Deep Reinforcement Learning for Optimal Hydropower Reservoir Operation

Wei Xu[1]; Fanlin Meng[2]; Weisi Guo[3]; Xia Li[4] and Guangtao Fu[5]

[1]Associate Professor, College of River and Ocean Engineering, Chongqing Jiaotong University, No.66 Xuefu Rd., Ran'an District, Chongqing, 400074, China. Email: xuwei19850711@163.com

[2]Research Fellow, Center for Water Systems, University of Exeter, Exeter EX4 4QF, UK. Email: M.Fanlin@exeter.ac.uk

[3]Professor, School of Aerospace, Transport and Manufacturing, Cranfield University, College Road, Bedford, MK43 0AL, Bedfordshire, UK. Email: Weisi.Guo@cranfield.ac.uk

[4]Associate Professor, College of River and Ocean Engineering, Chongqing Jiaotong University, No.66 Xuefu Rd., Ran'an District, Chongqing, 400074, China. Email: 154211570@qq.com

[5]Professor, Center for Water Systems, University of Exeter, Exeter EX4 4QF, UK; Turing Fellow, The Alan Turing Institute, 96 Euston Road, London, NW1 2DB, UK (Corresponding Author). ORCID: https://orcid.org/0000-0003-1045-9125. Email: g.fu@exeter.ac.uk

Abstract: Optimal operation of hydropower reservoir systems is a classical optimization problem of high dimensionality and stochastic nature. A key challenge lies in improving the interpretability of operation strategies, i.e., the cause-effect relationship between system outputs (or actions) and contributing variables such as states and inputs. Here we report for the first time a new Deep Reinforcement Learning (DRL) framework for optimal operation of reservoir systems based on Deep Q-Networks (DQN), which provides a significant advance in understanding the performance of optimal operations. DQN combines Q-learning and two deep ANN networks and acts as the agent to interact with the reservoir system through learning its states and providing actions. Three knowledge forms of learning considering the states, actions and rewards are constructed to improve the interpretability of operation strategies. The impacts of these knowledge forms and DRL learning parameters on operation performance are analysed. The DRL framework is tested on the Huanren hydropower system in China, using 400-year synthetic flow data for training and 30-year observed flow data for verification. The discretization levels of reservoir water level and energy output yield contrasting effects: finer discretization of water level improves performance in terms of annual hydropower generated and hydropower production reliability; however, finer discretization of hydropower production can reduce search efficiency and thus resulting DRL performance. Compared with benchmark algorithms including dynamic programming, stochastic dynamic programming, and decision tree, the proposed DRL approach can effectively factor in future inflow uncertainties when deciding optimal operations and generate markedly higher hydropower. This study provides new knowledge on the performance of DRL in the context of hydropower system characteristics and data input features, and shows promise of potentially being implemented in practice to derive operation policies that can be automatically updated by learning on new data.

Keywords: Artificial Intelligence; Deep Q-Network; Deep Reinforcement Learning; Hydropower System; Reservoir Operation

Introduction

Optimal real-time operation of hydropower reservoir systems has been widely studied and used as a classical optimization problem for testing new optimization and control algorithms (Yeh 1985;Giuliani et al. 2018). The popular algorithms include : 1) Hedging rules and operation rules-based approaches (Peng et al. 2015; Wan et al 2016; Ming et al.2017), which can be solved using evolutionary algorithms or other optimization methods; 2) various dynamic programming approaches based on the Bellman equation, including deterministic and stochastic approaches (Xu et al. 2014; Zhang et al. 2019); 3) data-driven algorithms such as decision trees (Xi et al. 2010; Zhang et al. 2017) and artificial neural networks (ANN) (e.g., Wang et al. 2010). These approaches are normally developed offline and cannot effectively update operation policies according to the dynamically changing flow conditions (Quinn et al., 2019). Real-time control systems such as model predictive control, which can collect and process data and update the control algorithm in real-time or near real-time, have been applied to industrial control problems including urban wastewater systems (e.g., Meng et al. 2017 & 2020). Only recently, however, they were developed for reservoir systems (e.g., Galelli et al. 2014; Ficchi et al. 2016; Vermuyten et al. 2018 & 2020).

Hydropower operation can be modelled as a Markov Decision Process (MDP) (Lee and Labadie 2007; Xu et al. 2014; Zhang et al. 2019), which is a Markov process with rewards and decisions. It can be argued that in some situations no perfect information on the system state is available, that is, the state is partially observable, so the operation problem is a partially observable MDP. For example, small reservoirs may not be fully monitored with high-resolution temporal and spatial water depth which are required for decision making. However, for simplicity, the reservoir operation problem is assumed as a fully observable MDP in this study. In the MDP, an agent (e.g., operator) interacts with the environment (e.g., the hydropower system) by taking an action (e.g., output of the turbines or reservoir release) depending on the current system states (e.g., water level), hydrological conditions (i.e., inflow) and rewards (e.g., hydropower benefit), which then affects the probability of the process moving into a new state. An MDP describes an environment for reinforcement learning (RL) where the agent can learn in real-time using new data to continuously improve its performance. Thus, RL is identified as one of the promising approaches for decision-making problems of MDP characteristics (Doltsinis et al. 2014). Indeed, it is particularly useful for optimal hydropower operation problems.

RL algorithms have been substantially improved in many aspects in the past decades, including balancing exploration and exploitation (Sutton and Barto 2018), search strategies (Lin 2015), learning behaviour (Sutton and Barto 2018), reward evaluation (Gao et al. 2019). However, there is lack of application to water resources systems or hydropower systems with a few studies using traditional RL such as Opposition-based learning, Q-learning or fitted Q-iteration (Lee and Labadie 2007; Castelletti et al. 2010 and 2013). Traditional RL uses state decision tables to map the relationship between states and actions (Lin 2015; Gao et al. 2019). With an increasing number of state variables, however, the decision table approach as in the traditional RL cannot effectively handle the large number of combinations of states and actions, resulting in the curse of dimensionality problem (Mnih et al. 2013; François-Lavet et al. 2018).

Recently, Deep Reinforcement Learning (DRL) was developed by combining traditional reinforcement learning with deep learning representation of non-linear high-dimensional mapping between system states and expected action rewards (Mnih et al. 2013; Mnih et al. 2015). The DRL was first presented by Mnih et al. (2013) for Atari games using the variants of the traditional Q-learning model (Watkins and Dayan 1992). Subsequently, Mnih et al. (2015) developed a novel deep Q-network (DQN) to enhance the capability of the DRL to play the classic Atari 2600 game, where two ANNs with the same structure were applied to construct relationships between states and actions, hence DRL is capable of handling high-dimensional states and actions. LeCun et al. (2015) regarded DRL as an important model for decision-making in the field of artificial intelligence (AI). DRL is the core algorithm of AlphaGo and used to consider the future effects of each action to maximize the probability of winning (Silver et al. 2016). The learning capacity of DRL in a complex environment has been further enhanced recently (Mnih et al. 2013; Mnih et al. 2015), which promoted its application in various fields, such as electrical grid systems , mechanical control and unmanned aerial vehicles . To the best of our knowledge, DQN based reinforcement learning has not been tested or applied to solve reservoir and hydropower operation problems.

In this study, we report for the first time a novel DRL framework for optimal hydropower operation and provide a significant advance in understanding its performance. The novelty of the DRL framework lies in the development of the DQN as an agent, consisting of two ANNs, to represent the relationships between states, actions and rewards, and definition of a decision value function for reward evaluation. Three forms of knowledge for DRL learning considering different system states are developed and compared. The Huanren Reservoir in North-eastern China is taken as an example to test the operation performance of the DRL framework. We benchmark our DRL results on decision tree (DT), dynamic programming (DP) and stochastic dynamic programming (SDP) models, which are already shown to be able to provide interpretability in their solutions. Interpretability is distinguished from the concept of explainability in this study. A model is defined as interpretable when a cause-effect relationship can be clearly observed within the system modelled. An explainable model focuses on describing the processing of the data or the representation of data inside a model, so it can explain how decisions are made inside the model. Through analysis of the results in terms of DRL performance and sensitivity to both input features and learning parameters, this study provides an in-depth understanding on the performance of DRL and an improved interpretability of reservoir operation, which helps to reveal the cause-effect relationship of reservoir operation. This study moves a step further towards building trustworthy intelligent operation systems for practical application.

Case Study

Huanren Hydropower System

Huanren Reservoir is located in the lower reaches of Hun River, in the north-eastern China. The reservoir basin covers an area within 124°43′~136°50′ E and 40°40′~42°15′ N, and the area is approximately 10,364 km^2. The annual average precipitation is 860 mm and 70% of precipitation is concentrated between May and September. Huanren Reservoir is regulated in an annual cycle and is mainly operated for hydropower generation. Its main characteristics are given in Table 1.

To generate a large training dataset, an Auto-Regressive and Moving Average (ARMA) model is used to simulate the inflows in the study basin, which was suggested by many studies (e.g., McLeod et al. 1983). The observed 10-day average inflows of Huanren Reservoir from 1980 to 2010 are used to construct an ARMA model. Then, a series of 400-year synthetic inflows are generated by the ARMA for DRL training. This time series is able to capture the variability of the river flow that drives reservoir operations. The observed inflows of Huanren Reservoir from 1980 to 2010 are used to verify the performance of the trained DRL model.

States and Actions

In this study, the states and actions are used in discrete forms. The water level range from the dead water level to the normal water level is discretized into ten intervals using a discretization size of 1m. One year is divided into 36 periods for simulation using a 10-day time step. Note the number of days in the third period of each month varies from 8 to 11 days depending on the month. The inflow is discretized into six intervals, and the turbine output as a decision variable is also divided into six levels according to the characteristics of the turbines, which constitute the action set, as shown in Table 2. Note that the inflow and output in each row in Table 2 are not necessarily linked, i.e., no relationship between inflow and output is suggested here.

Optimal Hydropower Operation

This section describes the problem of optimal hydropower operation and two classical solution methods for comparison with DRL, i.e., the SDP and DT.

Problem Formulation

In this study, the hydropower operation is to maximize the total power production as well as minimize the deviation from the required hydropower output to guarantee the stability of power supply. The hydropower benefit consists of two components: power production and penalty for deviation from system requirements as below

$$R(K_t, F_t, N_t) = E(K_t, F_t, N_t) - \{Max[(e - N_t), 0]\}^2 \tag{1}$$

$$E(K_t, F_t, N_t) = N_t \times \Delta t \tag{2}$$

$$F_{p,t} = \frac{N_t}{\eta \times H_t} \tag{3}$$

$$H_t = \frac{1}{2} \times \left[(K_t + K_{t+1}) - (D_t + D_{t+1}) \right] \tag{4}$$

$$V_{t+1} = V_t + \left(F_t - F_{p,t} - F_{s,t} \right) \times \Delta t \tag{5}$$

where R is the hydropower benefit; N_t is the hydropower output of the turbines at time step t and is the decision variable; $E(\cdot)$ is the generated energy; and $\{Max[(e - N_t), 0]\}^2$ is the penalty when N_t is less than the required firm output e, which is a constant value of 33 MW in the case study. F_t is the inflow at time step t; $F_{p,t}$ is the outflow for power generation at time step t, which is determined by N_t. $F_{s,t}$ is the amount of spilled water at time step t; V_{t+1} is the storage capacity, which is generated by the water balance equation Eq. (5); H_t is the average head difference during time step t; K_t is the water level at the beginning of time step t; K_{t+1} is the water level at the beginning of time step t+1 (i.e., the end of time step t); D_t and D_{t+1} are the downstream water levels of reservoir at the beginning and end of time step t, respectively. η is the turbine efficiency, which is 0.9 in this study. Δt is the simulation time interval and is 10 days in this study.

The constraints are as follows:

$$K_{\min} \le K_t \le K_{\max} \tag{6}$$

$$0 \le N_t \le N_M \tag{7}$$

$$0 \le F_t \le F_M \tag{8}$$

where K_{min} and K_{max} are the minimum and maximum water storage levels, respectively. N_M represents the installed capacity of the hydropower plant, and F_M represents the maximum release capacity of the turbines.

Decision Tree Model

The DT model (Bessler al. 2003; Wei and Hsu 2008; Xu et al. 2013) is used to benchmark the performance of the DRL model. DT is a type of implicit stochastic optimization and aims to determine the relationships between system states and actions (i.e., releases), i.e., to develop operation rules, through mining optimized operation policies from different inflow scenarios, which are obtained using a deterministic optimization model. DT models have a rather limited performance improvement compared to neural networks, but offer maximum interpretability to engineers as they build on revealing the cause-effect relationship between system states and actions (Bessler et al. 2003; Wei and Hsu 2008). It is not surprising that trusted DT data mining models are widely used for optimising hydropower operations since the 1990s (Xi et al. 2010; Xu et al. 2013; Hecht et al. 2020; Yang et al. 2020). In this study, the C5.0 decision tree (Quinlan 2020) is employed to develop operation policies using optimization results as samples. The samples consist of condition (i.e., state) and decision (i.e., action) attributes. In this study, the condition attributes are the water level and inflow at the current time step, and the 10-day inflow forecast at the next time step, and the decision attribute is the 10-day output of the turbines at the next time step.

The DT operation policies are generated using the following steps: 1) the operation policies are optimized using deterministic dynamic programming; 2) the operation policies at every time step are generated as operation samples, which are classified into four groups, i.e., dry season (November to April), prior-flood season (May to June), flood season (July to August) and post-flood season (September to October), to maintain the consistency of the sample decision-making methods; 3) the decision trees for each of the four seasons are developed using the C5.0 algorithm. Based on the decision trees, the operation policies of each season are generated from mining the results from the deterministic dynamic programming and used to simulate the hydropower operation.

Stochastic Dynamic Programming Model

SDP is developed from deterministic dynamic programming and has been extensively studied in hydropower operation (Yeh 1985; Xu et al. 2014; Zhang et al. 2019). The optimal operation policies of the hydropower reservoir are derived by the recursive equation, which is based on the Bellman equation. In the SDP model, the water level at the current time step and the 10-day inflow forecast in the future are used as state variables and the output of the turbines is used as a decision variable. The inflow and water level are discretized into intervals which are represented by representative values, and the randomness of inflows can be addressed by transition probabilities (Xu et al. 2014). The interval representative values of the inflow and reservoir storage are written as

$$\begin{cases} \hat{q}_{\mathrm{t}} = [q_t^1, q_t^2, \cdots\cdots, q_t^{\mu}] \\ \hat{K}_{\mathrm{t}} = [K_t^1, K_t^2, \cdots\cdots, K_t^{\varphi}] \end{cases} \tag{9}$$

where $\hat{q}_{\mathrm{t}}$ represents the inflow vector of the representative values at time step t; $\hat{K}_{\mathrm{t}}$ represents the storage intervals at the beginning of time step t. The superscripts of μ and φ are the total number of the inflow and storage intervals, respectively.

In the SDP model, it is assumed that the inflow constitutes a simple Markov process. Thus, the randomness of the inflow at time step t+1 is addressed through a Markov transition probability. The operation policies are derived using the backward Bellman equation by iterating until the ending storage reaches a steady state (Mujumdar and Nirmala 2007). The SDP model recursive equation is defined as

$$f_t\ (K_t, i) = Max\left\{R(K_t, i, K_{t+1}) + \sum_j P_t^{ij} \times f_{t+1}(K_{t+1}, j)\right\} \tag{10}$$

where f_t is the recursive equation at time step t. i and j are the intervals of the inflow at time steps t and t+1, respectively. P_t^{ij} is the Markov transition probability that the inflow of interval i at time step t transfers to interval j at time step t+1.

Deep Reinforcement Learning Framework

The main components of the DRL framework, as shown in Fig. 1, include an agent and the environment. The agent represented by the DQN interacts with its environment in discrete time steps.

At time t, the agent first receives the system states and inputs, i.e., the water storage level and inflow in this study. Then it selects an action with the maximum decision value from a set of available actions, according to the system states and inputs . Subsequently, the action is sent to the environment and implemented in the reservoir system to update the system states and evaluate the reward of the action. The states, rewards and actions are collected and stored to the computer memory, i.e., Random Access Memory (RAM), as the knowledge samples (Mnih et al. 2015). A knowledge sample is a tuple of different variables representing the states, rewards and actions. Three types of knowledge samples are tested in this study to investigate the cause-effect relationship between system states and actions. The samples are accumulated and updated by repeating the above simulation process, as shown by the solid lines in Fig. 1.

The DQN acts as the agent to generate actions given system states and replaces traditional operating rules, and it aims to learning the knowledge of the environment through exploration and exploitation. The learning starts after a specified number of samples are collected. That is, it begins to train the DQN, i.e., action network (AN) and target network (TN) with the collected samples. Through use of two networks, we can achieve stability and the agent can improve the decision-making ability through continuous learning (see details of implementation and reasoning below), thus derives optimal operations for hydropower systems. The DRL framework is explained below in detail.

Markov decision process

The DRL operations are an MDP, and the agent interacts with its environment in discrete time steps. The MDP is a discrete time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. An MDP is a 5-tuple (t, S, R, A, P), where t is time step, S is a set of states, R is the reward set, A is the action set, P is the state transition probability matrix.

In MDP, the decision maker chooses action a from A according to the initial state s at the beginning of time step t. The process responds at time step t+1 by randomly moving into a new state s' and giving the decision maker a corresponding reward. The transition probability is the likelihood that the system state moves from s to s' considering randomness. s' is influenced by the chosen action a

and the previous state s at time step t and is independent of all previous states and actions from earlier time steps. Thus, the state transition probability can be defined as below

$$P(S,S')=P(s_{t+1}=S'\mid s_t=S,a_t=a) \tag{11}$$

In hydropower operation, the decision maker chooses a decision action based on the initial state s. The variables in s and s' are specified in the knowledge forms described below. The output of the turbines is used as a decision action. The generated hydropower energy is the reward. The water level in the next state s' is determined by the water level, inflow and action (i.e., outflow) at time step t. The inflow at time step t+1 is unknown in real time operation. Thus, the state transition probability is normally used to address the randomness of inflow.

Deep Q-Network

In the DQN implemented here, the twin ANNs i.e., AN and TN, have been constructed with the same structure, i.e., one input layer, one output layer and hidden layers. However, their parameter values (i.e., neuron weights) are updated at different times. The AN has the latest weights and is used to evaluate the decision value of the action in real-time operation; the TN is updated only at a certain time step (e.g., every 5 iterations of training) using the AN weights, and is used to evaluate the benefit from the remaining simulation periods. The gradient descent method which is applied to optimize and update the network weights (François-Lavet et al. 2018). The main purpose of DRL training is to update the weights of the AN and TN networks.

The DQN mainly includes the following steps: (a) Building an agent including an AN and TN; (b) Training the AN; (c) Assigning the weights of the AN to the TN; (d) Selecting an action with the maximum Q value (i.e., the decision value of the action). The Q values of actions are generated using the AN with initial states (e.g. water level and forecast inflow) as inputs. During the above process, two techniques play a key role in improving the DQN performance:

(1) Experience Replay. The knowledge samples are stored in the memory, and the batch samples for training are drawn from the memory randomly (Schaul et al. 2015), which breaks the correlation between the samples and makes the neural network update more efficient.

(2) Target Network. If the weights of the AN are updated at each training, this would make the evaluation of the benefit from the remaining periods fluctuate greatly and impossible to converge.

Thus, the TN is used to ensure the stability of the DQN performance and should be updated less frequently than the AN.

State, action and reward

In this study, the reservoir storage level, inflow and operation period are used as the states of the reservoir system, and the output of the turbines is selected as the decision action. The hydropower energy benefit of an action is taken as the reward, which is evaluated using Eq. (1).

Selection of decision action

The DRL network takes the states (S) as inputs and the output is a vector corresponding to the Q values of all actions, i.e., $\left[Q(S,a_1),Q(S,a_2),\cdots,Q(S,a_n)\right]$, where n represents the total number of the actions. In real-time operation, the vector is generated by the AN, and the action with the maximum Q value is selected as the optimal action.

Knowledge form

The hydropower generation knowledge for agent learning is constructed by the states (S) at the beginning and end of time step t, the operation decision action (A_t) and reward (R_t) at time step t. Understanding knowledge forms can help to improve the interpretability of reservoir operation. So the following knowledge forms are built:

(1) Form A: the states (S_t) include the operation period (T_t) and the reservoir storage level (K_t) at the beginning of time step t. This form does not consider the inflow information and is represented as below:

$$\left\langle S_t=(T_t,K_t), Reward=R_t, Action=A_t, S_{t+1}=(T_{t+1},K_{t+1})\right\rangle \tag{12}$$

(2) Form B: the inflows F_t at time step t and F_{t+1} at time step t+1 are included in the states, as shown in Eq. (13). The inflow at time step t+1 needs to be known at time step t. Thus, the DRL model can be trained off-line with historical or synthetic data and used on-line when inflow forecasts at time step t and t+1 are available. In this study, the observed inflows are used as perfect forecasts to evaluate the performances of the models.

$$\langle S_t=(T_t,K_t,F_t),Reward=R_t,Action=A_t,S_{t+1}=(T_{t+1},K_{t+1},F_{t+1})\rangle \tag{13}$$

(3) Form C: Form C is proposed for the on-line operation scenario, which is more realistic in current real world reservoir operations. In this scenario, the inflow at the current time step t (F_t) is forecasted in real-time operation and included in the states (S_t); the inflow (F_{t+1}) at the next time step t+1 is unknown or has high uncertainty, thus is not included in the states as shown in Eq. (14). Note that the time step (i.e., forecast horizon) is 10 days in this study. At the beginning of the current time step t, F_t represents the flow in the next 10 days so it cannot be observed and has to be forecasted in a real-world condition, and thus is assumed as the flow forecast in this scenario. The second 10-day inflow forecast (F_{t+1}) is not used directly in Form C as it is assumed to be highly uncertain. Instead, it is evaluated with Markov transition probabilities and added into S_{t+1} to evaluate the decision value as explained in the section of *Q value* below.

$$\langle S_t=(T_t,K_t,F_t),Reward=R_t,Action=A_t,S_{t+1}=(T_{t+1},K_{t+1})\rangle \tag{14}$$

***Q* value**

In DRL, the immediate reward represents the performance of the action at the current time step, but the *Q* value reflects the performance of multiple time steps. Note that the DRL is based on the MDP, the decision value is constructed by the Bellman equation (Doltsinis et al. 2014), as shown in Eq. (15). In learning, the decision values of the training samples are evaluated and used for updating the weights of the networks. The decision values consist of the reward at time step t and the hydropower benefit at the remaining periods. An action is chosen with an aim to achieve the maximum decision value at each time step. The hydropower benefit at the remaining periods is represented by the maximum *Q* value at time step t+1, which is generated from the TN network using the state S_{t+1}.

In the knowledge forms A and B, the state variables at time step t+1 can be obtained directly from the training sample and fed to the TN network to generate the *Q* value at time step t+1. Thus, the decision value function is defined as (Mnih et al. 2013; Doltsinis et al. 2014)

$$u(S_t,A_t)=R_t+\lambda\times\max_{A_{t+1}}\{Q(S_{t+1},A_{t+1})\} \tag{15}$$

where λ represents the discount rate. λ balances the reward at time step t and the benefit from the remaining periods. The smaller the λ value, the greater the effect of the immediate reward.

Fig. 2(a) shows the computational process of Eq. (15), i.e., knowledge forms A and B. Assuming that Action 2 is selected as the optimal action A_t using state S_t, the reward and the state S_{t+1} of this action are evaluated. Based on S_{t+1}, assuming Action *n* has the maximum *Q* value amongst actions, so it is taken as the benefit from the remaining periods.

Fig. 2 (*b*) shows the computational process of Form C. Inflow F_{t+1} could have multiple values materialized with different transition probabilities, so the expected Q value is calculated to consider predictive uncertainties.

In the knowledge form C, i.e., Eq. (14), the inflow at time step *t*+1 is unknown. To consider the high uncertainty of inflow at time step *t*+1, the Markov transition probability P_t^{ij} in Eq. (10) is used to represent the probability of inflow interval *i* at time step *t* to interval *j* at time step *t*+1. Then, S_{t+1} can be obtained using the probabilistic inflows, and the *Q* values of the states at time step *t*+1 are generated by the TN network. Finally, the expected *Q* value, which represents the benefit in the remaining periods, is evaluated. The decision value function is defined as below

$$u(S_t, A_t) = R_t + \lambda \times \sum_{j=0}^{\mu} P_t^{ij} \times \max_{A_{t+1}}\{Q(S_{t+1}, A_{t+1})\} \tag{16}$$

Where μ is the total number of inflow intervals and *i* should be determined at time step *t* and take a value from 1 to μ.

Q-value update

The *Q* value is evaluated by averaging the decision values in *J* time steps where *J* is the total number of simulation time steps, as shown in Eq. (17). Eq. (17) can be simplified as Eq. (18). During learning, Eq. (18) is applied to update the *Q* values based on the samples in the knowledge base (Mnih et al. 2013). In machine learning, one epoch is an iteration of training when the entire training dataset passes the ANN. When the training dataset is big, it is further divided into batches for training. The loss function, i.e., Eq. (19), calculates the difference in the *Q* values between two training iterations (epoch or batch) *k* and *k*-1, and is used to update the weight parameters using the gradient descent method (François-Lavet et al. 2018).

$$Q_k = \frac{1}{J}\sum_{j=1}^{J} u_j = \frac{1}{J}\left(u_j + \sum_{j=1}^{J-1} u_j\right) = \left(1 - \frac{1}{J}\right) Q_{k-1} + \frac{1}{J} u_j \tag{17}$$

$$Q_k(S_t,A_t)=(1-\alpha)Q_{k-1}(S_t,A_t)+\alpha\cdot u(S_t,A_t) \quad (18)$$

$$Loss(k)=Q_k-Q_{k-1} \quad (19)$$

where u represents the decision value function; Q_k represents the Q value at iteration k; α is the learning rate.

The algorithm

In DQN, the agent's intelligence is determined by the AN and TN networks. The pseudo code of the DQN training is shown in Algorithm 1.

In the algorithm, the parameters include the number of samples in the memory (W), the required minimum number of samples (w), batch size of training samples (D), training interval (L), greedy rate (ε), discount rate (λ) and weight update interval (β). W, w, L and D control the memory capacity and the conditions of learning, which are generally regarded as low sensitive to learning. By contrast, ε, λ and β are more sensitive. ε determines the probability of exploration by choosing an action randomly, which affects the search efficiency. Smaller values of λ make the DRL focus more on immediate benefits, and smaller values of β make more frequent to update TN weights and more difficult to converge. Both λ and β affect the stability of learning.

In the case study, the architecture of AN and TN is determined through trial and error as below: one input layer, one output layer and three hidden layers of 100 nodes each with an activation function of Rectified Linear Unit (ReLU): g(z) = max{0, z}, and it can well represent the relationships between states and actions as shown by preliminary analysis. A deep network with more hidden layers may be required for more complex problems such as cascade reservoir operation problems. The DRL training ends after 2000 epochs, i.e., LT=2000.

Algorithm 1 The Pseudo Code of the DQN-DRL Training

Initialization:

(1) Training epochs (LT=2000); (2) Total number of simulation time steps (J);
(3)Training interval (L); (4) Batch size of training samples (D); (5) Memory ($W=\Phi$) and minimum requirement (w); (6) TN weight update interval (β); (7) Greedy rate (ε);
(8) Discount rate (λ); (9) Weights (η) of the AN; (10) Weights (ψ) of the TN.

For k in Iteration count (LT):

Initialize States: $S_1=(T_1,K_1,F_1)$; Cycle count (i=0)

For t in simulation time steps (J):
  If Random number< Greedy rate (ε):
    Choose an action randomly from the set of actions: Action=A_t
  Else:
    Choose the action with the maximum Q value: *Action*=A_t
  Execute the chosen action to calculate the new system state:
    Use Eq. (1) to evaluate Reward (R_t)
    Use Eq. (5) to evaluate the water storage level (K_{t+1}) at the end of time step t
  Save sample: The knowledge example at time step t is saved in the Memory
  Learning:
    If (W > w) and the remainder of $(i\times J+t)/L$ is 0:
      Randomly get D samples from the Memory: $\langle S_t, R_t, A_t, S_{t+1} \rangle$
      Input S_{t+1} and R_t to evaluate the decision value using the TN, i.e., Eq. (15) or Eq. (16) depending on the chosen knowledge form
      Input S_t to evaluate $Q_{k-1}(S_t, A_t)$ using the AN
      Use Eq. (18) to update $Q_k(S_t, A_t)$
      Update the weights (η) of AN according to the *Loss*(k)
      k ++
    If the remainder of $(i\times J+t)/\beta$ is 0:
      Update the weights of TN: $\psi=\eta$
i++

Results and Discussion

In this study, the 400-year synthetic inflows are used to develop the DT, SDP and DRL models. The planning horizon is 10 days, i.e., one simulation time step ahead. These models are developed to obtain the maximum benefits over the period of 400 years. Their performance is tested using the observed flows from 1980 to 2010, from which the inflow forecasts are taken. In addition, Dynamic Programming (DP) is used as a benchmark model using the observed flows, as in principle it can provide the best solution with future inflows assumed to be known during simulations.

Impact of learning parameters

The DRL learning performance is controlled by the model parameters. These parameters can be divided into two categories: control parameters and learning efficiency parameters, as shown in Table 3.

The control parameters are generally low sensitive parameters. The learning efficiency parameters determine the learning stability, search ability and convergence speed, and are normally high sensitive parameters. Thus, the impacts of the learning efficiency parameters are analyzed using the training dataset. To compare search efficiencies, the rewards during the learning process are shown for different parameters in Fig. 3. A reward value represents the average reward of the generated samples at each time of training, and thus represents the operation performance after each training. With an increasing training epoch, the performance of the model improves and the reward values increase gradually.

Fig. 3(a) shows the reward variations with different values of greedy rate ε. ε determines the probability that the operation decisions moving from exploitation to exploration. For example, when the ε greedy value is 0.95, the probability of the exploration is only 0.05. Such a large greed value can limit the DRL to discover new knowledge samples with high Q values, thus, it provides a low learning efficiency, i.e., a very flat reward curve during the learning process. When a smaller greed value is used, for example ε =0.8, a larger number of exploratory knowledge samples are generated and stored in the memory. This makes samples in the memory more diverse, However, inferior samples can also be included in the exploratory knowledge. In this case, it takes more time to exploit the samples during the learning process and the learning efficiency and accuracy can be low, in particular when a large amount of the inferior samples retains in the memory for a long time. Fig. 3(a) shows that a good balance between exploitation and exploration is achieved when ε =0.9 as the reward values are substantially higher than the reward traces of other rates.

Fig. 3(*b*) shows the reward variations using different values of the discount rate λ. λ determines the impact of the benefit at the remaining periods on the decision value. A larger λ value implies that the benefit in the remaining periods has stronger influence on the decision value. When λ is 0.95, the decision value is predominantly determined by the benefit of the remaining periods and is only slightly influenced by the reward at the current time step. When λ is 0.75, the influence of the reward from the current action on the decision value becomes larger, and the networks of DRL pay more attention to the immediate benefit. In the learning, the discount rate λ balances the reward of the current action and the benefit of the remaining periods. The λ value of 0.85 achieves a good balance, thus has a high learning performance than other λ values.

Fig. 3(c) shows the reward variations using different learning rates (α). When the value of α is 0.001, the Q value is less affected by the decision value according to Eq. (18) and instead mainly affected by the historical Q value. It makes the change in the updated Q value relatively small, which is not effective to the learning. With an increasing value of α, the Q and reward values are more affected by the decision value. With an increasing training epoch, the networks become stable gradually, and the reward variation curves show the performances of the α values. When the α value is 0.03, the learning rate α has a higher performance than the others.

Fig. 3(d) shows the reward variations using different weight update intervals (β) of the TN network. When the β value is 10, it represents that the TN network weights are updated every 10 training epochs. When the β value is lower, the TN network weights are updated more frequently, making the Q value of the remaining periods more variable. Conversely, a larger β value increases the difference of the weights between the AN and TN networks, and thus increases the Q value distortion from the two networks. This can lead to slow and inefficient learning.

The best parameter values obtained are provided in Table 3 and used in other analyses unless otherwise stated. As analysed above, the learning performance of the DRL is substantially affected by the learning efficiency parameters. Sensitivity analysis of learning parameters should be taken as an important diagnostic tool for generating an effective DQN policy.

Impact of discretization

Similar to learning parameters, the impact of discretization on model performance is investigated using the training dataset. In addition to the discretization size of 1 m, seven other scenarios are tested regarding the water level discretization, ranging from 0.25m to 2m. Fig. 4 shows the annual hydropower generated (AHG) and hydropower production reliability of the DRL with different discretization sizes. Reliability is defined as the probability that the output is no lower than the required firm output in this study (Hashimoto et al. 1982). Results show that AHG and reliability are increasing with increasing discretization precision of water level. This is mainly because the model accuracy is higher with increasing discretization precision of water level, however this is at expense of increasing search space and thus computing time. By contrast, increasing discretization precision of hydropower output reduces slightly AHG and Reliability, which results from reduced learning efficiencies. The reward variations of the eight scenarios during the training are shown in Fig. 5. The 3D surface shows that the rewards are also increasing with increasing discretization precision of water level.

Similar to the learning parameters, the best performing discretization levels are used for further analysis and algorithm comparisons. Water level discretization should be considered in diagnostic analysis.

Knowledge form

Fig. 6 shows the results of three knowledge forms for the historical period 1980 - 2020. The reservoir water levels, shown in Fig. 6(a-c), can directly reflect the differences of the hydropower operations derived from different forms. The differences in water level between each of the three approaches and DP are shown in Fig. 6(d).

Comparison of the results in Fig. 6 shows that the water levels of Forms A and B are controlled at the dead water level for most of the operation periods. Only in a few periods when the inflow is particularly large, the water level can rise to the normal water level. The main reason is that the outputs determined by the two forms are too large, which makes the water level quickly decrease to the dead water level. This result can be further explained using the knowledge samples for decisions at the current time step t=3 in Table 4 as below.

In knowledge Form A, the inflow at the current time step (t=3) is not included in the states, though it is provided for each knowledge sample in Table 4 for the illustration purpose only. The samples in Table 4 have the same states at the 3rd time step, i.e., reservoir storage level K_3 = 292 m, however, they have different rewards for different inflow values (F_3). The states at the next period are the same (i.e., K_4 = 293 m), thus the Q values at the remaining periods (i.e., Q_{t+1}) are the same value (i.e., 6.0 MWH). However, the maximum inflow at the current period can generate greater hydropower energy using the action with higher output, and lead to a greater reward at the current step, which makes the decision value larger. That is, the sample with an inflow of F_3=400 m^3/s and action a6 with the maximum output of 11.5 MWH is learned by the DRL as the optimal action for time step t=3.

With the DQN, the decision is made with the information of one step ahead. That is, at the current time step t, the decision is determined in anticipation of the system state at t+1, i.e., S_{t+1}. In Form B, the state S_{t+1} is specifically related to the second 10-day flow forecast F_{t+1}. In Table 4, at the 3rd time step, the system state is (3, 292, 200), which means the current water level is 292 m and the flow forecast for this time step is 200 m^3/s. The decision a6 at the 3rd time step is chosen with the maximum accumulative benefit from the 3rd time step (5.5 MWH) and the remaining time steps (6.3 MWH), given the system state at the 4th time step being (4, 291, 600). Note the benefit (i.e., Q value) of 6.3 MWH is estimated by the DQN for the water level of 291 m and the flow of 600 m^3/s at the 4th time step. If the actual water level and flow are different at the 4th time step, however, the decision a6 may not be the best decision at the 3rd time step. There is also an uncertainty in the benefit estimation by the DQN.

In Form C, the state S_{t+1} includes the water level only. However, the benefit from the remaining time steps (i.e., Q_{t+1} in Table 4) is evaluated as the expected Q value considering all possible flows with the transition probabilities. For the same system state (3, 292, 200) at the 3rd time step as in Form B, the decision a2 is chosen because the Q value for the water level of 292 m at the 4th time step is estimated as 6.3 MWH. Compared with the Form B decision, the Form C decision reserves more water in the reservoir at the 3rd time step. This decision is more robust as it considers the flow uncertainty in the future time steps.

The results in Fig. 6 show that Form C achieves the closest water levels to those from the dynamic programming approach. This shows the flow transitions learned from the training data set can represent well the randomness of future inflows. Thus, Form C is regarded as the best knowledge form for deep learning in this case study and thus used in the following analyses.

Relationships between state, inflow and outflow

Operating rules or curves are commonly used for reservoir operation in practice due to their simplicity and ease to use. They generally define desired storage volumes (or water levels) or desired releases based on the time of year and the existing storage volume. Under the rules, releases or outflows are implicitly expressed as functions of system states and inflows. These functions typically remain deterministic without considering the dynamic nature of reservoir operation, and thus offer high interpretability regarding revealing the cause-effect relationship of reservoir operation. However, the three methods used in this study, i.e., DT, SDP and DRL, provide probabilistic relationships between system states and inflows. These relationships are represented by the three models. In the case of DRL, the relationships are represented by the ANNs. They can be revealed using the mapping from water level and inflow to outflow shown in Fig. 7. The box plots in Fig. 7 are obtained from the historical data. The DRL approach is implemented with Form C and parameters as shown in Table 3.

As revealed in Fig. 7, the outflows vary greatly for a certain water level ranging from 290 m to 300 m, however, the median outflows are very close for different water levels. The interquartile ranges of DRL (i.e., the distance between the first and third quantiles) are roughly the same for all water levels except the lowest and highest water levels (290m and 300m), and are wider than those of decision tree and SDP. At the highest water level 300m, the outflows from all three methods vary in a wide range, but the outflows from DRL are more varied than those from DT and SDP. This implies that DRL is more flexible and provides more varied outflows in order to maximize the total hydropower benefit in response to dynamic inflow conditions. By contrast, DT and SDP generate outflows of less variations and are unable to adjust outflows considering stochastic inflows. Note all three methods have a number of outliers at all water levels. This highlights that high outflows are needed even at low water levels, perhaps due to high inflows in the following time steps.

Fig. 7 also show the relationships between inflow and outflow. The median outflows increase with increasing inflows and their interquartile ranges are also increasing except for the highest inflow. When the inflow occurs in the 6th interval, the outflow is very likely to be high in order to maintain the water level. The results from the three methods are consistent and reflect our intuitive knowledge in reservoir operation.

To further explain the relationships between inflows, water levels and outflows, water level curves over an entire year are shown for two years: wet year 2010 and dry year 2002 in Fig. 8. Amongst the three methods, DT has the lowest water levels in the first six months (periods 1-16), which are dry periods, while DRL has the highest water levels and thus generate the highest hydropower benefits. In the wet year 2010, DRL increases outflows in periods 13-15 in anticipation of high inflows in July and August. This leads to the lowest water levels in periods 16-19 to prepare for high inflows and reduces the volume of spilled water over the year. In the dry year 2002, DRL releases less water to keep high water levels in periods 13-15 in anticipation of low flows in July and August. Note that the water level curves provide clear interpretability on why DRL outperforms other two methods.

Note that model interpretability focuses on describing the cause-effect relationship between inputs and outputs and making it simple and meaningful to users. By contrast, explainability is the extent to which the internal mechanics of a model can be explained in human terms. Increasing interpretability can effectively improve the model predictive ability given changes in inputs, thus improve the model trustworthiness for users. Interpretability is regarded as a key step towards explainability. In other words, explainable models must be interpretable, however, the reverse is not always true. The explainability of the DQN needs to be tackled in future research.

Performance evaluation of hydropower energy

The performances of the models, i.e., DRL, SDP, DT and DP, are shown in Table 5. As explained above, DRL is implemented with Form C and parameters as shown in Table 3, DRL outperforms the SDP and DT methods in the two metrics AHG and reliability. Note that the DP results are obtained with the assumption of known future inflows and thus represent the best performance that could be achieved with optimisation. The comparison in Table 5 demonstrates that DRL is effective in the development of optimal hydropower operations. The operations by DT has the worst performance on the efficiency and stability. This demonstrates a well-established trade-off: (1) DRL offers superior output and reliability performance, but very limited interpretability; whereas (2) DT models offer significantly worse output and reliability performance but provide more interpretable mapping from states to actions. Our attempts to evaluate the performance of DRL with respect to knowledge forms mitigate this trade-off and lead to improved understanding on the cause-effect relationship between energy output and system states, i.e., interpretability. In particular, this is illustrated through the knowledge samples developed from the 400-year synthetic inflows, which explain how a decision (i.e., action) is made by balancing the immediate reward from the current operation and the cumulative benefit from the future operations under a specific system state.

Fig. 9 (a) shows the inflow variations during the 36 operation periods from 1980 to 2010 and Fig. 9 (b) shows the 10-day hydropower output boxplots from the three models. Fig. 9 (b) shows that the AHG mainly comes from periods 6-33. To compare the performances of the models, the operation periods are divided into 3 stages: the first stage from 6 to 16, the second stage from 17 to 26 and the third stage from 27 to 33.

In the first stage, the snow in the basin begins to melt, and the inflow has the first peak, as shown in Fig. 9(a). Comparing the energies in Fig. 9(b), the boxes and black solid lines of decision tree are higher and longer than the others. This implies that the outputs of decision tree are larger, which make the water levels lower and the energy benefit at the following periods reduced.

In the second stage, the inflow during wet season has the second peak. The boxes of decision tree are longer than those of SDP, especially in periods from 20 to 22. In the operation process, the decision tree, SDP and DRL spill a volume of 10.9×10^5, 8.8×10^5 and 7.2×10^5 m^3 during this stage, respectively. The results indicate that the operation strategies of decision tree are highly variable with the worst performance. The operation strategies of SDP increase output to reduce spill water.

In the third stage, the three models have large performance differences. Due to the poor control ability of the decision tree in the second stage, it makes the reservoir spill more water and has a lower water level at the end of the second stage. Thus, the lower water level reduces the efficiencies of the turbines, and the hydropower generation decreased significantly in this stage. The DRL reduces the outputs obviously in periods from 23 to 27; it makes reservoir store more water and keep higher water levels. Thus, in the following periods from 29 to 33, the DRL can generate more hydropower energies, resulting in a substantially higher annual output.

Overall, the results in Fig. 9 reveal that the best performance achieved by DRL in comparison to other approaches lies in the good balance between the immediate rewards from the current operations and the cumulative benefits from the future operations. This is achieved through the appropriate knowledge form developed and the learning parameter values learn from the 400-year stochastic simulated inflows. Note that previous research has demonstrated the performance of Q-learning for hydropower operations in terms of accuracy and computational effectiveness in comparison to traditional stochastic dynamic programming (Lee and Labadie, 2007; Castelletti et al., 2010 and 2013). However, this study demonstrated for the first time the advantages of deep Q-networks in hydropower operations.

Conclusions

This study presented a novel deep reinforcement learning approach for reservoir operation using deep Q-networks. With the case study of Huanren reservoir, the new approach was trained using

400-year simulated inflows and was verified and evaluated according to the observed inflows from 1980 to 2010. The key research findings are as below.

(1) This study provides an insight into the learning efficiency of DRL considering the impacts of discretization sizes of water level and energy output. The results show that the hydropower energy and reliability improve with increasing discretization precision of water level. However, increasing discretization precision of energy output reduces the learning efficiency. This implies that increasing discretization precision of the system states can improve the DRL performance but increasing discretization precision of the actions can reduce the search efficiency and thus the DRL performance.

(2) The four learning parameters of DRL, i.e., the learning rate, discount rate, greedy rate and TN updating intervals affect the trade-offs between the immediate rewards from the current operation and the cumulative benefits from the future operations. Thus, the values of these parameters need to be carefully analyzed to improve the DRL performance.

(3) Three knowledge forms are developed and assessed for constructing effective deep reinforcement learning. When the future inflow is not considered in Form A or its forecast is considered as accurate without uncertainty in Form B, the operations chosen tend to generate large discharges and high hydropower output at the current time step. When the future inflow is considered as probabilistic using the Markov transition approach in Form C, however, the performance of DRL is significantly improved with the benefits from the remaining time steps well represented.

(4) Compared to classical decision tree and stochastic dynamic programming, the DRL approach can factor in future inflow uncertainties when deciding optimal operations, thus achieve the best performance in term of annual hydropower generation and reliability. The twin networks can represent well the relationships between inflows, states and outflows through training with a 400-year stochastic inflow time series in the case study

In summary, we contributed a deep reinforcement learning approach for hydropower operation, which outperforms the two classic hydropower operation approaches – decision tree and stochastic dynamic programming. This approach has the potential to be implemented in practice to derive optimal operation strategies that can be interpreted and automatically updated by learning on new data.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request. Data include the synthetic and observed flow time series. The code that has been used for the deep reinforcement learning is also available.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (Grant No. 51609025), the UK Royal Society through an industry fellowship to Guangtao Fu (Ref: IF160108) and an international collaboration project (Ref: IEC\NSFC\170249), the Open Fund Approval (SKHL1713, 2017), Chongqing technology innovation and application demonstration project (cstc2018jscx-msybX0274, cstc2016shmszx30002). Both Guangtao Fu and Weisi Guo are also supported by The Alan Turing Institute under the EPSRC (Grant EP/N510129/1). A special thank goes to Hun River cascade hydropower development company, Ltd and Dalian University of Technology for the case study data.

References

Bessler, F.T., Savic, D.A. and Walters G.A. (2003). Water reservoir control with data mining, *J. Water Resour. Plann. Manage.*, **129**, 26– 34.

Castelletti, A., Galelli, S., Restelli, M., & Soncini- Sessa, R. (2010). Tree- based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(9), W09507.

Castelletti, A., Pianosi, F., & Restelli, M. (2013). A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 49(6), 3476-3486.

Doltsinis, S., Ferreira, P., & Lohse, N. (2014). An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(9), 1125-1138.

Ficchi, A., L. Raso, D. Dorchies, F. Pianosi, P. O. Malaterre, P. J. van Overloop, and M. Jay-Allemand. (2016). “Optimal operation of the multireservoir system in the seine river basin using deterministic and ensemble forecasts.” *J. Water Resour. Plann. Manage.* 142 (1): 05015005. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4), 219-354.

Galelli, S., A. Goedbloed, D. Schwanenberg, and P. J. van Overloop. (2014). “Optimal real-time operation of multipurpose urban reservoirs: Case study in Singapore.” J. Water Resour. Plann. Manage. 140 (4): 511–523. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000342.

Gao, Y., Chen, J., Robertazzi, T., and Brown, K. A. (2019). Reinforcement learning based schemes to manage client activities in large distributed control systems. *Physical Review Accelerators and Beams*, 22(1), 014601.

Giuliani, M., Quinn, J., Herman, J., Castelletti, A., and P. Reed (2018). Scalable multi-objective control for large scale water resources systems under uncertainty. IEEE Transactions on Control Systems Technology, 26(4), 588 1492-1499.

Hashimoto, T., Stedinger, J. R., & Loucks, D. P. (1982). Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation. *Water resources research*, 18(1), 14-20.

Hecht, J.S., Vogel, R.M., McManamay, R. A., Kroll, C.N. (2020). Decision Trees for Incorporating Hypothesis Tests of Hydrologic Alteration into Hydropower–Ecosystem Tradeoffs. *Journal of Water Resources Planning and Management*, 146(5): 04020017.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Lee, J. H., & Labadie, J. W. (2007). Stochastic optimization of multireservoir systems via reinforcement learning. *Water resources research*, 43(11).

Lin, S. Y. (2015). Reinforcement learning-based prediction approach for distributed Dynamic Data-Driven Application Systems. *Information Technology and Management*, 16(4), 313-326.

McLeod, A. I., & Li, W. K. (1983). Diagnostic checking ARMA time series models using squared- residual autocorrelations. *Journal of time series analysis*, 4(4), 269-273.

Meng, F., Fu, G., Butler, D. (2017). Cost-effective River Water Quality Management using Integrated Real-Time Control Technology. *Environmental Science &* Technology, 51, 17, 9876–9886. DOI:10.1021/acs.est.7b01727.

Meng, F., Fu, G., Butler, D. (2020). Regulatory Implications of Integrated Real-Time Control Technology under Environmental Uncertainty. *Environmental Science & Technology*, 54(3), 1314-1325. DOI:10.1021/acs.est.9b05106.

Ming, B., Liu, P., Chang, J., Wang, Y., & Huang, Q. (2017). Deriving operating rules of pumped water storage using multiobjective optimization: Case study of the Han to Wei interbasin water transfer project, China. *Journal of Water Resources Planning and Management*, 143(10), 05017012.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv*: 1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H.; Kumaran, D., Wierstra, D., Legg, S.; Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.

Mujumdar, P. P., and B. Nirmala (2007), A Bayesian Stochastic Optimization Model for a Multi-Reservoir Hydropower System, Water Resources Management, 21: 1465–1485.

Peng, Y., Chu, J., Peng, A., & Zhou, H. (2015). Optimization operation model coupled with improving water-transfer rules and hedging rules for inter-basin water transfer-supply systems. *Water resources management*, 29(10), 3787-3806.

Quinlan, J. R. (2020). Data Mining Tools See5 and C5.0. Available at https://www.rulequest.com/see5-info.html (accessed 26th Jan 2021).

Quinn, J. D., Reed, P. M., Giuliani, M., & Castelletti, A. (2019). What is controlling our control rules? Opening the black box of multireservoir operating policies using time-varying sensitivity analysis. *Water Resources Research*, 55, 5962-5984.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv*: 1511.05952.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D. (2016) Mastering the game of Go with deep neural networks and tree search. *Nature*, 529 (7587), 484-9.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. Second Edition, *MIT press, Cambridge*.

Vermuyten, E., P. Meert, V. Wolfs, and P. Willems. (2018). "Combining model predictive control with a reduced genetic algorithm for real-time flood control." J. Water Resour. Plann. Manage. 144 (2): 04017083. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000859.

Vermuyten, E.; E. Van Uytven; P. Meert; V. Wolfs; P. Willems. (2020). "Real-Time River Flood Control under Historical and Future Climatic Conditions: Flanders Case Study." J. Water Resour. Plann. Manage. 146(1): 05019022.

Wan, W., Zhao, J., Lund, J. R., Zhao, T., Lei, X., & Wang, H. (2016). Optimal hedging rule for reservoir refill. *Journal of Water Resources Planning and Management*, 142(11), 04016051.

Wang, Y. M., Chang, J. X., & Huang, Q. (2010). Simulation with RBF neural network model for reservoir operation rules. *Water resources management*, 24(11), 2597-2610.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4), 279-292.

Wei, C, & Hsu, N. (2008). Derived operating rules for a reservoir operation system: comparison of decision trees, neural decision trees and fuzzy decision trees. *Water Resour Res*, 44 (2), W02428.

Xi, S., Wang, B., Liang, G., Li, X., & Lou, L. (2010). Inter-basin water transfer-supply model and risk analysis with consideration of rainfall forecast information. *Science China Technological Sciences*, 53(12), 3316-3323.

Xu, W., Peng, Y., & Wang, B. (2013). Evaluation of optimization operation models for cascaded hydropower reservoirs to utilize medium range forecasting inflow. *Science China Technological Sciences*, 56(10), 2540-2552.

Xu, W., Zhang, C., Peng, Y., Fu, G., & Zhou, H. (2014). A two stage Bayesian stochastic optimization model for cascaded hydropower systems considering varying uncertainty of flow forecasts. *Water Resources Research*, 50(12), 9267-9286.

Yang, T., Liu, X., Wang, L., Bai, P. Li, J. (2020). Simulating Hydropower Discharge using Multiple Decision Tree Methods and a Dynamical Model Merging Technique. *J. Water Resour. Plann. Manage.*,146(2): 04019072.

Yeh, W. W.-G. (1985) Reservoir Management and Operations Models: A State-of-the-Art Review. *Water Resources Research*, 21 (12): 1797-1818.

Zhang, K., Wu, X., Niu, R., Yang, K., & Zhao, L. (2017). The assessment of landslide susceptibility mapping using random forest and decision tree methods in the Three Gorges Reservoir area, China. *Environmental Earth Sciences*, 76(11), 405.

Zhang, X., Peng, Y., Xu, W., & Wang, B. (2019). An optimal operation model for hydropower stations considering inflow forecasts with different lead-times. *Water resources management*, 33(1), 173-188.

List of figure captions

Fig. 1. The deep learning framework for hydropower operation.

Fig. 2. Evaluation processes of decision value functions

Fig. 3. The effects of the learning parameters on DRL learning: (a) effect of greedy rate ε, (b) effect of discount rate λ, (c) effect of learning rate α, (d) effect of weight update interval β.

Fig. 4. The AHG and reliability for hydropower operation with different discretization sizes

Fig. 5. The reward variations of the DRL with different discretization sizes

Fig. 6. Performances of the DRL models in the historical period 1980 - 2020. (a) water levels of the DRL model with Form A; (b) water levels of the DRL model with Form B; (c) water levels of the DRL model with Form C; (d) differences in water level between the DP and each of three DRL models.

Fig. 7. Relationships between system states, inflows and outputs

Fig. 8. Water level variations under decision tree, stochastic dynamic programming and deep reinforcement learning. (a) wet year 2010 and (b) dry year 2002.

Fig. 9. The boxplots of hydropower energy and inflow during the 36 operation periods. The boxes show 25 and 75 percentiles and the lines in the boxes are the medians (50 percentile). The whiskers show the distances to the maximum and minimum values.

Table 1. The Basic Characteristics of Huanren Reservoir

Characteristic	Value	Characteristic	Value
Total Storage (10^9 m^3)	3.46	Installed Capacity (MW)	222
Usable Storage (10^9 m^3)	2.19	Firm Output of Turbines (MW)	33
Dead Storage (10^9 m^3)	1.38	Outflow Capacity of Turbines (m^3/s)	450
Normal Water Level (m)	300	Dead Water Level (m)	290

Table 2. The inflow intervals and output levels of Huanren reservoir

Interval No.	Inflow (m^3/s)	Output (MW)
1	[0,50)	15
2	[50,150)	33
3	[150,300)	50
4	[300,500)	70
5	[500,800)	150
6	≥800	222

Table 3. The parameters of the DRL model for the Huanren hydropower case study

Control parameters	Value	Learning efficiency parameters	Value
Maximum memory capacity (W)	*3000*	Learning rate (α)	*0.03*
Minimum sample requires (w)	*200*	Discount rate (λ)	*0.85*
Training interval (L)	*50*	Greedy rate (ε)	*0.9*
Batch of training samples (D)	*200*	Weight update interval (β)	*30*

Table 4. Examples of the sample structure and Q value estimation

Knowledge Form	Samples in Memory at t=3 <S_t, *reward*, *action*, S_{t+1}>	Q value at t=4	Decision value ($u_t=R_t+Q_{t+1}$; λ=1)[1]
Form A	< (3, 292), 4.5, a1, (4, 293) > [2]	6.0	4.5+6.0=10.5
	< (3, 292), 5.0, a2, (4, 293) > [3]	6.0	5.0+6.0=11.0
	…		…
	< (3, 292), 5.5, a6, (4, 293) > [4]	6.0	5.5+6.0=11.5[5]
Form B	< (3, 292, 200), 4.5, a1, (4, 293, 200) >	6.0	4.5+6.0=10.5
	< (3, 292, 200), 5.0, a2, (4, 292, 300) >	5.9	5.0+5.9=10.9
	…		…
	< (3, 292, 200), 5.5, a6, (4, 291, 600) >	6.3	5.5+6.3=11.8
Form C	< (3, 292, 200), 4.5, a1, (4, 293) >	5.9	4.5+5.9=10.4
	< (3, 292, 200), 5.0, a2, (4, 292) >	6.3	5.0+6.3=11.3
	…		…
	< (3, 292, 200), 5.5, a6, (4, 291) >	5.6	5.5+5.6=11.1

Note: [1]simplified from Eqs. 15 and 16, R_t is the reward at t=3 and Q_{t+1} is the Q value at t=4; [2]when F_3=200 m^3/s; [3]when F_3=300 m^3/s; [4]when F_3=400 m^3/s; [5]the chosen decision with the maximum decision value.

Table 5. The performances of the three operation models

Operation model	AHG (MWH)	Reliability (%)
DP	449.06	93.17
Decision Tree	426.47	76.82
SDP	428.47	86.46
DRL	441.13	92.54

Figure 1

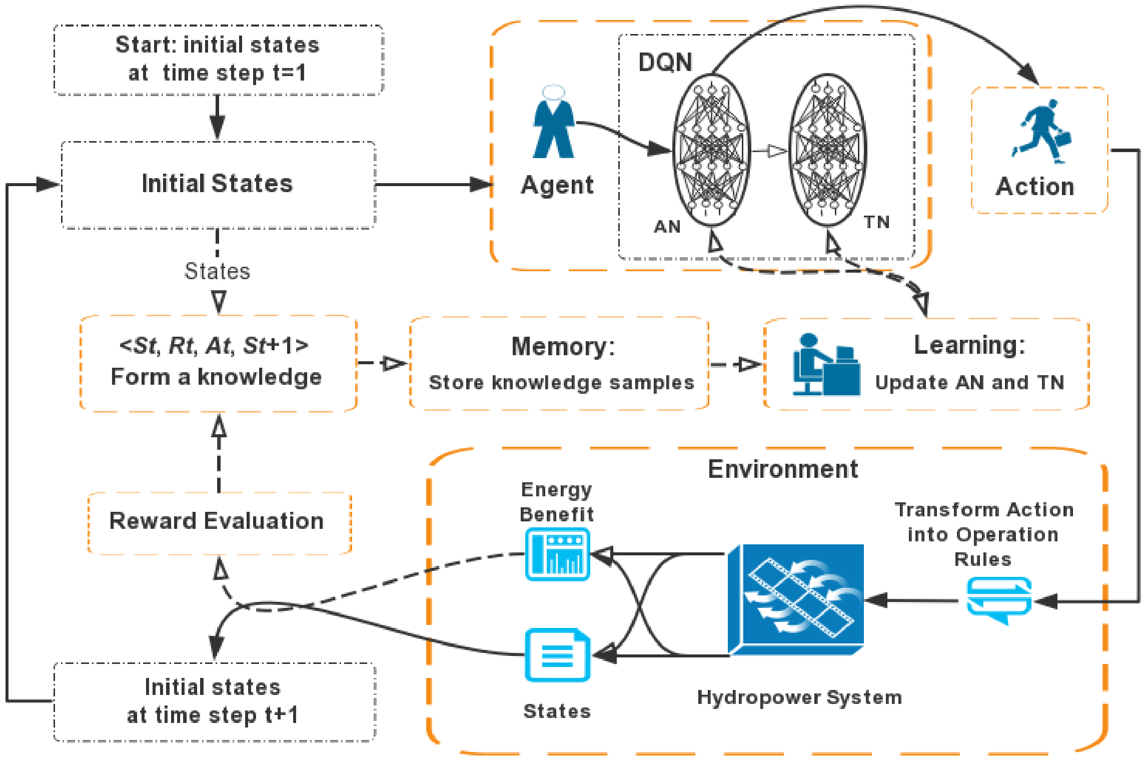

Start: initial states
at time step t=1
Initial States
States
<St, Rt, At, St+1>
Form a knowledge
Reward Evaluation
Initial states
at time step t+1
DQN
Agent
AN
TN
Action
Memory:
Store knowledge samples
Learning:
Update AN and TN
Environment
Energy
Benefit
Transform Action
into Operation
Rules
States
Hydropower System

Figure 2

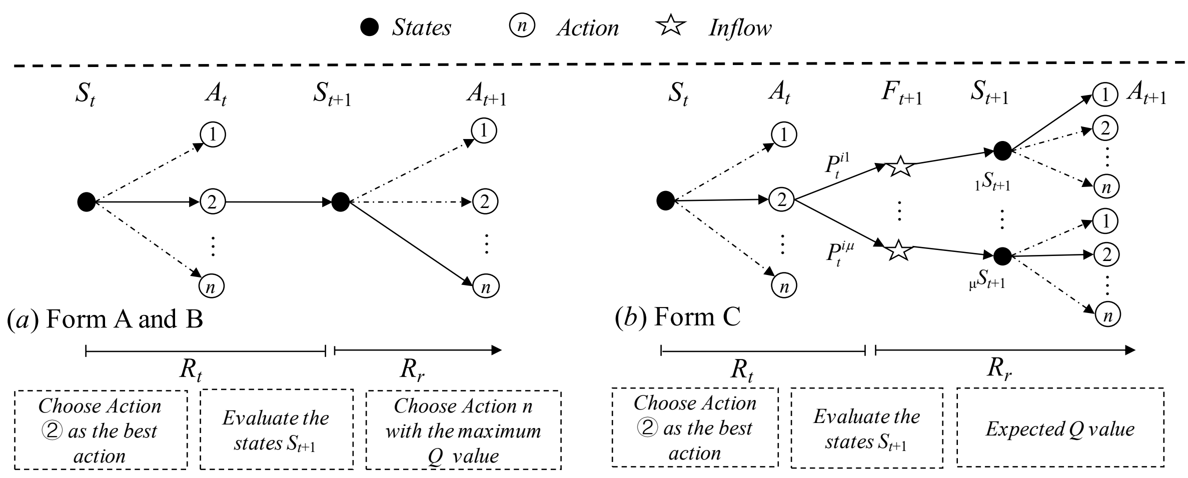

States
Action
Inflow
S_t
A_t
S_{t+1}
A_{t+1}
(a) Form A and B
R_t
R_r
Choose Action ② as the best action
Evaluate the states S_{t+1}
Choose Action n with the maximum Q value
F_{t+1}
P_t^{i1}
$P_t^{i\mu}$
${}_1S_{t+1}$
${}_\mu S_{t+1}$
(b) Form C
Expected Q value

Figure 3

Click here to access/download;Figure;Fig 3.pdf

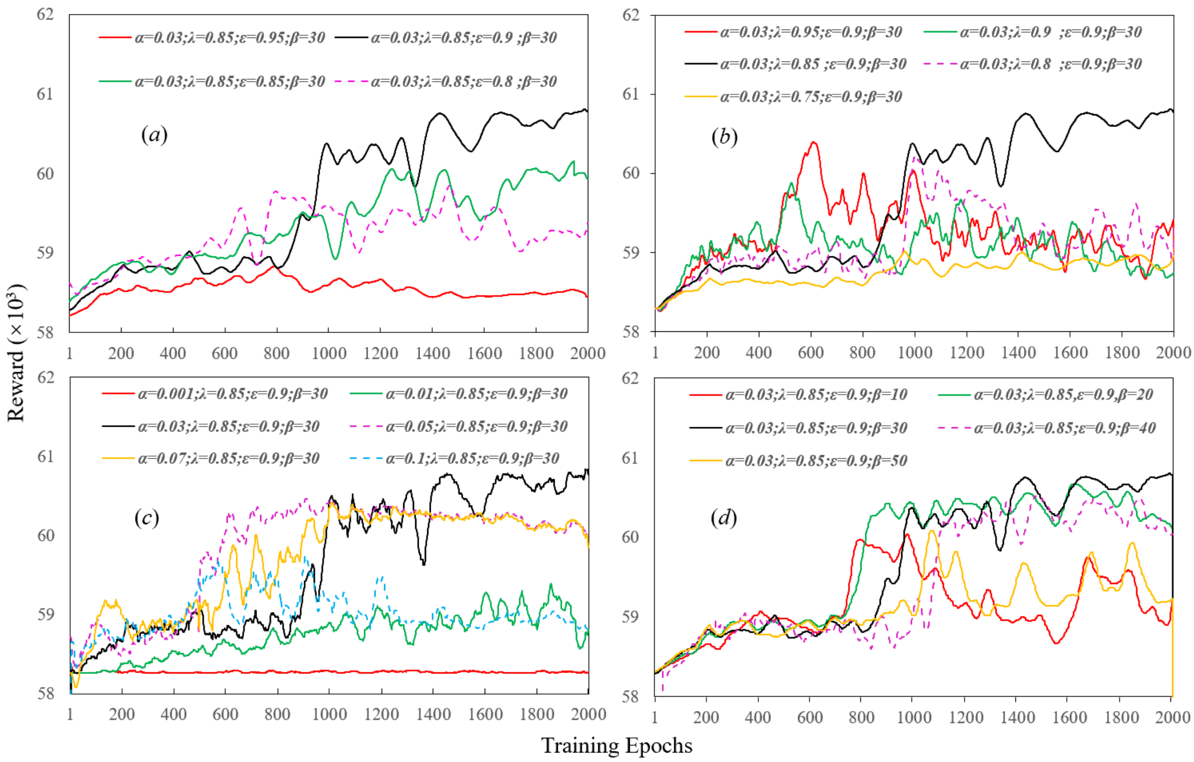
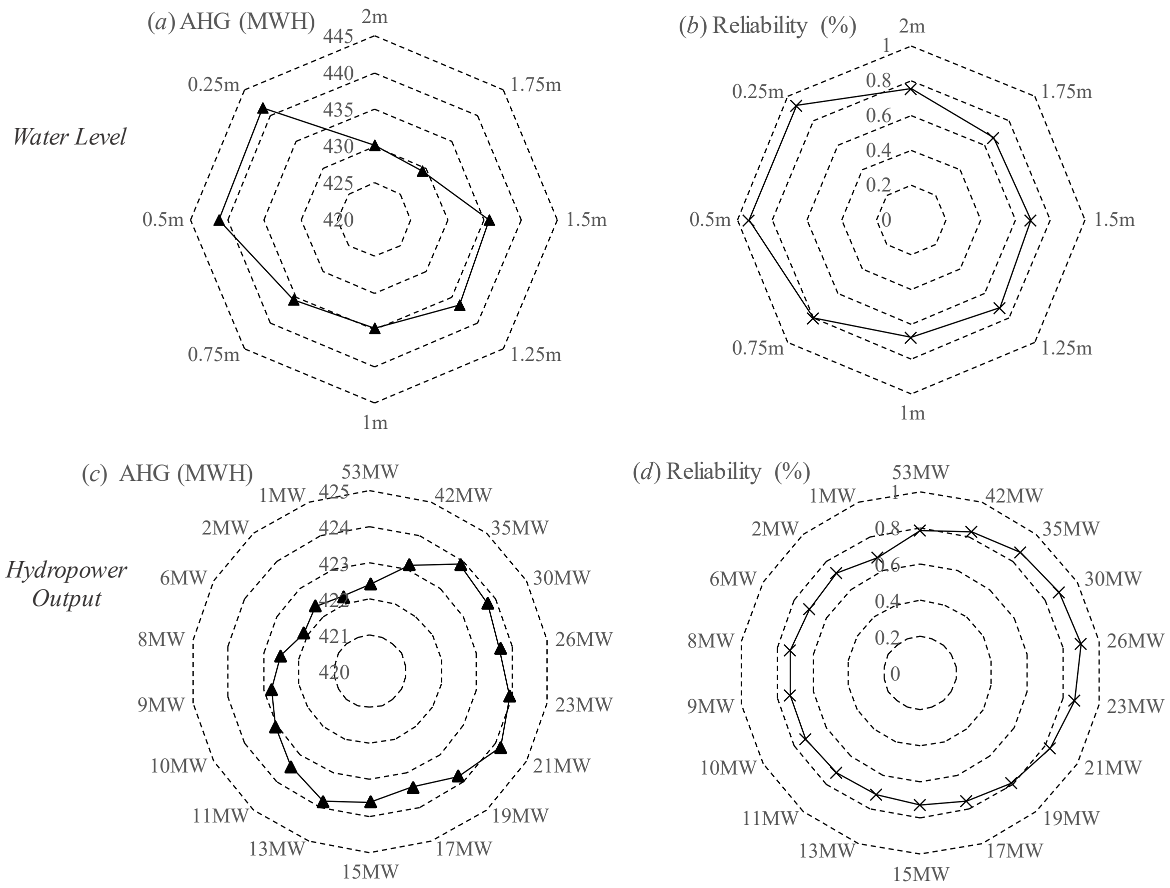
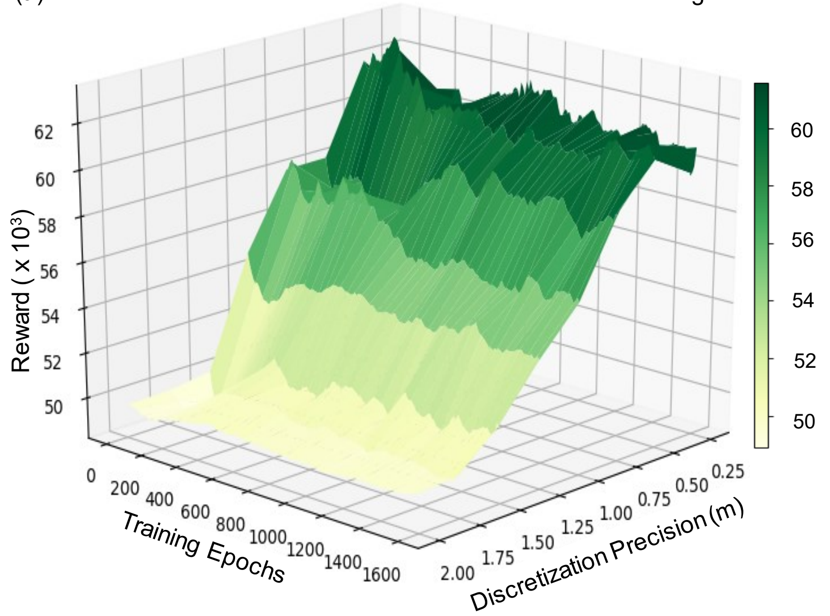


Figure 4

Click here to access/download;Figure;Fig 4.pdf



(*a*) The reward variations with water level discretization increasing.



(*b*) The reward variations with hydropower output discretization increasing.

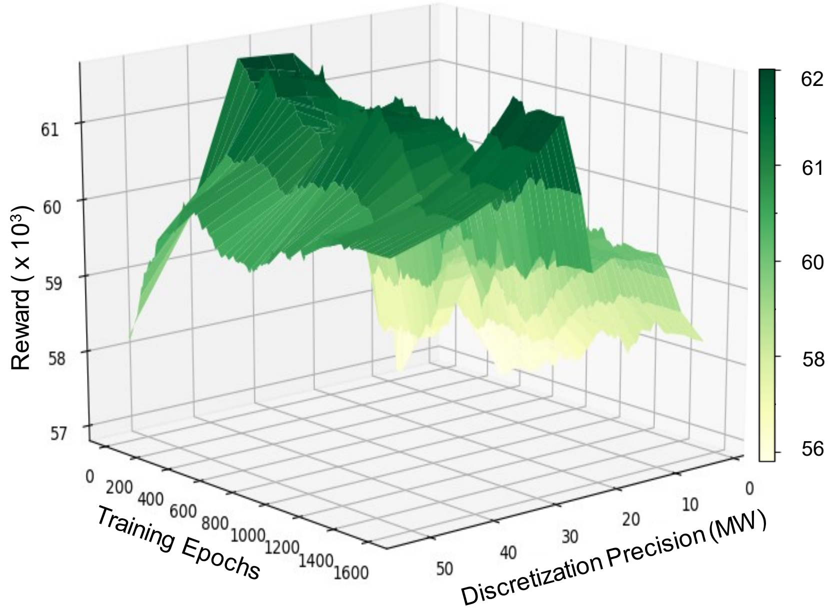


Figure 6

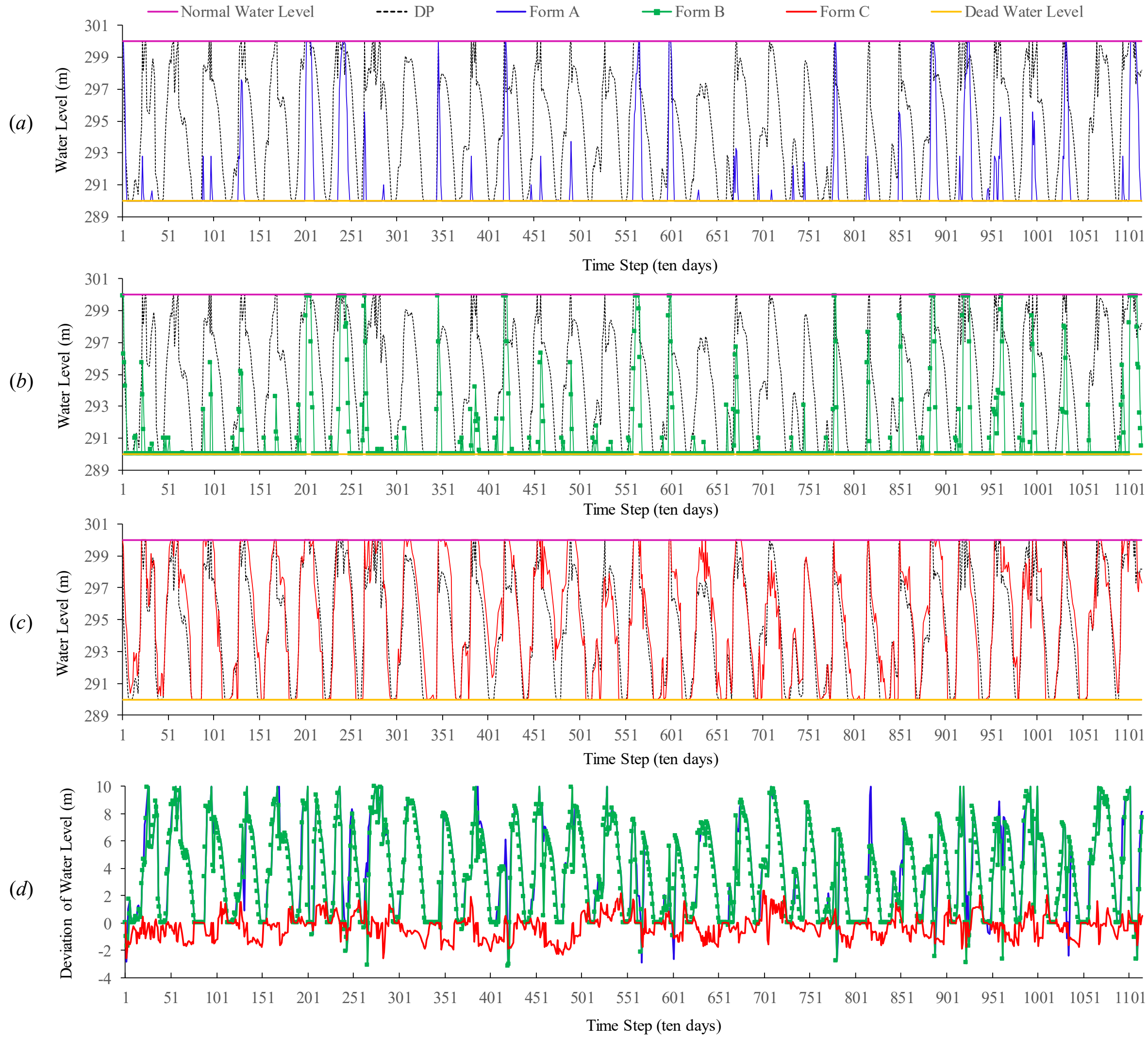

Normal Water Level
DP
Form A
Form B
Form C
Dead Water Level
(a)
(b)
(c)
(d)
Water Level (m)
Deviation of Water Level (m)
Time Step (ten days)
301
299
297
295
293
291
289
10
8
6
4
2
0
-2
-4
1
51
101
151
201
251
301
351
401
451
501
551
601
651
701
751
801
851
901
951
1001
1051
1101

Figure 7

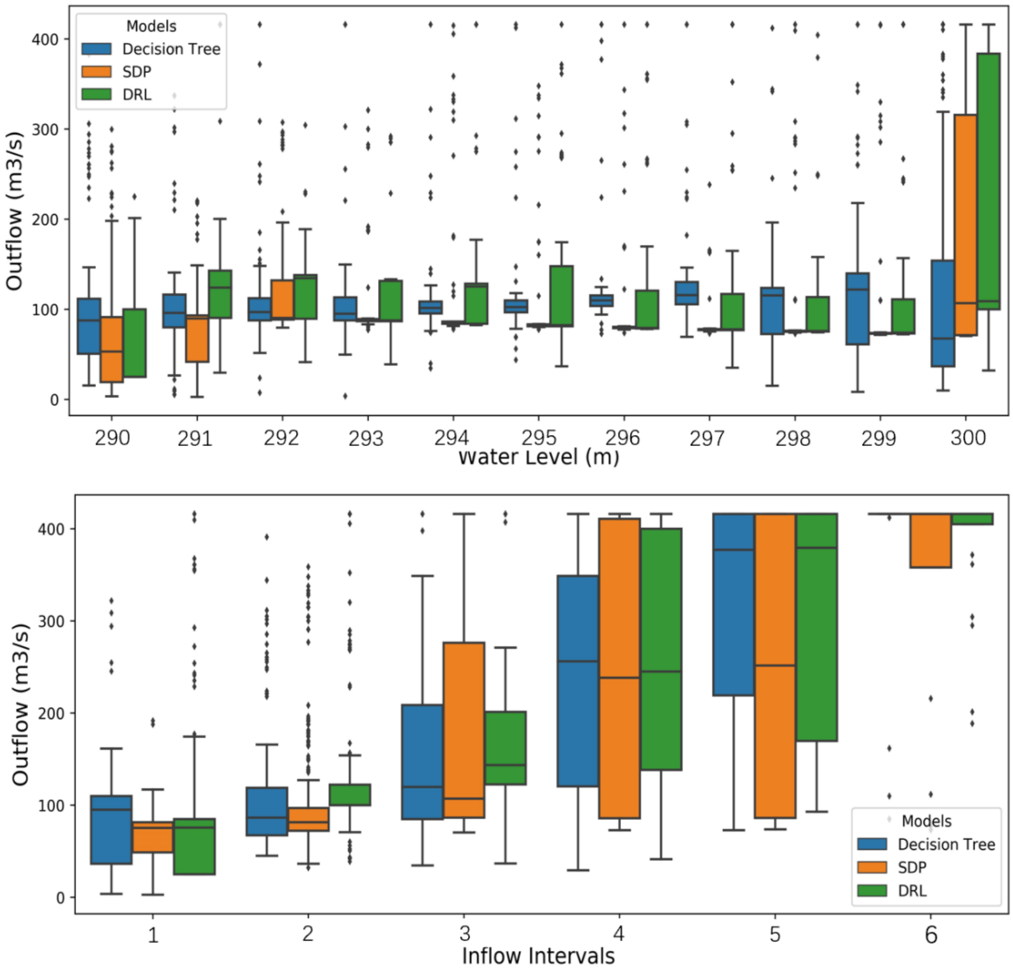
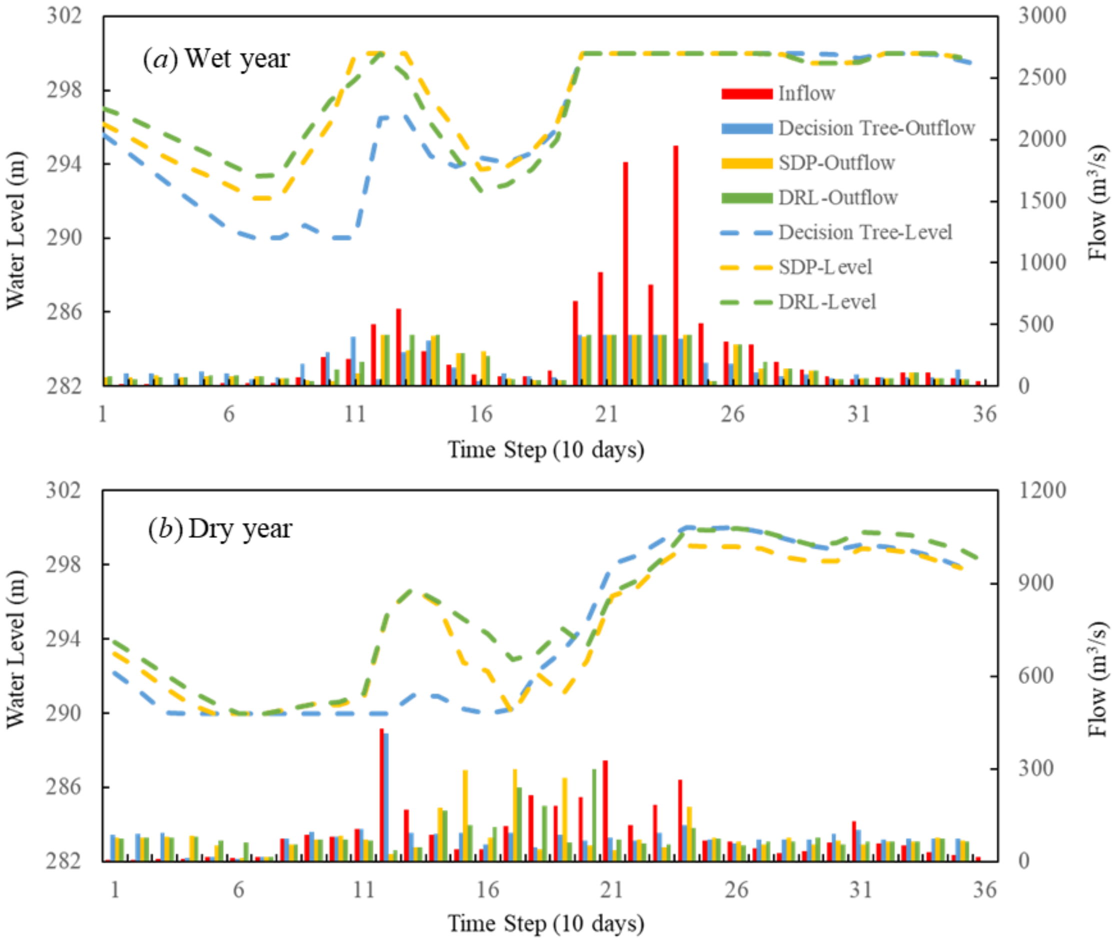
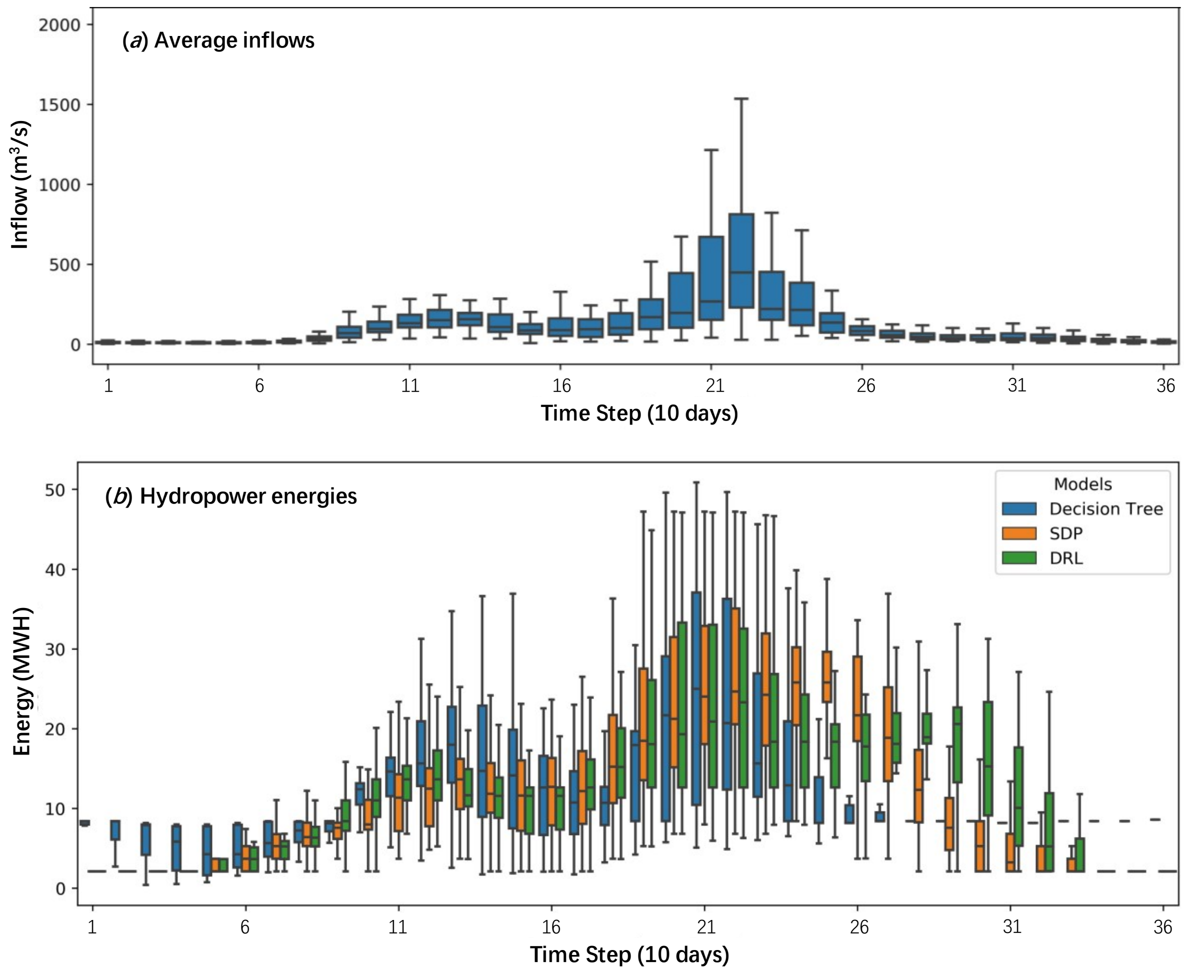

Models
Decision Tree
SDP
DRL
Outflow (m3/s)
0
100
200
300
400
290
291
292
293
294
295
296
297
298
299
300
Water Level (m)
Outflow (m3/s)
1
2
3
4
5
6
Inflow Intervals

Figure 8


(a) Wet year
Water Level (m)
Flow (m³/s)
Time Step (10 days)
Inflow
Decision Tree-Outflow
SDP-Outflow
DRL-Outflow
Decision Tree-Level
SDP-Level
DRL-Level
(b) Dry year
Water Level (m)
Flow (m³/s)
Time Step (10 days)



(a) Average inflows
Inflow (m³/s)
2000
1500
1000
500
0
1
6
11
16
21
26
31
36
Time Step (10 days)
(b) Hydropower energies
Energy (MWH)
50
40
30
20
10
0
Models
Decision Tree
SDP
DRL
1
6
11
16
21
26
31
36
Time Step (10 days)