Chapter 5
On Cutting Plane Algorithms and Dynamic Programming for Hydroelectricity Generation

Andy Philpott, Anes Dallagi, and Emmanuel Gallet

Abstract We consider *dynamic programming* (DP) approximations to hydroelectric reservoir scheduling problems. The first class of approximate DP methods uses decomposition and multi-modeling heuristics to produce policies that can be expressed as the sum of one-dimensional *Bellman functions*. This heuristic allows us to take into account non-convexities (appearing in models with head effect) by solving a MIP at each time stage. The second class of methods uses cutting planes and sampling. It is able to provide multidimensional policies. We show that the cutting plane methods will produce better policies than the first DP approximation on two convex problem formulations of different types. Modifying the cutting plane method to approximate the effect of reservoir head level on generation also yields better results on problems including these effects. The results are illustrated using tests on two river valley systems.

5.1 Introduction

The mid-term hydrothermal scheduling problem involves determining a policy of releasing water from reservoirs for hydroelectricity generation and generating from thermal plant over some planning horizon of months or years so as to meet the future demand for electricity at the lowest expected fuel cost. The first models (dating back to [8, 9]) for these problems used dynamic programming, a tool that was confined to systems with one or two reservoirs, unless reservoir aggregation heuristics (see, e.g., [2, 13]) are used.

A. Philpott (✉)
University of Auckland, Private Bag 92019, Auckland, New Zealand
e-mail: a.philpott@auckland.ac.nz

A. Dallagi • E. Gallet
EDF R&D, 1 avenue du Général de Gaulle, 92141 Clamart Cedex, France
e-mail: anes.dallagi@edf.fr; emmanuel.gallet@edf.fr

R.M. Kovacevic et al. (eds.), *Handbook of Risk Management in Energy Production and Trading*, International Series in Operations Research & Management Science 199,
DOI 10.1007/978-1-4614-9035-7__5,

An effort to model systems with multiple reservoirs led to the development in the 1980s and 1990s of various multi-stage stochastic linear programming models (see, e.g., [7]) using scenario trees. Stochastic dual dynamic programming (SDDP) [10] was developed as a response to the problem of dealing with a rapidly growing scenario tree. This method approximates the future cost function of dynamic programming using a piecewise linear outer approximation, defined by cutting planes or *cuts* computed by solving linear programs. This avoids the curse of dimensionality that arises from discretizing the state variables. The intractability arising from a branching scenario tree is avoided by essentially assuming stagewise independent uncertainty. This allows cuts to be shared between different states, effectively collapsing the scenario tree.

There has been little published work comparing the SDDP methodology with classical dynamic programming. A relatively old paper by Archibald et al. [1] shows that nested Benders decomposition outperforms classical dynamic programming in some computational tests on models with a small number of stages and scenarios, but becomes intractable as these grow. Our contribution is to demonstrate some advantages of SDDP-type algorithms in comparison with dynamic programming when the problem has many stages, so that nested Benders decomposition is computationally intractable, at least in its standard scenario-tree form.

In some electricity systems hydrothermal scheduling problems are solved using price decomposition (see, e.g., [6]). In a deterministic setting this method gives a subproblem to be solved for each thermal unit and each hydro river-chain. These subproblems can in principle be coordinated by price to yield an overall generation plan that meets demand in every period at minimum total cost. The coordination problem has been less well studied under inflow uncertainty (although see, e.g., [3]). Here subproblems must be coordinated by a random price process to yield an overall generation plan that meets the demand at minimum expected cost. A similar set of (sub)problems arises when river-chains are operated by different agents in a competitive electricity pool market, where the prices over time come from the pool. In both these settings the river-chain optimization subproblem is a challenging problem to solve, since it must handle uncertainties in prices and inflows. Indeed it is a variant of the hydrothermal scheduling problem in which the electricity price is modeled as the (random) marginal cost of an infinitely large thermal unit.

In this paper we compare two approaches to solving this problem and analyze their differences. The first approach uses a stochastic dynamic programming heuristic applied to a low-dimensional approximation of the state space. We discuss the form of the approximation and demonstrate biases in the marginal values of water. The second method is the dynamic outer approximation sampling algorithm (DOASA) described in [11], which is a special version of SDDP. We describe how the DOASA algorithm overcomes some of the biases of the former. The optimal value functions derived from the two methods are tested numerically on two river-chains.

The paper is laid out as follows. In the next section we describe the stochastic control problem that we wish to solve. This problem is illustrated in the following section by two example river-chains (RC1 and RC2) operated by EDF in France. The networks describing these river-chains have different topologies and so one can use different heuristics to compute approximately optimal release policies for the reservoir control problem, with a view to computing estimates of water values for short-term optimization.[1] The heuristics (called MORGANE) are described in Sect. 5.3. In Sect. 5.4 we give a brief description of the DOASA algorithm and then describe how the forward simulation step of this method can be used to evaluate both the DOASA release policy and the corresponding release policies that arise by applying MORGANE. In Sect. 5.5 we present the results of applying the MORGANE heuristics to the example systems and compare this with the results of applying DOASA.

5.2 The Hydroelectric River-Chain Model

We consider a river-chain represented by a network of n nodes (reservoirs and junctions) and m arcs (canals or river reaches). The topology of the network can be represented by the $n \times m$ incidence matrix A, where

$$a_{ij} = \begin{cases} 1, & \text{if node } i \text{ is the tail of arc } j, \\ -1, & \text{if node } i \text{ is the head of arc } j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

By adding dummy nodes if necessary, we can ensure that every pair of nodes is joined by at most one arc. Let $x(t)$ denote a vector of reservoir storages in each node at the beginning of week t and $\omega(t)$ a vector of uncontrolled reservoir inflows (in cubic meters) that have occurred in week t. We let $h(t)$ be a vector of flow rates (cubic meters per hour) in the arcs in the network and $p(t)$ a vector of electricity prices at time t. Here we adopt the convention that these prices are applied to each flow in the network and are adjusted to account for conversion factors. Thus if arc j does not represent a generating station then $p_j(t) \equiv 0$, and if j is a station then $p_j(t)$ is the spot price of electricity multiplied by a scale factor η_j for that station (converting cubic meters of water passing though the station into MWh). Some flows represent spill (with $\eta_j = 0$) from reservoirs to the river reach below a station.

We also need to allow for more than one price period within a week. In our experiments we use $B = 21$ blocks in a week, each of duration $(d_b)_{b=1,\dots,B}$ hours. These account for variations in price between peak and off-peak times. To model this we assume that $p(t)$ and $h(t)$ are of dimension mB, where the first B components of each vector correspond to arc 1, and so on. We then define the $m \times mB$ matrix

$$D = \begin{bmatrix} d_1\ d_2\ \dots\ d_B & 0\ \ 0\ \ \dots\ 0 & \dots\dots & 0\ \ 0\ \ \dots\ 0 \\ 0\ \ 0\ \ \dots\ 0 & d_1\ d_2\ \dots\ d_B & \dots\dots & \vdots\ \ \vdots\ \ \vdots\ \ \vdots \\ \vdots & & \dots\dots & 0\ \ 0\ \ \dots\ 0 \\ 0\ \ 0\ \ \dots\ 0 & & & d_1\ d_2\ \dots\ d_B \end{bmatrix} \tag{5.2}$$

[1] A deterministic optimization is operated on the short-term (within a day) using a more accurate model in order to provide the actual feasible releases to be performed.

so that the total quantity of flow through arc j in week t is $Dh(t)$, and the revenue earned is $p(t)^\top h(t)$, where component $B(j-1)+k$ of $p(t)$ now equals the electricity price/MWh in block k in week t multiplied by both d_b and η_b.

The hydroelectric river-chain problem we wish to solve seeks to construct a policy for generating electricity from the river-chain so as to maximize the expected revenue. We assume in this paper that the uncontrolled reservoir inflows are the only uncertain parameters and that these are stagewise independent. In each realization of uncertainty, the policy we seek will give a set of reservoir releases defined by $h(t)$ that generate electricity. The policy is defined in terms of a dynamic programming Bellman function $\mathbb{E}[V_t(x,\omega(t))]$ where $V_t(x,\omega(t))$ gives the maximum expected revenue that can be earned in weeks $t, t+1, \ldots$ when reservoir storage $x(t) = x$ and week t's inflow is known to be $\omega(t)$. Here

$$\begin{aligned} V_t(x,\omega(t)) = \max\ & p(t)^\top h(t) + \mathbb{E}[V_{t+1}(x(t+1),\omega(t+1))] \\ \text{s.t.}\quad & x(t+1) = x - ADh(t) + \omega(t), \\ & 0 \le h(t) \le b, \quad 0 \le x(t+1) \le r. \end{aligned} \tag{5.3}$$

We place a limit on the time horizon of $t = T$ and specify a future value function $V_{T+1}(x(T+1),\omega(T+1))$ to give a bounded problem.

5.3 River-Chain Optimization and Multi-modeling

In this section we present the MORGANE dynamic programming heuristics as applied to two river-chains. These heuristics are applied in order to keep some advantages of the stochastic dynamic programming method. In fact, they allow us to model non-convexities in the problem without, as far as the stage problems are solved, losing the theoretical validity of the method. Nevertheless dynamic programming is faced with the curse of dimensionality, and so the heuristics seek to avoid this by reducing the dynamic programming calculations to a sequence of one-dimensional problems.

The river-chains are called the RC1 Valley and the RC2 river-chain. Due to confidentiality issues the names of the river-chains and reservoirs are anonymized. Schematic representations of these are shown in Figs. 5.1 and 5.2. The RC1 system has three main reservoirs feeding a chain of two essentially run-of-the-river generating stations. The RC2 system is a chain of stations some of which have headpond reservoirs with significant storage capacity.

The MORGANE model treats these river-chains slightly differently. For the RC1 system, MORGANE computes marginal water values for each of the three main reservoirs by performing a stochastic dynamic programming algorithm for each of these reservoirs considering that the storage levels of the other reservoirs are constrained to be at given levels at the end of each week. In the RC2 system, MORGANE applies a different heuristic that consists of decomposing the river-chains into two parts: an upstream one and a downstream one. Applying the first heuristic

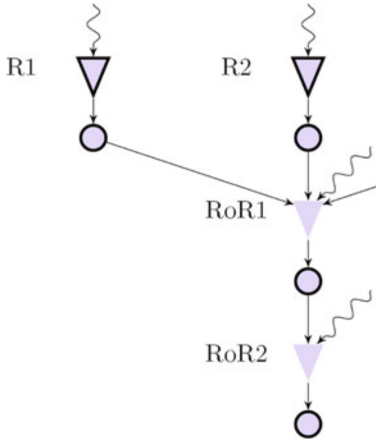

Fig. 5.1 The RC1 river system

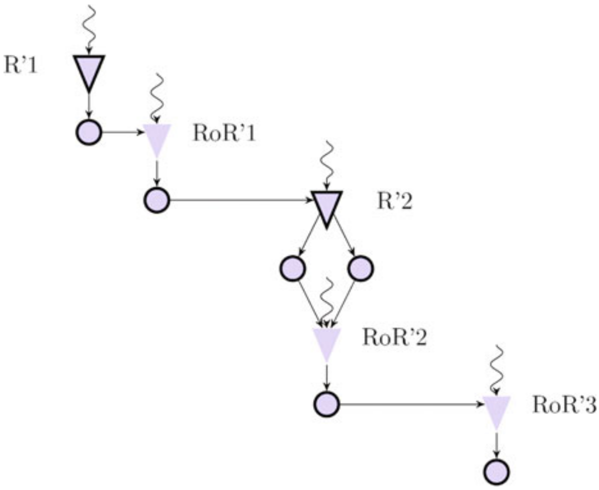

Fig. 5.2 The RC2 river system

to the upstream part generates a controlled flow feeding the downstream part. Again the first heuristic is applied to the downstream section in order to compute water values for the reservoir at the top of this section.

5.3.1 Heuristic 1: Constrained Multi-modeling

The curse of dimensionality means that computing multidimensional policies using dynamic programming is numerically intractable. The idea of heuristic 1 is to compute a multidimensional policy as the sum of one-dimensional ones. In this approach, each reservoir present in the considered river-chain is operated as if the other reservoirs have storage level targets to reach at the start and end of each stage. This target is an a priori fixed proportion of the maximum storage capacity.

Solving a hydroelectric river-chain model, composed of n interconnected reservoirs, thus amounts to solving n subproblems. Each subproblem is solved using dynamic programming, resulting in n revenue-to-go functions.

We denote by $V_t^{[i]}(x_i, \omega(t))$ the maximum expected revenue that can be earned in weeks $t, t+1, \ldots, T$, when reservoir i storage is $x_i(t) = x_i$ and reservoir $j \neq i$ storage is $x_j(t) = x_j(t+1) = \alpha_j r_j, \quad \alpha_j \in [0,1]$ at the start and end of each time stage. Week t's inflow is known to be $\omega(t)$. Here

$$
\begin{aligned}
V_t^{[i]}(x_i, \omega(t)) = \max\ & p(t)^\top h(t) + \mathbb{E}[V_{t+1}^{[i]}(x_i(t+1), \omega(t+1))] \\
\text{s.t.}\ \ & x_i(t) = x_i, \\
& x(t+1) = x(t) - ADh(t) + \omega(t), \\
& x_j(t) = \alpha_j r_j, \quad \forall j \neq i \\
& x_j(t+1) = \alpha_j r_j, \quad \forall j \neq i \\
& 0 \leq h(t) \leq b, \quad 0 \leq x_i(t+1) \leq r_i.
\end{aligned}
\tag{5.4}
$$

The multidimensional Bellman function that will allow us to recompute an overall policy will be the sum of one-dimensional Bellman functions:

$$
V_t(x(t), \omega(t)) = \sum_{i=1}^{n} V_t^{[i]}(x_i(t), \omega(t)). \tag{5.5}
$$

It is obvious that this heuristic is suboptimal in the sense that the subproblems have less flexibility than the original problem formulation. Nevertheless, this heuristic allows us to maintain the advantages of classical dynamic programming. In particular, no convexity assumptions are needed in order to perform this heuristic as long as we can solve the (MIP) subproblems with a reasonable amount of computational effort.

5.3.2 Heuristic 2: Geographical Decomposition

The previous heuristic is adapted to parallel reservoirs feeding into the same turbine or run-of-the-river installations. In fact in that kind of configuration the strategies of the different reservoirs defined by the heuristic are optimal if the downstream constraints are never binding. We discuss this issue in more depth in the Appendix.

For a cascade-type river-chain this decoupling is typically not possible, and a geographical decomposition can be applied between downstream and upstream using the special structure of the river network.

For the RC2 river-chain the matrices describing the network can be decomposed as an upstream part containing $p = 2$ reservoirs and a downstream part containing $n - p = 3$ reservoirs, so

$$A = \begin{bmatrix} A^{[1]} & B^{[1]} \\ B^{[2]} & A^{[2]} \end{bmatrix} = \left[\begin{array}{ccc|ccc} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right] \tag{5.6}$$

and

$$D = \begin{bmatrix} D^{[1]} & D^{[2]} \end{bmatrix}. \tag{5.7}$$

Heuristic 2 looks for strategies depending only on two reservoir levels: the topmost one (reservoir 1) and the most valuable[2] one in the middle of the chain (reservoir $p+1$).

First, we use heuristic 1 in order to compute a policy for the topmost reservoir while fixing the releases of the other reservoirs in order to meet an a priori fixed target at the start and end of each time stage. We denote by $V_t^{[1]}(x_1, \omega(t))$ the maximum expected revenue that can be earned in weeks $t, t+1, \ldots, T$, when the topmost reservoir storage is $x_1(t) = x_1$ and reservoir $j \neq 1$ storage is $x_j(t) = x_j(t+1) = \alpha_j r_j, \quad \alpha_j \in [0,1]$. Week t's inflow is known to be $\omega(t)$. Here

$$\begin{aligned} V_t^{[1]}(x_1, \omega(t)) = \max\ & p(t)^\top h(t) + \mathbb{E}[V_{t+1}^{[1]}(x_1(t+1), \omega(t+1))] \\ \text{s.t.}\ & x_1(t) = x_1, \\ & x(t+1) = x(t) - ADh(t) + \omega(t), \\ & x_j(t) = \alpha_j r_j, \quad \forall j \neq 1 \\ & x_j(t+1) = \alpha_j r_j, \quad \forall j \neq 1 \\ & 0 \leq h(t) \leq b, \quad 0 \leq x_1(t+1) \leq r_1. \end{aligned} \tag{5.8}$$

Second, we use the previously computed releases $h^{[1]}(t) \in \mathbb{R}^p$ feeding the downstream part, in order to optimize the second part of the river-chain. Hence, we denote by $V_t^{[2]}(x_{p+1}, \omega(t))$ the maximum expected revenue that can be earned in weeks $t, t+1, \ldots, T$, when the storage of the topmost reservoir (in the downstream section) is $x_{p+1}(t) = x_{p+1}$ and reservoir $j > p$ storage is $x_j(t) = x_j(t+1) = \alpha_j r_j, \quad \alpha_j \in [0,1]$. Week t's inflow in the downstream part of the river is denoted by $\omega^{[2]}(t)$. Here

[2] It could be the biggest one or the most valuable one from the operator's point of view.

$$
\begin{aligned}
V_t^{[2]}(x_{p+1}, \omega(t)) = \max\ & p^{[2]}(t)^\top h^{[2]}(t) + \mathbb{E}[V_{t+1}^{[2]}(x_{p+1}(t+1), \omega^{[2]}(t+1))] \\
\text{s.t.}\ \ & x_{p+1}(t) = x_{p+1}, \\
& x^{[2]}(t+1) = x^{[2]}(t) - A^{[2]}D^{[2]}h^{[2]}(t) \\
& \qquad\qquad -B^{[2]}D^{[1]}h^{[1]}(t) + \omega^{[2]}(t), \\
& x_j(t) = \alpha_j r_j, \quad \forall j > p \\
& x_j(t+1) = \alpha_j r_j, \quad \forall j > p \\
& 0 \leq h^{[2]}(t) \leq b, \quad 0 \leq x_{p+1}(t) \leq r.
\end{aligned}
\tag{5.9}
$$

The Bellman function that defines our policy will then be the sum of the one-dimensional Bellman functions:

$$
V_t(x(t), \omega(t)) = V_t^{[1]}(x_1(t), \omega(t)) + V_t^{[2]}(x_{p+1}(t), \omega(t)). \tag{5.10}
$$

While simulating the generated policy for the remaining reservoirs, we add to the simulator the target constraints at the beginning and end of each stage.

As in the previous case, this heuristic is suboptimal in the sense that the subproblems have less flexibility than the original problem. But, as in the previous heuristic, we are able to keep some of the dynamic programming advantages (dealing with non-convexities).

5.4 Dynamic Outer Approximation Sampling Algorithm

The DOASA code [11] is based on the SDDP technique of Pereira and Pinto [10]. To solve a maximization problem, DOASA approximates $\mathbb{E}[V_t(x, \omega(t))]$ using a piecewise linear outer approximation that is updated using samples of the inflow process. Of course, this approximation is available only if convexity assumptions are made on the considered problem. Weekly prices are represented by a price duration curve with 21 blocks, and $\omega(t)$ is sampled from historical inflow observations.

The DOASA code yields an outer approximation to $\mathbb{E}[V_t(x, \omega(t))]$ at each stage t by solving the single-stage approximating problem:

$$
\begin{aligned}
SP(x, \omega(t))\text{: } \max\ & p(t)^\top h(t) + \theta_{t+1} \\
\text{s.t.}\ \ & x(t+1) = x - ADh(t) + \omega(t), \qquad [\pi_t] \\
& 0 \leq h(t) \leq b, \quad 0 \leq x(t+1) \leq r, \\
& \alpha_{t+1}^k + \left(\beta_{t+1}^k\right)^\top x(t+1) \geq \theta_{t+1}, k \in \mathscr{C}(t+1).
\end{aligned}
\tag{5.11}
$$

The flow balance constraints at each node i have dual variables π_t. These are used to compute the inequality constraints

$$\alpha_t^k + \left(\beta_t^k\right)^\top x(t) \geq \theta_t, \quad k \in \mathscr{C}(t) \tag{5.12}$$

which are the cuts defining an outer approximation of the future value function. For simplicity of notation, we have represented this construction as applying to all nodes i, but in practice we compute cuts only for the reservoirs with positive storage capacity.

At iteration k of the algorithm the cut coefficients are computed as follows:

1. Solve the stage problem $SP(x^k(1), \omega(1))$ for the known realization $\omega(1)$, giving optimal solution $x^k(2)$ and optimal value $V_1(x^k(1), \omega(1))$.
2. For $t = 2, 3, \ldots, T$, solve the stage problem $SP(x^k(t), \omega(t))$ for a sample realization of ω, recording the solution $x^k(t+1)$.
3. For $t = T, T-1, \ldots, 2$, solve the stage problem $SP(x^k(t), \omega(t))$ for every realization of $\omega(t)$, recording the solution value $V_t(x^k(t), \omega(t))$ and duals $\pi_t(x^k(t), \omega(t))$, and adding the cut

$$\theta_t \leq \alpha_t + \beta_t^\top x(t), \tag{5.13}$$

to every problem at stage $t-1$, where

$$\beta_t = \mathbb{E}[\pi_t(x^k(t), \omega(t))], \tag{5.14}$$

and

$$\alpha_t = \mathbb{E}[V_t(x^k(t), \omega(t))] - \beta_t^\top (x^k(t)). \tag{5.15}$$

The algorithm terminates after little progress is observed in the upper bound estimate $V_1(x^k(1), \omega(1))$. The policy defined by the cuts can then be simulated, and its expected revenue estimated by a sample average. This gives an estimate of a lower bound on the optimal value that can be checked against the upper bound.

In contrast to SDDP that runs many forward passes simultaneously, DOASA performs a single forward pass in each iteration. This can be shown to be more effective at delivering good solutions when stopping the algorithm early [4].

5.5 Experiments

DOASA and MORGANE were applied to data from the two valley systems RC1 and RC2. In both systems, MORGANE was applied subject to the following restrictions[3]:

1. There are no constraints on reservoir levels apart from them ranging between 0 and their full capacity.

[3] These restrictions are not in favor of the MORGANE heuristics. In fact the heuristics are able to take into account stochasticity and time dependency on the prices, end non-convexity constraints.

2. All generating plant are assumed to be available at full capacity throughout the year. Real problems include randomness on the availability of these plants.
3. Electricity prices were assumed to be known in advance, and set to their average over all the scenarios that had been generated. Each river-chain used a different set of average prices. Prices were for 21 blocks of time during the week representing peak, off-peak, and shoulder periods. The number of hours in each block varies with the day of week and the river valley.
4. A set of 41 weekly inflow sequences was used to represent the reservoir inflow distribution, assumed to be stagewise independent.
5. The factors for converting flow to electricity in each station were set to be constant (i.e., we assume that they do not vary with reservoir head level) in a first experiment and vary in a second one.

Under these restrictions MORGANE was solved over a 104-week horizon and marginal water values were computed for each reservoir at the end of each week. The marginal water values for week 52 were then converted into cuts and provided to DOASA as end conditions. DOASA was then applied over the first 52 weeks of the planning horizon.

5.5.1 Converting Marginal Water Values to Cuts

Upon terminating, MORGANE provides marginal water values for each reservoir as a function of their level. This function for reservoir $i = 1,\dots,n$ is represented as a list of reservoir levels and marginal water values denoted (x_{ik}, β_{ik}), $k = 0,1,\dots,K$. This can be converted into cuts by choosing an arbitrary future value α_{i0} from an empty reservoir at time t and then computing α_{ik} recursively using

$$\alpha_{ik} = \alpha_{ik-1} + (\beta_{ik-1} - \beta_{ik})x_{ik}, \quad k = 1,2,\dots,K, \tag{5.16}$$

as shown in Fig. 5.3.

The future value function $\mathbb{E}[V_{T+1}(x(t+1),\omega(t+1))]$ is then represented as the sum of the individual reservoir value functions. Thus

$$\mathbb{E}[V_{t+1}(x(t+1),\omega(t+1))] = \sum_{i=1}^{n} \max_{k=0,1,\dots,K} \{\alpha_{ik} + \beta_{ik}x_i(t+1)\}. \tag{5.17}$$

We model this in DOASA using a multi-cut representation

$$\theta_{t+1} = \sum_{i=1}^{n} \theta_{t+1}(i) \tag{5.18}$$

$$\alpha_{ik} + \beta_{ik}x_i(t+1) \geq \theta_{t+1}(i), \quad k = 0,1,\dots,K,\ i = 1,2,\dots,n. \tag{5.19}$$

Depending on the choice of α_{i0} this representation gives a different level for $\mathbb{E}[V_{t+1}(x(t+1),\omega(t+1))]$, but accurately reproduces the marginal water values for

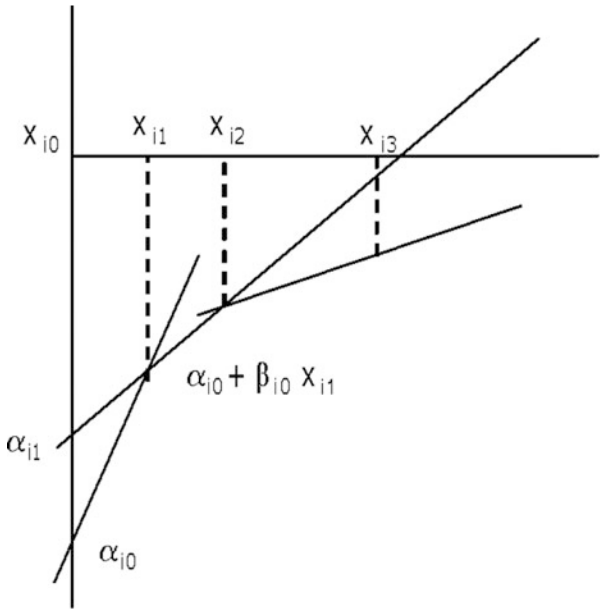

Fig. 5.3 Converting marginal water values to cuts

each storage level at which they are recorded by MORGANE, under the assumption that the MORGANE files of marginal water values give these values at the lowest possible storage level that they apply (i.e., the step function is continuous from the right). We remark that the above construction can be used for representing the MORGANE policy at any stage $t \leq T$. We thus use this both for providing a future cost function to DOASA at $t = 52$ and for simulating the MORGANE policies in our experiments, so that the same code is used to simulate both MORGANE and DOASA policies.

5.5.2 Modeling Variations in Head

Recall that for generating stations $p_j(t)$ denotes the spot price of electricity multiplied by a scale factor η_j for that station (converting cubic meters of water passing through the station into MWh). In practice the scale factor η_j is a function of net head, namely the difference in height of the headwater and tailwater of the turbine generating electricity. A common model (see [12]) assumes that η_j is net head times some efficiency term that depends on flow rate. If we assume that the tailwater height and flow rate is constant then η_j is linear with head level. The headwater height then depends on the volume of water x stored in the reservoir upstream of the station as a concave function since the area of the reservoir increases with head height.

In MORGANE the relationship between conversion factor and head is expressed using a finite set of hydro production functions that depend on reservoir level x. Each production function is modeled using two linear pieces defined by the most efficient flow rate h_e and the maximum flow rate h_m, both of which depend on x. When the reservoir volume is x, the power generated by flow rate h is

$$\begin{aligned} E(h,x) = \max_{h_1,h_2}\ & \eta_e(x)h_1 + \eta_m(x)h_2, \\ \text{s.t. }\ & h_1 + h_2 = h, \\ & h_1 \le h_e(x), \quad h_2 \le h_m(x) - h_e(x). \end{aligned} \tag{5.20}$$

If values of $\eta_e(x)$ and $\eta_m(x)$ are specified for a range of reservoir volumes then we may compute $E(h,x)$ by linearly interpolating these values and solving the maximization problem above.

The term $p(t)^\top h(t)$ in the objective function of our stage problem becomes $p(t)^\top E(t)$, where $p(t)$ and $E(t)$ are of dimension mK and have components that correspond, respectively, to spot price at time t in each time block and network arc and to $E(h,x)$, where h is the station flow in that time block and network arc, and x is the storage in the upstream reservoir at time t.

The DOASA code now yields an outer approximation to $\mathbb{E}[V_t(x,\omega(t))]$ at each stage t by solving the approximating problem:

$$\begin{aligned} SP(x,\omega(t))\colon \max\ & p(t)^\top E(t) + \theta_{,t+1} \\ \text{s.t.}\quad & x(t+1) = x - ADh(t) + \omega(t), && [\pi_t] \\ & 0 \le h(t) \le b, \quad 0 \le x(t+1) \le r, \\ & E(t) = \eta_e(x)h_1 + \eta_m(x)h_2, \\ & h_1 + h_2 = h(t), \\ & h_1 \le h_e(x), && [\rho_t] \\ & h_2 \le h_m(x) - h_e(x), && [\sigma_t] \\ & \alpha_{t+1}^k + \left(\beta_{t+1}^k\right)^\top x(t+1) \ge \theta_{t+1},\ k \in \mathscr{C}(t+1). \end{aligned} \tag{5.21}$$

Even though $SP(x,\omega(t))$ is a convex problem for fixed x, this does not guarantee that the optimal value function $V_t(x,\omega(t))$ is concave as a function of x, even if $\eta_e(x)$ and $\eta_m(x)$ are concave. Indeed it is conceivable that the marginal value of water might increase as reservoirs fill up if the increase in head makes each cubic meter of water more valuable for generation.

A valid outer approximation of $V_t(x,\omega(t))$ by cutting planes requires that it is concave. If we apply the DOASA algorithm to this model then the value of the first-stage problem can no longer be guaranteed to be an upper bound on the value of an optimal policy. Of course it is possible to construct a concave approximation of each production function (see [5]) and use this. However we simply assume that the lack of concavity of these functions is not too severe, so that the cuts computed by DOASA give a reasonably good policy even if the value function is not concave.

To account for head effects in stage 3 of DOASA we now solve $SP(x,\omega(t))$ giving $h_1^k(\omega(t))$ and $h_2^k(\omega(t))$ and duals $\pi_t(x^k(t),\omega(t))$. We can then add the cut

$$\theta_t \leq \alpha_t + \beta_t^\top x(t), \tag{5.22}$$

to every problem at stage $t-1$, where

$$\beta_t = \mathbb{E}[\pi_t(x^k(t),\omega(t)) + p(t)^\top g(x^k(t),\omega(t))], \tag{5.23}$$

and

$$g(x,\omega) = \eta_e'(x)h_1^k(\omega) + \eta_m'(x)h_2^k(\omega). \tag{5.24}$$

This calculation ignores the dependence of h_e and h_m on x which could be included as extra terms in β_t, namely

$$\mathbb{E}[\Big(\rho_t(x^k(t),\omega(t)) - \sigma_t(x^k(t),\omega(t))\Big)^\top h_e'(x^k(t)) + \sigma_t(x^k(t),\omega(t))^\top (h_m'(x^k(t)))], \tag{5.25}$$

where ρ_t and σ_t are the dual variables corresponding to the bounds on h_1 and h_2 as shown in the formulation above.

When head effects are modeled, the forward pass in DOASA must also be changed so as to use the appropriate interpolated production functions corresponding to the state variable that is visited in the forward simulation.

The policies obtained by computing 100 of these cuts in DOASA were simulated against the policies obtained from MORGANE to give the plots shown in Fig. 5.7 for the RC1 and Fig. 5.8 for RC2 systems.

5.6 Numerical Results

We present in this section two different experiments over the two river-chains (RC1 and RC2). In the first experiment we assume that the head is a fixed constant in the production functions. The results concerning this first experiment will allow us to study the proposed MORGANE heuristics versus DOASA. The other experiments are presented to illustrate Sect. 5.5.2 where we take into account head variations.

5.6.1 Experiment 1: Fixed Head

In this experiment DOASA was run for 100 iterations, giving 100 cuts at each stage. Since MORGANE has approximately 40 marginal water values for each of three reservoirs, we claim that this gives a commensurate level of discretization. DOASA takes about one minute per iteration, and so it must be run for nearly two hours on this problem, while MORGANE takes about 10 min. The progress of the DOASA upper bound for a fixed head level model is shown in Fig. 5.4.

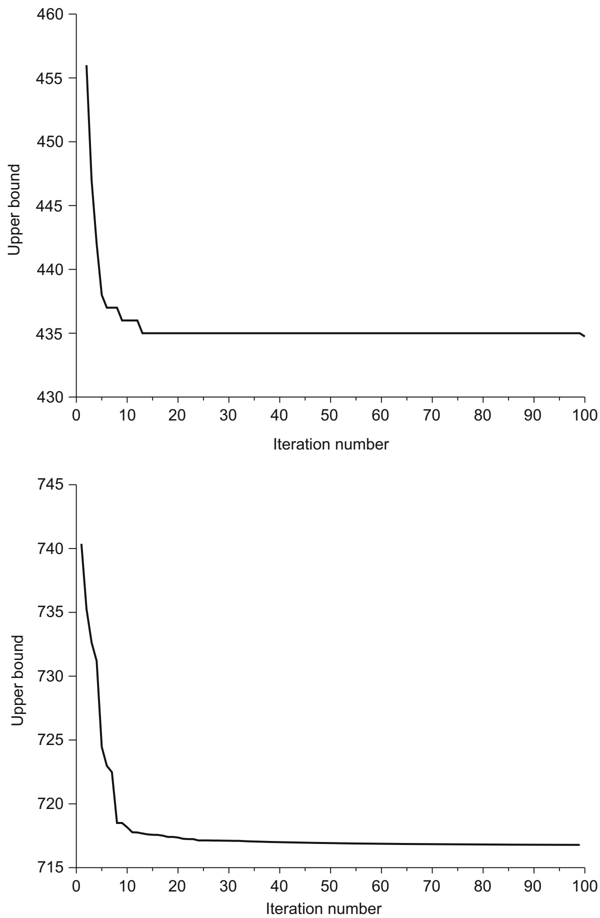

Fig. 5.4 Upper bound for revenue plus true value for RC1 DOASA policy assuming fixed head (RC1 *above*, RC2 *below*)

- For the RC1 system, the minimum (scaled[4]) upper bound is 434.740. The DOASA policy when simulated over 100 out-of-sample stagewise independent inflow sequences gives an expected value of 434.600 with a standard error of 0.139, which indicates that the DOASA policy is close to optimal.
- For the RC2 system,[5] the minimum (scaled) upper bound is 716.785. The DOASA policy when simulated over 100 out-of-sample stagewise independent inflow sequences gives an expected value of 716.454 with a standard error of 0.834, which indicates that the DOASA policy is close to optimal.[6]

The DOASA policy was then simulated over the 41 inflow sequences that were used to construct the policy. Since these sequences display some stagewise dependence, this is a sterner test of DOASA which assumes that inflows are stagewise independent.

- For the RC1 river system, the average optimal value over these scenarios was 434.161 with a standard error of 0.521 for the RC1 river system. This is lower than the out-of-sample expected value of 434.600 but very close. Recall that the upper bound of 434.740 is an upper bound on the value of the DOASA policy that assumes independence, so we cannot deduce that this is a bound on the best policy that took advantage of information about possible persistence in inflows. The policy from MORGANE was also simulated over the same inflow sequences. In 95 % of scenarios the values from DOASA were larger (see Fig. 5.5).
- For the RC2 river system, the average optimal value over these scenarios was 714.068 with a standard error of 3.451. The policy from MORGANE was also simulated over the same inflow sequences. This required some care as MORGANE provides water values only for reservoir R'1 and reservoir R'2. We therefore assumed in computing each week's releases that the terminal values of the reservoir levels of the other reservoirs are set at 50 % of their capacities. The simulation of the MORGANE policy then gives an average optimal value of 712.543 with a standard error of 3.389. In 85 % of scenarios the values from DOASA were larger (see Fig. 5.5).

Some insights into the reason for the difference between the policies can be seen by examining the marginal water values. Since both policies share the same value function at stage T, their stage T optimization problems should deliver the same water value functions at stage $T-1$. The actual values of these are shown for two sets of reservoir storage levels in Tables 5.1 and 5.2.

- For RC1, the marginal values computed by DOASA are lower than those computed by MORGANE. This can be explained by observing that MORGANE

[4] For confidentiality purposes, the units of measurement are omitted.

[5] This has a different network topology, and so it provides a useful comparison of the effect of river-chain topology on the policies from the two methodologies. As before, DOASA was run for 100 iterations, giving 100 cuts at each stage.

[6] An experiment with 200 cuts gave a smaller upper bound of 716.700 and an estimated value of 716.467 with a standard error of 0.833.

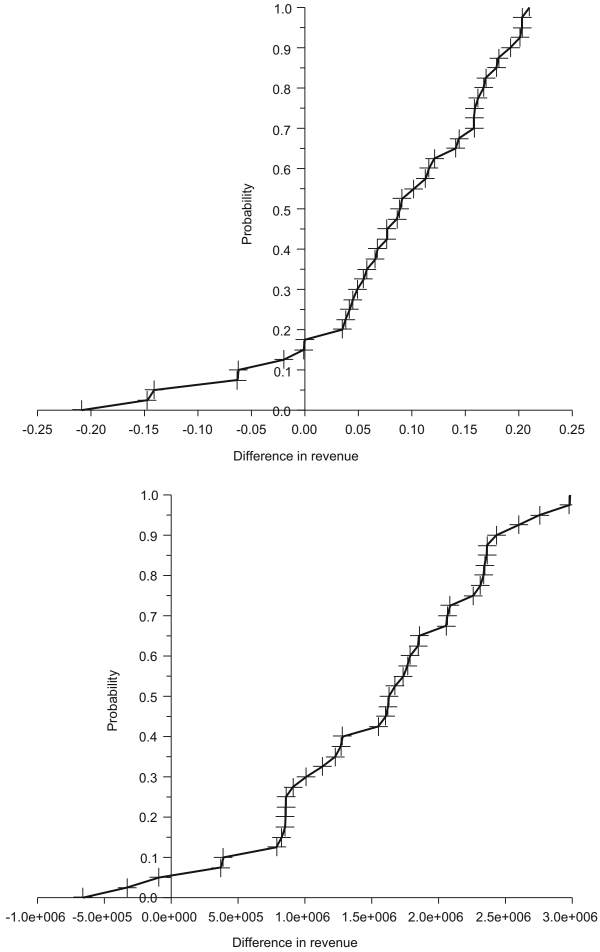

Fig. 5.5 Cumulative plot of the difference in value between the DOASA solution and the MORGANE solution for RC1 (*above*) and RC2 (*below*) over 41 scenarios assuming fixed head

assumes that all other reservoirs terminate at 50 % of capacity when computing marginal values for the reservoir in question. This adds an extra constraint on releases from these reservoirs over the week. Suppose that these constraints result in no other capacity constraints being binding. Then the marginal value of extra water in reservoir R1 (serving the most efficient station) will be the revenue earned by passing this down the river-chain through generating stations. If however the optimal policy means that river reach capacity constraints below the run-of-the-river RoR1 station are binding then the marginal value of extra water in reservoir R1 will be the revenue earned by it minus the loss in revenue from reducing the flow in the other stations that are less efficient, so as to satisfy the constraint. In this way the marginal water values for each reservoir can de-

Table 5.1 Marginal water values (euros per cubic meter) computed from DOASA and MORGANE for end of stage 51 for each of the storage reservoirs assuming fixed head

	Reservoir R1	Reservoir R2	Reservoir R3
Storage (m^3)	13,818,000	3,577,000	14,520,000
DOASA (€/m^3)	0.151	0.176	0.150
MORGANE (€/m^3)	0.159	0.190	0.155
Storage (m^3)	13,818,000	7,114,000	14,520,000
DOASA (€/m^3)	0.154	0.161	0.148
MORGANE (€/m^3)	0.159	0.163	0.155

Table 5.2 Marginal water values (euros per cubic meter) computed from DOASA and MORGANE for end of stage 51 for each of the storage reservoirs assuming fixed head

	Reservoir R'1	Reservoir R'2
Storage (m^3)	203,500,000	79,000,000
DOASA (€/m^3)	0.0359	0.0167
MORGANE (€/m^3)	0.0344	0.0159

pend on the water levels in *other* reservoirs, as well as on the level of their own. In other words, if the other reservoirs are full it is more likely that the capacity constraints will bind and so the water value in reservoir R1 will be lower than the value obtained when the other reservoirs are empty. We examine this nonseparability of the value function in more detail in the Appendix.

- In the RC2 system, the water values for reservoir R'1 under the DOASA policy are higher than those for MORGANE, since if reservoir R'2 is constrained to be at 50 % of its value at the end of the current week, then this precludes transferring water through reservoir R'2 to later weeks. Thus if prices are high now and reservoir R'2 is constrained downstream, then we cannot generate in period 1 without spilling, which could be avoided if water can be transferred by storing it till a later period. We examine the separability of the value function for the cascaded system in the Appendix.

Despite the differences in marginal water values, the policies of DOASA and MORGANE perform similarly. The average storage levels over the 41 scenarios are shown in Fig. 5.6.

5.6.2 Experiment 2: Variation in Head

In this experiment DOASA was run for 100 iterations, giving 100 cuts at each stage on the same reference as the previous section. Head effect was taken into account while optimizing the river-chains and while simulating the obtained policies as

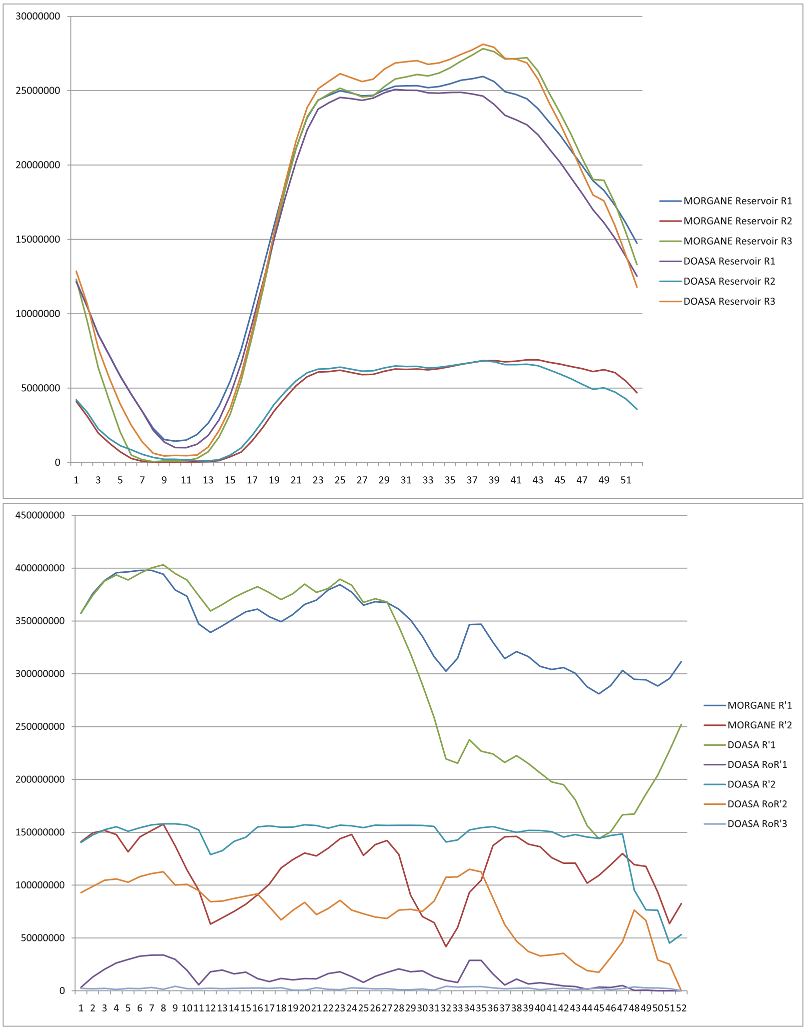

Fig. 5.6 Comparison of average stock levels in each reservoir (over 41 scenarios) for the DOASA and MORGANE policies (RC1 *above* and RC2 *below*) assuming fixed head

described in Sect. 5.5.2. For the RC2, cuts using DOASA are computed assuming that the three smallest reservoirs (i.e., those without storage) can vary their level between bounds. The MORGANE policy assumes that these reservoirs are fixed at their midpoint levels at the end of each week. So for comparison we conducted two computational experiments on the RC2 as follows:

1. We simulate the MORGANE policy, first with free endpoints on the three small reservoirs, and then with fixed weekly endpoints.

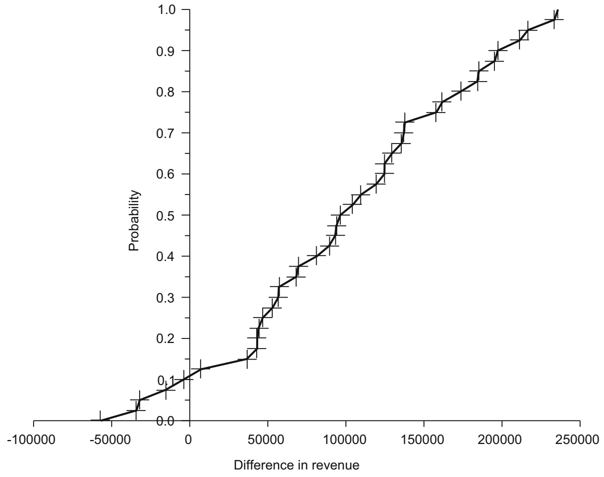

Fig. 5.7 Cumulative plot of the difference in value between the DOASA solution and the MORGANE solution for RC1 with head effect over 41 scenarios

2. We simulate the DOASA policy, first with free endpoints on the three small reservoirs, and then with fixed weekly endpoints.

In these experiments, DOASA behaves better than the MORGANE heuristics, especially if allowed more flexibility in the small reservoirs. However even if this is restricted, the DOASA policy is better than MORGANE showing on average:

- DOASA constrained saves 2M over MORGANE constrained
- DOASA unconstrained saves 3M over MORGANE unconstrained
- DOASA unconstrained saves 3.79M over MORGANE constrained

5.7 Conclusions

The aim of this paper is to compare a stochastic dynamic programming-based heuristic that can handle non-convexities appearing in real problems (MORGANE) to an outer approximation method that needs some convexity assumptions. The set of experiments chosen in this paper demonstrates that constructing DP policies using multivariate Bellman functions gives better results than methods that ignore the cross terms. The source of DOASA's advantage is from using a polyhedral surface that is not a separable sum of one-dimensional curves. When this feature is absent, for example when downstream constraints are never binding, the policies are equivalent. In practice, however there are always periods when these constraints are significant, and these periods cause the policies to diverge. We still need to conduct experiments in order to confirm these results while prices are stochastic and while

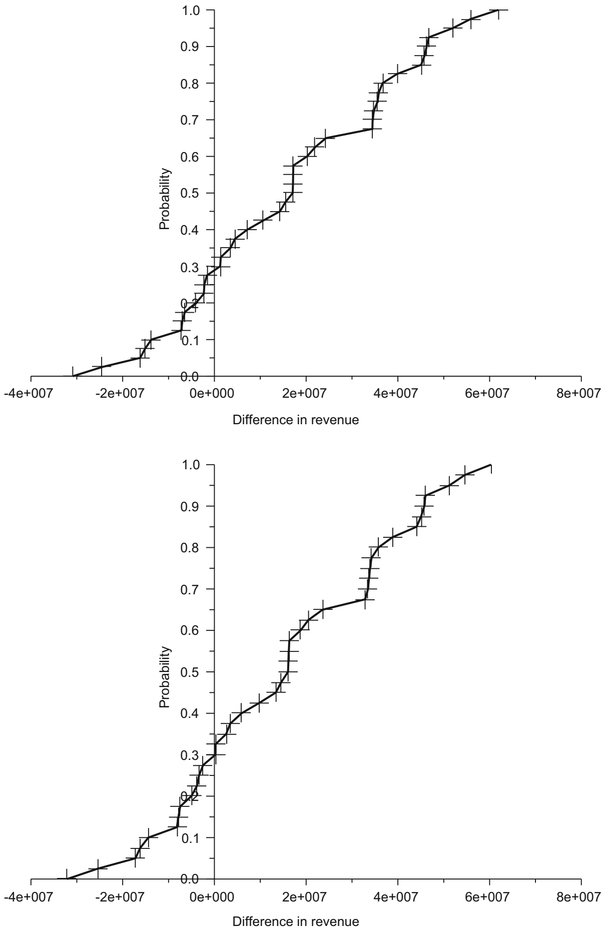

Fig. 5.8 Cumulative plot of the difference in value between the DOASA solution and the MORGANE solution for RC2 with free endpoints (*above*) and constrained endpoints (*below*) over 41 scenarios

the time dependency is taken into account (MORGANE can handle this dependency but not DOASA due to convexity issues). Experiments while we have more complicated constraints (storage constraints, solid water, etc.) are still to be made.

5.8 Appendix: Approximations Made by the Multi-modeling Heuristic

In this appendix we examine the implications of the approximations used by MORGANE for computing marginal water values. The analysis is different for each system, so we discuss each in turn.

5.8.1 RC1 River Chain

The MORGANE approximation of the RC1 system assumes that $V_t(x_1,x_2,x_3)$ is additively separable. To examine the separability of $V_t(x_1,x_2,x_3)$, consider a simplified version of RC1 with two reservoirs having capacity a_1 and a_2 and releases y_1 and y_2 through stations with price per unit of water flow of p_1 and p_2. Like RC1, the tailwater shares a common channel with capacity k. We wish to investigate the form of $V_t(x_1,x_2)$ when

$$V_{t+1}(x_1,x_2) = -(x_1-a_1)^2 - (x_2-a_2)^2 . \tag{5.26}$$

This gives

$$\begin{aligned} V_t(x_1,x_2) = \max\nolimits_{y_1,y_2} \;& p_1y_1 + p_2y_2 - (x_1-y_1-a_1)^2 - (x_2-y_2-a_2)^2 \\ \text{s.t.} \;& y_1 \le x_1 \\ & y_2 \le x_2 \\ & y_1+y_2 \le k \\ & y \ge 0 \end{aligned} \tag{5.27}$$

After some algebra, it can be shown that $V_t(x_1,x_2)$ can be separated into a sum $V_t^1(x_1)+V_t^2(x_2)$ for all values of x_1,x_2 except for those satisfying the following three conditions:

$$\begin{bmatrix} \left(x_2-\left(a_2-\tfrac{1}{2}p_2\right)\right)+\left(x_1-\left(a_1-\tfrac{1}{2}p_1\right)\right) > k \\ \left(x_1-\left(a_1-\tfrac{1}{2}p_1\right)\right), \left(x_2-\left(a_2-\tfrac{1}{2}p_2\right)\right) > 0 \\ \left|\left(x_2-\left(a_2-\tfrac{1}{2}p_2\right)\right)-\left(x_1-\left(a_1-\tfrac{1}{2}p_1\right)\right)\right| < k \end{bmatrix}. \tag{5.28}$$

In this exceptional case,

$$\begin{aligned} V_t(x_1,x_2) = & -\frac{1}{2}(x_1+x_2)^2 \qquad (5.29)\\ & + \left(k+a_1+a_2+\frac{1}{2}p_1-\frac{1}{2}p_2\right)x_1 + \left(k+a_1+a_2-\frac{1}{2}p_1+\frac{1}{2}p_2\right)x_2 \\ & + \left(\begin{array}{l} \tfrac{1}{2}kp_1 - ka_1 - ka_2 - \tfrac{1}{2}k^2 + \tfrac{1}{2}kp_2 - \tfrac{1}{2}a_1^2 - a_1a_2 - \tfrac{1}{2}a_1p_1 \\ +\tfrac{1}{2}a_1p_2 - \tfrac{1}{2}a_2^2 + \tfrac{1}{2}a_2p_1 - \tfrac{1}{2}a_2p_2 + \tfrac{1}{8}p_1^2 - \tfrac{1}{4}p_1p_2 + \tfrac{1}{8}p_2^2 \end{array}\right) \end{aligned}$$

which contains the cross term $-x_1x_2$, and so is not separable.

5.8.2 RC2 River Chain

We now look at the water value calculations that MORGANE makes for a system like the RC2. We show by example that fixing the final water levels of other reservoirs (as MORGANE does) in order to compute the marginal water value of a given reservoir can lead to a smaller number than the true value. Consider a system of two reservoirs in cascade as shown in Fig. 5.2.

Let $x(t) \in \mathbb{R}^2$ be the stock level at the start of period t in each reservoir (having capacity $a \in \mathbb{R}^2$)and let $u(t) \in \mathbb{R}^2$ denote the release from each reservoir, $h(t) \in \mathbb{R}^2$ the stochastic inflow and $s(t) \in \mathbb{R}^2$ the spill. All of these depend on time. Suppose that we control this system using a value of water for reservoir 1 only and setting a target in each stage on the level in reservoir 2. Given an expected future value function $V_{t+1}(x_1)$, the releases $u(\omega)$ are chosen to solve

$$\begin{array}{ll} \max & p_1u_1 + p_2u_2 + V_{t+1}(x_1 - u_1 - s_1 + h_1(\omega)) \\ \text{s.t.} & a_2/2 + h_2(\omega) + u_1 - u_2 - s_2 = a_2/2 \\ & 0 \le u_i \le b_i \\ & 0 \le s_i \le d_i \end{array} \tag{5.30}$$

We then compute

$$V_t(x_1) = \mathbb{E}(p_1u_1(\omega) + p_2u_2(\omega) + V_{t+1}(x_1 - u_1(\omega) - s_1(\omega) + h_1(\omega)) \tag{5.31}$$

We will test this approximation in a deterministic framework and study the marginal water value of reservoir 1. To do this consider two periods $t = 1,2$ with no residual water value, and consider the additional value of an extra amount δ of water in reservoir 1 at the start of period 1. This can be computed by finding

$$\begin{array}{ll} Q(\delta)= \max & \sum_{t=1}^2 \sum_{i=1}^2 p(t)u_i(t) \\ \text{s.t.} & x_1 + \delta + h_1(1) - u_1(1) - s_1(1) = x_1(2) \\ & x_2 + h_2(1) + u_1(1) - u_2(1) - s_2(1) = x_2(2) \\ & x_1(1) + h_1(2) - u_1(2) - s_1(2) = x_1(3) \\ & x_2(1) + h_2(2) + u_1(2) - u_2(2) - s_2(2) = x_2(3) \\ & 0 \le x_i(t) \le a_i \\ & 0 \le u_i(t) \le b_i \\ & 0 \le s_i(t) \le d_i \end{array} \tag{5.32}$$

Suppose the problem data are given as

	$p(t)$	$x_1(t)$	$x_2(t)$	$h_1(t)$	$h_2(t)$
$t=1$	2	1	0	0	0
$t=2$	1	–	–	0	2

(5.33)

Then

$$\begin{array}{ll} Q(0)= \max & 2u_1(1) + 2u_2(1) + u_1(2) + u_2(2) \\ \text{s.t.} & 1 - u_1(1) = x_1(2) + s_1(1) \\ & 0 + 1 + u_1(1) - u_2(1) = x_2(2) + s_2(1) \\ & x_1(1) - u_1(2) = x_1(3) + s_1(2) \\ & x_2(1) + u_1(2) - u_2(2) = x_2(3) + s_2(2) \\ & 0 \le x_i(t) \le 1 \\ & 0 \le u_1(t) \le 1, \quad 0 \le u_2(t) \le 2 \\ & 0 \le s_i(t) \le 3 \end{array} \tag{5.34}$$

This has solution

	$u_1(t)$	$u_2(t)$	$x_1(t+1)$	$x_2(t+1)$	$s_1(t)$	$s_2(t)$
$t=1$	1	2	0	1	0	0
$t=2$	0	1	0	0	0	0

(5.35)

with return 7. Now if we increase x_1 to $x_1+\delta$, then $Q(\delta)=7+3\delta$, so the marginal water value at reservoir 1 is 3. However, if we constrain the storage to be one-half capacity at the end of period 1, then we have $Q(\delta)=7+2\delta$, so the marginal water value at reservoir 1 is 2, which is less than its value in the unconstrained case.

References

1. Archibald T, Buchanan C, McKinnon K, Thomas L (1999) Nested Benders decomposition and dynamic programming for reservoir optimisation. J Oper Res Soc 50(5):468–479
2. Archibald T, McKinnon K, Thomas L (2006) Modeling the operation of multireservoir systems using decomposition and stochastic dynamic programming. Nav Res Log (NRL) 53(3):217–225
3. Barty K, Carpentier P (UMA), Cohen G (CERMICS), Girardeau P (UMA, CERMICS) (2010) Price decomposition in large-scale stochastic optimal control. http://arxiv.org/pdf/1012.2092v2 (submitted)
4. De Matos V, Philpott A, Finardi E, Guan Z (2010) Solving long-term hydrothermal scheduling problems. Technical report, Electric Power Optimization Centre, University of Auckland
5. Diniz AL, Maceira MEP (2008) A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects. IEEE Trans Power Syst 23(3):1298–1308
6. Dubost L, Gonzalez R, Lemaréchal C (2005) A primal-proximal heuristic applied to the French unit-commitment problem. Math Program (A 104):129–151
7. Jacobs J, Freeman G, Grygier J, Morton D, Schultz G, Staschus K, Stedinger J (1995) SOCRATES: a system for scheduling hydroelectric generation under uncertainty. Ann Oper Res 59:99–133
8. Little, J. (1955), The use of storage water in a hydroelectric system, J Oper Res Soc Amer 3(2):187–197.
9. Masse P (1946) Les réserves et la régulation de l'avenir dans la vie économique. Hermann, Paris
10. Pereira MVF, Pinto LMVG (1991) Multi-stage stochastic optimization applied to energy planning. Math Program 52:359–375
11. Philpott A, Guan Z (2008) On the convergence of stochastic dual dynamic programming and other methods. Oper Res Lett 36:450–455
12. Philpott A, Craddock M, Waterer H (2008) Hydro-electric unit commitment subject to uncertain demand. Eur J Oper Res 125:410–424
13. Turgeon A (1980) Optimal operation of multireservoir power systems with stochastic inflows. Water Resour Res 16(2):275–283