A novel nested stochastic dynamic programming (nSDP) and nested reinforcement learning (nRL) algorithm for multipurpose reservoir optimization

Blagoj Delipetrev, Andreja Jonoski and Dimitri P. Solomatine

ABSTRACT

In this article we present two novel multipurpose reservoir optimization algorithms named nested stochastic dynamic programming (nSDP) and nested reinforcement learning (nRL). Both algorithms are built as a combination of two algorithms; in the nSDP case it is (1) stochastic dynamic programming (SDP) and (2) nested optimal allocation algorithm (nOAA) and in the nRL case it is (1) reinforcement learning (RL) and (2) nOAA. The nOAA is implemented with linear and non-linear optimization. The main novel idea is to include a nOAA at each SDP and RL state transition, that decreases starting problem dimension and alleviates curse of dimensionality. Both nSDP and nRL can solve multi-objective optimization problems without significant computational expenses and algorithm complexity and can handle dense and irregular variable discretization. The two algorithms were coded in Java as a prototype application and on the Knezevo reservoir, located in the Republic of Macedonia. The nSDP and nRL optimal reservoir policies were compared with nested dynamic programming policies, and overall conclusion is that nRL is more powerful, but significantly more complex than nSDP.

Key words | algorithm, optimal reservoir operation, reinforcement learning, stochastic dynamic programming

Blagoj Delipetrev (corresponding author)
University Goce Delcev,
Krste Misirkov bb,
2000 Shtip,
Republic of Macedonia
E-mail: *blagoj.delipetrev@ugd.edu.mk*

Andreja Jonoski
Dimitri P. Solomatine
UNESCO-IHE Institute for Water Education,
Delft,
The Netherlands

Dimitri P. Solomatine
Water Resources Section,
Delft University of Technology,
Delft,
The Netherlands

INTRODUCTION

Optimal reservoir operation (ORO) is a multi-objective problem that is often solved by dynamic programming (DP) and stochastic dynamic programming (SDP). These two methods suffer from the so-called 'dual curse' which forbids them to be employed in reasonably complex water systems. The first one is the 'curse of dimensionality' that is characterized with an exponential growth of the computational complexity with the state–decision space dimension (Bellman 1957). The second one is the 'curse of modelling' that requires an explicit model of each component of the water system (Bertsekas & Tsitsiklis 1996) to calculate the effect of each system's transition. The curse of dimensionality limits the number of state–action variables and prevents DP and SDP being used in complex reservoir optimization problems.

doi: 10.2166/hydro.2016.243

There have been various attempts to overcome the curses (Castelletti *et al.* 2012; Li *et al.* 2013; Anvari *et al.* 2014), or earlier DP-based on successive approximations (Bellman & Dreyfus 1962), incremental DP (Larson 1968) and differential DP (DDP) (Jacobson & Mayne 1970). The DDP starts with an initial guess of values and policies for the goal and continues with improving the policy using different techniques (Atkeson & Stephens 2007). The incremental DP attempts to find a global solution to a DP problem by incrementally improving local constraint satisfaction properties as experience is gained through interaction with the environment (Bradtke 1994).

In the last decade, there has been significant RL research and applications in ORO. Researchers from the Polytechnic University of Milan (Italy) have developed

SDP and a number of RL implementations in ORO (Castelletti *et al.* 2001, 2007). The study by Castelletti *et al.* (2002) proposes a variant of Q-learning named Qlp (Q-learning planning) to overcome the limitations of SDP and standard Q-learning by integrating the off-line approach, typical for SDP and model-free characteristic of Q-learning. The vast state–action space in most cases is extremely difficult to express with a lookup table so often a generalization through a function approximation (for example by a neural network) is required (see e.g., Bhattacharya *et al.* 2003). A similar approach, proposed by Ernst *et al.* (2006), called 'fitted *Q*-iteration', combines RL concepts of off-line learning and functional approximation of the value function. Recent RL methods (Castelletti *et al.* 2010) have been using tree-based regression for mitigating the curse of dimensionality. The application of various ORO methods are reviewed in Yeh (1985) and for multireservoir systems in Labadie (2004).

This paper address the multi-objective ORO problem of satisfying multiple objectives related to: (1) reservoir releases to satisfy multiple downstream users competing for water with dynamically varying demands; (2) deviations from target water levels in the reservoir (recreation and/or flood control); and (3) hydropower production that is a combination of the reservoir water level and the reservoir releases. This problem when posed has multiple objectives (eight in our case study) and decision variables (six in our case study), and it is unsolvable with standard ORO algorithms because of the curse of dimensionality. The main objective is to research and develop algorithms that can solve previously mentioned multi-objective ORO problems, alleviating the curse of dimensionality.

We have developed two new algorithms named nested SDP (nSDP) and nested RL (nRL). These algorithms are similar to an already published nested dynamic programming (nDP) algorithm (Delipetrev *et al.* 2015) that is compared with already existing DP methods. At each state transition of the nSDP and nRL, an additional nOAA is executed to allocate optimal releases to individual water users, which lowers the starting problem dimension and successfully alleviates the curse of dimensionality. The nOAA is implemented with (1) simplex for linear allocation and (2) weighted quadratic deficits for non-linear allocation.

The nSDP and nRL algorithms were tested at the Knezevo multipurpose reservoir of the Zletovica hydro-system located in the Republic of Macedonia. The Zletovica hydro-system is a relatively complex water resource system, including one reservoir, Knezevo, significant tributary inflow downstream of the reservoir, several intake points, several water supply and irrigation users, and hydropower. The specific problem addressed here is how to operate the Knezevo reservoir, to satisfy as much as possible water users and other objectives. The main issue is to include five water users, two towns and two agricultural users, ecological demand, minimum and maximum reservoir critical levels, and hydropower, creating an optimization problem with a total of eight objectives and six decision variables.

This article is organized in six sections. The next section describes the ORO problem and the nSDP and nRL algorithms followed by a section explaining the Zletovica case study and optimization problem formulation. As the Zletovica case study is not a classical single reservoir optimization problem, in the next section we describe the nSDP and nRL implementation and then the experimental settings. Results and discussion follow and finally the conclusions.

THE ORO PROBLEM AND NOVEL ALGORITHMS

nSDP

The classical SDP ORO presented in Loucks & Van Beek (2005, pp. 244–251) is based on the following Bellman equation (Bellman 1957):

$$V(x_t) = \min\{g(x_t, x_{t+1}, a_t) + \gamma^t \cdot \sum_j p_{q_{t+1}|q_t} \cdot V(x_{t+1})\} \quad (1)$$

where $V(x_t)$ represents state value function, $g\ (x_t, x_{t+1}, a_t)$ represents the reward function of transition between state x_t and state x_{t+1}, a_t is the decision–action vector including releases for multiple users, γ is the discount factor to ensure convergence and $p_{q_{t+1}|q_t}$ is the probability P_{ij}^t for a reservoir inflow q_t that is in interval i in time step t, to become q_{t+1}^j that is in interval j in time step $t+1$, also

shown in Equation (2). The minimization is performed on state and action variables:

$$P_{ij}^{t} = P\left\{ q_{t+1}^{j} \text{ in interval j} | q_{t}^{i} \text{ in interval i} \right\} \quad (2)$$

$$\sum_{i} P_{ij}^{t} = 1 \quad (3)$$

Often the state x_t includes the reservoir storage s_t, and the hydrometeorological information; which, in our case, is the reservoir inflow q_t in the time $[t,\ t+1]$, making the state vector $x_t = \{s_t,\ q_t\}$. The reservoir is governed by the mass balance equation:

$$s_{t+1} = s_t + q_t - r_t - e_t \quad (4)$$

where r_t is the reservoir release and e_t is the reservoir evaporation.

The action vector a_t defines the policy p based on the state x_t as shown in Equation (5):

$$a_t = p(x_t) \quad (5)$$

The objective functions (OFs) can be aggregated into single-objective aggregated weighted sum function as show in Equation (6):

$$g_t(x_t, x_{t+1}, a_t) = \sum_{i=1}^{n} w_{it} \cdot g_{it}(x_t, x_{t+1}, a_t) \quad (6)$$

where $g_t\ (x_t, x_{t+1}, a_t)$ is the aggregated reward of n objectives at time step t, w_{it} is the objective weight at time step t and $g_{it}\ (x_t, x_{t+1}, a_t)$ is the step reward of each objective at time step t.

A specific characteristic of the problem considered in this article is that this release is to be divided between n competing users, and this multiplies the total number of decision variables. This bring us to the main novelty and idea of the nested algorithms, and that is to execute nOAA at each state transition, optimizing releases for the water demand users, considering at the same time other objectives and constraints. That is the only difference between the nSDP and classical SDP, or the executing of the nOAA at each state transition, as shown in Figure 1. The $d_{1t} \ldots d_{nt}$ are the water demands, while $r_{1t} \ldots r_{nt}$ are their respected releases at each time step, as shown in Figure 1. The reward $g_t\ (x_t, x_{t+1}, a_t)$ is calculated at each nSDP transition, taking into account all objectives including starting and ending storage volumes s_t and s_{t+1}, release r_t that is divided by nOAA to all users and functions $r_{1t} \ldots r_{nt}$ depending on their demands $d_{1t} \ldots d_{nt}$ and priorities. Additional explanations about nesting and its features can be found in the nDP paper (Delipetrev *et al.* 2015).

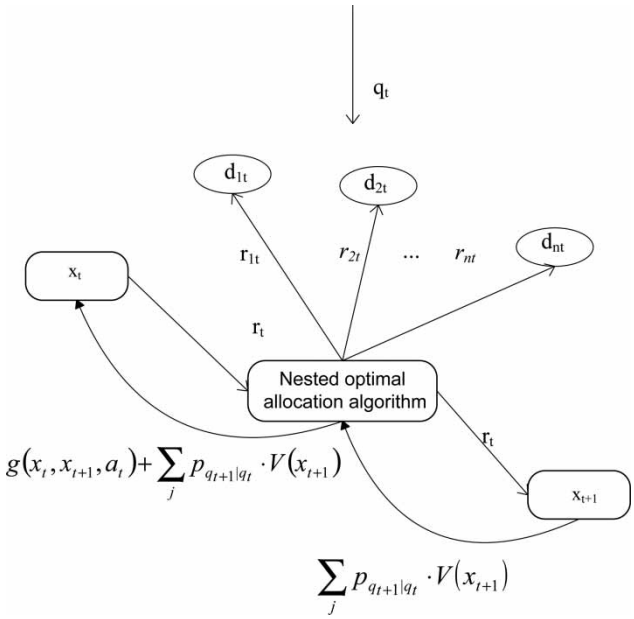

Figure 1 | Transition at time step t of the nSDP algorithm.

The nSDP pseudo code is shown in Algorithm 1:

(1) Discretize the reservoir inflow q_t into L intervals i.e., q_{lt} $(k = 1, 2\ldots, L)$.

(2) Create the transition matrices TM that describe the transition probabilities $p_{q_{t+1}|q_t}$.

(3) Discretize storage s_t and s_{t+1} in m intervals, i.e., s_{it} $(i = 1, 2, \ldots, m)$, s_{jt+1} $(j = 1, 2, \ldots, m)$ (in this case $x_t = s_t$) and set $k = 0$.

(4) Set time $t = T - 1$ and $k = k + 1$.

(5) Set reservoir level $i = 1$ (for time step t).

(6) Set reservoir level $j = 1$ (for time step $t + 1$).

(7) Set inflow cluster $l = 1$ (for time step t).

(8) Calculate the total release r_t using Equation (4).

(9) Execute the nested optimization algorithm to allocate the total release to all users $\{r_{1t},\ r_{2t}, \ldots r_{nt}\}$ in order to meet their individual demands.

(10) Calculate the $g\ (x_t,\ x_{t+1},\ a_t)$ and update $V(x_t)$.

(11) $l = l + 1$.

(12) If $l \leq L$, go to step 8.

(13) $Jj = j + 1$.

(14) If $j \leq m$, go to step 7.

(15) Select the optimal actions (decision variables) $\{a_{1t},\ a_{2t}\ldots a_{nt}\}_{opt}$, which consist of the optimal transition $\{x_{t+1}\}_{opt}$ and the users releases $\{r_{1t},\ r_{2t}\ldots r_{nt}\}_{\mathrm{opt}}$ that give minimal value of $V(x_t)$.

(16) $i = i + 1$.

(17) If $i \leq m$, go to step 6.

(18) If t > 0

(19) $t = t - 1$

(20) Go to step 4.

(21) If t = 0, check if the optimal actions (decision variables) $\{a_{1t},\ a_{2t}\ldots a_{nt}\}_{opt}$, are changed from the previous episode (or in the last three consecutive episodes)? If they are changed, go to step 4, otherwise stop.

Underlined step 9 in Algorithm 1 is the nOAA. Algorithm 1 presents the general nSDP algorithm that depending on the case study needs to be adjusted, as will be demonstrated in the following sections.

nRL

Reinforcement learning (RL) is a machine learning method that maps situation and actions to maximize the cumulative reward signal. The RL components are an agent, an environment, and a reward function. The environment is observable to the agent through state x_t (state variables). The agent observes state x_t and takes action a_t. The environment reacts to this action, and based on the changes in the environment, gives a reward $g\ (x_t,\ x_{t+1},\ a_t)$ to the agent. The main difference between the RL and SDP is while SDP makes an exhaustive optimization search over all possible state–action space, the RL optimization is incremental for the currently visited state, or SDP applies breadth-first search, while RL applies single-step depth-first search (Lee & Labadie 2007).

Although there are several RL methods for solving Markov decision problems, the most popular is the Q-learning method (Sutton & Barto 1998). The Q-learning updates the state–action value function incrementally, rather than performing a complete replacement:

$$Q(x_t, a_t) = Q(x_t, a_t) + \alpha \cdot [g(x_t, x_{t+1}, a_t) + \gamma \cdot \max Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)] \quad (7)$$

where $Q\ (x_t,\ a_t)$ is the state–action value function, α is the learning rate coefficient, x_t, a_t, γ and $g\ (x_t,\ x_{t+1},\ a_t)$ have been described before. The maximization is performed on state and action variables.

The nRL design can support several state x_t and action a_t variables. One of the possible nRL design is to define the state $x_t = \{t,\ s_t,\ q_t\}$, action $a_t = \{s_{t+1}\}$ and reward $g\ (x_t,\ x_{t+1},\ a_t)$. Often, the RL action is defined by the reservoir release $a_t = \{r_t\}$, but since these variables are dependent in the mass balance Equation (4) it does not make any conceptual difference, e.g., when the next reservoir volume s_{t+1} is known, the release r_t can be calculated, or vice versa. There are RL action implementation differences between the next reservoir volume s_{t+1} and the reservoir release r_t, which are discussed in the section 'nRL application to the case study'.

If we assume there are N years of available historical time series data of reservoir inflow, these data are divided appropriately in N episodes. The RL agent includes several parameter settings as previously described: α – the learning rate; γ – the discount factor; M – the maximum number of episodes that defines the maximum number of episodes the agent will perform (this is the stopping criterion preventing the RL infinite loop); LT – learning threshold and LR – learning rate. LR is the sum of all the learning updates $|Q\ (x_{t+1},\ a_{t+1}) - Q\ (x_t,\ a_t)|$ in one episode, as shown in Equation (8). If LR is below some predefined threshold named LT, then the RL should stop learning:

$$LR = \sum_{t=0}^{T} |Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)| \quad (8)$$

The nRL pseudo code is presented in Algorithm 2:

(1) Divide the inflow into N episodes for each year.

(2) Discretize the reservoir inflow q_t into L intervals, making L intervals centers q_{lt} $(k = 1, 2\ldots, L)$.

(3) Discretize storage s_t in m intervals, making m discretization levels s_{it} $(i = 1, 2, \ldots, m)$.

(4) Set initial variables: α, γ, maximum number of episodes – M, learning threshold – LT.

(5) Set T as period that defines the number of time steps t in episode (in our case 52 for weekly and 12 for monthly).

(6) Set $LR = 0$.

(7) Set $n = 1$ (number of an episode).

(8) Set $t = 1$ (time step of a period).

(9) Define initial state x_t with selecting a starting reservoir volume s_{it}.

(10) Get the reservoir inflow q_{lt} and t from the current episode.

(11) Select action a_t (exploration, or exploitation) and make transition x_{t+1}.

(12) Calculate the reservoir release r_t based on x_t, x_{t+1}, q_{kt}, and Equation (4).

(13) Execute the nested optimization with distributing the reservoir release r_t between water demand users using linear or quadratic formulation, calculate the deficits and other objectives, and calculate the reward $g(x_t, x_{t\pm1}, a_t)$.

(14) Calculate the state action value $Q(x_t, a_t)$.

(15) Calculate learning update $|Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)|$ and add it to LR.

(16) $t = t + 1$ and move agent to state x_{t+1}.

(17) If $t < T$ then go to step 10.

(18) If $t = T$ then $n = n + 1$.

(19) If $n < N$ then set new episode data and go to step 8.

(20) If $n = N$ and $LR > LT$ then go to step 6.

(21) If $n = N$ and $LR < LT$ then stop.

(22) If $n = M$ then stop.

The main nRL feature is step 13 that executes the nOAA. The nRL design can support the additional state variables, as will be demonstrated in the case study implementation presented in the following sections.

Approaches to nested optimization of step-wise resource allocation

The nOAA are the same as in Delipetrev *et al.* (2015) where two methods are used to optimally allocate the total reservoir release r_t between n water users: simplex method in the case of linear problem and weighted quadratic deficit for non-linear problem. Each water user is described with its demand d_{it} and corresponding weight W_{it} at time step t. For the nested optimal allocation, the following variables are relevant: d_{1t}, d_{2t}…d_{nt} are users' demands; W_{1t}, W_{2t}… W_{nt} are the corresponding demands' weights; r_t is the reservoir release; $r_{1t..}r_{nt}$ are the users' releases; v is the release discretization value.

Note that at the beginning of each nested optimization, the nOAA check if the release r_t can satisfy the aggregated demand of all users in Equation (9):

$$\text{If } \sum_{i=0}^{n} d_{it} < r_t \text{ then } r_{1t} = d_{1t}, r_{2t} = d_{2t}, r_{nt} = d_{nt} \tag{9}$$

If the release r_t can satisfy the aggregated demand of all users, then the optimal allocation is not performed since all the releases can be set to their demands.

Linear method

In the considered reservoir optimization problem, the simplex method is used for solving the following linear programming optimization problem:

$$\min \sum_{i=1}^{n} W_{it} \cdot (d_{it} - r_{it}) \tag{10}$$

subject to:

$$r_{1t} + r_{2t} \ldots + r_{nt} \leq r_t \tag{11}$$

$$r_{1t} \leq d_{1t}, r_{2t} \leq d_{2t..}, r_{nt} \leq d_{nt} \tag{12}$$

$$r_t, d_{t1}, d_{t2..}, d_{nt} \geq 0 \tag{13}$$

Minimization of the optimization problem is performed on the release variable r_{it}.

Non-linear method

The weighted quadratic deficit is used when the OF is non-linear – this is the case when the squared weighted deficit of the demand objectives is to be minimized. The reservoir release r_t is assumed to be discretized in v levels; this value is set at the beginning and stays the same over nDP

execution. The quadratic OF is to minimize:

$$\min \sum_{i=1}^{n} W_{it} \cdot (d_{it} - r_{it})^2 \tag{14}$$

with the same constraints previously described in Equations (11)–(13). Again, the minimization is performed on the release variable r_{it}.

CASE STUDY

The Zletovica hydro-system is located in the eastern part of the Republic of Macedonia and it uses the water resources of the River Zletovica and its tributaries. It is composed of the Knezevo reservoir, several water distribution canals used for delivering water to downstream users and associated infrastructure structures, as shown in the schematic representation in Figure 2. The Knezevo reservoir is a multipurpose reservoir, and its main objective is to provide drinking water to several towns and populated areas in the region, as well as to provide environmental flows in Zletovica, water for agriculture and hydropower (in the exact order of decreasing priority).

There are five towns in this region (Kratovo, Probishtip, Zletovo, Shtip and Sveti Nikole) and two large agriculture irrigation regions named upper and lower zone. The region is characterized by mountainous topography; the system is also designed to include several small hydropower plants, most of which are located downstream of the Knezevo reservoir as derivational power plants that utilize the natural head differences created by the topography. The hydropower system is still in development and according to the feasibility study report (GIM 2010) the plan is to build eight hydropower plants.

The GIM (2008) report contains a detailed hydrological model of the Zletovo river basin with four river flow measurement points on the River Zletovica. The monthly time series data of the river flow measurement points from the year 1951 to 1991 are used in this research. There is a significant tributary inflow between the Knezevo reservoir and the last branching point. First, the tributary inflow is used to satisfy the water demand objectives, and if additional water quantities are needed then they are released from the reservoir.

The r_{it} represent each water user release quantity. The numbering (r_{3t} to r_{7t}) is selected to fit the optimization formulation in which objectives related to reservoir water level are numbered with indexes 1 and 2, as will be shown below. The solid line represents the main Zletovica river, and the significant left tributary. Some of the presented variables are further explained in the following subsection.

The hydro-system is modelled in a lumped way such that water from the tributary inflow q_t^{Tr} is first allocated to all users, and after that the reservoir releases are used to satisfy the remaining user demands. The tributary inflow q_t^{Tr} is calculated as a difference between the last river measurement point q_{3t} and reservoir inflow q_t. The analysis proved that this assumption holds and it is possible to consider tributary inflow q_t^{Tr} as the total water quantity available to all users. This approach decreases the number of system variables characterizing the water users (these two variables are used however for some hydropower calculations). In this system, the main water users are the towns Shtip and Sveti Nikole and both agricultural zones.

The maximum water level in the Knezevo reservoir considered for this study is $H_{max} = 1{,}061.5$ m amsl, which is, in fact, the normal operational level. This level corresponds to max storage volume of $V_{max} = 23.5 \times 10^6$ m^3. The minimum storage volume (dead storage) in the Knezevo reservoir is $V_{min} = 1.50 \times 10^6$ m^3 corresponding to $H_{min} = 1{,}015.0$ m amsl water level, leaving effectively 22.0×10^6 m^3 of storage volume in the Knezevo reservoir for balancing available inflows with downstream water demands.

Formulation of the optimization problem

The Knezevo reservoir optimization problem has eight objectives and six decision variables. Each objective is described by its target value and its corresponding weight at each time step. In this study, we aggregate all objectives into one OF being the weighted sum of squared deviations over the entire time horizon; referring to the Bellman equation the reward function has the following form:

$$g_t(s_t, s_{t+1}, a_t) = \sum_{i=1}^{8} w_{it} \cdot D_{it}^2 \tag{15}$$

where w_{it} is the objective weight for a given objective i and

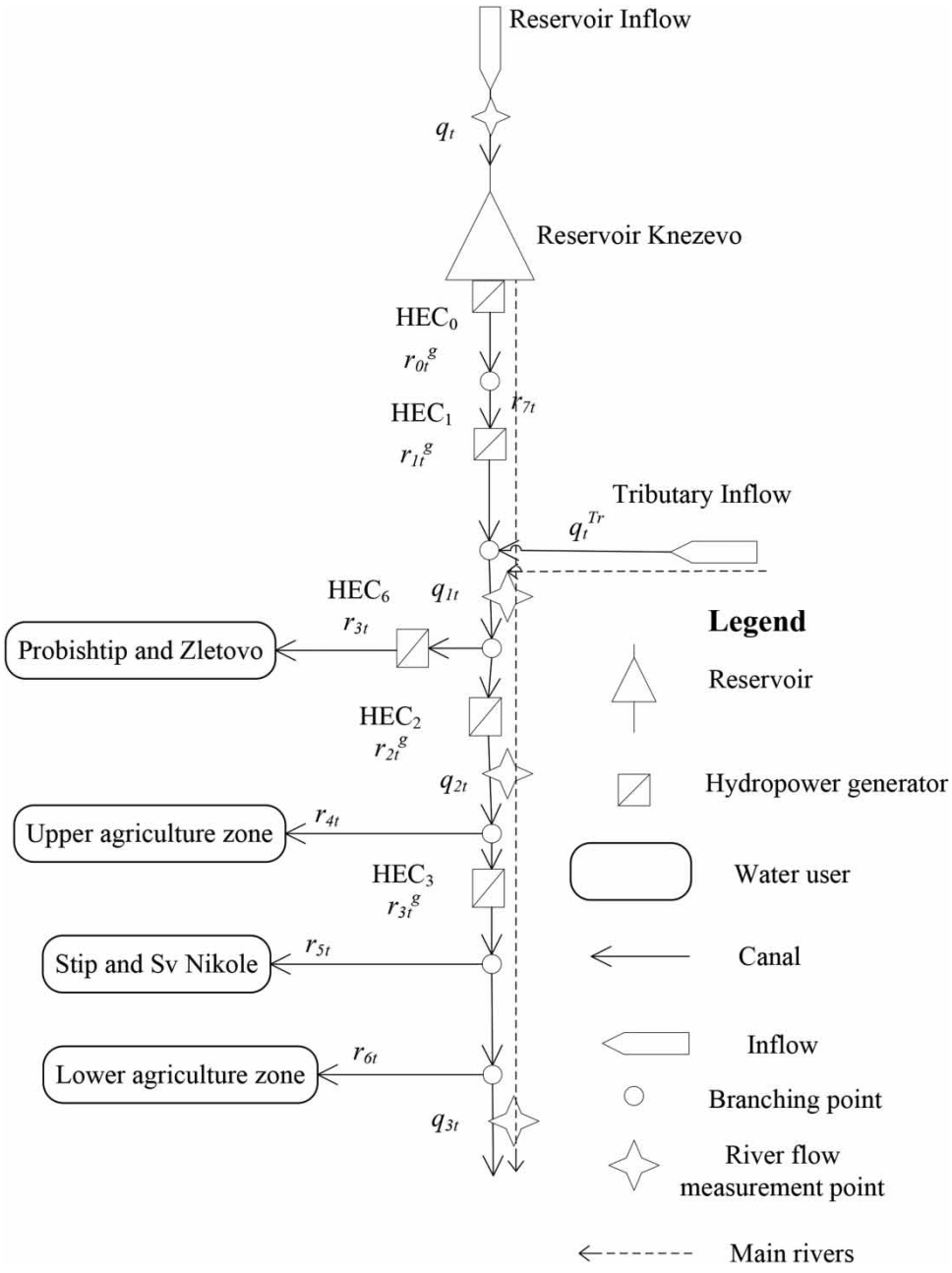

Figure 2 | Schematic representation of the Zletovica hydro-system.

time step t and D_{it} is the difference between the target value and decision variable for a given objective i and time step t.

The first two objectives relate to deviations from recreation and flood control water level targets:

$$D_{1t} = \begin{cases} 0, \text{ if } \quad l_t \geq \min l_t \\ \min l_t - l_t, \text{ if } \quad l_t < \min l_t \end{cases} \tag{16}$$

$$D_{2t} = \begin{cases} 0, \text{ if } \quad l_t \leq \max l_t \\ l_t - \max l_t, \text{ if } \quad l_t > \max l_t \end{cases} \tag{17}$$

where $min\ l_t$ and $max\ l_t$ are the recreation and flood water level targets. The purpose of the recreation level target is to preserve a minimum reservoir volume, especially as additional storage volume in case of emergency. The flood

level target prevents the reservoir reaching the reservoir spill height and avoids possible uncontrolled spills.

Based on the hydro-system configuration, our formulation has five users with water demand-related objectives. These are the following users: (1) the towns of Zletovo and Probishtip, (2) the upper agricultural zone, (3) the towns of Shtip and Sveti Nikole, (4) the lower agricultural zone, and (5) the minimum environmental flow, with their respective demands d_{3t}, d_{4t}, d_{5t}, d_{6t}, d_{7t}. The objectives deficits are calculated using Equation (18):

$$D_{it} = \begin{cases} 0, \text{ if } \quad d_{it} \le r_{it} \\ d_{it} - r_{it}, \text{ if } \quad d_{it} > r_{it} \end{cases} \tag{18}$$

Here, r_{it} is the release (decision variable) for a given objective (i) and time step (t).

The last objective is related to hydropower. Its corresponding formulation uses w_{8t} as the hydropower energy production weight and D_{8t} is calculated from:

$$D_{8t} = \begin{cases} 0, \text{ if } \quad \mathrm{h_t} \le p_t \\ h_t - p_t, \text{ if } \quad h_t > p_t \end{cases} \tag{19}$$

where h_t is the hydropower target demand and p_t is the hydropower production.

The five hydroelectric power plants (HEC), named HEC_0–HEC_3 and HEC_6 are dependent on the reservoir operation and they are the only ones considered in the optimization. HEC_0 is positioned at the Knezevo reservoir and the entire reservoir release r_t goes through the turbines of HEC_0. The reservoir release r_t is compared with the generator maximum water capacity HEC_{0max}, as in Equation (20):

$$r_{0t} = \begin{cases} r_1, \text{ if } \quad r_t \le HEC_{0\max} \\ HEC_{0\max}, \text{ if } \quad r_t > HEC_{0\max} \end{cases} \tag{20}$$

The r_{0t} is the reservoir release water quantity that goes over the turbines of HEC_0. The energy generated by HEC_0 (MWh) is:

$$HEC_{0t} = gen_0 \cdot r_{0t} \cdot \frac{l_t + l_{t+1}}{2} \cdot 24 \cdot \text{days in month} \tag{21}$$

where gen_0 is the coefficient that includes all conversion coefficients and total efficiency and h_t and h_{t+1} are the reservoir levels in time step t and $t+1$.

Other HEC are calculated in the same way as HEC_0, with the very important notion that HEC_2 and HEC_3 are using q_{1t} and q_{2t} variables. All HEC coefficients are taken from GIM (2010). All the hydropower plants together produce the total energy p_t.

The action vector a_t consist of six actions or decision variables: s_{t+1}, r_{3t}, r_{4t}, r_{5t}, r_{6t}, r_{7t} which are the next optimal reservoir state and water user releases at each time step. Using these decision variables, it is possible to calculate all other variables and OFs.

NSDP AND NRL ALGORITHMS IMPLEMENTATION

nSDP application to the case study

The nSDP was adjusted to accommodate the case study optimization problem formulation presented in the section 'Case study'. The main implementation issue in applying nSDP is how to include the four stochastic variables q, q_{1t}, q_{2t} and q_t^{Tr}, as shown in Figure 2. There is no example of numerical solution of SDP with four stochastic variables without provoking the curse of dimensionality. Perhaps it is possible to design it mathematically, but the practical implementation will probably be very difficult and impractical.

The alternative approach is to investigate the correlation between the reservoir q_t and the q_t^{Tr} tributary inflow. The correlation coefficient between these two variables is about 0.9 on weekly data. The high correlation gives the opportunity to include the tributary inflow q_t^{Tr} as another stochastic variable in the nSDP algorithm. If this were not the case (low correlation coefficient), then another approach would be needed. The nSDP with the two stochastic variables can be implemented only if the reservoir q_t and tributary inflow q_t^{Tr} belong to the same cluster at each time step. It is worth noting that the high correlation coefficient typically suggests that the values of both variables belong to the same cluster interval at each time step over the entire modelling period.

The correlation analysis between reservoir inflow q_t and tributary inflow q_t^{Tr} bring us to a possible solution to discard

other stochastic variables q_{1t} and q_{2t} and simplify the optimization problem formulation. The stochastic variables q_{1t} and q_{2t} are only used in calculation of the hydropower OF, and do not affect other OF. The consequence of optimization problem simplification and adjustment is the impossibility to calculate HEC_2 and HEC_3 power production (and the total hydropower production as well) using nSDP. Therefore, the hydropower aspect is not included in nSDP.

Algorithm 3 adds and changes several steps of Algorithm 1 to implement the Zletovica case study and is shown below:

(1a) Discretize the tributary inflow q_t^{Tr} into L intervals i.e., $q_{lt}^{Tr}(l = 1, 2 \ldots, L)$.

(2a) Create the transition matrices *TM* that describe the transition probabilities $p_{q_{t+1}^{Tr}|q_t^{Tr}}$ of tributary inflow q^{Tr}.

(7) Set reservoir inflow and tributary inflow cluster $l = 1$ (for time step t) (the reservoir and tributary inflow clusters are the same).

(7a) Distribute the tributary inflow q_{kt}^{Tr} using nested optimization between water demand users and calculate their remaining deficits.

(9) Execute the nested optimization algorithm to allocate the total release to all users $\{r_{3t}, r_{4t}, r_{5t}, r_{6t}, r_{7t}\}$ in order to meet their remaining deficits and calculate D1, D2, D3, D4, D5, D6, D7 and D8.

Algorithm 3 has additional steps (1a) and (2a) that are added after Algorithm 1 steps (1) and (2) correspondingly, and these steps calculate the tributary inflow q_t^{Tr} variable. Both variables, the reservoir inflow q_t and tributary inflow q_t^{Tr}, are discretized using K-means algorithm. Algorithm 3 step 7 replaces Algorithm 1 step 7, where the same cluster is set for the reservoir and tributary inflow. Steps (8a), (9) and (9a) are adding/replace steps as described previously.

nRL application to the case study

The nRL includes all case study variables (q_t^{Tr}, q_{1t} and q_{2t}), and implements the optimization problem formulation as described in the case study section. The nRL executes multiple episodes with deterministic variables time series data, where each episode is 1 year. The nRL implementation is very specific for this ORO problem described in the case study. That is why often designing and implementing RL (and other machine learning techniques) is an art, because the modellers construct the entire system, define variables, states, actions, rewards, etc.

The primary design decision in the nRL (and RL in general) is to determine the state, the action and the reward variables. Three different approaches to define states x_t were tested: (1) $x_t = \{t, s_t\}$, (2) $x_t = \{t, s_t, q_t\}$, (3) $x_t = \{t, s_t, q_t, q_t^{Tr}\}$. The nRL action and reward were the same in all three approaches. The action a_t is described with the next storage volume $a_t = \{s_{t+1}\}$ and consequently 'nested' releases $a_t = \{s_{t+1}, r_t, r_{3t}, r_{4t}, r_{5t}, r_{6t}, r_{7t}\}$. The reward $g(x_t, a_t, x_{t+1})$ is defined as an optimization problem formulation or Equation (15). The only difference is that deviation is with a negative sign and the nRL OF is to maximize negative deviation. The maximal gain is 0 when the objective is satisfied.

As mentioned before, the action can be described as the reservoir release $a_t = \{r_t\}$. This does not make any conceptual difference considering equations, since the next state s_{t+1} can be calculated from the mass balance equation, but it is more complicated to implement. In our case, the reservoir storage s_t and reservoir inflow q_t are discretized, and the evaporation e_t is calculated using s_t and s_{t+1}. If a reservoir release action r_t is selected, then based on the mass balance equation the calculated next reservoir volume s_{t+1} will fall in between two discretized storage volumes.

Instead, it is much more convenient and easier to implement the next reservoir volume as action $a_t = \{s_{t+1}\}$. In that case, the start and next discrete storage volumes s_t and s_{t+1}, and discretized reservoir inflow q_t are defined, and the evaporation e_t and the reservoir release r_t can be easily calculated.

The state space grows exponentially with the additional state variables. The state space directly influences the computational time and the agent's ability to learn. However, the action space stays the same due to the 'nested' methodology. A third approach was used, that increased the state vector dimension to about 94,900 cells (52 weeks × 73 reservoir level × 5 reservoir inflow discretization × 5 tributary inflow discretization). Because the agent explores/exploits the possible actions over the modelling period, it is very likely that some of the $Q(x_t, a_t)$ in the matrix will be unused. The solution selected for dealing with this issue was to use the HashMap function supported in Java. Both

nSDP and nRL are developed in Java. The nRL implementation pseudo code on the case study is shown in Algorithm 4 below:

(2a) Discretize the tributary inflow q_t^{Tr} into L intervals i.e., $q_{lt}^{Tr}(k = 1, 2 \ldots, K)$.

(9) Define initial state x_t with an initial reservoir volume s_t, read the reservoir q_{kt}, tributary inflow cluster value q_{kt}^{Tr}, q_{1t} and q_{2t} from the current *episode*, and the time step t.

(12a) Distribute the tributary inflow q_{kt}^{Tr} using nested optimization between water demand users and calculate their remaining deficits.

(13) Execute the nested optimization with distributing the reservoir release r_t between water demand users $\{r_{3t}, r_{4t}, r_{5t}, r_{6t}, r_{7t}\}$ satisfying the remaining deficits, calculate D_1, D_2, D_3, D_4, D_5, D_6, D_7 and D_8, and calculate the reward $g(x_t, x_{t+1}, a_t)$.

Algorithm 4 steps (2a) and (12a) are added after Algorithm 2 steps (2) and (12). Steps (9) and (13) of Algorithm 4 replace the same step from Algorithm 2.

EXPERIMENTAL SETTINGS

The available 55 years' weekly data are separated into two parts: (1) training and (2) testing. The data from 1951 to 1994 (2,340 time steps) are used for training and 1994–2004 (520 time steps) for testing. The nSDP training data consist of reservoir inflow q_t and tributary inflow q_t^{Tr} in the previously mentioned period. The nRL training data consist of reservoir q_t and tributary inflow q_t^{Tr}, and the two other flows q_{1t} and q_{2t} are used for hydropower calculation. The nSDP and nRL data for minimum and maximum levels, water supply, irrigation demands, ecological flow and hydropower are set to the 2005 weekly data presented in the case study section, and they are the same in the training and testing periods. The reservoir operation volume is discretized in 73 equal levels (300×10^3 m^3). The minimum level was set at 1,021.5 m amsl and the maximum level at 1,060 m amsl. The weights applied in these experiments are shown in Table 1. At the beginning, the nested optimization algorithm (linear or quadratic) and the number of clusters (in our case five) are selected in both nSDP and nRL. Both nSDP and nRL have the same experimental settings.

The main OF combines three distinct objective types: the minimum and maximum reservoir critical levels that are measured in m, the water user demands that are measured in 10^3 m^3/per time step (week or month) and the hydropower energy production that is measured in MWh/per time step (week or month). This is the main reason why the weights have different magnitudes. A similar approach is taken in other previous research studies (e.g., Pianosi & Soncini-Sessa 2009; Rieker 2010; Quach 2011).

The weights are set according to the objective importance and create the ORO policy. The most important objective is the environmental flow (w_7) followed by cities' water demands (w_3 and w_5), agriculture demands (w_4 and w_6), and lastly hydropower production (w_8). The hydropower weights were set extremely low for two main reasons: (1) the hydropower objective by the reports is considered as a by-product from reservoir operation and not its main feature; and (2) to lower as much as possible the influence of hydropower in ORO, and have a valid ground in comparing nSDP and nRL.

The nDP results of testing data are 'the optimal operation', meaning that nDP is a deterministic optimization and calculates the ORO. The nSDP and nRL, on the other hand, are trained on training data, producing a policy, and have not seen the testing data. The nDP results are used as a benchmark for the nSDP and nRL policies. The closer the policies derived by nSDP and nRL are to nDP, the better they are.

The algorithms are additionally labelled to denote the deficit formulations used in the nOAA. For example, nDP-L_5 stands for nDP using the linear deficits' formulation, and nDP-Q_5 stands for nDP using the quadratic deficits' formulation. The nRL parameters at the beginning are set at: $\alpha_0 = 0.8$, $\gamma = 0.5$ and $\varepsilon = 0.8$. The parameter α is set to decrease linearly with the number of episodes. The

Table 1 | nDP-L_5, nDP-Q_5, nSDP-L_5, nSDP-Q_5, nRL-L_5 and nRL-Q_5 experiments weights

Experiments	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
nDP-L_5, nDP-Q_5; nSDP-L_5, nSDP-Q_5; nRL-L_5, nRL-Q_5	2,000,000	2,000,000	200	1	200	1	300	0.01

maximum number of episodes is set to $M = 400{,}000$. There were various approaches tested for decreasing ϵ, and the one used in the experiments is decreasing ϵ on each 100,000 episodes by half. Starting at $\epsilon_0 = 0.8$ and with increasing the number of episodes the agent is making less exploration and more exploitation actions, and insuring convergence to the optimal solutions.

RESULTS AND DISCUSSION

The nRL agent learns, and after a number of episodes (10,000), the policy is tested. The nRL-L_5 agent learns the optimal policy and gets closer to nDP-L_5 ORO as the number of episodes increases, as shown in Figure 3. However, after a large number of episodes, the learning slightly deteriorates. The nRL-L_5 agent optimal reservoir policy is poor at 30,000 episodes, as shown in Figure 3. From 50,000 to 80,000 episodes, the nRL-L_5 policy improves, and between 80,000 and 160,000, the policy is the closest to the nDP-L_5 ORO. After 250,000 episodes the policy slightly deteriorates. The possible reason for this is overtraining, which is a known issue when using machine learning algorithms.

Another benchmark for the nRL optimal reservoir policy is to sum up the absolute difference between nDP optimal reservoir volume and nRL optimal reservoir volume at each time step in the testing period. The formula used is presented here:

$$S_n = \sum_{t=1}^{T} \left| s_t^{nDP} - s_t^{nRL} \right| \tag{22}$$

where s_t^{nDP} is the nDP-L_5 reservoir volume at time t and s_t^{nRL} is the reservoir volume of nRL-L_5 at time t.

The absolute difference between the nRL-L_5 and nDP-L_5 optimal reservoir volumes can be used as the stopping criterion. Obviously, the nRL-L_5 optimal reservoir policy performs best between 80,000 and 160,000 episodes of training. Afterwards, the policy somewhat deteriorates, although it is still relatively good.

The period 1999–2001 is very dry, because of low reservoir and tributary inflow, as shown in Figure 4. The

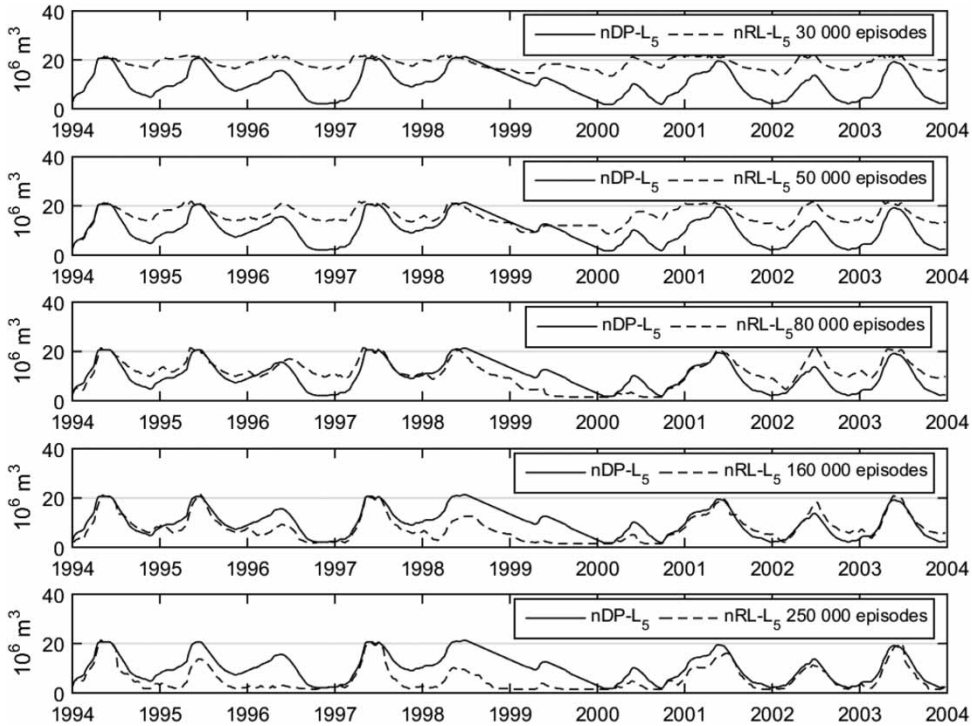

Figure 3 | nRL-L_5 agent learning with increasing the number of episodes: nDP-L_5 target reservoir storage (blue in online version) and nRL-L_5 obtained reservoir storage (red in online version) (testing period). Please refer to the online version of this paper to see this figure in colour: http://dx.doi.org/10.2166/hydro.2016.243.

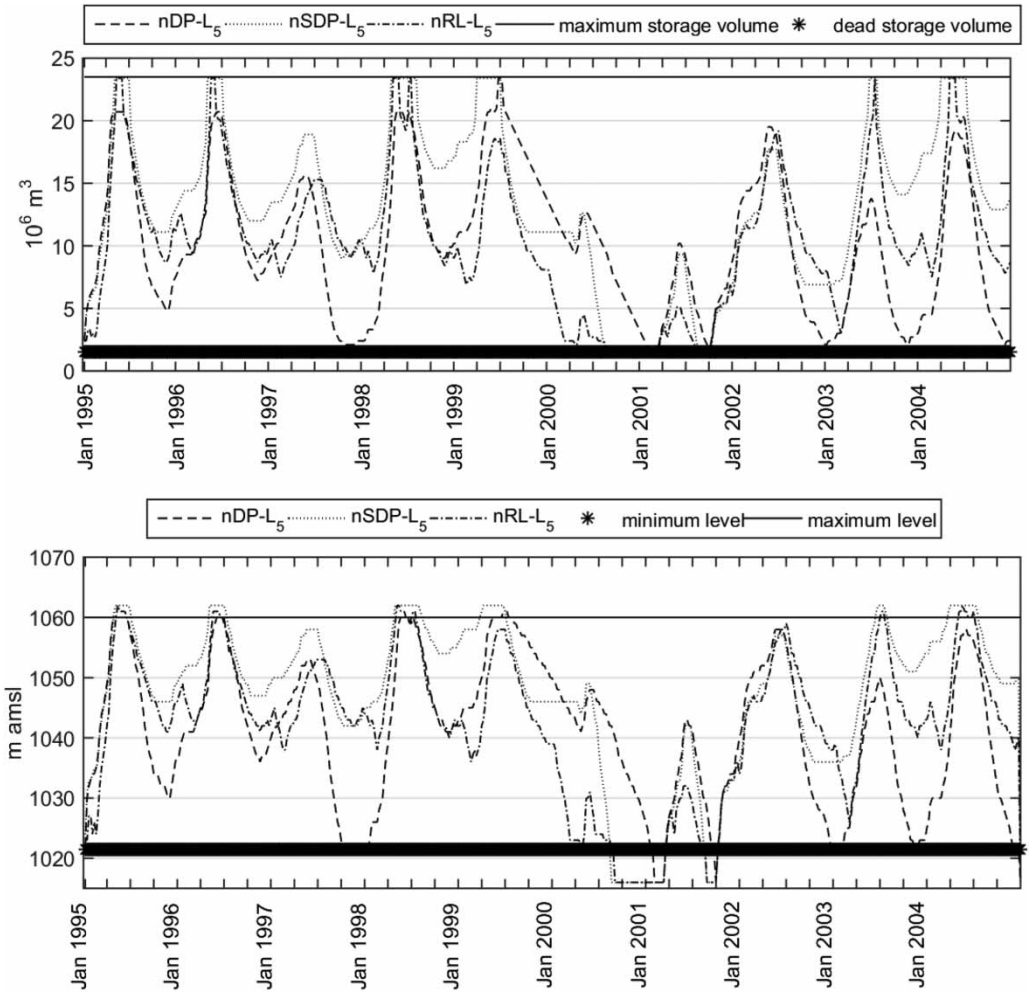

Figure 4 | nDP-L_5, nSDP-L_5 and nRL-L_5 optimal reservoir volume, minimum and maximum storage, optimal reservoir level and minimum and maximum levels in the testing period 1994–2004.

reservoir level (and reservoir volume) in all algorithms goes to the lowest possible level. During this period, the nDP-L_5 has a very limited minimum level violation, while nSDP-L_5 and nRL-L_5 have significant minimum level violation, as shown in Figure 4. The reason for this behaviour is that nDP-L_5 is a deterministic optimal algorithm that has perfect knowledge (forecast) of the future. When nDP-L_5 is applied to the 1994–2004 testing data, it means that an exhaustive search is performed to find the optimal deterministic solution. Therefore, nDP-L_5 as shown in Figure 4, performs an optimization policy in the period 1999–2001 that is leaving the reservoir on the minimum level only for a very short period of time. On the other side, both nSDP-L_5 and nRL-L_5 train/learn on training data and produce policies. The nSDP-L_5 and nRL-L_5 policies are assessed on the testing data. The training data 1951–1994 are the combination of wet, average and dry years that are fed to nSDP-L_5 and nRL-L_5 algorithms. Based on these data, both algorithms derive the optimal policy, which is a universal 1-year policy (for wet, average and dry years). Due to that, the nSDP-L_5 and nRL-L_5 performance in the period 1999–2001 leaves the reservoir empty for a longer period, and it is noticeable that nRL-L_5 performs better than nSDP-L_5. The overspills are considered and calculated and can be partially observed in the periods May 1999 and 2000 in Figure 4. The nDP-Q_5, nSDP-Q_5 and nRL-Q_5 results are similar to the nDP-L_5, nSDP-L_5 and nRL-L_5 correspondingly.

The nDP-L_5, nSDP-L_5, nRL-L_5, nDP-Q_5, nSDP-Q_5 and nRL-Q_5 optimization results and comparison of the sum of

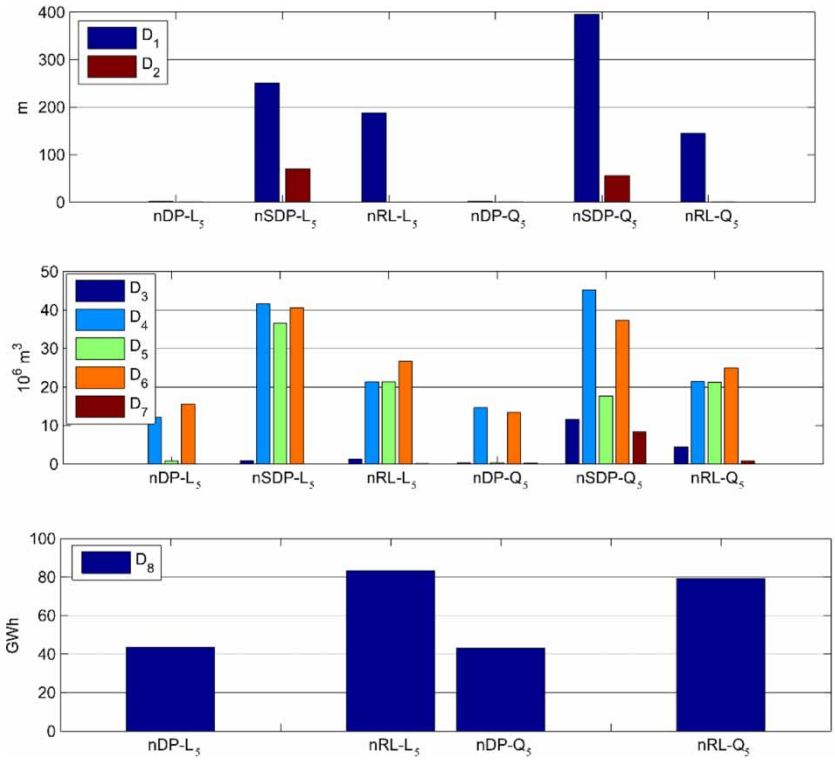

Figure 5 | nDP-L_5, nDP-Q_5, nSDP-L_5, nSDP-Q_5, nRL-L_5 and nRL-Q_5 comparison of the sum of minimum level (D_1) and maximum level (D_2) deviations, sum of users' deficit ($D_{3\text{-}7}$) and sum of hydropower deficit (D8) in the testing period 1994–2004.

minimum level (D_1) and maximum level (D_2) deviations, sum of users' deficit ($D_{3\text{-}7}$) and sum of hydropower deficit (D_8) in the testing period 1994–2004 are shown in Figure 5.

Figure 5 results show that nDP-L_5 and nDP-Q_5, which are the target, have very low D_1 and D_2 deviation, and that nRL-L_5 and nRL-Q_5 are better than nSDP-L_5 and nSDP-Q_5. The same applies in D_3–D_7 deviations, while in D_8 nSDP-L_5 and nSDP-Q_5 are not calculated. The nRL-L_5 and nRL-Q_5 produce better ORO policies than the nSDP-L_5 and nSDP-Q_5, as shown in Figure 5.

The overall results shown in Figure 5 correspond to the results shown in Figure 4, where it is obvious that the nRL policy performs better than nSDP, and it is closer to the target set by nDP.

CONCLUSIONS

The paper presented nSDP and nRL novel ORO algorithms that can solve problems with multiple decision variables, successfully alleviating the curse of dimensionality. These algorithms were implemented and tested in the case of the Zletovica hydro-system with eight objectives and six decision variables.

The nSDP has issues in implementing several state variables without provoking the curse of dimensionality, thus adjustments were needed to fit nSDP to the case study optimization problem requirement. The nRL showed its true power with including all four stochastic variables implementing the complete optimization problem formulation, but its implementation and tuning requires additional effort. The main conclusion from the implementation of the algorithms is that nDP can implement complex optimization problem formulations without significant problems. The nSDP has limitations when additional optimization problem variables are included. The nRL is very powerful in implementing complex optimization problems, but needs tuning concerning its design, parameters, action list, convergence criteria, etc.

The presented nSDP and nRL and their implementation in the case study of the Zletovica river basin confirmed that

in some situations the curse of dimensionality and computational complexity can be overcome. There could be situations where, for example, in applying nSDP and the correlation between the reservoir and tributary inflow is low, this proposed solution is not applicable. This restricts the possibility of nSDP application to a subset of reservoir problems. In this particular case, the two stochastic variables' approximation in nSDP was the best approach. In any case, the nSDP algorithm is limited in few stochastic variables before breaking the curse of dimensionality and becoming computationally unsolvable. On the other hand, nRL demonstrated its full capacity in including multiple stochastic variables and solving this problem and, at least on the conceptual level, can be applied to much more complex single and multireservoir problems.

The nSDP and nRL were used to derive 1-year weekly optimal reservoir policy. The available weekly data (1951–2004) were divided into a training (1951–1994) and testing part (1994–2004). The nSDP and nRL optimized/learned the optimal reservoir policy on training data, and their policy was examined on the testing data. The nDP solved the ORO problem in the testing period (1994–2004) and this solution was used as a target for both nSDP and nRL. Interesting results were to observe how the nRL agent learns with the increase of the number of episodes. The nRL optimal reservoir policy is best between 80,000 and 160,000 learning episodes. The nSDP and nRL policies were benchmarked against the nDP results and it was found that the nRL performs better than nSDP overall and for all objectives separately. The main conclusion is that the nRL is a better choice than the nSDP, at least for the considered case study.

The presented nSDP and nRL algorithms are successfully tested on a relatively complex single ORO problem, as the Zletovica case study is. Generally, nSDP cannot be applied in two and more multireservoir systems because of the curse of dimensionality. On the other hand, nRL supports several stochastic variables, as demonstrated in this case study, and in our opinion could be a potential solution for multireservoir ORO problems. However, as stated previously, the nRL implementation is difficult and highly problem specific.

The developed nested algorithms are computationally efficient and can be run on standard personal computers. For the considered case study, on a standard PC, nDP executes in 1–3 min, nSDP in 2–5 min, while nRL is 8–15 min (the longest is nRL-Q).

The ORO is a multi-objective problem by its nature because often different objectives (water demands, hydropower and reservoir levels) are concerned. In this research, it was first reduced to a single objective optimization problem by employing the single-objective aggregated weighted sum function. It is possible to make several single-objective optimization algorithms that are executed multiple times with several weight sets, i.e., multi-objective optimization by a sequence of single-objective optimization searches. This method can be applied to nDP, nSDP and nRL, which will create fully fledged multi-objective algorithms. Future research will focus on designing and developing multi-objective variants on the nDP, nSDP and nRL producing Pareto set.

REFERENCES

Anvari, S., Mousavi, S. J. & Morid, S. 2014 Sampling/stochastic dynamic programming for optimal operation of multi-purpose reservoirs using artificial neural network-based ensemble streamflow predictions. *Journal of Hydroinformatics* **16** (4), 907–921.

Atkeson, C. & Stephens, B. 2007 Random sampling of states in dynamic programming. *Advances in Neural Information Processing Systems*, 33–40, NIPS Foundation.

Bellman, R. 1957 *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA.

Bellman, R. & Dreyfus, S. 1962 *Applied Dynamic Programming.* RAND Corporation, Santa Monica, CA, USA.

Bertsekas, D. P. & Tsitsiklis, J. N. 1996 *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, USA.

Bhattacharya, B., Lobbrecht, A. H. & Solomatine, D. P. 2003 Neural networks and reinforcement learning in control of water systems. *Journal of Water Resources Planning and Management* **129** (6), 458–465.

Bradtke, S. J. 1994 Incremental Dynamic Programming for On-Line Adaptive Optimal Control. PhD Dissertation, University of Massachusetts at Amherst, MA, USA.

Castelletti, A., Corani, G., Rizzoli, A., Soncini-Sessa, R. & Weber, E. 2001 A reinforcement learning approach for the operational management of a water system. In: *Proceedings of IFAC Workshop Modelling and Control in Environmental Issues*. 22–23 August, Elsevier, Yokohama, Japan.

Castelletti, A., Corani, G., Rizzolli, A., Soncinie-Sessa, R. & Weber, E. 2002 Reinforcement learning in the operational management of a water system. In: *IFAC Workshop on*

Modeling and Control in Environmental Issues. Keio University, Yokohama, Japan, pp. 325–330.

Castelletti, A., de Rigo, D., Rizzoli, A. E., Soncini-Sessa, R. & Weber, E. 2007 Neuro-dynamic programming for designing water reservoir network management policies. *Control Engineering Practice* **15** (8), 1031–1038.

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. & Soncini-Sessa, R. 2012 Stochastic and robust control of water resource systems: concepts, methods and applications. In: *System Identification, Environmental Modelling, and Control System Design*. L. Wang & H. Garnier (eds), Springer, London, UK, pp. 383–401.

Delipetrev, B., Jonoski, A. & Solomatine, D. 2015 A novel nested dynamic programming (nDP) algorithm for multipurpose reservoir optimization. *Journal of Hydroinformatics* **17** (4), 570–583.

Ernst, D., Geurts, P. & Wehenkel, L. 2006 Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* **6** (1), 503–556.

GIM 2008 *Project for Irrigation System of Multipurpose Hydro System Zletovica*. Technical Report. GIM.

GIM 2010 *Feasibility Project for 8 Hydropower Centrals of Hydro System Zletovica*. Technical Report. GIM.

Jacobson, D. & Mayne, D. 1970 *Differential Dynamic Programming*. Elsevier, New York, USA.

Labadie, J. W. 2004 Optimal operation of multireservoir systems: state-of-the-art review. *Journal of Water Resources Planning and Management* **130** (2), 93–111.

Larson, R. E. 1968 *State Increment Dynamic Programming*. Elsevier, New York, USA.

Lee, J. H. & Labadie, J. W. 2007 Stochastic optimization of multireservoir systems via reinforcement learning. *Water Resources Research* **43** (11), W11408.

Li, J.-Q., Zhang, Y.-S., Ji, C.-M., Wang, A.-J. & Lund, J. R. 2013 Large-scale hydropower system optimization using dynamic programming and object-oriented programming: the case of the Northeast China power grid. *Water Science & Technology* **68** (11), 2458–2467.

Loucks, D. P. & Van Beek, E. 2005 *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*. UNESCO Publishing, Paris, France.

Pianosi, F. & Soncini-Sessa, R. 2009 Real-time management of a multipurpose water reservoir with a heteroscedastic inflow model. *Water Resources Research* **45** (10), W10430.

Quach, X. 2011 Assessing and Optimizing the Operation of the Hoabinh Reservoir in Vietnam by Multi-Objective Optimal Control Techniques. PhD Dissertation, Polytechnic University of Milan, Milan, Italy.

Rieker, J. D. 2010 A Reinforcement Learning Strategy for Reservoir Operational Improvement of Riverine Water Quality. PhD Dissertation, Colorado State University, Fort Collins, CO, USA.

Sutton, R. & Barto, A. 1998 *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.

Yeh, W. 1985 Reservoir management and operations models: a state-of-the-art review. *Water Resources Research* **21** (12), 1797–1818.

First received 23 December 2015; accepted in revised form 10 August 2016. Available online 17 September 2016