_t, Qc_t)$ is generated energy which represents the immediate reward function, $p(s_{t+1} = s' / s_t = s, a_t = a) = p(s, a, s')$ denotes the state transition probability from state $s_t = s$ at time t to state $s_{t+1} = s'$ at time $t+1$ following the action $a_t = a$, η is the overall hydropower efficiency which is assumed to be 0.9. k is a hydropower parameter which is equal to 8.5. Q_t represents the water release (m^3 / s) in each time period t; H_j is the water level (m) which is given by the reservoir regulation curve between up water level Z_{sy} and down water level Z_{xy}.

$$H_j = Z_{\mathrm{sy}} - Z_{\mathrm{xy}} = (Z_t + Z_{t+1}) / 2 - Z_{\mathrm{xy}} \quad \textbf{(24)}$$

In this case study, one hydrological year is assumed to start from October and end at September. The simulation horizon is over 68 years (October 1933 to September 2000) of runoff time series. The performance attained by simulating these policies in this time horizon by applying Reinforcement Learning Methods have been compared with those achieved by the Stochastic Dynamic Programming method over the same optimization period. In the Reinforcement Learning model system, the state variable s_t is taken as the reservoir storage, the decision variable a_t is assumed to be the release in each month t, including the pumping time tc_t in each time t. The state transition function is the reservoir's water balance equation that provides the storage on month $t+1$ as a function of storage in time period t, the inflow in time period t and the release decision, taking into account the rule curves of the reservoir bottom gates and spillways.

$$s_{t+1} = s_t + \left[q_t - Q_t - sp_t\right] * \Delta T_t + Qc_t * tc_t \quad \textbf{(25)}$$

The immediate reward associated with the hydroelectric generation is a scalar quantity and represent the monthly energy (GWh) which can be produced by the hydropower station following some specified decision policies. This is computed as a function of the release from the reservoir and its water level, which determines the hydraulic head (equation 23). The final water level at the end of each month is the final state of the system in specific time and the initial state for the following month. To introduce the uncertainty of the inflow series, the state transition probabilities were computed based on simple Markov Decision Process. The main characteristics of the runoff in the river basin are represented in table 3. The inflow classes were randomly selected based on Pearson III Probability Distribution Function. [7-9]

Coefficients For Person III Distribution

Coefficient	October	November	December	January	February	March	April	May	June	July	August	September
μ	135.70	99.25	49.06	38.84	35.35	68.34	353.79	296.06	328.71	390.95	489.60	326.51
σ	75.45	47.847	14.059	7.8993	7.4482	29.866	135.70	135.20	219.39	295.50	385.58	260.83
Cv	0.56	0.4820	0.2865	0.2033	0.2107	0.4370	0.3835	0.3727	0.6674	0.7558	0.7875	0.7988
Cs	1.44	1.4462	0.4756	0.3356	0.3476	0.7245	0.6367	0.6188	1.1179	1.2093	1.2600	1.2781

K=Cs/Cv	2.6	3	1.66	1.65	1.65	1.66	1.66	1.68	1.56	1.6	1.6	1.6

Table 1. Statistical Parameters for Pearson III Probability Distribution Function: μ Mean value of inflow time series; σ Standard deviation of inflow time series; Cv Coefficient of variation; Cs Standard deviation Coefficient

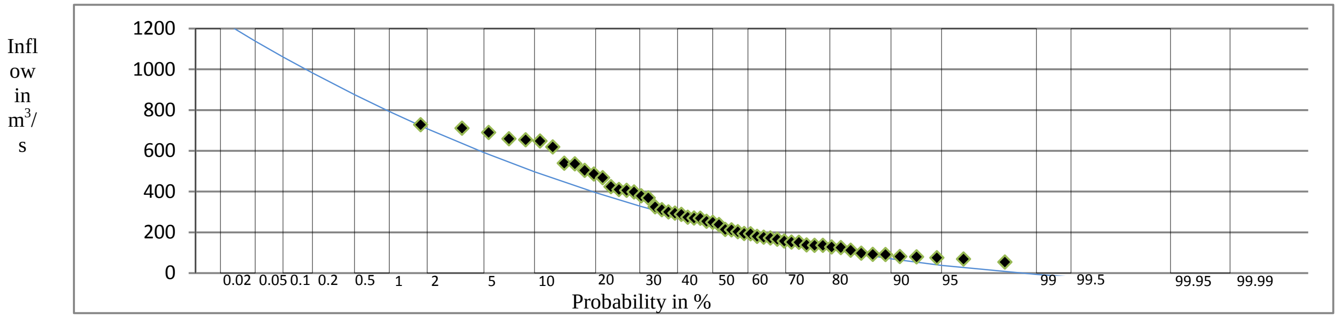

Figure 1. Probability Distribution of inflow series in October

3. 1. Results and Discussion

The storage level as a state of the system, after each interaction with the environment belongs to one of the discrete values. Furthermore, as previously mentioned, the action that the agent takes can lead to a release, which is completely different from the action taken. However, the process of updating action-value functions was performed based on this action. The computation results are shown in Tables 2 and 3 respectively.

Reservoir	**SDP Algorithm**	**SARSA Algorithm**	**Q Learning Algorithm**
Non Pumping (10^9 kWh)	4.43464	4.16495	4.24326
Pumping 10 hours/day (x10^9 kWh)	4.78020	4.63405	4.61662
Incremental (x10^9 kWh)	0.34556	0.4691	0.47335
Computation Time (Seconds)	1979.83	876.59	734.36

Table 2. Optimal operation of hydropower station. Pumping time 10 hours/day. Discount Factor γ=0.95 and greedy=0.01

Reservoir	**SDP Algorithm**	**SARSA Algorithm**	**Q Learning Algorithm**
Non Pumping (10^9 kWh)	4.43464	4.3546	4.41866
Pumping 10 hours/day	4.98020	4.8189	4.92346

(x10^9 kWh)			
Incremental (x10^9 kWh)	0.54556	0.4643	0.5048
Computation Time (Seconds)	1979.83	876.59	734.36

Table 3. Optimal operation of hydropower station. Pumping time 10 hours/day. Discount Factor γ=0.95. and greedy=0.1.

After several iterations and updates of the value function in each time t, the Reinforcement Learning algorithm converges to the optimal policy $\pi^*(s_t)$. The selected optimal action following the optimal decision policy, leads the system to transit from state s_t to the final state s_{t+1} based on the mass balance equation. Figure **2** represents the water level at the end of each month, following the optimal decision policy for this case study. Based in the computation results, we can conclude that pumping water can improve the efficiency of hybrid pumped power station and increase the expected energy generation of the power plant.Furthermore, from the results shown in tables 2 and 3, it is verified that for anyone of the applied algorithms, results in a significant increase of the guaranteed output energy by over **11.4%** , comparing the operation of reservoirs with pumping system instead of operation without pumping water. Based on the computation results performance, the SDP method demands a higher computational time compared to Reinforcement Learning method. The greater gain obtained by applying Reinforcement Learning Algorithm in solving stochastic optimization problems is the capability of obtaining near optimal solution with a shorter computational time compared to other iterative methods such as Stochastic Dynamic Programming algorithm, as proved in this case study.

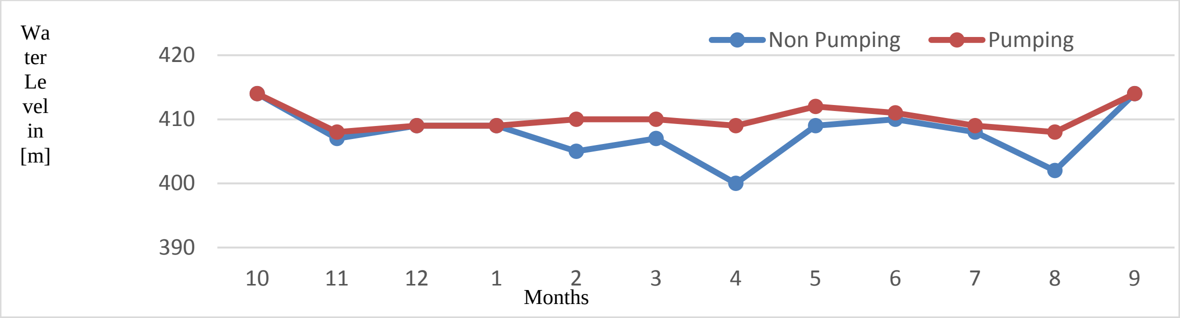

Figure 2. Monthly water level following the optimal operation policy. Optimal pumping time of 10 hours/day.

Tables 2 and 3 respectively, show the performance results of the reinforcement learning algorithm and SDP Algorithm. In table 2, the greedy policy in Reinforcement Learning algorithm were set to 0.01 while in table 3 the greedy policy was increased to 0.1. The results of this experiment show that by applying the SDP algorithm for solving the reservoir optimization problem, an annual production of 4435 *GWh* can be achieved while by applying Reinforcement Learning (Q Learning and SARSA) algorithm, gives an annual production of 4418*GWh* and 4355 *GWh*, respectively. The Q Learning and SARSA algorithms have a deviation error of *0.38%* and *1.8%* respectively compared with the result obtained by SDP algorithm. Table 3 shows that the application of greedy policy improvement, Reinforcement Learning can achieve higher performance in maximization of the value function.

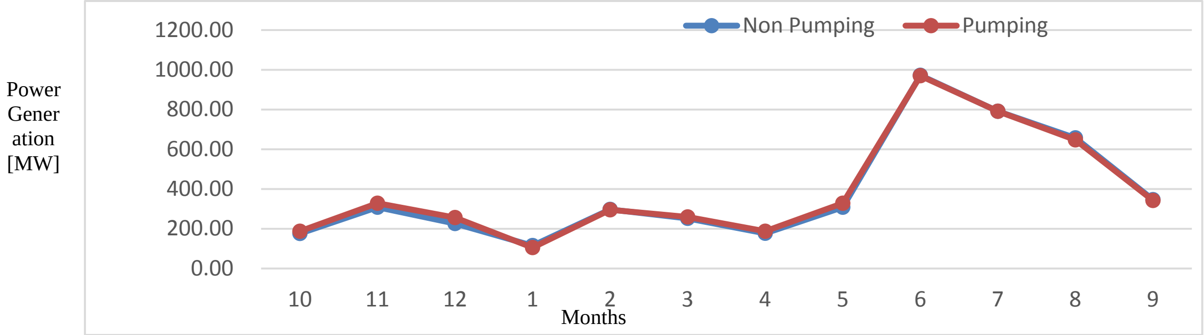

Figure 2 Monthly Power Generation in Reservoir. Optimal pumping time of 10 hours/day.

The quantity of water discharged from upstream reservoir to downstream channel plays an important role for several activities along downstream channel. Based on computation results, we can conclude that the application of hybrid pumped unity increases the water head of the upstream reservoir, consequently the discharge capacity of this reservoir increases, improving the expected energy generation. However, the discharge from the upstream reservoir has a direct impact in water supply for many other applications in the downstream channel. During the operation of cascaded reservoirs with hybrid pumped unity, special attention should be given to the pumping time and the amount of water to be pumped so that there is no shortage of water in the downstream channel, especially in the dry season or weak precipitation period.

4. Conclusion

In this study, the robustness of Reinforcement Learning is shown by its ability to solve nonlinear optimization models such as long-term reservoir operation problems. The case study on hybrid pumped reservoir illustrates that Reinforcement Learning techniques can acquire approximately the optimal solutions with higher robustness. The application of the hybrid pumped unity increases the net energy of hybrid pumped station and the expected generation. It has also been shown that Reinforcement Learning algorithm can achieve higher performance in terms of computation time compared with Stochastic Dynamic Programming. Finally, it can be stated that reinforcement learning method proposed in this study is a progressively promising algorithm for solving complex non-linear problems in water resource systems such as multi-purpose reservoir operation problem or cascaded systems.

References

1. Abdalla, Alaa Eatzaz. (2007), Reinforcement Learning Algorithm for operation planning of hydroelectric power multireservoir system. Unpublished PhD; Thesis. University Of British Columbia.

2. Castelletti A. Galelli S. Restelli M., et al, (2010), Tree-based reinforcement learning for optimal Water reservoir operation. Journal of Water Resource.

3. Defourny B, Ernst D, Wehenkel L. (2013) Scenario Trees and Policy Selection for Multistage Stochastic Programming using Machine Learning. Informs J on Computing.1–27.

4. Durante JL, Nascimento J, Powell et al. (2017), Approximate Dynamic Programming with Hidden Semi-Markov Stochastic Models in Energy Storage Optimization. Princeton University, Princeton NJ: Technical report.

5. Kaufmann E, Cappé O, Garivier A. (2016), On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models. Journal of Machine Learning Research. 17:1–42

6. Lee JH, Labadie JW. (2007), Stochastic optimization of multi-reservoir systems via Reinforcement Learning. Journal of Water Resource.

7. Li Wenwu, Hung Jin, Guo Xihai. (2012), Mid-Long term optimization of reservoir operation for hybrid pumped storage power plant, *Automation of Electric Power Systems*, vol. 32, no. 4, pp. 104–108.

8. Li Wenwu, Hung Jin. (2012), Mid-Long term optimization of cascade reservoir operation for hybrid pumped storage power station, *Journal of Automation of Electric Power Systems*,

9. Li Wenwu, Wu Xixi, Hung Jin, Guo Xihai. (2013), Mid-long term optimization of reservoir operation for hybrid pumped storage power station based on stochastic dynamic programming, *Power System Protection and Control*.

10. Mahootchi M., Tizhoosh H. R., and Ponnambalam K.. (2007), Reservoir Operation Optimization by Reinforcement Learning. Journal of Water Management Modeling. 227–308.

11. Powell WB. (2007), Approximate Dynamic Programming. John Wiley and Sons.

12. Powell WB, Meisel S. (2016), Tutorial on Stochastic Optimization in Energy Part II: An Energy Storage Illustration. IEEE Transactions on Power Systems. .

13. Powell WB, Meisel S. (2016), Tutorial on Stochastic Optimization in Energy II: An energy storage illustration. IEEE Transactions on Power Systems. 31(2):1459–1467.

14. Salas D, Powell WB. (2017), Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems. Informs J on Computing. 30(1):1–41.

15. Sen S, Zhou Z. (2014), Multistage stochastic decomposition: A bridge between stochastic programming and approximate dynamic programming. SIAM J Optimization. 24(1):127–153.

16. Sutton R, Barto A. (2017), Reinforcement Learning: An Introduction.

17. Zugno M, Conejo AJ. A (2015), Robust optimization approach to energy and reserve dispatch in electricity markets. European Journal of Operational Research. 247(2):659–671