DOCUMENT DE TRAVAIL 2002-013

A NEURO-DYNAMIC PROGRAMMING APPROACH FOR
STOCHASTIC RESERVOIR MANAGEMENT

Abdeslem Boukhtouta,
Bernard Lamond
A Neuro-Dynamic Programming Approach for Stochastic Reservoir Management

A. Boukhtouta\textsuperscript{1} \& B. F. Lamond\textsuperscript{2}

\textsuperscript{1}Defence Research and Development Canada Valcartier, Canada
\textsuperscript{2}Département Opérations et systèmes de décision, Université Laval, Canada

Abstract

We propose an approach based on neural networks for optimizing a single hydroelectric reservoir. A stochastic neuro-dynamic programming algorithm is used to approximate the future value function by a neural function. The latter is used in deriving the optimal policy. The approximation architecture, based on the feed-forward network, gives very smooth approximate functions even with a coarse discretization of the state and action variables. The hydroelectric reservoir model presented in our study assumes a piecewise linear reward of the electricity produced and takes into account the turbine head effects and the stochastic inflows. The method is illustrated with a numerical example.

1 Introduction

We consider a mathematical model for optimizing the expected discounted rewards of a hydroelectric reservoir with random inflows and turbine head variations, and for which the one-period benefits are given by a concave, piecewise linear function of the energy produced. Such a model can be used by a public utility, for instance, for planning its electricity production over a planning horizon of several years, with monthly or quarterly time steps, say. In Lamond [1], this model was solved by dynamic programming using a piecewise polynomial approximation of the value function, which gives the expected discounted future rewards. Here, we solve the same model using an adaptation of the neuro-dynamic programming (or NDP) method of Bertsekas and Tsitsiklis [2].

The special structure of optimal decision rules for a reservoir with stochastic inflows and concave, piecewise linear rewards was first obtained by Gessford and
Karlin [3], whose model neglected the turbine head variations. Under this simplifying assumption, they showed the value function to be concave at every time step if the state variable is the volume of water in storage at the beginning of the period. The special structure was extended by Lamond [1] to the less restrictive situation in which the turbine efficiency increases with the water head. In this case, concavity of the value function does not hold with respect to the volume of water in storage, but it was shown to hold with respect to the equivalent potential energy. A piecewise polynomial approximation of the function of expected discounted future rewards was then constructed, using the special structure of the optimal decision rules. Here, we approximate the expected discounted rewards using a neural network approach, and we exploit the special structure of optimal decision rules for training the neural network (i.e., estimating its parameters) using only a coarse discretization of the state space. The NDP approach thus provides an easy to compute, accurate approximation of the value function and optimal operating policy.

2 Reservoir optimization model

We use the notation of Lamond [1]. The planning horizon comprises $T$ periods, denoted by the index $t = 1, \ldots, T$, each period representing a time step of, say, one month or one quarter. We assume the operation of the reservoir obeys the laws of a Markov decision process (MDP) (see, e.g., Lamond and Boukhtouta [4]). The evolution of the system is described with three variables at each period:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_t$</td>
<td>state (volume of water in the reservoir at the beginning of period $t$)</td>
</tr>
<tr>
<td>$A_t$</td>
<td>action (volume of water in the reservoir after electricity generation)</td>
</tr>
<tr>
<td>$D_t$</td>
<td>random noise (volume of uncontrolled inflows into the reservoir),</td>
</tr>
</tbody>
</table>

where the random variables $D_1, D_2, \ldots, D_T$ are assumed i.i.d. (independent and identically distributed).

In this model, the volume of water flowing through the penstocks in period $t$, and thus contributing to electricity generation, is equal to $S_t - A_t$. We assume no upper limit on the turbine capacity, but this could be included in the model by adding an extra line segment with null slope to the piecewise linear reward function eq. (3) below, as in Lamond, Monroe and Sobel [5]. On the other hand, we assume the natural inflows received in the reservoir during period $t$ are not available for electricity production until the beginning of period $t + 1$, and that the action $A_t$ is selected based on the knowledge of the observed state $S_t$ but not the random inflow $D_t$ which is observed only at the end of period $t$. Moreover, we assume there is an upper limit $U$ to the reservoir storage capacity such that any amount of water in excess of $U$ is lost instantly through the sluice, without producing electricity. Hence the system obeys the transition equation

$$S_{t+1} = (A_t + D_t) \land U,$$

where the notation “$x \land y$” denotes “$\min\{x, y\}”.

2
We also assume a reward \( R_t \) is received in period \( t \) depending on the quantity of electricity \( E_t \) generated during the period. Let the function \( \vartheta(s) \) give the water head on the turbines when a volume \( s \) of water is stored in the reservoir. Then, assuming the instantaneous power is proportional to both the head and the flow of water, the quantity of energy generated during period \( t \) is given by

\[
E_t(s_t, a_t) = \int_{s_t}^{a_t} \vartheta(x) \, dx = \Theta(s_t) - \Theta(a_t),
\]

where \( \Theta(s) = \int_0^s \vartheta(x) \, dx \) is interpreted as the potential energy associated with a volume \( s \) of water in storage. The reward is assumed to be obtained by selling the first \( Q_t \) units of electricity at a price \( h_1t \) dollars per unit, and by selling all exceeding electricity at a price \( h_2t \) dollars per unit, where \( h_1t > h_2t > 0 \), as in Figure 1.

Defining the piecewise linear function

\[
\rho_t(E_t) = \begin{cases} 
  h_1t E_t & \text{if} \ 0 \leq E_t \leq Q_t, \\
  h_1t Q_t + h_2t (E_t - Q_t) & \text{if} \ E_t > Q_t,
\end{cases}
\]

the reward in period \( t \) is given by

\[
R_t = r_t(S_t, A_t) = \rho_t(\Theta(S_t) - \Theta(A_t)).
\]

The objective of the reservoir operator is to set the action variables \( A_1, A_2, \ldots, A_T \) in such a way as to maximize the expected discounted total reward

\[
\mathbf{E}\left[ \sum_{t=1}^{T} \beta^{t-1} R_t \right],
\]

with “\( \mathbf{E}[X] \)” denoting the mathematical expectation of a random variable \( X \), and where \( \beta \) is the one-period discount factor, with \( 0 \leq \beta \leq 1 \). According to MDP
theory, for each period $t$, there is an optimal decision rule $\pi_t(s_t)$ such that the action $a_t = \pi_t(s_t)$ should be chosen whenever the state $s_t$ is observed.

### 3 Structure of optimal policies

Mathematically, the optimal decision rules can be defined using Bellman’s principle of optimality and the method of backward induction. Assuming the function $V_{T+1}(s_{T+1})$ of terminal rewards is defined for all $s_{T+1} \in [0, U]$, then for $t = T, \ldots, 1$, we get

\[
W_t(a_t) = E[V_{t+1}((a_t + D_t) \wedge U)] \quad \forall a_t \in [0, U] \tag{5}
\]

\[
V_t(s_t) = \max_{0 \leq a_t \leq s_t} r_t(s_t, a_t) + \beta W_t(a_t) \quad \forall s_t \in [0, U]. \tag{6}
\]

A decision rule for period $t$, such that the action $a_t = \pi_t(s_t)$ is optimal if state $s_t$ is observed, is then defined by

\[
\pi_t(s_t) = \arg \max_{0 \leq a_t \leq s_t} r_t(s_t, a_t) + \beta W_t(a_t) \quad \forall s_t \in [0, U].
\]

Moreover, if the function $\pi_t(s_t)$ is given, then substitution in eq. (6) gives the expression

\[
V_t(s_t) = r_t(s_t, \pi_t(s_t)) + \beta W_t(\pi_t(s_t)) \tag{7}
\]

for the optimal value function.

The functions $V_t(s_t)$ are not concave in general, but it was shown in Lamond [1] that the change of variables $x_t = \Theta(s_t)$ yields functions $v_t(x_t) = V_t(\Theta^{-1}(x_t))$ that are concave, provided $v_{T+1}(x_{T+1})$ is concave and $\theta'(s)/\theta(s)$ is nonincreasing. An analysis similar to that of Gessford and Karlin [3], but using the potential energy $x_t$ as state variable rather than the stored water volume $s_t$, and then back transforming into $s_t$, reveals the structure of the optimal decision rules (see Lamond [1] for proofs).

First, let the function $\varphi_t(s_t)$ give the action $a_t$ such that $E(s_t, a_t) = Q_t$. This action thus corresponds to a hydroelectric production of exactly $Q_t$ units of energy. When storage is insufficient, we define $\varphi_t(s_t) = 0$. Then

\[
\varphi_t(s_t) = \begin{cases} 
\Theta^{-1}(\Theta(s_t) - Q_t) & \text{if } \Theta(s_t) \geq Q_t, \\
0 & \text{else.}
\end{cases} \tag{8}
\]

For example, with constant turbine efficiency ($\theta(s) = \theta_0$), we have

\[
\varphi_t(s_t) = (s_t - Q_t/\theta_0)^+,
\]

where “$(x)^+$” denotes the positive part of $x$. When the turbine efficiency is an affine function of storage ($\theta(s) = \theta_0 + \theta_1 s$), we have that $a = \varphi_t(s_t)$ satisfies the
quadratic equation
\[ \frac{\theta_1}{2}a^2 + \theta_0a - C_t = 0, \]
where
\[ C_t = \theta_0s_t + \frac{\theta_1}{2}s_t^2 - Q_t, \]
provided \( C_t \geq 0 \). Thus we have
\[ \varphi_t(s_t) = -\theta_0 + \sqrt{\theta_0^2 + 2\theta_1C_t}/\theta_1. \]

Next, for each \( t = 1, \ldots, T \), there are two critical numbers \( \hat{a}_{1t} \) and \( \hat{a}_{2t} \), given by
\[ \hat{a}_{it} = \arg \max_{0 \leq a \leq U} -h_{it}\Theta(a) + \beta W_t(a), \]
for \( i = 1, 2 \), such that \( \hat{a}_{1t} \leq \hat{a}_{2t} \). Also, let
\[ \hat{s}_{it} = \Theta^{-1}(\Theta(\hat{a}_{it}) + Q_t). \]

Then the optimal decision rules are given in Table 1, where the state space in period \( t \) is partitioned into four zones and a simple interpretation is given. We note, in particular, that the optimization of eq. (6) for all states \( s_t \in [0, U] \) can be performed simply by doing a pair of line searches to find \( \hat{a}_{1t} \) and \( \hat{a}_{2t} \). The resulting shape of the decision rules is illustrated graphically in Figure 3 below.

<table>
<thead>
<tr>
<th>Zone</th>
<th>State ( (s_t) )</th>
<th>Action ( (\pi_t(s_t)) )</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([0, \hat{a}_{1t}])</td>
<td>( s_t )</td>
<td>no energy production</td>
</tr>
<tr>
<td>2</td>
<td>([\hat{a}<em>{1t}, \hat{s}</em>{1t}])</td>
<td>( \hat{a}_{1t} )</td>
<td>discharge to ( \hat{a}_{1t} ) ( (E_t &lt; Q_t) )</td>
</tr>
<tr>
<td>3</td>
<td>([\hat{s}<em>{1t}, \hat{s}</em>{2t}])</td>
<td>( \varphi_t(s_t) )</td>
<td>produce energy ( E_t = Q_t )</td>
</tr>
<tr>
<td>4</td>
<td>([\hat{s}_{2t}, U])</td>
<td>( \hat{a}_{2t} )</td>
<td>discharge to ( \hat{a}_{2t} ) ( (E_t &gt; Q_t) )</td>
</tr>
</tbody>
</table>

The special structure of optimal decision rules was exploited in Lamond [1] to obtain a piecewise polynomial approximation of the value functions \( \mathcal{V}_t(s_t) \) which, in turn, was exploited to evaluate the functions \( W_t(a_t) \), the expectation in eq. (5) being obtained by solving analytically the corresponding integrals. Here, we present another approach based on neural networks.
4 Solution by neuro-dynamic programming

Our NDP approach is also based on backward induction with eq. (5) and (6), but at every step \( t \), instead of approximating \( V_t(s_t) \) by a piecewise polynomial function, we approximate \( W_t(a_t) \) by a weighted sum of sigmoidal functions plus a constant. This can be represented schematically as a feedforward neural network with an input layer, a hidden layer and a single output layer, as in Figure 2.

In neural network terminology, a formal neuron simulates the behaviour of a biological neuron whose dendrites collect the energy from its input signals and whose axon transmits a signal to other neurons. In the formal neuron, the energy from the dendrites is represented by a weighted sum of the input variables, and the axon transmission is represented by applying a transfer function to the weighted sum of inputs. In our model, the input layer comprises the two input signals, which are the action variable \( a_t \) and the constant 1 (usually called bias). The hidden layer consists of three formal neurons, labelled \( k = 1, 2, 3 \). Each neuron first combines the two input signals through its “dendrites”, giving an “energy” \( \delta_k = b_k + r_k a \), and then applies a transfer function (the sigmoidal function, in this case) to transmit the signal

\[
f(\delta_k) = \frac{1}{1 + e^{-\delta_k}}
\]

on to the next layer. The output layer comprises a single neuron whose “dendrites” combine the three signals from the neurons of the hidden layer plus a bias term,
and whose “axon” transmits the output signal

\[ W = b' + \sum_{k=1}^{3} r'_k f(\delta_k). \]

The ten biases and weights \( b', b_k, r_k \) and \( r'_k \), for \( k = 1, 2, 3 \), are model parameters whose determination is discussed below. Let \( b \) and \( r \) denote respectively the bias and weight vectors. Then, for all \( a_t \in [0, U] \), our neural network approximates \( W_t(a_t) \) by the function

\[ \tilde{W}_t(a_t; b, r) = b' + \sum_{k=1}^{3} \frac{r'_k}{1 + e^{-(b_k + r_k a_t)}}. \] (11)

In neural network modelling, the weights are determined in a training step. In our NDP approach, we perform a training step at every period \( t = T, \ldots, 1 \) in the backward induction of DP. Thus the training step for period \( t \) must determine the ten parameters of the function \( \tilde{W}_t(a_t; b, r) \). To do this, we chose a finite subset \( S^m = \{0, m, 2m, \ldots, U\} \) of the continuous state space \([0, U]\), such that \( U \) is an integer multiple of \( m \), and we approximate the continuous distribution of the random variable \( D_t \) of natural inflows by a discrete distribution with probability mass function \( \alpha_{im} \), for \( i = 0, 1, 2, \ldots, U/m \). We then proceed by backward induction.

In period \( t \), we first approximate the value function \( V_{t+1}(s) \) at every point \( s \in S^m \) by some numbers \( V^*_t(s) \). For \( t = T \), we take the terminal values which are assumed given in the model data:

\[ V^*_T(s) = V_T(s). \]

For \( t = T - 1, \ldots, 1 \), we exploit the special structure of the optimal decision rules as follows. We compute the (approximate) critical points \( \tilde{a}_{1,t+1} \) and \( \tilde{a}_{2,t+1} \) using eq. (9) with \( t \) replaced by \( t + 1 \) and with \( W \) replaced by \( \tilde{W} \). Taking the decision rule \( \pi_{t+1}(s) \) of Table 1 with \( t \) replaced by \( t + 1 \), we obtain \( V^*_{t+1}(s) \) from eq. (7) for all \( s \in S^m \). Then an approximation \( \tilde{W}_t^*(a_t) \) of \( W_t(a_t) \) is obtained by evaluating the expectation in eq. (5), for every \( a_t \in S^m \), using the discrete inflow distribution on \( S^m \) and replacing \( V \) by \( V^* \). We get

\[ W_t^*(a_t) = \sum_{i=0}^{u^m(a_t)} \alpha_{im} V_{t+1}^*(a_t + im) + \gamma^m(a_t) V_{t+1}^*(U), \]

where \( u^m(a_t) = (U - a_t - m)/m \) and \( \gamma^m(a_t) = 1 - \sum_{i=0}^{u^m(a_t)} \alpha_{im} \). Finally, the model parameters \( b \) and \( r \) can now be estimated by the method of least squares:

\[ \min_{b, r} \sum_{s \in S^m} \left( W_t^*(a) - \tilde{W}_t(a; b, r) \right)^2. \]
5 Numerical Results

To investigate the quality of the approximation obtained with the neural network, we solved a numerical example both by discrete dynamic programming (DP) on a discretized state space $S^0 = \{0, 1, 2, \ldots, U\}$, with $U = 200$ water units, and with our neuro-dynamic programming (NDP) approach using a discrete set $S^m$ with $m = 10$. A planning horizon of $T = 12$ periods is considered, and we assume all model data to be same in all periods $t = 1, \ldots, T$, with a discount factor of $\beta = 0.95$. The final reward is $V_{13}(s) = 0$ for all $s \in [0, U]$. The reward function has a unit price of $h_{1t} = $1.00 up to $Q_t = 100$ energy units, and $h_{2t} = $0.30 per extra energy unit. The turbine efficiency (or head) function is

$$\vartheta(s) = 1 + 0.005s,$$

so the turbine is twice as efficient with a full reservoir ($\vartheta(200) = 2$) than with an empty reservoir ($\vartheta(0) = 1$). The random variables $D_t$ of natural inflows follow a normal distribution with a mean of 60 water units and a standard deviation of 20 water units.

Our model was implemented in Matlab using the Neural Network Toolbox. More specifically, the training step was performed using the Levenberg-Marquardt
method with Bayesian regularization (to avoid overfitting), as prescribed in the user guide by Demuth and Beale [6].

Table 2: Comparison of the critical points obtained by DP and NDP

<table>
<thead>
<tr>
<th>Period</th>
<th>DP</th>
<th>NDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89</td>
<td>135</td>
</tr>
<tr>
<td>2</td>
<td>88</td>
<td>135</td>
</tr>
<tr>
<td>3</td>
<td>87</td>
<td>134</td>
</tr>
<tr>
<td>4</td>
<td>85</td>
<td>133</td>
</tr>
<tr>
<td>5</td>
<td>82</td>
<td>132</td>
</tr>
<tr>
<td>6</td>
<td>77</td>
<td>130</td>
</tr>
<tr>
<td>7</td>
<td>69</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>59</td>
<td>125</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>122</td>
</tr>
<tr>
<td>10</td>
<td>28</td>
<td>118</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>115</td>
</tr>
</tbody>
</table>

The optimal policies obtained by DP and NDP are plotted in Figure 3 for some of the periods, and the critical points are compared in Table 2. Table 3 gives the mean relative error between the optimal actions obtained by DP \(\pi_t(s_t)\) and NDP \(\tilde{\pi}_t(s_t)\), which is defined as

\[
\frac{1}{U + 1} \sum_{s_t=0}^{U} \frac{|\pi_t(s_t) - \tilde{\pi}_t(s_t)|}{\pi_t(s_t)}.
\]

The error decreases with \(t\) so the further away one is from the end of the planning horizon, the better is the approximation.

Finally, Figure 4 compares graphically the function \(\tilde{W}_t(a_t)\) obtained by NDP with the function \(W_t(a_t)\) obtained by DP. For all practical purposes, these curves are superimposed. In fact, the mean relative error between \(W_t(a_t)\) and \(\tilde{W}_t(a_t)\) is less than 0.1% in our example.

In conclusion, our numerical results indicate a rather good quality of solutions. Because this NDP approach exploits the special structure of optimal solutions, it

Table 3: Relative error of optimal policies

<table>
<thead>
<tr>
<th>Period</th>
<th>Error (%)</th>
<th>Period</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t = 2)</td>
<td>0.30</td>
<td>(t = 8)</td>
<td>0.83</td>
</tr>
<tr>
<td>(t = 5)</td>
<td>0.37</td>
<td>(t = 11)</td>
<td>3.12</td>
</tr>
</tbody>
</table>
is computationally economical. Consequently, we feel the NDP approach offers some potential for addressing the (more difficult) case of multireservoir systems.

References