

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/346701882>

Efficient Reservoir Management through Deep Reinforcement Learning

Preprint · December 2020

DOI: 10.48550/arXiv.2012.03822

CITATION

1

READS

233

10 authors, including:



Tarun Nair

Amrita Vishwa Vidyapeetham

17 PUBLICATIONS 191 CITATIONS

SEE PROFILE



Srinivas Vaidyanathan

91 PUBLICATIONS 1,942 CITATIONS

SEE PROFILE



Nachiket Kelkar

Wildlife Conservation Trust

74 PUBLICATIONS 1,335 CITATIONS

SEE PROFILE



Rajat Nayak

Foundation for Ecological Research Advocacy and Learning

11 PUBLICATIONS 230 CITATIONS

SEE PROFILE

Efficient Reservoir Management through Deep Reinforcement Learning

Xinrun Wang¹, Tarun Nair², Haoyang Li¹, Yuh Sheng Reuben Wong¹, Nachiket Kelkar²

Srinivas Vaidyanathan³, Rajat Nayak³, Bo An¹, Jagdish Krishnaswamy², Milind Tambe⁴

¹Nanyang Technological University, {xwang033, liha0016, wong1109, boan}@ntu.edu.sg

²ATREE, Bangalore, India, {tarun.nair, nachiket.kelkar, jagdish}@atree.org

³FERAL, Bangalore, India, {srinivasv, rajat}@feralindia.org

⁴Google Research, India, milindtambe@google.com

Abstract

Dams impact downstream river dynamics through flow regulation and disruption of upstream-downstream linkages. However, current dam operation is far from satisfactory due to the inability to respond the complicated and uncertain dynamics of the upstream-downstream system and various usages of the reservoir. Even further, the unsatisfactory dam operation can cause floods in downstream areas. Therefore, we leverage reinforcement learning (RL) methods to compute efficient dam operation guidelines in this work. Specifically, we build offline simulators with real data and different mathematical models for the upstream inflow, i.e., generalized least square (GLS) and dynamic linear model (DLM), then use the simulator to train the state-of-the-art RL algorithms, including DDPG, TD3 and SAC. Experiments show that the simulator with DLM can efficiently model the inflow dynamics in the upstream and the dam operation policies trained by RL algorithms significantly outperform the human-generated policy.

1 Introduction

Dams are intended to serve multiple functions, including irrigation, hydropower, and water supply. While some dams can provide flood control functions, in the Indian context, they are largely designed to maximize storage, and operated to avoid floods following extreme storm events. While dam/reservoir operations are largely governed by project objectives and user agreements, they are also guided by contingencies such as floods. Operation policies may be based on the Standard Linear Operating Policy, Storage Zoning, Rule Curves or System Engineering Techniques such as Simulation and Optimization [3]. Dam management, however, suffers from design and operational inefficiencies, e.g., sediment trapping, non-adaptive rule curves. In addition, flow regulation and the disruption of upstream-downstream linkages due to dams also impact river dynamics and threaten aquatic ecosystems and organisms.

While dams designed for flood control can be effective at reducing peak discharge associated with storm events and increasing discharge during dry periods, these dams also cause floods in downstream areas due to poor flood forecasting and faulty reservoir storage practices. For example, extreme rainfall and high reservoir storage combined to cause the 2018 Kerala floods (India) which resulted in the death of more than 400 people, affected 56,844.44 ha of cropped area, with economic damage exceeding USD 3 billion [8, 11]. The Bansagar Dam (Madhya Pradesh, India) is another example of reservoir mismanagement, and was implicated in severe downstream flooding in 2016 and 2018

due to the practice of reaching full reservoir level (FRL) in the early part of the monsoon and the subsequent lack of a flood cushion in this period [16]. Ecological impacts (fish species loss and breeding disruptions in endangered species) downstream of Bansagar are also documented [4, 10].

The major challenges in dam operation (i.e., reservoir management) include (a) the increasing trend of extreme rainfall events makes the prediction of rainfalls inaccurate, (b) the changes in catchment precipitation and land use cannot be immediately detected by the operator, and (c) the complicated downstream dynamics and the various water usages (e.g., ecology, agriculture, fisheries, navigation) make the operation difficult to be optimized. The efficient reservoir management requires the inflow forecasting of the upstream and the modeling of the complex dynamics in the downstream [17].

To address this problem, we leverage reinforcement learning (RL) approaches to explore efficient dam operation policy. Specifically, we built offline simulators with the reservoir inflow model using rainfall estimates from rain gauge and satellite observations (CHIRPS) for the reservoir catchment. As hydro-climatic systems are highly dynamic, we used the Dynamic Linear Model (DLM) [5, 6, 12] in which the parameters evolve with incoming data using Kalman filter and Bayesian update of parameter distributions and the forecast error at any time-step can be estimated and incorporated into the evolution of the model. We use this model to forecast the changes of the storage of the reservoir due to inflow from the catchment upstream due to rainfall. Experiments show that the simulator with DLM can efficiently model the inflow dynamics and the dam operation policies trained by RL algorithms significantly outperform the human-generated policy.

2 Motivating Scenario

Here we show a simple example for the dam management problem in Figure 1, where there is a catchment area at the upstream of the reservoir and a dam which can determine the volume of water being released to the downstream. The rainfall in the reservoir catchment area will flow into the reservoir, and depends on catchment rainfall, temperature, and land use / land cover. The reservoir has two main functions: i) irrigation in the dry season, and ii) hydropower generation. The floods in the downstream can cause the death of humans and livestock, and damage crop and property. Flood damage depends on river conditions and discharge from the dam. The objective of this work is to leverage RL methods to compute the efficient dam operation policy with trained by simulators built on real data. The algorithms in this work can play a central role for the future AI-based dam operation system, which can help human operator to mitigate the downstream floods, as well as maximize various functions of the reservoir, e.g., hydropower and irrigation potentials.

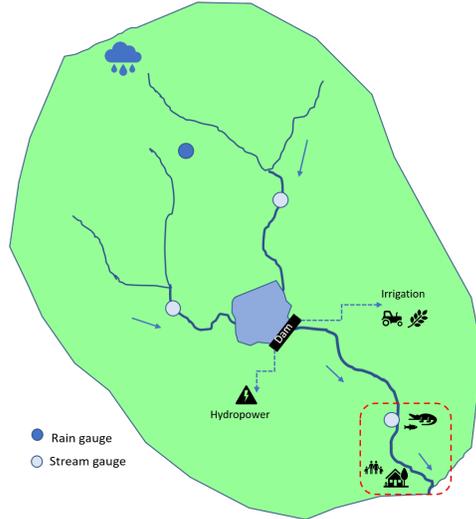


Figure 1: An example of dam management.

3 Problem Statement

The dam management problem is formulated as an MDP, denoted as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma, s_0 \rangle$, where \mathcal{S} is the set of states of the dam system, \mathcal{A} is the set of actions that the dam operator can take, i.e., the amount of water being released, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, γ is the discount factor and s_0 is the initial state. The agent's policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, parameterized by θ , specifies the probability of an action being taken at each state. The return of the agent at time step t is $r_t = \gamma^t R(s_t, a_t)$ where s_t , a_t and s_{t+1} are the state and the action at time step t , respectively.

Specifically, the state $s \in \mathcal{S}$ consists the following information: i) the water level of the reservoir h and ii) the rainfall of K previous steps $\mathbf{rf} = \langle rf_k, k = 1, \dots, K \rangle$ where rf_k is the rainfall of k -th previous day. The transition function P depends on the rainfall data at the current step rf' and the dam action $a \in \mathcal{A}$: given that the current state is $s = \langle h, \mathbf{rf} \rangle$, the successive state s' is defined as

$$h' = g(g^{-1}(h) + f(rf', \mathbf{rf}) - a), \quad \mathbf{rf}' = \langle rf', rf_k \rangle, k = 1, \dots, K - 1 \quad (1)$$

where g is the function which maps the storage (i.e., the volume of water stored) to the water level of the reservoir (accordingly, g^{-1} is the function mapping the water level to the storage), f is the function which predict the amount of water flowing into the reservoir given the rainfall. The reward function R includes three parts: i) Irrigation potentials including R_{rice}^I which is the potential rice productivity and R_{wheat}^I which is the potential wheat productivity, ii) Hydropower potential R^H and iii) Flood damage R^F . The objective is to maximize the accumulated return $J(\theta) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} r_t | s_0] = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0]$.

4 Compute Dam Operation Policy

We leverage the reinforcement learning methods to compute the dam operation policy. RL is a powerful method to compute the optimal policy of an MDP problem. Combining with powerful deep neural network approximators, deep RL [9] has achieved remarkable successes on challenging discrete or continuous decision making problems [14]. For the problem with continuous states and/or actions, the parameterized policy π_θ can be updated by taking the derivative of the return $\nabla_\theta J(\theta)$. In the actor-critic method, the policy, known as actor, can be updated through the deterministic policy gradient algorithm [13]:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_a Q(s, a) \nabla_\theta \pi_\theta(s)] \quad (2)$$

where $Q(s, a)$ is the estimated return by the critic. On the other hand, the critic can be updated through temporal difference learning, which is an update rule based on Bellman equation [15]. The Bellman equation describes a fundamental relationship between the value of a state-action pair (s, a) and the successive (s', a') :

$$Q(s, a) = r + \gamma \mathbb{E}_{a' \sim \pi(s')} [Q(s', a')] \quad (3)$$

For large state space, the value can be estimated by a differentiable function approximator, parameterized by ϕ . In Deep Deterministic Policy Gradient (DDPG) [7], both actor and critic are parameterized by networks and the critic is updated following the deep Q-learning [9], where the network is updated by using temporal difference learning with a secondary frozen target network $Q_{\phi'}(s, a)$ to maintain a fixed objective y over multiple updates

$$y = r + \gamma Q_{\phi'}(s', a'), \quad a' \sim \pi_{\theta'}(s') \quad (4)$$

where a' is sampled from a target actor network $\pi_{\theta'}$. To reduce the overestimation of Q values in DDPG, Twin Delayed Deep Deterministic policy gradient algorithm (TD3) [1] uses two Q-functions instead of one (hence “twin”), and uses the smaller of the two Q-values to form the targets in the Bellman error loss functions and updates the policy (and target networks) less frequently than the Q-function. On the other hand, Soft Actor-Critic (SAC) [2] maximizes a trade-off between expected return and entropy where the entropy is a measure of randomness in the policy, resulting in the improvement of the stability during training.

5 Experimental Results

Bansagar Dam Dataset. An overview of Bansagar Dam’s landscape is displayed in Figure 2. The dataset contains data on Bansagar Reservoir water level and Rainfall received in the upper catchment area from 1 Jan 2012 to 12 Dec 2019.

Simulators. We build four simulators with different upstream models, i.e., $f(\cdot)$: i) $f(\cdot)$ is a Generalized Least Squares (GLS) with the rainfall data, ii) $f(\cdot)$ is a Dynamic Linear Model (DLM) with the rainfall data, iii) $f(\cdot)$ is a Dynamic Linear Model (DLM) with the rainfall data and the output of the GLS model and iv) $f(\cdot)$ is computed by the real data. For the other functions, we have: i) Storage-water level model $g^{-1} := 0.3653h - 119.78$ where h is the water level, ii) Flood damage model $R^F := e^{-981} \cdot \hat{h}^{170}$ where \hat{h} is the maximum reservoir level observed during the 14 days period iii) Irrigation potentials with a) potential rice productivity $R_{rice}^I = 0.1315h - 43.121$ and b) potential wheat productivity $R_{wheat}^I = 0.2642h - 86.641$ where the outputs are million tonnes and iv) Hydropower potential $R^H = 5.1927h - 1342.5$ where the output is million watts. Other parameters of the simulator is displayed in Table 1.

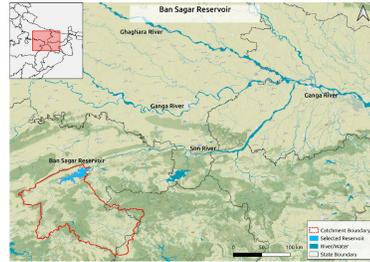


Figure 2: Bansagar dam

Table 1: Simulator’s parameters

dam_cap	342.934	dam_break_damage	80
dam_base_water	0.1	water_year_start	01 June
water_year_end	31 May	dry_season_months	Nov-Jun
discount_factor	0.999	max_step	365
flooded_area_slope	0.00006	power_potential_slope	0.003
wheat_slope	30	rice_slope	30

Evaluation. For the RL algorithms, we choose the widely used DDPG and two state-of-the-art methods TD3 and SAC. We first build all the four simulators with the data from 2012 to 2018, which is used for training the RL policies, and test the trained policies on the simulator where $f(\cdot)$ is computed with the data of 2019. We compare the performances of the computed polices with a human-generated discharge policy, which is displayed in Appendix A.

Table 2: NSE of Different inflow model

	GLS	DLM	GLS+DLM
NSE	0.4045	0.9843	0.9843

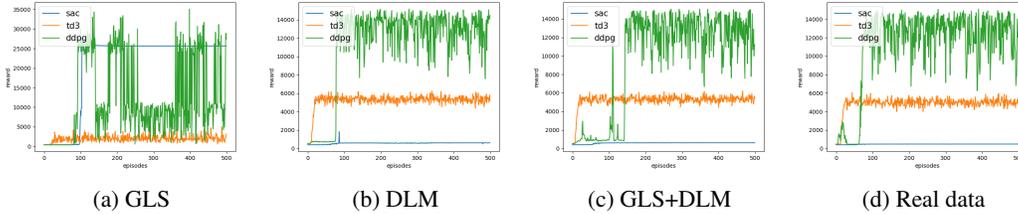


Figure 3: Training curves of DDPG, TD3 and SAC in different simulators

Results. We first evaluate the different prediction model with Nash–Sutcliffe model efficiency coefficient (NSE). The results are displayed in Table 2¹, where we can observe that the GLS cannot provide a good prediction of the inflow from the upstream and the DLM significantly outperforms the GLS. We do not observe the improvement of including both GLS and DLM into the simulator. We present the training results in Figure 3. We observe that the training of SAC is not successful and SAC does not perform well in most of the simulators. The training of TD3 and DDPG is efficient and DDPG can significantly outperform TD3 during the training, although TD3 is more stable. We show the testing results of the trained policies, as well as the baseline strategy, in Figure 4. We find that SAC is not perform well because the training is not successful. The policies trained by TD3 and DDPG is much better than the baseline strategy. Although DDPG performs better during the training, TD3 obtains nearly identical results during test and outperform DDPG in the simulator with real data. Interestingly, the DDPG policy trained by the simulator with the real data does not perform well when testing use the real data. One possible reason is that the policy overfits to the data used in the simulator for training, which indicates the necessity of using model for the rainfall rather than the real data in the simulator.

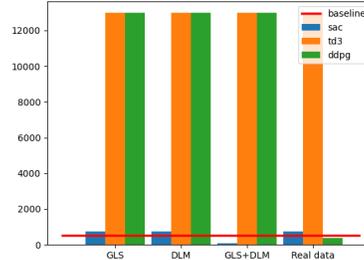


Figure 4: Testing of trained policies

6 Conclusion

In this work, we fist build simulators with the real data and mathematical models and use the simulator to train the efficient water release policy of the dam through deep RL. Experiments show that our methods outperform the existing baselines. Future works include building simulators with more realistic flood damage model of the downstream and more data for predicting rainfalls in the upstream and ultimately deploying to the real dams for evaluations. We target to build the AI-based dam operation system and deploy to the real dam, e.g., Bansagar dam, to improve the efficiency.

¹We try 5 seeds and tune the algorithms’ parameters, then choose the best one to plot and test.

Acknowledgements

This work is supported by NTU, ATREE and Google Research (Google grant on AI for Social Good). The ATREE team also acknowledges the support from India's National Mission on Biodiversity and Human Well-Being.

References

- [1] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *ICML*, pages 1587–1596, 2018.
- [2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, pages 1861–1870, 2018.
- [3] Sharad K. Jain. Introduction to reservoir operation. Technical report, National Institute of Hydrology: Roorkee, India, 2019.
- [4] KD Joshi, DN Jha, A Alam, SK Srivastava, Vijay Kumar, and AP Sharma. Environmental flow requirements of river sone: impacts of low discharge on fisheries. *Current Science*, pages 478–488, 2014.
- [5] Jagdish Krishnaswamy, Michael Lavine, Daniel D Richter, and Karl Korfmacher. Dynamic modeling of long-term sedimentation in the yadkin river basin. *Advances in Water Resources*, 23(8):881–892, 2000.
- [6] Jagdish Krishnaswamy, Srinivas Vaidyanathan, Balaji Rajagopalan, Mike Bonell, Mahesh Sankaran, RS Bhalla, and Shrinivas Badiger. Non-stationary and non-linear influence of ENSO and Indian Ocean Dipole on the variability of Indian monsoon rainfall and extreme rain events. *Climate Dynamics*, 45(1-2):175–184, 2015.
- [7] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.
- [8] Vimal Mishra, Saran Aadhar, Harsh Shah, Rahul Kumar, Dushmanta Ranjan Pattanaik, and Amar Deep Tiwari. The Kerala flood of 2018: Combined impact of extreme rainfall and reservoir storage. *Hydrology and Earth System Sciences Discussions*, pages 1–13, 2018.
- [9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [10] Tarun Nair, Ravi Chellam, and Jagdish Krishnaswamy. Estimation of ecological flows in the son gharial sanctuary (son river, madhya pradesh) with a focus on gharials *gavialis gangeticus*. <https://drive.google.com/file/d/0B3EXxrcBsJQxa0tza1BCCeDiLVE/view>, 2016.
- [11] Government of Kerala. Soil health status of Kerala in post flood scenario. <https://sdma.kerala.gov.in/wp-content/uploads/2020/08/Soil-Flood.pdf>, 2018.
- [12] Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. Dynamic linear models. In *Dynamic Linear Models with R*, pages 31–84. Springer, 2009.
- [13] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, pages 387–395, 2014.
- [14] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.
- [15] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.
- [16] Himanshu Thakkar. Tale of a disaster foretold: Bansagar dam ready to create another flood disaster along Sone and Ganga. <https://sandrp.in/2018/09/10/tale-of-a-disaster-foretold-bansagar-dam-ready-to-create-another-flood-disaster-along-sone-and-ganga/>, 2018.
- [17] Robyn J Watts, Brian D Richter, Jeffrey J Opperman, and Kathleen H Bowmer. Dam reoperation in an era of climate change. *Marine and Freshwater Research*, 62(3):321–327, 2011.

A Details of the Baseline Discharge Strategy

Table 3: Baseline discharge strategy from Nov to Jun. For other months, the discharge strategy will be releasing the water from the upstream, i.e., keeping the water level stable.

Ten-Daily Period	Discharge (cumecs)	Ten-Daily Period	Discharge (cumecs)
Nov 01-10	73.1	Mar 01-10	28.1
Nov 11-20	68.6	Mar 11-20	27.0
Nov 21-30	64.1	Mar 21-31	23.8
Dec 01-10	59.6	Apr 01-10	20.5
Dec 11-20	55.1	Apr 11-20	17.3
Dec 21-31	48.5	Apr 21-30	16.1
Jan 01-10	41.8	May 01-10	14.9
Jan 11-20	35.2	May 11-20	13.7
Jan 21-30	33.6	May 21-31	12.5
Feb 01-10	31.9	Jun 01-10	11.3
Feb 11-20	30.3	Jun 11-20	57.6
Feb 21-28	29.2	Jun 21-30	104.0