Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion*

A.B. Philpott† and V.L. de Matos‡

October 7, 2011

Abstract

We consider the incorporation of a time-consistent coherent risk measure into a multi-stage stochastic programming model, so that the model can be solved using a SDDP-type algorithm. We describe the implementation of this algorithm, and study the solutions it gives for an application of hydro-thermal scheduling in the New Zealand electricity system. The performance of policies using this risk measure at different levels of risk aversion is compared with the risk-neutral policy.

1 Introduction

Multi-stage stochastic linear programming models have been solved using decomposition for over thirty years, originating with the seminal work of [2], but there are still very few implementations of these models in commercial settings. The classical version of this model constructs a scenario tree that branches at each stage. Even with a small number of outcomes per stage, the size of the scenario tree grows exponentially with the number of stages. In two-stage problems with many scenarios, the sample average approximation approach enables large-scale problems to be solved within reasonable error bounds [11]. However, as argued by [22], the exponential growth of the scenario tree makes all but the smallest instances of multi-stage problems intractable for sample average approximation.

One area in which multi-stage stochastic linear programming models are widely applied is in the long-term scheduling of water resources, in particular in hydro-thermal electricity systems. This problem involves determining a policy of releasing water from reservoirs for hydro-electricity generation and generating from thermal plant over some planning horizon

*This research was carried out during a visit by the second author to the Electric Power Optimization Centre. Grant support from CAPES and Tractebel Energia GDF Suez - P&D ANEEL (PE-0403-0108/2009) and the New Zealand Marsden Fund under contract UOA719WIP is gratefully acknowledged.

†Electric Power Optimization Centre, University of Auckland, New Zealand: a.philpott@auckland.ac.nz

‡Laboratório de Planejamento de Sistemas de Energia Elétrica, Universidade Federal de Santa Catarina: vitor@labplan.ufsc.br

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 [14],[12]) 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. [24]) were used.

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. [10]) using scenario trees. Stochastic Dual Dynamic Programming (SDDP) [17] 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 stage-wise independent uncertainty. This allows cuts to be shared between different states, effectively collapsing the scenario tree.

The ability to share cuts under some specific forms of stage-wise dependency as discussed by Infanger and Morton [9] is now included in most commercial implementations of the SDDP algorithm. Monte Carlo sampling is also used in estimating bounds. These features make SDDP look more like an approximate dynamic programming method than a multi-stage stochastic linear programming algorithm. Commercial implementations of SDDP are in widespread use around the world, and are used to schedule hydro-electric plant in a number of South American countries including Brazil and Chile.

The standard implementations of SDDP are risk neutral, in that they seek policies that minimize expected cost. In hydro-thermal systems this cost comes about from thermal fuel and penalty costs, such as shortages. A cost-minimizing system operator would accept occasional shortages in electricity if this made the long-run cost of fuel a minimum. In practice, shortages do not occur very often, but when they do, they are so disruptive that politicians and system operators would wish to avoid them. It therefore makes sense to compute hydro-thermal scheduling policies that are risk averse. In some circumstances it is possible to have a significantly less risky policy with a modest increase in expected cost.

In this paper we describe a version of SDDP that models risk. Our work is based on the recent paper by Shapiro [23], but draws also on work by [21] and [15]. Our measure of risk in each stage is a convex combination of expectation and conditional value at risk [19], [20]. This makes it coherent as defined by [1]. The risk measure we use also satisfies a dynamic programming recursion, and so it is time-consistent in the sense defined by [21]. The recursive nature of its definition, and its convexity also admits approximation using cutting planes, and so we can modify SDDP to accommodate this.

Several other authors have developed SDDP implementations that account for risk. In [8], Iliadis *et al* describe a hydro-thermal scheduling model that accounts for the conditional value at risk of accumulated revenue shortfall at the end of the planning horizon, however there are few details in this paper about its implementation in the SDDP method. Guigues and Sagastizabal [5] study a rolling horizon model that repeatedly solves and implements the solution to a single stage problem with chance constraints. Guigues and Romisch [4]

present a general framework for extended polyhedral risk measures in the context of SDDP. The general risk measure they use makes use of a state space augmented by a vector of costs representing a possible history up to the current time. In contrast the model proposed by Shapiro [23] uses one extra state variable in each stage and so is more straightforward to compute. As we shall see, even in this case the algorithm takes some time to converge to a good solution.

Our aim in this paper is to demonstrate that risk-averse policies for this class of large-scale stochastic programming problems can be computed reasonably easily using SDDP-type methods. Moreover, by simulating these policies on a representation of a real hydro-thermal system, we are able to draw some conclusions about the value of these models as decision tools.

The paper is laid out as follows. In the next section, for completeness, we describe the risk-neutral SDDP algorithm, and describe a version of this model using a Markov chain to represent stage-wise dependence in our model. This section can be skipped by readers who are familar with this class of algorithms. In Section 3, we define conditional value at risk and describe how this is implemented in a multi-stage context in Section 4. We describe this in some detail, starting with a two-stage model to build the reader's intuition. The final model that we discuss in this section uses a Markov chain with states that can be used to adapt the level of risk aversion. In Sections 5 and 6 we describe a model of the New Zealand electricity system and some computational results of experiments where this approach is applied to this system, respectively. Section 7 concludes the paper.

2 Multi-stage stochastic linear programming

In this section we review the Stochastic Dual Dynamic Programming (SDDP) algorithm proposed by [17] as a solution strategy for risk-neutral multi-stage stochastic linear programming. For a more detailed discussion of this algorithm, the reader is referred to [18] and [23]. The class of problems we consider have T stages, denoted $t = 1, 2, \ldots, T$, in each of which a random right-hand-side vector $b_t(\omega_t) \in \mathbb{R}^m$ has a finite number of realizations defined by $\omega_t \in \Omega_t$. We assume that the outcomes ω_t are stage-wise independent, and that Ω_1 is a singleton, so the first-stage problem is

$$\begin{aligned} z = \min \quad & c_1^\top x_1 + \mathbb{E}[Q_2(x_1, \omega_2)] \\ \text{s.t.} \quad & A_1 x_1 = b_1, \\ & x_1 \geq 0, \end{aligned} \tag{1}$$

where $x_1 \in \mathbb{R}^n$ is the first stage decision and $c_1 \in \mathbb{R}^n$ a cost vector, A_1 is a $m \times n$ matrix, and $b_1 \in \mathbb{R}^m$.

We denote by $Q_2(x_1, \omega_2)$ the second stage costs associated with decision x_1 and realization $\omega_2 \in \Omega_2$. The problem to be solved in the second and later stages t, given decisions x_{t-1} and realization ω_t, can be written as

$$
\begin{aligned}
Q_t(x_{t-1}, \omega_t) = \min \quad & c_t^\top x_t + \mathbb{E}[Q_{t+1}(x_t, \omega_{t+1})] \\
\text{s.t.} \quad & A_t x_t = b_t(\omega_t) - E_t x_{t-1}, \quad [\pi_t(\omega_t)] \\
& x_t \geq 0,
\end{aligned} \tag{2}
$$

where $x_t \in \mathbb{R}^n$ is the decision in stage t, c_t its cost, and A_t and E_t denote $m \times n$ matrices. Here $\pi_t(\omega_t)$ denotes the dual variables of the constraints. In the last stage we assume either that $\mathbb{E}[Q_{T+1}(x_T, \omega_{T+1})] = 0$, or that there is a convex polyhedral function that defines the expected future cost after stage T. For all instances of (2) we assume relatively complete recourse, whereby (2) at stage t has a feasible solution for all values of x_{t-1} that are feasible for the instance of (2) at stage $t-1$. Relatively complete recourse can be ensured by introducing artificial variables with penalty terms in the objective.

2.1 Stochastic dual dynamic programming

The SDDP algorithm performs a sequence of major iterations, each consisting of a *forward pass* and a *backward pass* through all the stages, to build an approximately optimal policy. In each forward pass, a set of N scenarios is sampled from the scenario tree and decisions are taken for each stage of those N scenarios, starting in the first stage and moving forward up to the last stage. In each stage, the observed values $\bar{x}_t(s)$ of the decision variables x_t, and the costs of each stage in all scenarios s are saved.

The SDDP algorithm builds a policy that is defined at stage t by a polyhedral outer approximation of $\mathbb{E}[Q_{t+1}(x_t, \omega_{t+1})]$. This approximation is constructed using cutting planes called Benders cuts, or just *cuts*. In other words in each tth-stage problem, $\mathbb{E}[Q_{t+1}(x_t, \omega_{t+1})]$ is replaced by the variable θ_{t+1} which is constrained by a set of linear inequalities

$$
\theta_{t+1} - \bar{g}_{t+1,k,s}^\top x_t \geq \bar{h}_{t+1,k,s}, \qquad k = 1, 2, ..., K, \quad s = 1, 2, ..., N, \tag{3}
$$

where K is the number of backward passes that have been completed.

With this approximation, the first-stage problem is

$$
\begin{aligned}
z = \min \quad & c_1^\top x_1 + \theta_2 \\
\text{s.t.} \quad & A_1 x_1 = b_1, \\
& \theta_2 - \bar{g}_{2,k,s}^\top x_1 \geq \bar{h}_{2,k,s}, \qquad k = 1, 2, ..., K, \\
& \qquad\qquad\qquad\qquad\quad s = 1, 2, ..., N, \\
& x_1 \geq 0,
\end{aligned} \tag{4}
$$

and the t-th stage problem becomes

$$
\begin{aligned}
\tilde{Q}_t(x_{t-1}, \omega_t) = \min \quad & c_t^\top x_t + \theta_{t+1} \\
\text{s.t.} \quad & A_t x_t = b_t(\omega_t) - E_t x_{t-1}, \qquad [\pi_t(\omega_t)] \\
& \theta_{t+1} - \bar{g}_{t+1,k,s}^\top x_t \geq \bar{h}_{t+1,k,s}, \qquad k = 1, 2, ..., K, \\
& \qquad\qquad\qquad\qquad\qquad\quad s = 1, 2, ..., N, \\
& x_t \geq 0.
\end{aligned} \tag{5}
$$

where we interpret the set of cuts as being empty when $K = 0$.

At the end of the forward pass, a convergence criterion is tested, and if it is satisfied then the algorithm is stopped, otherwise it starts the backward pass, which is defined below. In the standard version of SDDP (see [17]), the convergence test is satisfied when z, the lower bound on the expected cost at the first stage (called the *Lower Bound*), is statistically close to an estimate of the expected total operation cost (called the *Upper Bound*) obtained by averaging the cost of the policy defined by the cuts when applied to the N sampled scenarios. In this simulation the total operation cost for each scenario is the sum of the present cost ($c_t^\top x_t$) over all stages t. For completeness we have descibed this test in our mathematical description of SDDP, but in our computational experiments we adopt a different approach in which the algorithm is terminated after a fixed number of iterations. This has proved to be more reliable than the standard test for the problems we are solving. (See [7],[23] for a discussion of the drawbacks of the standard convergence criterion.)

If the convergence criterion is not satisfied, then SDDP amends the current policy using a backward pass that adds N cuts to each stage problem, starting at the penultimate stage and working backwards to the first. To compute the coefficients for the cuts, we solve the next stage problems for all possible realizations (Ω_{t+1}) in each stage t and scenario s. The cut for (5), the t-th (approximate) stage problem in scenario s, is computed after its solution $\bar{x}_t^k(s)$ has been obtained in the forward pass immediately preceding backward pass k. Solving the $t+1$-th (approximate) stage problem for every $\omega_{t+1} \in \Omega_{t+1}$ gives $\bar{\pi}_{t+1,k,s} = \mathbb{E}[\pi_{t+1}(\omega_{t+1})]$, which defines the cut gradient

$$\bar{g}_{t+1,k,s} = -\bar{\pi}_{t+1,k,s}^\top E_{t+1} \tag{6}$$

and its intercept

$$\bar{h}_{t+1,k,s} = \mathbb{E}[Q_{t+1}(\bar{x}_t^k(s), \omega_{t+1})] + \bar{\pi}_{t+1,k,s}^\top E_{t+1} \bar{x}_t^k(s). \tag{7}$$

The SDDP algorithm is initialized by setting $\theta_t = -\infty$, $t = 2, ..., T$, $K = 0$, $k = 1$. Thereafter the algorithm performs the following three steps repeatedly until the convergence criterion is satisfied.

1. Forward Pass

   For $t = 1$, solve (4) and save $\bar{x}_1^k(s) = x_1$, $s = 1, ..., N$, and $\bar{z}^k = z$;

   For $t = 2, ..., T$ and $s = 1, ..., N$,

   Solve (5) setting $x_{t-1} = \bar{x}_{t-1}^k(s)$, and save $\bar{x}_t^k(s)$ and $\tilde{Q}_t(\bar{x}_{t-1}^k(s), \omega_t)$.

2. Standard Convergence Test (at $100(1-\alpha)\%$ confidence level).

   Calculate the Upper Bound: $z_u = \frac{1}{N} \sum_{s=1}^{N} \sum_{t=1}^{T} c_t^\top \bar{x}_t^k(s)$

   $$\sigma_u = \sqrt{\frac{1}{N} \sum_{s=1}^{N} \left(\sum_{t=1}^{T} c_t^\top \bar{x}_t^k(s) \right)^2 - z_u^2}.$$

Calculate the Lower Bound: $z_l = \bar{z}^k$;

Stop if

$$z_l > z_u - \frac{Z_{\frac{\alpha}{2}}}{\sqrt{N}}\sigma_u,$$

where Z_α is the $(1-\alpha)$ quantile of the standard normal distribution; otherwise go to the backward pass.

3. Backward Pass

For $t = T, ..., 2$, and $s = 1, ..., N$,

For $\omega_t \in \Omega_t$, solve (5) using $\bar{x}^k_{t-1}(s)$ and save $\bar{\pi}_{t,k,s} = \mathbb{E}[\pi_t(\omega_t)]$ and $\tilde{Q}_t(\bar{x}^k_{t-1}(s), \omega_t)$;

Calculate the kth cut for s in stage $t-1$ using (6) and (7).

Set $K = K + 1$, $k = k + 1$.

2.2 Markov process in the SDDP algorithm

The algorithm described above assumes that the random variables are stage-wise independent. In many settings this is not a suitable model, and there is some correlation over time. A popular approach to dealing with this is to model the random variables as an autoregressive process with independent errors (see e.g. [13]). As long as one assumes that the random variables on the constraint right-hand sides are affine functions of the errors, this approach can accommodate cut sharing using the approach discussed in [9]. In many settings, the right-hand side random variables are best modelled by a nonlinear transformation of an autoregressive process. For example, in hydro-thermal scheduling problems, the inflows are often transformed by a shifted logarithmic function to give close to normal disturbances before their correlation structure is extracted. In these circumstances, the affine assumption is not valid, and so cut sharing is not admissible.

In this paper, we describe a different approach in which the random variables have a probability distribution that depends on an underlying state which follows a Markov process. This method was originally developed in [16] to model randomly varying objective coefficients. When the state is continuous (as in an autoregressive process) we require that the future cost function is convex as a function of this state. This is usually not the case, and so the state is discrete (as in a finite Markov chain), and we must enumerate a future cost function for each value that the state may take, thus increasing the dimension of the dynamic program to be solved. This is the main computational disadvantage of this approach.

On the other hand, the Markov process approach has some modelling advantages that are not shared by the autoregressive process, that might make the extra effort worthwhile. In some systems, the realizations of the random variables do not match the fitted autoregressive process very well. The historical data for these systems can indicate the existence of random regime changes that are not captured well by considering variations around historical averages. These regime changes are better represented by a point process, around which

disturbances can be modelled. So-called *hidden* Markov models such as that described in [6] can be quite sophisticated and incorporate autocorrelated disturbances (such as an ARMA model) around an unobserved Markov state to deliver synthetic hydrological inflow sequences that match observed data very well.

The states of the Markov process can be quite general, and encapsulate all historical information that may be relevant to the actions taken at each stage of the optimization problem. For example, in the hydro-thermal scheduling context, the Markov state could include hydrological information from the current stage, market information, and factors influencing demand. One aim of this paper is to develop and test methodologies for incorporating a Markov model into SDDP, in both a risk-neutral and risk-averse framework. For our numerical experiments, we have chosen a very simple model with a small number of inflow states, and stage-wise independent variation about these. This structure enables us to use SDDP with cut sharing in this framework, without being concerned with disturbances that are not affine (as modelled in [13] and [6]).

To give a formal description of our approach, suppose that the process W_t, $t = 1, 2, \ldots T$, is a Markov chain with transition matrices $P^{(t)}$. For simplicity we denote the realizations of W_t by integers $i = 1, 2, \ldots, S$. In our hydro-thermal scheduling model, each Markov state realization i at stage t corresponds to a set Ω_{ti} of reservoir inflow outcomes ω_{ti}. Thus, the particular inflow outcome ω_{ti} that occurs in stage t is conditioned on the realization of W_t. We can simulate a realization of the overall process by alternately generating an outcome i for W_t and then an outcome for ω_{ti} chosen randomly from Ω_{ti}. Since in our computational results in section 6 we restrict attention to an implementation where $N = 1$, we shall assume henceforth that the forward pass contains only one scenario. This serves also to ease the notation needed for the description of the approach.

In the first-stage problem, we assume the system is in known state s_1, and Ω_{1s_1} is a singleton, giving

$$\begin{aligned} z = \min \quad & c_1^\top x_1 + \textstyle\sum_{j=1}^{S} P^{(1)}_{s_1 j} \mathbb{E}[Q_{2j}(x_1, \omega_{2j}) \mid W_2 = j] \\ \text{s.t.} \quad & A_1 x_1 = b_1, \\ & x_1 \geq 0, \end{aligned}$$

where $Q_{2j}(x_1, \omega_{2j})$ represents the second stage costs associated with decision x_1 and realization $\omega_{2j} \in \Omega_{2j}$. The problems to be solved in the second and later stages t, given the decision variables x_{t-1} from stage $t-1$, and given Markov state i and realization ω_{ti} can be written as

$$\begin{aligned} Q_{ti}(x_{t-1}, \omega_{ti}) = \min \quad & c_t^\top x_t + \textstyle\sum_{j=1}^{S} P^{(t)}_{ij} \mathbb{E}[Q_{t+1,j}(x_t, \omega_{t+1,j}) \mid W_{t+1} = j] & \\ \text{s.t.} \quad & A_t x_t = b_t(\omega_{ti}) - E_t x_{t-1}, & [\pi_t(\omega_{ti})] \\ & x_t \geq 0. & \end{aligned}$$

The forward pass of SDDP (with $N = 1$) now consists of a sequence of alternately sampled Markov state realizations and conditional inflow outcomes. At each stage t we solve a stage problem given the Markov state realization and observed inflow at this stage, using

a cutting-plane approximation of the future cost. This then yields a sequence of Markov states and values for decision variables $\bar{x}_t$, $t = 1, 2, \ldots, T-1$, that optimize each of the approximate stage problems.

The backward pass computes cuts at stage t at the point $\bar{x}_{t-1}$, where a different cut is recorded for each state $i = 1, 2, \ldots, S$ of the Markov chain. Thus in the backward pass at each stage $t > 1$, we solve $\sum_{i=1}^{S} |\Omega_{ti}|$ linear programs, each having a right-hand-side vector $b_t(\omega_{ti}) \in \mathbb{R}^m$, $\omega_{ti} \in \Omega_{ti}$. Note that each of these problems is solved at the same $\bar{x}_{t-1}$ computed in the preceding forward pass.

In each stage problem, there are several possible ways to define the future cost function using cuts. In the *single-cut* version, for each Markov state i at stage t after K iterations, we compute a solution to an outer approximation of the stage problem at the point $\bar{x}_{t-1}$ using

$$\begin{array}{rll} \tilde{Q}_{ti}(\bar{x}_{t-1}, \omega_{ti}) = \min & c_t^\top x_t + \theta_{t+1,i} & \\ \text{s.t.} & A_t x_t = b_t(\omega_{ti}) - E_t \bar{x}_{t-1}, & [\pi_t(\omega_{ti})] \\ & \theta_{t+1,i} + \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j,k}^\top E_{t+1} x_t \geq \sum_{j=1}^{S} P_{ij}^{(t)} \bar{h}_{t+1,j,k}, & k = 1, 2, \ldots K, \\ & x_t \geq 0. & \end{array} \tag{8}$$

where, for each $k = 1, 2, \ldots, K$,

$$\bar{\pi}_{t+1,j,k} = \mathbb{E}[\pi_{t+1}(\omega_{t+1,j}) \mid W_{t+1} = j] \text{ evaluated at iterate } \bar{x}_t^k, \tag{9}$$

$$\bar{h}_{t+1,j,k} = \mathbb{E}[\tilde{Q}_{t+1,j}(\bar{x}_t^k, \omega_{t+1,j}) \mid W_{t+1} = j] + \bar{\pi}_{t+1,j,k}^\top E_{t+1} \bar{x}_t^k. \tag{10}$$

The optimal value function $\tilde{Q}_{t+1,j}(x, \omega_{t+1,j})$ for the subproblem solved in state j and outcome $\omega_{t+1,j}$ at stage $t+1$ is a convex polyhedral function of x, having subgradient $-E_{t+1}^\top \pi_{t+1}(\omega_{t+1,j})$ at $x = \bar{x}_t$. The conditional expectation of the optimal value function at x given state j is $\mathbb{E}[\tilde{Q}_{t+1,j}(x, \omega_{t+1,j}) \mid W_t = j]$ which is convex with subgradient

$$\mathbb{E}[-E_{t+1}^\top \pi_{t+1}(\omega_{t+1,j}) \mid W_t = j] = -E_{t+1}^\top \bar{\pi}_{t+1,j}$$

at $x = \bar{x}_t$. The (approximate) future cost function $\theta_{t+1,i}(x)$ evaluated at x in state i with outcome ω_{ti} at stage t is then the expectation of the optimal value function in each Markov state that might occur in the next stage. This is

$$\theta_{t+1,i}(x) = \sum_{j=1}^{S} P_{ij}^{(t)} \mathbb{E}[\tilde{Q}_{t+1,j}(x, \omega_{t+1,j}) \mid W_t = j],$$

which is convex with subgradient $-E_{t+1}^\top \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j}$ at $x = \bar{x}_t$. So this future cost function satisfies

$$\theta_{t+1,i}(x) \geq \theta_{t+1,i}(\bar{x}_t) - \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j,k}^\top E_{t+1}(x - \bar{x}_t)$$

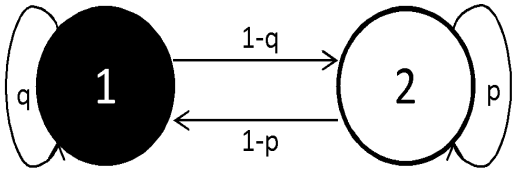

Figure 1: Example of state transitions.

showing that

$$\theta_{t+1,i} + \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j,k}^{\top} E_{t+1} x \geq \sum_{j=1}^{S} P_{ij}^{(t)} \bar{h}_{t+1,j,k}$$

is a valid cut for the approximate future cost.

In the *multi-cut* version of this method, we represent the future cost by cuts for each of the possible Markov states in the next stage. Thus at stage t, we compute

$$\begin{array}{rll} \tilde{Q}_{ti}(x_{t-1}, \omega_{ti}) = \min & c_t^{\top} x_t + \sum_{j=1}^{S} P_{ij}^{(t)} \theta_{t+1,j} & \\ \text{s.t.} & A_t x_t = b_t(\omega_{ti}) - E_t x_{t-1}, & [\pi_t(\omega_{ti})] \\ & \theta_{t+1,j} + \bar{\pi}_{t+1,j,k}^{\top} E_{t+1} x_t \geq \bar{h}_{t+1,j,k}, & \begin{array}{l} j = 1, 2, \ldots S, \\ k = 1, 2, \ldots K, \end{array} \\ & x_t \geq 0, & \end{array} \tag{11}$$

where $\bar{\pi}_{t+1,j,k}^{\top}$ and $\bar{h}_{t+1,j,k}$ are defined by (9) and (10) respectively. A similar analysis to the above shows that this defines a valid outer approximation to the future cost function using cutting planes.

In both single-cut and multi-cut versions of the algorithm it is necessary to maintain S sets of cuts at each stage. In the multi-cut version of the algorithm, each of the S subproblems will use one set of cuts, so the stage optimization problems will be larger, whereas in the single-cut case, each node will use only one set of cuts. Although the size of each stage problem grows more quickly in the multi-cut case, this strategy is expected to require fewer iterations to achieve convergence.

This construction can be illustrated with an example. Suppose that the Markov chain has two states, 1 and 2, which are shown in Figure 1 with the transition probabilities q, $1-q$, and p, $1-p$. Here and henceforth we will colour nodes in state 1 black and nodes in state 2 white.We augment the state space with a new state variable which takes values 1 and 2. Suppose that the random realizations in each stage can take only four values, ξ_a, ξ_b, ξ_c, ξ_d, and that these can be classified into two states as shown in Figure 2.

For a three-stage problem, this corresponds to a scenario tree as shown in Figure 3, in which the black nodes correspond to state 1 and the white nodes represent state 2. From Figure 3 it is possible to see that the set of descendant nodes is the same for any given stage, but they may have different probabilities depending on the value of the current state.

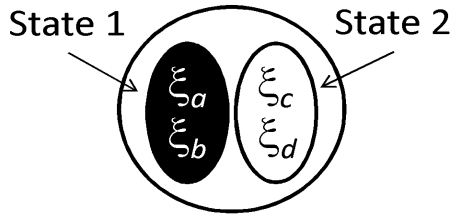

Figure 2: Markov process states. The outcomes ξ_a and ξ_a are equally likely conditional on being in state 1, and ξ_c and ξ_d are equally likely conditional on being in state 2.

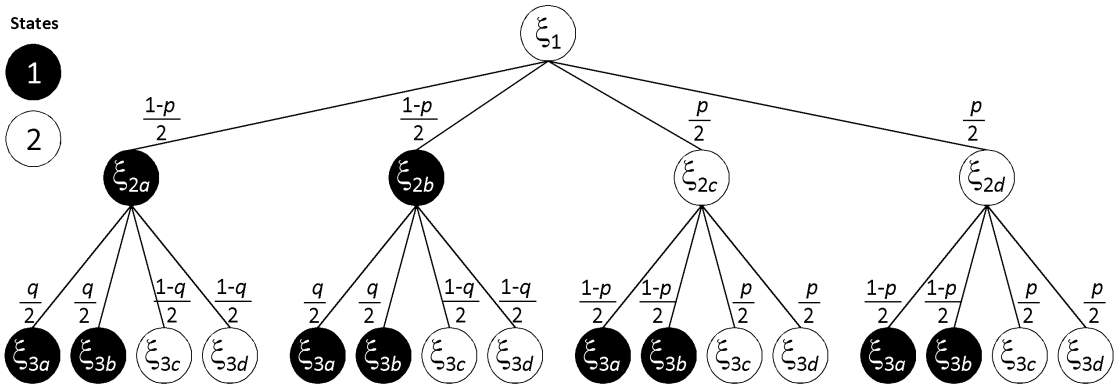

Figure 3: Scenario tree with the Markov Process.

Therefore, cuts cannot be shared directly. However as the dual solutions in one node are valid for all nodes with the same realization in that stage, one can use the solutions to compute a cut for each state by using the appropriate probabilities.

The single-cut approach constructs the conditional expectation of each cut with the appropriate transition probabilities and stores a single extra cut in each state at stage 2. This gives

$$\begin{aligned}
\theta_{31} + \left[q\tfrac{\pi_{3a}+\pi_{3b}}{2} + (1-q)\tfrac{\pi_{3c}+\pi_{3d}}{2}\right]^\top E_3 x_2 \geq\ & \left[q\tfrac{Q(\xi_{3a})+Q(\xi_{3b})}{2} + (1-q)\tfrac{Q(\xi_{3c})+Q(\xi_{3d})}{2}\right] \\
& + \left[q\tfrac{\pi_{3a}+\pi_{3b}}{2} + (1-q)\tfrac{\pi_{3c}+\pi_{3d}}{2}\right]^\top E_3 \bar{x}_2. \quad \text{for state 1} \\
\theta_{32} + \left[(1-p)\tfrac{\pi_{3a}+\pi_{3b}}{2} + p\tfrac{\pi_{3c}+\pi_{3d}}{2}\right]^\top E_3 x_2 \geq\ & \left[(1-p)\tfrac{Q(\xi_{3a})+Q(\xi_{3b})}{2} + p\tfrac{Q(\xi_{3c})+Q(\xi_{3d})}{2}\right] \\
& + \left[(1-p)\tfrac{\pi_{3a}+\pi_{3b}}{2} + p\tfrac{\pi_{3c}+\pi_{3d}}{2}\right]^\top E_3 \bar{x}_2. \quad \text{for state 2}
\end{aligned}$$

In the multi-cut approach the third stage problems would generate the following pair of cuts for the second stage

$$\begin{aligned}
\theta_{3ab} + \left(\tfrac{\pi_{3a}+\pi_{3b}}{2}\right)^\top E_3 x_2 &\geq \tfrac{Q(\xi_{3a})+Q(\xi_{3b})}{2} + \left(\tfrac{\pi_{3a}+\pi_{3b}}{2}\right)^\top E_3 \bar{x}_2, \\
\theta_{3cd} + \left(\tfrac{\pi_{3c}+\pi_{3d}}{2}\right)^\top E_3 x_2 &\geq \tfrac{Q(\xi_{3c})+Q(\xi_{3d})}{2} + \left(\tfrac{\pi_{3c}+\pi_{3d}}{2}\right)^\top E_3 \bar{x}_2,
\end{aligned}$$

and the future cost at stage 2 is represented by

$$\theta_{31} = q\theta_{3ab} + (1-q)\theta_{3cd}$$

when the node corresponds to state 1 (black) and by

$$\theta_{32} = (1-p)\theta_{3ab} + (p)\theta_{3cd}$$

when the node corresponds to state 2 (white).

We observe that the version of SDDP described in this section will be computationally effective only in cases where S is small (our experiments in Section 5 are limited to a model with $S = 4$), and in its current form, it may not be suitable for large S. Nevertheless, we can use the approach to test the efficacy of using risk-averse models to avoid high costs in situations where there is some stage-wise dependence in the random variables that might make a solution that assumed independence and risk neutrality perform quite badly (even on average).

3 Risk measures

In this section we begin our discussion of how to make the policies generated by SDDP risk averse in the sense that they penalize large losses, without compromising the expected cost too much. One common approach to measure the risk of a loss distribution of a given random variable Z is the $1-\alpha$ *value at risk*, $\mathsf{VaR}_{1-\alpha}[Z]$, that is defined by [20] as

$$\mathsf{VaR}_{1-\alpha}[Z] = \inf_u\{u : \Pr(Z \le u)\} \ge 1-\alpha,$$

where α is typically chosen to be some small probability e.g. 0.05. This means that $\mathsf{VaR}_{1-\alpha}[Z]$ is the left-side $(1-\alpha)$th percentile of the loss distribution. It is well known that even when Z is a convex function of a decision x, the function $\mathsf{VaR}_{1-\alpha}[Z]$ is not guaranteed to be convex in x, which makes optimization difficult in general, and impossible in SDDP. The tightest convex safe approximation of $\mathsf{VaR}_{1-\alpha}[Z]$ is called the *conditional value at risk*. This can be written [20] as

$$\mathsf{CVaR}_{1-\alpha}[Z] = \inf_u\{u + \alpha^{-1}\mathbb{E}[(Z-u)_+]\},$$

where we write $(a)_+$ for $\max\{a, 0\}$.

In this paper we study a combination of the expected total cost and the conditional value at risk, as suggested by Shapiro [23]. Therefore, we use a risk measure

$$\rho(Z) = \beta\mathbb{E}[Z] + \gamma\mathsf{CVaR}_{1-\alpha}[Z], \tag{12}$$

where β and γ are nonnegative. In practice it makes sense to choose $\beta > 0$ since $\mathsf{CVaR}_{1-\alpha}[Z]$ on its own will disregard the effect of decisions on expected outcomes, which might result in expensive policies on average that we would wish to avoid if cheaper ones were possible with the same level of CVaR.

Conditional value at risk is an example of a coherent risk measure. According to [1] a function $\rho : \mathbb{R}^n \to \mathbb{R}$ is a *coherent risk measure* if ρ satisfies the following axioms for Z_1 and $Z_2 \in \mathbb{R}^n$.

- **Convexity:** $\rho(\alpha Z_1 + (1-\alpha)Z_2) \leq \alpha\rho(Z_1) + (1-\alpha)\rho(Z_2)$, for $\alpha \in [0,1]$;
- **Monotonicity:** If $Z_1 \leq Z_2$, then $\rho(Z_1) \leq \rho(Z_2)$;
- **Positive homogeneity:** If $U \in \mathbb{R}$ and $U > 0$, then $\rho(UZ_1) = U\rho(Z_1)$;
- **Translation equivariance:** If $U \in \mathbb{R}$, then $\rho(\mathbb{I}U + Z_1) = U + \rho(Z_1)$.

The risk measure defined in (12) satisfies the first three axioms, and in order to satisfy the fourth (translation equivariance), we have

$$\begin{aligned} U + \rho(Z_1) &= \rho(\mathbb{I}U + Z_1) \\ &= \beta\mathbb{E}[\mathbb{I}U + Z_1] + \gamma\mathsf{CVaR}_{1-\alpha}[\mathbb{I}U + Z_1] \\ &= \beta U + \beta\mathbb{E}[Z_1] + \gamma U + \gamma\mathsf{CVaR}_{1-\alpha}[Z_1] \\ &= (\beta+\gamma)U + \beta\mathbb{E}[Z_1] + \gamma\mathsf{CVaR}_{1-\alpha}[Z_1] \\ &= (\beta+\gamma)U + \rho(Z_1). \end{aligned}$$

so $\beta + \gamma = 1$. Therefore, we replace β and γ by $(1-\lambda)$ and λ, respectively to give

$$\rho(Z) = (1-\lambda)\mathbb{E}[Z] + \lambda\mathsf{CVaR}_{1-\alpha}[Z] \tag{13}$$

yielding

$$\rho(Z) = (1-\lambda)\mathbb{E}[Z] + \lambda \inf_u \{u + \alpha^{-1}\mathbb{E}[(Z-u)_+]\}.$$

The risk measure $\rho(Z)$ is equivalent to the *mean deviation from quantile* proposed by Miller and Ruszczynski [15], bearing in mind that in our setting we are minimizing Z. In this setting, the mean deviation from quantile measure is

$$\rho_d(Z) = \mathbb{E}[Z] + \lambda \min_\eta \sum_{i=1}^{N} p_i \max\left(\frac{1-\alpha}{\alpha}(z_i - \eta), \eta - z_i\right) \tag{14}$$

in which N is the number of realizations of the discrete random variable Z. We have

$$\max(\frac{1-\alpha}{\alpha}(z_i-\eta), \eta - z_i) = (\eta - z_i) + \max\left(\frac{1-\alpha}{\alpha}(z_i-\eta) - (\eta - z_i), 0\right)$$

and

$$\frac{1-\alpha}{\alpha}(z_i-\eta) - (\eta - z_i) = \frac{1}{\alpha}\left[(z_i-\eta) - \alpha(z_i-\eta) + \alpha(z_i-\eta)\right] = \frac{1}{\alpha}(z_i-\eta).$$

So,

$$\max(\frac{1-\alpha}{\alpha}(z_i-\eta), \eta - z_i) = (\eta - z_i) + \max\left(\frac{1}{\alpha}(z_i-\eta), 0\right) \tag{15}$$

Therefore, by replacing (15) in (14) we obtain

$$\begin{aligned}\rho_d(Z) &= \mathbb{E}[Z] + \lambda \min_{\eta} \sum_{i=1}^{N} p_i \left[(\eta - z_i) + \max\left(\frac{1}{\alpha}(z_i - \eta), 0 \right) \right] \\ &= \mathbb{E}[Z] - \lambda \mathbb{E}[Z] + \lambda \min_{\eta} (\eta + \sum_{i=1}^{N} p_i \max\left(\frac{1}{\alpha}(z_i - \eta), 0 \right)) \\ &= \rho(Z)\end{aligned}$$

The measure ρ as defined is a single period measure, which is extended in [23] to a *dynamic risk measure* $\rho_{t,T}$ over $t = 1, 2, \ldots, T$ following the general theory of [21]. To help the reader interpret the computational results it is worthwhile presenting a brief summary of this general construction. Given a probability space $(\Omega, \mathcal{F}, P)$, a dynamic risk measure applies to a situation in which we have a random sequence of costs $(Z_1, Z_2, \ldots, Z_T)$ which is adapted to some filtration $\{0, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \ldots \subset \mathcal{F}_T \subset \mathcal{F}$ of σ-fields, where Z_1 is assumed to be deterministic. A dynamic risk measure is then defined to be a sequence of *conditional risk measures* $\{\rho_{t,T}\}$, $t = 1, 2, \ldots, T$. Given a dynamic risk measure, we can derive a corresponding single-period risk measure using

$$\rho_t(Z_{t+1}) = \rho_{t,T}(0, Z_{t+1}, 0, \ldots, 0).$$

By [21, Theorem 1], any time-consistent dynamic risk measure can then be constructed in terms of single-period risk measures ρ_t by the formula

$$\rho_{t,T}(Z_t, Z_{t+1}, \ldots, Z_T) = Z_t + \rho_t(Z_{t+1} + \rho_{t+1}(Z_{t+2} + \ldots + \rho_{T-2}(Z_{T-1} + \rho_{T-1}(Z_T)) \ldots))).$$

In the next section we describe this construction in the special case in which we choose the single-period risk measure

$$\rho_t(Z) = (1 - \lambda_{t+1})\mathbb{E}[Z \mid \mathcal{F}_t] + \lambda_{t+1} \inf_u \{u + \alpha^{-1}\mathbb{E}[(Z - u)_+ \mid \mathcal{F}_t]\}$$

where λ_{t+1} is a parameter chosen to be measurable with respect to $\mathcal{F}_t$.

4 Implementing a CVaR risk measure in SDDP

In this section we present the modelling strategy to optimize the coherent risk measure discussed in Section 3. This can be considered to be one of the main contributions of this paper, because although our approach is similar to the ones shown in [23] and [15], there are some important differences related to our solution strategy. In this section we omit a description of the basic SDDP algorithm, because the algorithm is exactly the same as the one presented in Section 2 except for the problems to be solved and the cut calculations.

4.1 A two-stage model

To help understand how the stage problems are affected by our risk measure, we first consider a two-stage linear problem that aims to minimize the first stage cost plus the risk measure applied to the second stage costs. Here the first stage is deterministic and the second stage random variable has finite support Ω_2. In this paper the stochastic process is going to be modelled by random variables only in the constraint right-hand side. This problem can be written as follows:

$$\begin{array}{rll} \text{SP: min} & c_1^\top x_1 + (1-\lambda)\mathbb{E}[c_2^\top x_2] + \lambda u_2 + \lambda\alpha^{-1}\mathbb{E}[\left(c_2^\top x_2 - u_2\right)_+] \\ \text{s.t.} & A_1 x_1 = b_1, \\ & A_2 x_2(\omega) + E_2 x_1 = b_2(\omega), & \text{for all } \omega \in \Omega_2, \\ & x_1 \geq 0, \quad x_2(\omega) \geq 0, & \text{for all } \omega \in \Omega_2. \end{array}$$

We then replace $\left(c_2^\top x_2 - u_2\right)_+$ by $v_2(\omega)$ where

$$\begin{array}{rcll} v_2(\omega) & \geq & c_2^\top x_2(\omega) - u_2, & \text{for all } \omega \in \Omega_2, \\ v_2(\omega) & \geq & 0, & \text{for all } \omega \in \Omega_2. \end{array}$$

As a consequence, the new 2-stage problem can be written as the following linear program:

$$\begin{array}{rll} \text{SP: min} & c_1^\top x_1 + (1-\lambda)\mathbb{E}[c_2^\top x_2] + \lambda u_2 + \lambda\alpha^{-1}\mathbb{E}[v_2] \\ \text{s.t.} & A_1 x_1 = b_1, \\ & A_2 x_2(\omega) + E_2 x_1 = b_2(\omega), & \text{for all } \omega \in \Omega_2, \\ & v_2(\omega) \geq c_2^\top x_2(\omega) - u_2, & \text{for all } \omega \in \Omega_2, \\ & x_1 \geq 0, \quad x_2(\omega) \geq 0, \quad v_2(\omega) \geq 0, & \text{for all } \omega \in \Omega_2. \end{array}$$

Observe in SP that there are two first-stage decisions to be made, x_1, and the level u_2 that attains $\inf_u\{u+\alpha^{-1}\mathbb{E}\left[\left(c_2^\top x_2 - u\right)_+\right]\}$. Given choices of $x_1 = \bar{x}_1$ and $u_2 = \bar{u}_2$ the second-stage problem becomes

$$\begin{array}{rll} \text{SP}(\bar{x}_1, \bar{u}_2)\text{: min} & (1-\lambda)\mathbb{E}[c_2^\top x_2] + \lambda\alpha^{-1}\mathbb{E}[v_2] \\ \text{s.t.} & A_2 x_2(\omega) = b_2(\omega) - E_2\bar{x}_1, & \text{for all } \omega \in \Omega_2, \\ & v_2(\omega) - c_2^\top x_2(\omega) \geq -\bar{u}_2, & \text{for all } \omega \in \Omega_2, \\ & x_2(\omega) \geq 0, \quad v_2(\omega) \geq 0, & \text{for all } \omega \in \Omega_2. \end{array}$$

This decouples by scenarios to give

$$\begin{array}{rll} Q(\bar{x}_1, \bar{u}_2, \omega) = \min & (1-\lambda)c_2^\top x_2 + \lambda\alpha^{-1} v_2 \\ \text{s.t.} & A_2 x_2 = b_2(\omega) - E_2\bar{x}_1, & [\pi_2(\omega)] \\ & v_2 - c_2^\top x_2 \geq -\bar{u}_2, & [\phi_2(\omega)] \\ & x_2 \geq 0, \quad v_2 \geq 0. \end{array}$$

The optimal dual multipliers are shown in brackets on the right. By strong duality the optimal dual solution satisfies

$$Q(\bar{x}_1, \bar{u}_2, \omega) = \pi_2(\omega)^\top(b_2 - E_2\bar{x}_1) - \phi_2(\omega)\bar{u}_2.$$

SP can now be represented by

$$\begin{array}{rll} \text{SP: min} & c_1^\top x_1 + \lambda u_2 + \mathbb{E}[Q(x_1, u_2, \omega)] \\ \text{s.t.} & A_1 x_1 = b_1, \\ & x_1 \geq 0, \end{array}$$

and Benders decomposition can be used to compute its solution, by solving

$$\begin{array}{rll} \text{MP: min} & c_1^\top x_1 + \lambda u_2 + \theta_2 \\ \text{s.t.} & A_1 x_1 = b_1, \\ & \theta_{2k} + \bar{\pi}_{2k}^\top E_2 x_1 + \bar{\phi}_{2k} u_2 \geq \bar{h}_{2k}, \quad k = 1, 2, \ldots, K \\ & x_1 \geq 0, \end{array}$$

where k counts the cuts that are added to the Benders master problem,

$$\begin{aligned} \bar{\pi}_{2k} &= \mathbb{E}[\pi_{2k}(\omega)], \\ \bar{\phi}_{2k} &= \mathbb{E}[\phi_{2k}(\omega)], \\ \bar{h}_{2k} &= \mathbb{E}[Q_2(\bar{x}_{1k}, \bar{u}_{2k}, \omega)] + \bar{\pi}_{2k} E_2 \bar{x}_{1k} + \bar{\phi}_{2k} \bar{u}_{2k}, \end{aligned}$$

and $\bar{x}_{1k}$and $\bar{u}_{2k}$ are the values of first-stage variables at which cut k is evaluated.

4.2 A multi-stage model

We can generalize this method to a T-stage problem, which we illustrate using notation for a three-stage problem. We consider a probability space $(\Omega, \mathcal{F}, P)$, and a random sequence of right-hand sides $(b_1, b_2, \ldots, b_T)$ adapted to some filtration $\{0, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_T \subset \mathcal{F}$ of σ-fields, where b_1 is assumed to be deterministic. We assume in this section that all random parameters are stage-wise independent, and that parameters $\lambda_2, \ldots, \lambda_T$ are deterministic. In the case where $T = 3$, SP can be written as follows:

$$\begin{array}{rll} & c_1^\top x_1 + (1-\lambda_2)\mathbb{E}[c_2^\top x_2 + (1-\lambda_3)\mathbb{E}[c_3^\top x_3 \mid \mathcal{F}_2] + \lambda_3 u_3 + \lambda_3 \alpha_3^{-1} \mathbb{E}[\left(c_3^\top x_3 - u_3\right)_+ \mid \mathcal{F}_2]] \\ \text{SP: min} & +\lambda_2 u_2 + \lambda_2 \alpha_2^{-1} \mathbb{E}\left[\left(\begin{array}{c} c_2^\top x_2 + (1-\lambda_3)\mathbb{E}[c_3^\top x_3 \mid \mathcal{F}_2] + \lambda_3 u_3 \\ +\lambda_3 \alpha_3^{-1} \mathbb{E}[\left(c_3^\top x_3 - u_3\right)_+ \mid \mathcal{F}_2] - u_2 \end{array} \right)_+\right] \\ \text{s.t.} & A_1 x_1 = b_1, \\ & A_2 x_2(\omega_2) + E_2 x_1 = b_2(\omega_2), \qquad \text{for all } \omega_2 \in \Omega_2, \\ & A_3 x_3(\omega_3) + E_3 x_2(\omega_2) = b_3(\omega_3), \qquad \text{for all } \omega_2 \in \Omega_2 \text{ and } \omega_3 \in \Omega_3, \\ & x_1 \geq 0, \qquad x_2(\omega_2) \geq 0, \qquad x_3(\omega_3) \geq 0, \qquad \text{for all } \omega_2 \in \Omega_2 \text{ and } \omega_3 \in \Omega_3. \end{array} \tag{16}$$

Since the random parameters b_3 are assumed to be independent of b_2, the third-stage problem can be formulated as

$$\begin{array}{rlll} Q_3(x_2, u_3, \omega_3) = \min & (1-\lambda_3) c_3^\top x_3 + \lambda_3 \alpha_3^{-1} v_3 & \\ \text{s.t.} & A_3 x_3 = b_3(\omega_3) - E_3 x_2, & [\pi_3(\omega_3)] \\ & v_3 - c_3^\top x_3 \geq -u_3, & [\phi_3(\omega_3)] \\ & x_3 \geq 0, \qquad v_3 \geq 0. & \end{array}$$

Stage-wise independence allows us to denote $\mathbb{E}[Q_3(x_2, u_3, \omega_3) \mid \mathcal{F}_2]$ by the function $\mathcal{Q}_3(x_2, u_3)$, which is measurable with respect to $\mathcal{F}_2$ through its dependence on x_2 and u_3. Thus we can write SP as follows:

$$\begin{array}{rll} \text{SP: min} & c_1^\top x_1 + (1-\lambda_2)\mathbb{E}[c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2,u_3)] + \lambda_2 u_2 & \\ & +\lambda_2\alpha^{-1}\mathbb{E}\left[(c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2,u_3) - u_2)_+\right] & \\ \text{s.t.} & A_1 x_1 = b_1, & \\ & A_2 x_2(\omega_2) + E_2 x_1 = b_2(\omega_2), & \text{for all } \omega_2 \in \Omega_2, \\ & x_1 \geq 0, \quad x_2(\omega_2) \geq 0, & \text{for all } \omega_2 \in \Omega_2. \end{array}$$

We then replace $(c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2, u_3) - u_2)_+$ by $v_2(\omega_2)$ where

$$\begin{array}{rcll} v_2(\omega_2) & \geq & c_2^\top x_2(\omega_2) + \lambda_3 u_3(\omega_2) + \mathcal{Q}_3(x_2(\omega_2), u_3(\omega_2)) - u_2 & \text{for all } \omega_2 \in \Omega_2, \\ v_2(\omega_2) & \geq & 0 & \text{for all } \omega_2 \in \Omega_2. \end{array}$$

As a consequence, the new 2-stage problem can be written as

$$\begin{array}{rll} \text{SP: min} & c_1^\top x_1 + (1-\lambda_2)\mathbb{E}[c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2,u_3)] + \lambda_2 u_2 + \lambda_2\alpha^{-1}\mathbb{E}[v_2] & \\ \text{s.t.} & A_1 x_1 = b_1, & \\ & A_2 x_2(\omega_2) + E_2 x_1 = b_2(\omega_2), & \text{for all } \omega_2 \in \Omega_2, \\ & v_2(\omega_2) \geq c_2^\top x_2(\omega_2) + \lambda_3 u_3(\omega_2) + \mathcal{Q}_3(x_2(\omega_2), u_3(\omega_2)) - u_2, & \text{for all } \omega_2 \in \Omega_2, \\ & x_1 \geq 0, \quad x_2(\omega_2) \geq 0, \quad v_2(\omega_2) \geq 0, & \text{for all } \omega_2 \in \Omega_2. \end{array}$$

Given choices of $x_1 = \bar{x}_1$ and $u_2 = \bar{u}_2$ the second stage problem becomes

$$\begin{array}{rll} \text{SP}(\bar{x}_1, \bar{u}_2)\text{: min} & (1-\lambda_2)\mathbb{E}[c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2,u_3)] + \lambda_2\alpha^{-1}\mathbb{E}[v_2] & \\ \text{s.t.} & A_2 x_2(\omega_2) = b_2(\omega_2) - E_2\bar{x}_1, & \text{for all } \omega_2 \in \Omega_2, \\ & v_2(\omega_2) - c_2^\top x_2(\omega_2) - \lambda_3 u_3(\omega_2) - \mathcal{Q}_3(x_2(\omega_2), u_3(\omega_2)) \geq -\bar{u}_2, & \text{for all } \omega_2 \in \Omega_2, \\ & x_2(\omega_2) \geq 0, \quad v_2(\omega_2) \geq 0, & \text{for all } \omega_2 \in \Omega_2. \end{array}$$

This decouples by scenario to give

$$\begin{array}{rll} Q_2(\bar{x}_1, \bar{u}_2, \omega_2) = \text{min} & (1-\lambda_2)(c_2^\top x_2 + \lambda_3 u_3 + \mathcal{Q}_3(x_2,u_3)) + \lambda_2\alpha^{-1} v_2 & \\ \text{s.t.} & A_2 x_2 = b_2(\omega_2) - E_2\bar{x}_1, & \\ & v_2 - c_2^\top x_2 - \lambda_3 u_3 - \mathcal{Q}_3(x_2,u_3) \geq -\bar{u}_2, & \\ & x_2 \geq 0, \quad v_2 \geq 0. & \end{array}$$

Now if $\mathcal{Q}_3(x_2, u_3)$ can be approximated by K_3 cuts, then we obtain a lower bound approximation to $Q_2(\bar{x}_1, \bar{u}_2, \omega_2)$ denoted

$$\begin{array}{rll} \tilde{Q}_2(\bar{x}_1, \bar{u}_2, \omega_2) = \text{min} & (1-\lambda_2)(c_2^\top x_2 + \lambda_3 u_3 + \theta_3) + \lambda_2\alpha^{-1} v_2 & \\ \text{s.t.} & A_2 x_2 = b_2(\omega_2) - E_2\bar{x}_1, & [\pi_2(\omega_2)] \\ & v_2 - c_2^\top x_2 - \lambda_3 u_3 - \theta_3 \geq -\bar{u}_2, & [\phi_2(\omega_2)] \\ & \theta_3 + \bar{\pi}_{3k}^\top E_3 x_2 + \bar{\phi}_{3k} u_3 \geq \bar{h}_{3k}, & k = 1, 2, \ldots, K_3, \\ & x_2 \geq 0, \quad v_2 \geq 0. & \end{array}$$

In general the approximate optimal value of the tth stage of SP can be represented at any x_{t-1}, u_t by

$$\begin{array}{rll} \tilde{Q}_t(x_{t-1}, u_t, \omega_t) = \min & (1-\lambda_t)(c_t^\top x_t + \lambda_{t+1}u_{t+1} + \theta_{t+1}) + \lambda_t \alpha^{-1} v_t & \\ \text{s.t.} & A_t x_t = b_t(\omega_t) - E_t x_{t-1}, & [\pi_t(\omega_t)] \\ & v_t - (c_t^\top x_t + \lambda_{t+1}u_{t+1} + \theta_{t+1}) \geq -u_t, & [\phi_t(\omega_t)] \\ & \theta_{t+1} + \bar{\pi}_{t+1,k}^\top E_{t+1} x_t + \bar{\phi}_{t+1,k} u_{t+1} \geq \bar{h}_{t+1,k}, & k = 1, 2, \ldots, K_{t+1}, \\ & x_t \geq 0, \qquad v_t \geq 0. & \end{array} \tag{17}$$

where k counts the cuts that are added to the tth-stage Benders master problem,

$$\begin{aligned} \bar{\pi}_{t+1,k} &= \mathbb{E}[\pi_{t+1,k}(\omega_{t+1})], \\ \bar{\phi}_{t+1,k} &= \mathbb{E}[\phi_{t+1,k}(\omega_{t+1})], \\ \bar{h}_{t+1,k} &= \mathbb{E}[Q_{t+1}(\bar{x}_{tk}, \bar{u}_{t+1,k}, \omega_{t+1})] + \bar{\pi}_{t+1,k} E_{t+1} \bar{x}_{tk} + \bar{\phi}_{t+1,k} \bar{u}_{t+1,k}, \end{aligned}$$

and $\bar{x}_{tk}$and $\bar{u}_{t+1,k}$ are the values of the tth-stage variables at which cut k is evaluated.

Since each stage model is a linear program with uncertainty appearing on the right-hand side, we can apply the standard form of SDDP to solve the risk-averse model. Moreover the algorithm satisfies all the conditions in [18], and so it converges almost surely to the optimal policy, under mild conditions on the sampling process (e.g. independence).

One practical difficulty is obtaining reliable estimates of the upper bound on the cost of an optimal policy. The multi-stage setting with CVaR requires a conditional sampling process to estimate the cost of any policy, which would be prohibitively expensive for problems with many stages.

The absence of a good upper-bound estimate makes it difficult to check the convergence of the method. One possible approach is to stop the algorithm if the lower bound has not changed significantly for some iterations, but this does not guarantee that the current policy is close to optimal, even if one is interested only in the first stage action. Our approach is to run the algorithm until the risk-neutral version of the code has converged, and then use the same number of iterations for the risk-averse model.

Observe that the formulation above remains valid if $\lambda_2, \ldots, \lambda_T$ are not deterministic but adapted to the filtration $\{\mathcal{F}_t\}$ (with λ_2 deterministic and λ_{t+1} $\mathcal{F}_t$-measurable). If, in addition, the parameters $\lambda_2, \ldots, \lambda_T$ are stage-wise independent, then the SDDP algorithm can still be applied, but now each stage problem has some randomness appearing in the recourse matrix (in the coefficient of u_{t+1}) and so the almost sure convergence of the method is an open question. In the next section we show how SDDP can be modified to handle a specific form of randomness in λ_{t+1} in which it depends on the state of the Markov chain discussed in Section 2.2.

4.3 Risk aversion with Markov-chain uncertainty

In Section 2.2 we discussed how to integrate a Markov chain model into the SDDP algorithm to solve a risk-neutral problem in which the uncertain data have some stage-wise dependence.

In our risk-averse model, the Markov chain can be implemented in exactly the same way, whereby we calculate one set of cuts for each Markov state, where each cut is an affine function of u and x.

Using Markov states to represent stage-wise dependence in the uncertain parameters provides an opportunity to make the risk measure depend on the state of the Markov chain. This type of model using *Markov risk measures* is explored in a general setting in [21]. In our model the analysis in [21] simplifies, since in our Markov chain model, the actions we take do not have any effect on the (discrete) state of the system, which merely serves to model some stage-wise dependence in the random right-hand sides. As in the previous subsection, these are still adapted to a filtration $\{0, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \ldots \subset \mathcal{F}_T \subset \mathcal{F}$, but this now has a specific form of dependence.

Recall the risk-neutral problems to be solved in the second and later stages t, given Markov state i, previous decision x_{t-1}, and realization ω_{ti} can be written as

$$\begin{array}{rll} Q_{ti}(x_{t-1}, \omega_{ti}) = \min & c_t^\top x_t + \sum_{j=1}^{S} P_{ij}^{(t)} \mathbb{E}[Q_{t+1,j}(x_t, \omega_{t+1,j}) \mid W_{t+1} = j] & \\ \text{s.t.} & A_t x_t = b_t(\omega_{ti}) - E_t x_{t-1}, & [\pi_t(\omega_{ti})] \\ & x_t \geq 0. & \end{array}$$

Let us define

$$P_{ij}^{(t)}(\omega) = P_{ij}^{(t)} \Pr(\omega_{t+1,j} = \omega \mid W_{t+1} = j).$$

At stage t the single-period coherent risk measure we use is

$$\begin{aligned} \rho_{t|i}(Z_{t+1}) &= (1 - \lambda_{t+1}(i)) \sum_{j=1}^{S} \sum_{\omega} P_{ij}^{(t)}(\omega) Z_{t+1,j}(\omega) \\ &\quad + \lambda_{t+1}(i) \inf_{u \in \mathbb{R}} \{u + \frac{1}{\alpha} \sum_{j=1}^{S} \sum_{\omega} P_{ij}^{(t)}(\omega)(Z_{t+1,j}(\omega) - u)_+\}. \end{aligned}$$

Observe that although $\lambda_{t+1}(i)$ is a parameter used to compute the risk of outcomes in stage $t+1$, it is measurable with respect to $\mathcal{F}_t$, because the observed Markov state i (which determines our choice of λ) is measurable with respect to $\mathcal{F}_t$. In other words at stage t we choose the parameter $\lambda_{t+1}(i)$ to weight the expectation and conditional value at risk of outcomes from $t+1$ onwards.

The *single-cut* version of the algorithm with λ_{t+1} depending on Markov state i is misnamed as it now requires S cuts for Markov state i at stage t, one for each possible value of the Markov state in the previous stage. To compute cut $s \in \{1, 2, \ldots, S\}$ in Markov state i

we use parameter $\lambda_t(s)$ and solve

$$\begin{array}{rll}
\tilde{Q}_{tsi}(x_{t-1}, u_t, \omega_{ti}) = \min & (1-\lambda_t(s))(c_t^\top x_t + \lambda_{t+1}(i)u_{t+1} + \theta_{t+1,i}) & \\
& \quad +\lambda_t(s)\alpha^{-1}v_t & \\
\text{s.t.} & A_t x_t = b_t(\omega_{ti}) - E_t x_{t-1}, & [\pi_{ts}(\omega_{ti})] \\
& v_t - (c_t^\top x_t + \lambda_{t+1}(i)u_{t+1} + \theta_{t+1}) \geq -u_t, & [\phi_{ts}(\omega_{ti})] \\
& \theta_{t+1} + \sum_{j=1}^S P_{ij}^{(t)} \bar{\pi}_{t+1,i,j,k}^\top E_{t+1} x_t & \\
& \quad + \sum_{j=1}^S P_{ij}^{(t)} \bar{\phi}_{t+1,i,j,k} u_{t+1} \geq \sum_{j=1}^S P_{ij}^{(t)} \bar{h}_{t+1,i,j,k}, & k = 1, 2, \ldots, K_{t+1}, \\
& x_t \geq 0, \qquad v_t \geq 0. &
\end{array}$$

where at the kth iteration

$$\begin{array}{rcl}
\bar{\pi}_{t+1,i,j,k} & = & \mathbb{E}[\pi_{t+1,i,}(\omega_{t+1,j}) \mid W_t = i, W_{t+1} = j], \\
\bar{\phi}_{t+1,i,j,k} & = & \mathbb{E}[\phi_{t+1,i,}(\omega_{t+1,j}) \mid W_t = i, W_{t+1} = j], \\
\bar{h}_{t+1,i,j,k} & = & \mathbb{E}[Q_{t+1,i,,j}(\bar{x}_t^k, \bar{u}_{t+1}^k, \omega_{t+1,j}) \mid W_t = i, W_{t+1} = j] \\
& & +\bar{\pi}_{t+1,i,j,k}^\top E_{t+1} \bar{x}_t^k + \bar{\phi}_{t+1,i,k} \bar{u}_{t+1}^k
\end{array}$$

Here $\bar{x}_t^k$ and $\bar{u}_{t+1}^k$ are the values of x_t and u_{t+1} obtained at the kth forward pass of SDDP (assuming that $N = 1$). The cuts in Markov state i at stage t that correspond to the previous state s are only valid for realizations of the Markov chain that visit s at stage $t-1$. They could be computed along with the $S-1$ cuts for the other states whenever the Markov chain visits state i in stage t. Alternatively we can add the cut corresponding to s one at a time to those stored for Markov state i whenever a forward pass of the algorithm visits state s immediately before i.

As an illustration of the model with λ dependence, consider the example presented in Section 2.2, and let us assume that one would like to be less risk averse when the system is in Markov state 1. For example, we might choose $\lambda_{t+1} = 0.25$, when the realization at stage t belongs to state 1 and $\lambda_{t+1} = 0.75$, when the realization at stage t belongs to state 2. Figure 4 shows the scenario tree with the Markov Chain and the λ_{t+1} value shown for each Markov state at stage $t+1$, assuming that we start in a state 2 realization.

This figure shows that since the system is in Markov state 2 at stage 1, we have $\lambda_2 = 0.75$ in both states at stage 2. We do not consider outcomes for the case where the system is in Markov state 1 at stage 1. When the system is in Markov state 1 at stage 2, then we have $\lambda_2 = 0.25$ in all states at stage 3, that have come from state 1 at stage 2. When the system is in Markov state 2 at stage 2, then we have $\lambda_2 = 0.75$ in all states at stage 3, that have come from state 2 at stage 2.

The fact that λ_{t+1} depends on the state in stage t makes cut sharing more complicated. As discussed in Section 2.2, when λ_{t+1} does not depend on the observed state in stage t, the set of possible realizations and the formulation are the same for all states. Therefore, the stage $t+1$ solutions in the backward pass are used to generate one cut for each state in stage t by using the appropriate transition probabilities. On the other hand, when λ_{t+1} depends on the observed state in stage t, the formulation of problem (17) will depend on

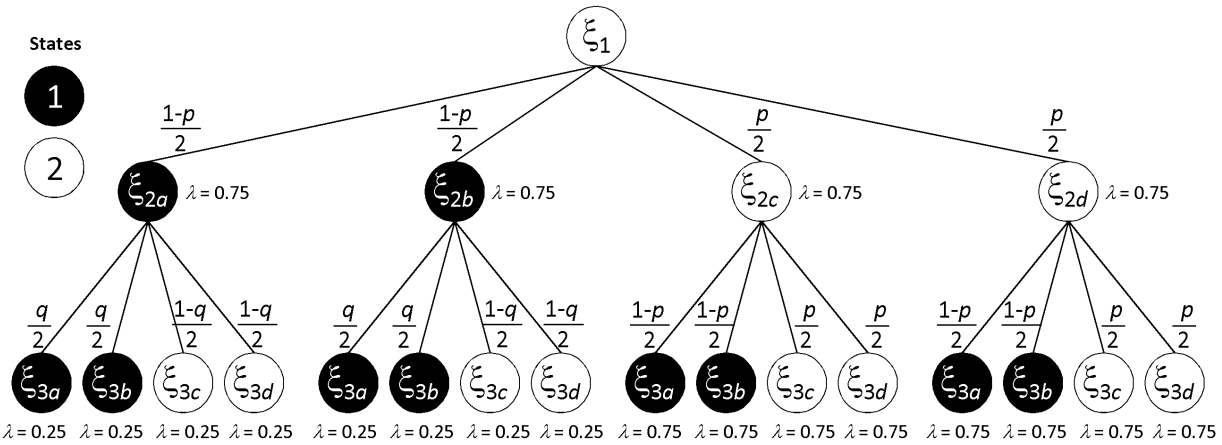

Figure 4: Risk-averse Markov process tree with risk aversion dependent on state in previous stage.

the observed state in stage t because of λ_{t+1}. This dependence means that we compute cuts only for the Markov states that are visited in the sampled scenarios from the forward pass, which means that the cut calculated in a specific scenario can only be added to the observed state. As a consequence, assuming that λ_{t+1} depends on the observed state in stage t incurs some additional computational cost when compared to the independent λ_{t+1}, as S times as many forward simulations will be needed to generate cuts for the same number of states as before. This effect can be seen in the computational experiments we describe in section 6.

The effectiveness of this model at controlling risk relies on a Markov chain that has a certain amount of state persistence, so a realization of the expensive state in stage t is likely to persist into stage $t + 1$. In the case of the previous example, a persistent model would have $p > 0.5$ and $q > 0.5$. If, on the other hand, the process is stage-wise independent, then this choice of λ_{t+1} would not make sense as there would be no reason to change our risk attitude for outcomes that occur at stage $t + 1$, based on the realization of the Markov state at stage t.

5 Application: long-term hydrothermal scheduling

In this section we describe the application of the risk-averse SDDP algorithm to a hydro-electric scheduling model developed for the New Zealand electricity system. The model consists of 33 hydro plants (5400 MW) and 12 thermal plants (2800 MW). We use a simplified transmission network $\mathcal{N}$ comprising three buses: one for the South Island (South), one for the lower North Island (Hay) and one for the upper North Island (North) as shown in Figure 5. We model storage in eight hydro reservoirs in the South Island, and a single reservoir at the head of the Waikato chain of eight stations in the North Island. All other hydro stations

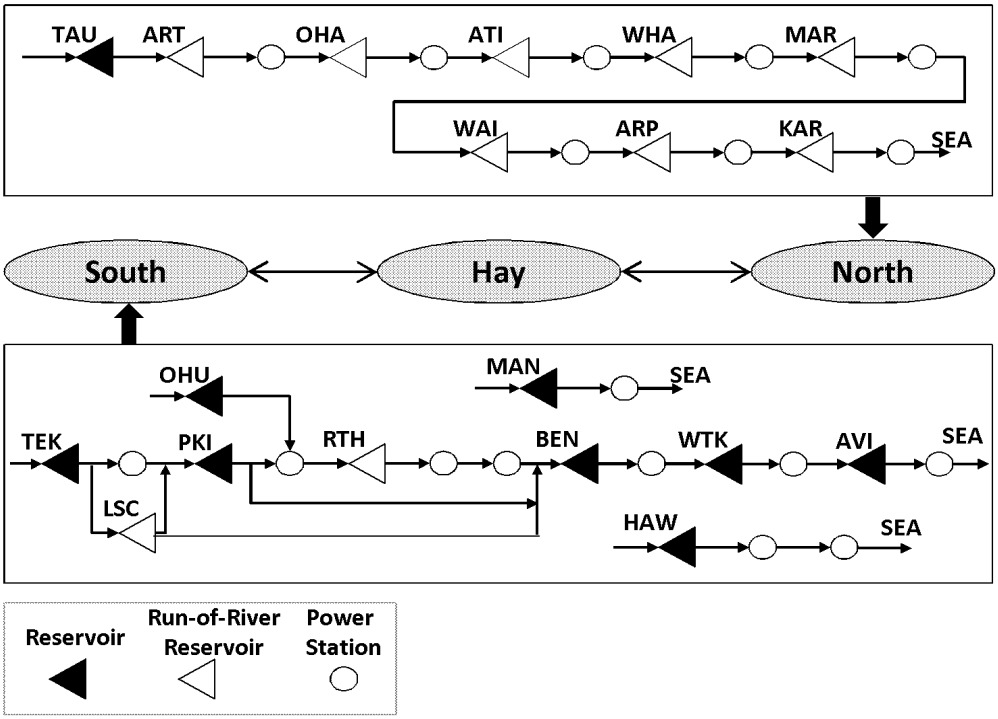

Figure 5: Representation of the New Zealand hydro-thermal scheduling model.

are assumed to be run-of-river. All thermal plants are located in the upper North Island.

The formulation we solve is a stochastic dynamic programming problem in which at each stage $t = 1, 2, \ldots, T$, the Bellman equation is approximated by a linear program. We first describe the general model (18) and then describe how the data specializes to the particular instance we solve[1]. For simplicity, the description shown in this section is for the risk-neutral problem with a single Markov state.

The objective of the model is to minimize the cost to meet the demand D_{it} in stage t at each bus $i \in \mathcal{N}$ plus the expected cost $\mathbb{E}[Q_{t+1}(v_t, \omega_{t+1})]$ over periods $t+1, t+2, \ldots, T$. Here v_t is the vector of reservoir storage levels at the end of period t, where initial storage v_0 is given, and ω_{t+1} denotes the random outcome (scenario) in period $t+1$. Observe that the first stage is deterministic and so ω_1 is known. We discriminate between thermal generation f_{pt}, at thermal plant $p \in \mathcal{T}(i)$ (that has capacity a_p and incurs a fuel cost η_p), and hydro generation $\gamma_m h_{mt}$, at hydro station $m \in \mathcal{H}(i)$ (that has capacity b_m, and is essentially free). The parameter γ_m, which varies by generating station m, converts the flow of water h_{mt} into electric power. We assume that load shedding is modelled as (dummy) thermal generators with higher marginal costs than the most expensive thermal unit. This gives the following

[1]Details of the New Zealand hydro-thermal system used in this model can be found at http://www.epoc.org.nz.

formulation at stage t:

$$
\begin{aligned}
&Q_t(v_{t-1}, \omega_t) = \min \left(\textstyle\sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{T}(i)} \eta_p f_{pt} + \mathbb{E}[Q_{t+1}(v_t, \omega_{t+1})] \right) \\
&\text{s.t.} \\
&w_i(y_t) + \textstyle\sum_{p \in \mathcal{T}(i)} f_{pt} + \sum_{m \in \mathcal{H}(i)} \gamma_m h_{mt} \geq D_{it}, i \in \mathcal{N}, \\
&v_t = v_{t-1} - A(h_t + s_t) + \xi_t(\omega_t), \\
&0 \leq f_{pt} \leq a_p, \quad p \in \mathcal{T}(i),\, i \in \mathcal{N}, \\
&0 \leq h_{mt} \leq b_m, \quad 0 \leq s_{mt} \leq c_m, \quad m \in \mathcal{H}(i), \\
&0 \leq v_{mt} \leq r_m, \quad m \in \mathcal{H}(i),\, i \in \mathcal{N}, \quad y \in Y.
\end{aligned}
\tag{18}
$$

The components of the vector y measure the flow of power in each transmission line. We denote the flow in the directed line from i to k by y_{ik}, where by convention we assume $i < k$. A negative value of y_{ik} denotes flow in the direction from k to i. In general we require that this vector lies in some convex set Y, which may model DC-load flow constraints arising from Kirchhoff's laws and thermal flow limits. The function $w_i(y)$ defines the amount of power arriving at node i for a given choice of y. In many models this is chosen to be concave, allowing one to model increasing marginal line losses. In the New Zealand model with three buses there are no loops, so Y represents line capacities only. We also assume that there are no line losses which gives

$$w_i(y) = \sum_{k<i} y_{ki} - \sum_{k>i} y_{ik}.$$

The water balance constraints are represented by

$$v_t = v_{t-1} - A(h_t + s_t) + \xi_t(\omega_t)$$

where s_t denotes spill in period t, and $\xi_t(\omega_t)$ is the uncontrolled inflow into the reservoir in period t. Storage, release and spill variables are subject to capacity constraints. The node-arc incidence matrix A represents all river-valley networks, and aggregates controlled flows that leave from each reservoir by spilling or generating or enter a reservoir by spilling or generating electricity upstream. In other words row i of $A(h_t + s_t)$ gives the total controlled flow out of the reservoir (or river junction) represented by row i, this being the sum of any release and spill of reservoir i minus the release and spill of any immediately upstream reservoir.

The time horizon of our model is one year with weekly time steps, so $T = 52$. Each weekly stage problem (without cuts) has a total of 476 constraints and 540 variables.We are using data from the calendar year 2006 and assume that in each stage the set of possible inflows is given by the historical inflows from 1987 to 2006, inclusive. As a consequence,

Markov State	North Island	South Island
WW	Wet	Wet
DW	Dry	Wet
WD	Wet	Dry
DD	Dry	Dry

Table 1: Markov states for the New Zealand LTHS problem.

in our problem we have a scenario tree that has 20 random realizations per stage (called *openings*) and 52 stages given a total of more than 2.25×10^{66} scenarios.

The inflows were modelled by estimating transitions between four states of a Markov chain as follows. First, historical inflows were aggregated into two groups corresponding to the South Island and North Island. We classified two possible states (wet and dry[2]) for each island, to give a total of four states as shown in Table 1. After grouping the outcomes into four sets corresponding to each state, the transition probabilities are estimated from the historical inflow sequence from 1987 to 2006, inclusive.

An inflow sequence is simulated by constructing a random sequence of 52 states, and then randomly sampling a weekly inflow record from the group of historical outcomes representing the simulated state in each week. To test the performance of candidate policies on a common benchmark, we assume throughout this paper that the Markov chain we construct in this way represents the true stochastic process of inflows. (It is certainly interesting to test how different approximations of the real inflow process affect the policies being computed, but we see this as a different modelling exercise to be explored in a separate study.)

As mentioned above, we assume that nine reservoirs (with a total capacity of 7.5 billion cubic metres) can store water from week to week, and the remaining reservoirs are treated as run-of-river plant with limited intra-week flexibility. In some cases we also have minimum or maximum flow constraints that are imposed by environmental resource consents. When this is the case total discharge limits are added to the model, and deviations of flows outside these limits are penalized in the objective function. Weekly demand is aggregated from historical records, and represented by a load duration curve with three blocks representing peak, off-peak, and shoulder periods.

6 Computational experiments

In this section we present the results of computational experiments on the New Zealand hydro-thermal scheduling model to evaluate the performance of the SDDP algorithms discussed throughout this paper. Our SDDP code, which applies the single-cut version of the algorithm (as described in section 4), was implemented in Microsoft Visual C++ Version 9.0

[2]A historical outcome is considered to be a dry state if the sum of all inflows in the island is smaller than the historical average of the sum of all inflows. Otherwise, it is considered to be wet.

using the CPLEX 11.2 callable library. As alluded to above, we always choose $N = 1$ in our implementation (i.e. we use only one scenario in each forward pass). For the New Zealand problem, this gives better policies than the standard choice ($N = 200$) when the algorithm is terminated early (see [3]).

In all experiments where λ_t is independent of the state at stage $t-1$, we choose λ_t to be a fixed constant L for all stages t. Additionally, we assume $\lambda = 0$ for all cases in stage 1 and $T+1$. We denote by "4 Ls" the model where λ_t can take different values depending on which of the four inflow states is realized in the previous stage. We use $L = 0$, 0.5, and 0.9 to represent risk neutrality, mild risk aversion, and strong risk aversion respectively. This gives seven models as shown in Table 2.

Case	λ_t	Markov Chain
L = 0	0	no
L = 0.5	0.5	no
L = 0.9	0.9	no
L = 0 (M)	0	yes
L = 0.5 (M)	0.5	yes
L = 0.9 (M)	0.9	yes
4 Ls	0, 0.5, 0.9	yes

Table 2: Cases.

In the first three models we construct a candidate policy using SDDP under the assumption that inflows are stage-wise independent. In the second set of three models (denoted "(M)") the policy is constructed using four Markov states. In the case "4 Ls", λ_t takes three values as shown in Table 3. In all cases we use the Markov chain inflow model to simulate

State in $t-1$	λ_t
WW	0
DW	0.5
WD	0.5
DD	0.9

Table 3: Lambda for each Markov state in case 4 Ls.

the performance of the policy.

To validate our code, we first applied the seven risk settings to a model with $T = 4$ and 4 openings, giving a scenario tree with 64 scenarios. This allowed us to compare the lower bounds obtained with SDDP and the solution obtained from solving the deterministic equivalent linear program directly using CPLEX 11.2. The results, shown in Table 4, indicate that the algorithm performs as expected on small problems.

In our experiments we also choose to run the SDDP algorithm for a fixed number of iterations, rather than using the standard stopping criterion [17] described in section 2. The standard test can be misleading (see e.g. [7],[23]) and lead to premature termination of

Case	DLP Optimal Solution(10^6 \$)	Lower Bound (10^6 \$)
L = 0	6.5689816843	6.568981684267
L = 0.5	11.950358776	11.950358775737
L = 0.9	16.683566128	16.683566127608
L = 0 (M)	6.5604147645	6.560414764476
L = 0.5 (M)	11.937104988	11.937104987936
L = 0.9 (M)	16.708486732	16.708486731945
4 Ls	14.768832334	14.768832334269

Table 4: Validation of the implementation.

SDDP. For each experiment we run SDDP for a maximum of 4000 iterations, except for the case "4 Ls" where it is run for 16000 iterations so that each Markov state has approximately 4000 cuts. As discussed above, in the case "4 Ls", when λ_{t+1} depends on the Markov state in stage t, in the backward pass we compute cuts only for the Markov states at stage $t+1$ that have come from this state. Thus we require more iterations to give a similar number of cuts in each stage as we obtain in the other cases. All tests were run on a 3.0 GHz Core 2 Extreme computer (model QX9650) with 4 cores, 4GB RAM and a 12MB cache. The computation times in seconds are shown in Table 5.

Case	Time(s)
L=0	30737
L=0.5	30348
L=0.9	33877
L=0 (M)	28784
L=0.5 (M)	52376
L=0.9 (M)	57355
4 Ls	129105

Table 5: Computation times in seconds for each case.

Figure 6 depicts the lower bound for the first 16000 iterations of the case "4 Ls", and the first 4000 iterations of all the other cases. This indicates that assuming convergence after 4000 cuts per Markov state seems to be reasonable.

In order to compare the policies obtained in each case, we sampled 4000 inflow scenarios using the Markov chain, and tested the policies in each scenario. The results are presented in Table 6. Here the expected total operating cost measures fuel cost, shortage cost, and penalties for violating river flow constraints, in other words the risk-neutral objective function. Since the risk-neutral Markov chain model produces the smallest expected total operation cost, we list the increases in expected total operating cost of the other policies in the third column.

These numbers show several things. First of all, there is a small increase in expected cost (\$1.27M) from assuming stage-wise independent inflows, when they are really Markovian.

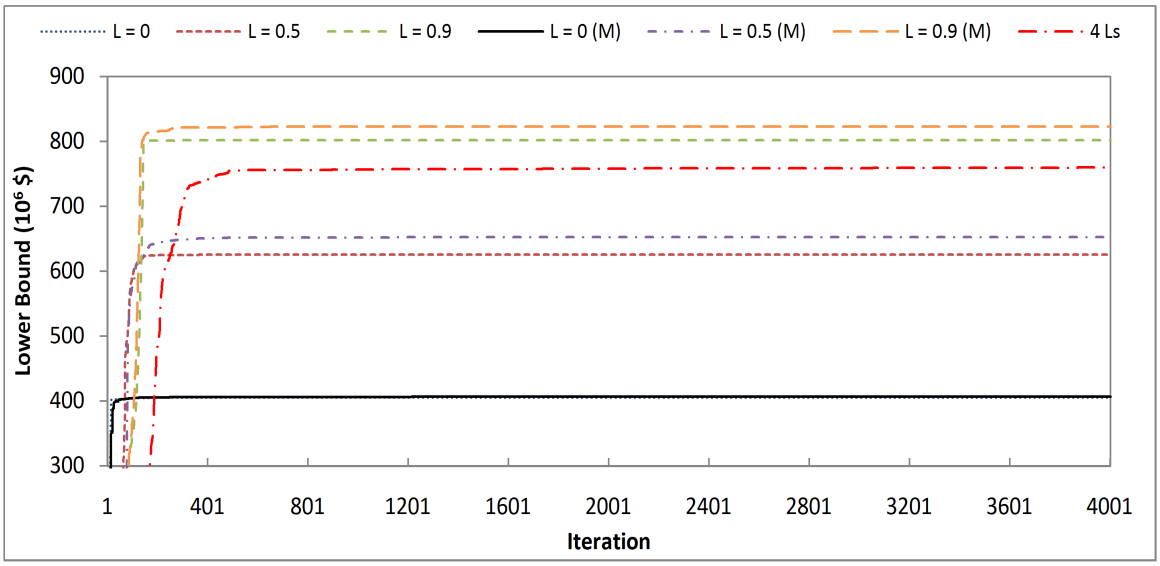

Figure 6: Lower Bound for all cases. The scale on the horizontal axis indicates the average number of cuts per Markov state per stage, so the curve for the "4L's" model corresponds to 16000 iterations.

Case	Expected Total Operation Cost(10^6 \$)	Increase in Cost from Case 4 (10^6 \$)
L = 0	407.071	1.272
L = 0.5	409.991	4.193
L = 0.9	418.557	12.759
L = 0 (M)	405.798	-
L = 0.5 (M)	409.538	3.740
L = 0.9 (M)	418.833	13.035
4 Ls	418.618	12.820

Table 6: Expected total operation cost.

The SDDP model that assumes Markovian inflows should perform better in a benchmark simulation of these inflow sequences than a model that ignores this feature. The numbers also show the tradeoffs in expected cost that are incurred by an increase in risk aversion. As a percentage of the total cost, these numbers appear to be quite modest.

To investigate the potential benefits of the risk-averse models, we focus on some of the extreme scenarios that give large costs for the risk-neutral model. Table 7 gives the total operation cost in the worst case scenario for each policy. Case L = 0 (which is risk neutral

Case	Total Operation Cost(10^6 \$)
L = 0	823.822
L = 0.5	643.570
L = 0.9	630.791
L = 0 (M)	631.020
L = 0.5 (M)	627.750
L = 0.9 (M)	628.265
4 Ls	628.273

Table 7: Most expensive scenario.

and assumes independence) is substantially more expensive. It is interesting to see that a policy computed using Markov states, performs creditably in this worst case.

We can extend this examination to the 200 worst scenarios in each case represented by the distributions of total cost shown in Figure 7. These plots represent the cumulative distribution functions of total cost under each policy, where the scenario counter references different scenarios for each plot, so they should not be interpreted as one policy uniformly dominating another over all scenarios. Nevertheless it is easy to see that the risk-neutral stage-wise independent model is stochastically dominated to first order by the other policies, at least over the 200 worst outcomes. The other policies seem to produce comparable distributions of total cost.

It is worth remarking that our multi-period risk measure ρ is not designed to control for total cost incurred over the year, and so the plots we show of total cost outcomes might mislead us. The exact interpretation of ρ (with $\lambda > 0$) is less obvious, but at any stage it controls high values of cost in the future, so the policy focuses on avoiding high future costs that might be incurred in the next few periods, whether they come from imminent shortages or the accumulated risk of a future shortage or constraint violation. This is arguably different from focusing on the distribution of total annual cost, and controlling the extent of its upper tail.

In addition to the most expensive cases of total cost we can examine the least expensive cases (corresponding to high inflows). Figure 8 depicts the least expensive total cost scenarios for each policy, in which one can notice that more risk averse policies incur higher costs when inflows are plentiful. This is reflected in Figure 9, which shows the expected national reservoir levels in terms of stored energy. Here one can observe that the risk averse policies save more water in the first half of the year to protect against low inflows later in the year. As a

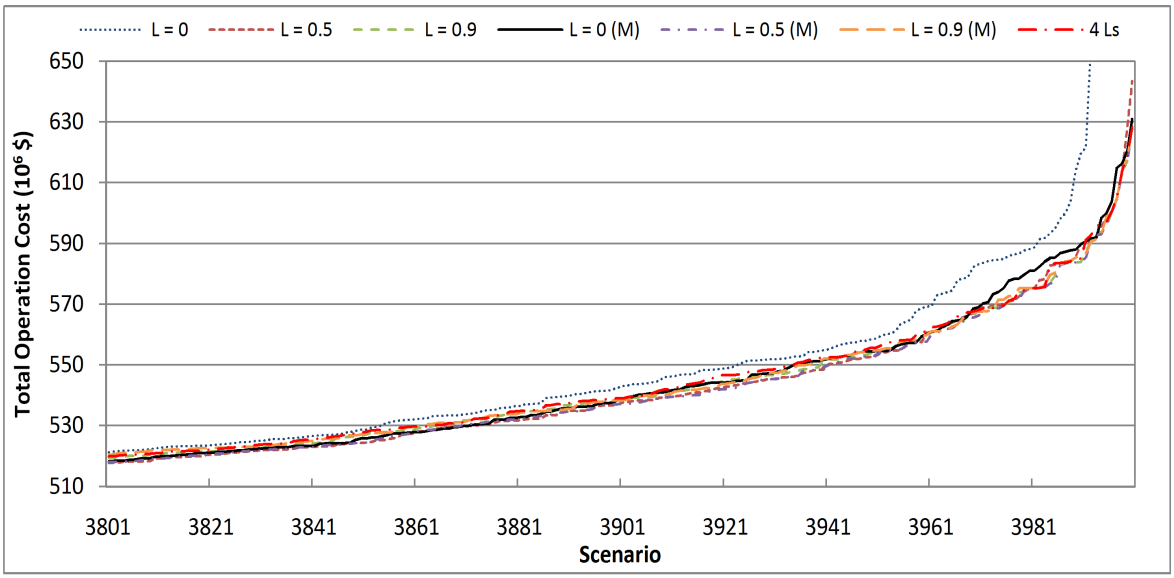

Figure 7: Most expensive scenarios in the simulation.

consequence, the risk averse models tend to use more thermal generation in the first half of the year when it might not be necessary and finish the year with reservoirs at higher levels than their risk-neutral counterparts.

Many formulations of the long-term hydro scheduling problem used in practice include a reservoir danger zone or *minzone* as part of the model. This is a trajectory of (possibly aggregate) reservoir storage that represents a minimum level for security of supply. Such a zone can be computed using simulation over future inflow sequences, and choosing a reservoir level that avoids shortages in a given high percentage of these sequences, assuming full commitment of all thermal plant. Once a minzone has been determined, penalty terms can be added to SDDP to discourage policies from allowing reservoir storage to enter the minzone.

Since we have chosen to ignore the New Zealand minzone in our model, it is interesting to see to what extent our model of risk aversion can prevent trajectories from passing into it. Figure 10 shows the number of scenarios that the national storage level was below the minzone in each stage. Both risk-neutral models were unable to avoid breaching the minzone in a large number of outcomes. The risk-averse models were more conservative, with only the $L = 0.5$ independent model showing some violation of the minzone.

We finish this section by studying the effect of risk aversion on load shedding (disconnecting consumers in times of shortage). Table 8 presents the probability of load shedding for each model, and its expected contribution to the cost. Both of these are very low in the risk-neutral cases, and zero for all risk averse cases, although these figures are computed assuming a Markov inflow model, and may be substantially higher if inflows have a higher degree of stage-wise dependence.

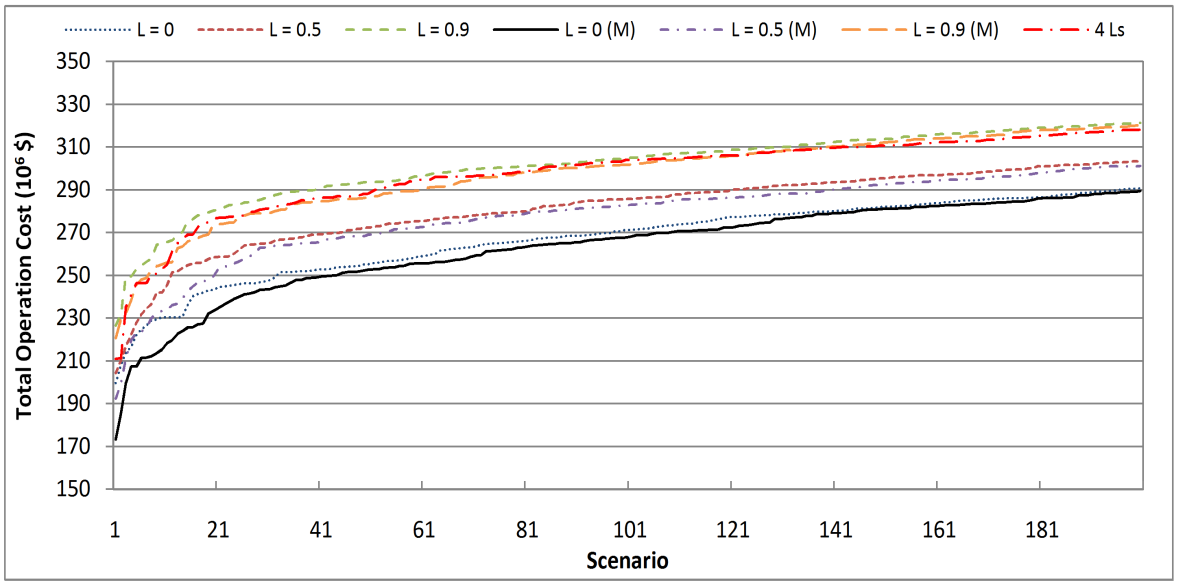

Figure 8: Least expensive scenarios in the simulation.

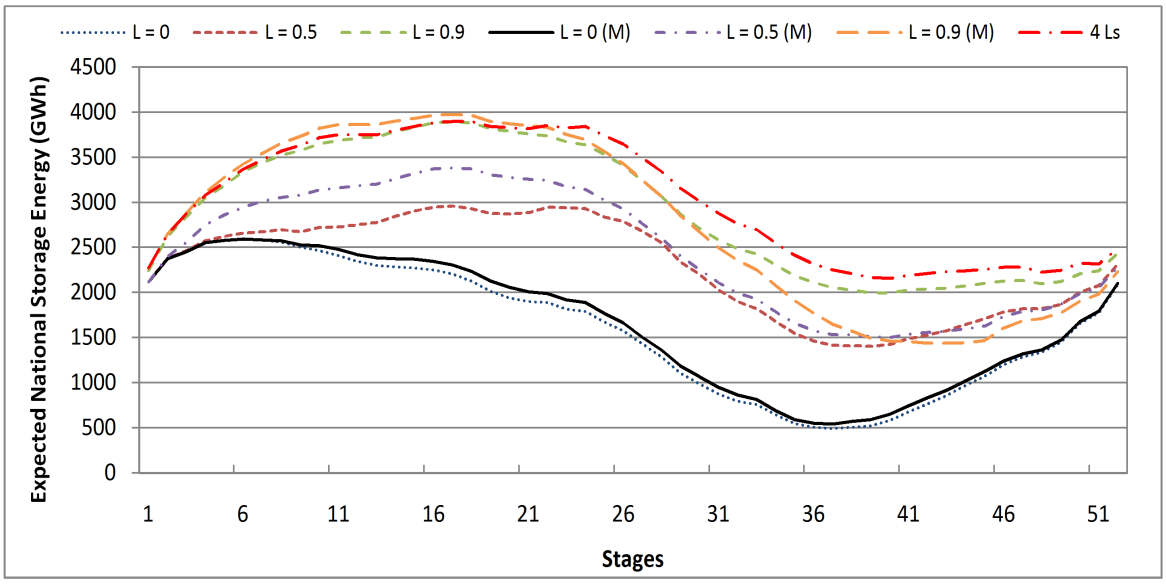

Figure 9: Expected national storage level in terms of energy.

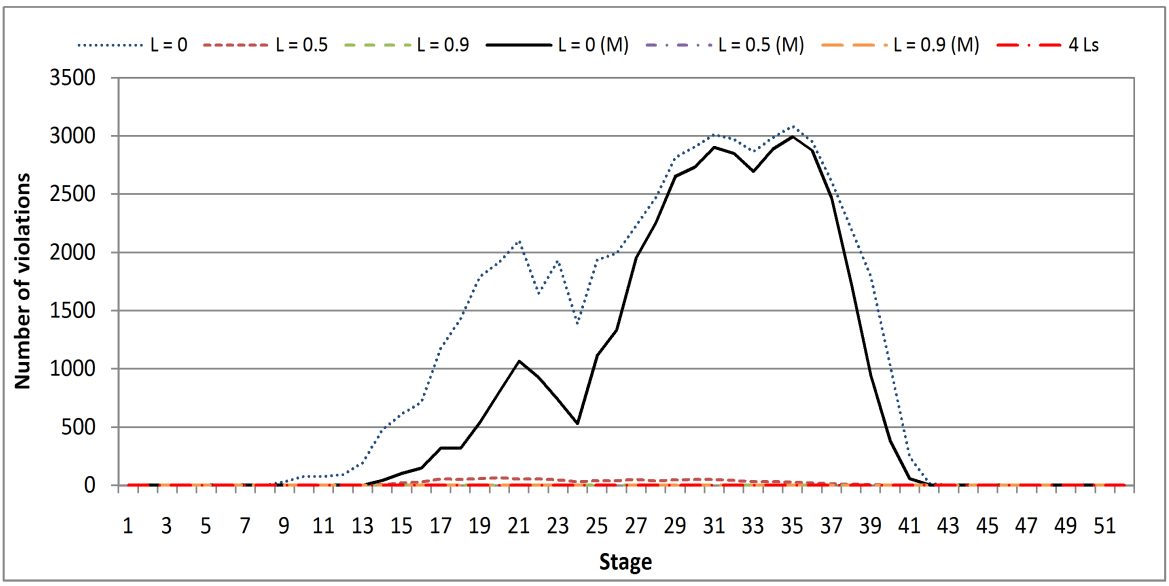

Figure 10: Number of scenarios where the national storage level was below the minimum level.

Case	Risk of Load Shedding (%)	Expected Cost of Load Shedding (10^6 \$)
L = 0	0.085	0.421
L = 0.5	0	0
L = 0.9	0	0
L = 0 (M)	0.025	0.0005
L = 0.5 (M)	0	0
L = 0.9 (M)	0	0
4 Ls	0	0

Table 8: Risk and Expected cost of Load Shedding.

7 Conclusions

It appears from our limited experiments that risk aversion can be incorporated into multi-stage models with surprising ease. Given an appropriate level of risk aversion it is possible to reduce the probability of bad outcomes with only mild degradation in overall cost. In our experiments we chose λ_t to take the same value L throughout the year. Varying λ_t throughout the year to be low at times when inflows and reservoir levels are typically high, and high otherwise is likely to yield improvements in performance, but we have not attempted this here, as these choices are problem dependent, and settings obtained for the New Zealand system would be unlikely to apply generally.

The risk-averse methodology we have described provides a promising alternative to the use of minzones for controlling risk. In SDDP it is impossible to impose minzones as hard constraints, because that will violate the assumption of relatively complete recourse. For this reason soft constraints with violation penalties are preferred. However it is often not obvious what penalties one should place on minzone violations to give appropriate safeguards. When a real constraint is actually violated (for example when load must be shed) the costs incurred are also real, rather than penalties used to control risk, and so they are easier to justify even though they may be challenging to estimate. Moreover, since each stage problem can be a relatively large linear program, naive choices of these penalties can give counterintuitive results. For example, one would not want to shed load in order to meet a minzone constraint, which might happen with a poor choice of penalties. Our results show that with appropriate choices of λ_t, a dynamic risk measure can meet constraints with very high probability with a modest increase in expected operation cost.

A potential weakness with our approach is the difficulty in estimating the value of a candidate policy. This value can be interpreted as an equivalent payment we would make in any state to avoid incurring future costs. However, even without this estimate, we can still estimate parameters for the distributions of parameters of interest. We see that risk-averse models are performing as expected, by saving more water in the reservoirs, reducing the costs in the most expensive scenarios and reducing the risk of load shedding. As expected, the system has to pay a price for the risk aversion, which is a function of the risk aversion level.

Acknowledgement

The authors would like to thank the anonymous referees for their helpful suggestions that have improved the presentation of this paper.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999.

[2] J.R. Birge. Decomposition and partitioning methods for multi-stage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.

[3] V. De Matos, A.B. Philpott, E.C. Finardi, and Z. Guan. Solving long-term hydro-thermal scheduling problems. Technical report, Electric Power Optimization Centre, University of Auckland, 2010.

[4] V. Guigues and W. Romisch. Sampling-based decomposition methods for risk-averse multistage stochastic programs. *Optimization Online*, 2010.

[5] V. Guigues and C. Sagastizabal. The value of rolling horizon policies for risk-averse hydro-thermal planning. *Optimization Online*, 2010.

[6] D. Harte, M. Pickup, and P. Thomson. Hidden Markov models for New Zealand hydro catchment inflows: a preliminary analysis. Technical report, Electricity Commission of New Zealand, 2007.

[7] T. Homem-de Mello, V. De Matos, and E.C. Finardi. Quasi-Monte Carlo methods for a multi-stage stochastic programming model in the long-term hydrothermal scheduling. Technical report, Laboratório de Planejamento de Sistemas de Energia Elétrica, Universidade Federal de Santa Catarina, 2010.

[8] N.A. Iliadis, M.V.F. Pereira, S. Granville, R.M. Chabar, and L.-A. Barroso. Portfolio optimization of hydroelectric assets subject to financial indicators. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –8, June 2007.

[9] G. Infanger and D.P. Morton. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.

[10] J. Jacobs, G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus, and J. Stedinger. SOCRATES: A system for scheduling hydroelectric generation under uncertainty. *Annals of Operations Research*, 59:99–133, 1995.

[11] J. Linderoth, A. Shapiro, and S. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142:215–241, 2006.

[12] J.D.C. Little. The use of storage water in a hydroelectric system. *J. of the Operations Research Society of America*, 3(2):187–197, 1955.

[13] M. E. P. Maceira and J. M. Damázio. Use of the PAR(p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the Brazilian hydropower system. *Probab. Eng. Inf. Sci.*, 20(1):143–156, 2006.

[14] P. Masse. *Les réserves et la régulation de l'avenir dans la vie économique*. Hermann, Paris, 1946.

[15] N. Miller and A. Ruszczyński. Risk-averse two-stage stochastic linear programming: modeling and decomposition. *Operations Research*, opre.1100.0847, 2010.

[16] B. Mo, A. Gjelsvik, and A. Grundt. Integrated risk management of hydro power scheduling and contract management. *IEEE Transactions on Power Systems*, 16(2):216 –221, May 2001.

[17] M. V. F. Pereira and L. M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.

[18] A.B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and other methods. *Operations Research Letters*, 36:450–455, 2008.

[19] R.T. Rockafellar and S. Uryasev. Optimization of conditional value at risk. *The Journal of Risk*, 2:21–41, 2000.

[20] R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471, 2002.

[21] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125:235–261, 2010.

[22] A. Shapiro. On complexity of multistage stochastic programs. *Operations Research Letters*, 34(1):1 – 8, 2006.

[23] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.

[24] A. Turgeon. Optimal operation of multireservoir power systems with stochastic inflows. *Water Resources Research*, 16(2):275–283, 1980.