SDDP with stagewise-dependent objective coefficient uncertainty

A. Downward, O. Dowson and R. Baucke

Department of Engineering Science
University of Auckland

ISMP 2018, Bordeaux
July 4, 2018

Outline

Outline

Background

Motivation

Example: hydroelectric generator

Price-state interpolation

- Static interpolation
- Dynamic interpolation

Example: hydroelectric generator

- Price-taking generator
- Price-making generator

Remarks

Background

Multistage stochastic programming

Solution techniques for multistage stochastic programming problems are an active area of research. However, many techniques are some extension of Benders decomposition.

For convex problems, *stochastic dual dynamic programming* (`SDDP`) is the most well known algorithm. However, there are many variants on this: e.g. `DOASA`, `CUPPS` etc.

Background

Traditional SDDP assumptions

The following assumptions are typical of traditional SDDP implementations.

- Finite number of stages, $t \in \{1, \ldots, T\}$, with a terminal cost-to-go.

Background

Traditional SDDP assumptions

The following assumptions are typical of traditional SDDP implementations.

- Finite number of stages, $t \in \{1, \ldots, T\}$, with a terminal cost-to-go.
- Stage-problem is a linear program.

Background

Traditional SDDP assumptions

The following assumptions are typical of traditional SDDP implementations.

- Finite number of stages, $t \in \{1, \ldots, T\}$, with a terminal cost-to-go.
- Stage-problem is a linear program.
- Optimal objective function of stage-problem is convex with respect to some state-vector x.

Background

Traditional SDDP assumptions

The following assumptions are typical of traditional SDDP implementations.

- Finite number of stages, $t \in \{1, \ldots, T\}$, with a terminal cost-to-go.
- Stage-problem is a linear program.
- Optimal objective function of stage-problem is convex with respect to some state-vector x.
- Noise $\omega \in \Omega_t$ is discrete and stagewise independent.

Background

Traditional stage-problem

In stage t, we solve:

$$\begin{aligned} V_t\left(x_{t-1}, \omega\right) = \min_{x_t} \quad & c_t^{\omega\top} x_t + \mathbb{E}_{\omega' \in \Omega_{t+1}}[V_{t+1}(x_t, \omega')] \\ \text{s.t.} \quad & A_t^{\omega} x_t - x_{t-1} = a_t^{\omega} \\ & x_t \geq 0, \end{aligned}$$

where x_{t-1} is the incoming state variable for stage t, and x_t is the outgoing state variable (in our general formulation the vector of states also includes controls).

Background

Traditional SDDP algorithm

SDDP typically consists of a *forward pass*. In our case, we will use an approximate cost-to-go to find a sequence of states:

$$\hat{x}_t^k, \quad \forall t \in \{1, \ldots, T\}.$$

Background

Traditional SDDP algorithm

SDDP typically consists of a *forward pass*. In our case, we will use an approximate cost-to-go to find a sequence of states:

$$\hat{x}_t^k, \quad \forall t \in \{1, \ldots, T\}.$$

At the end of the forward pass, a backward pass generates cuts based on the marginal future cost of changes to the outgoing state variables.

$$\mathcal{V}_{t+1}(x_t) \approx \theta_{t+1} \geq \theta_{t+1}^{k\omega} + \beta_{t+1}^{k\omega}(x_t - \hat{x}_t^k), \ \forall \omega \in \Omega_{t+1}, \ \forall t \in \{1, \ldots, T-1\}.$$

Background

Traditional SDDP algorithm

SDDP typically consists of a *forward pass*. In our case, we will use an approximate cost-to-go to find a sequence of states:

$$\hat{x}_t^k, \quad \forall t \in \{1, \ldots, T\}.$$

At the end of the forward pass, a backward pass generates cuts based on the marginal future cost of changes to the outgoing state variables.

$$\mathcal{V}_{t+1}(x_t) \approx \theta_{t+1} \geq \theta_{t+1}^{k\omega} + \beta_{t+1}^{k\omega}(x_t - \hat{x}_t^k), \ \forall \omega \in \Omega_{t+1}, \ \forall t \in \{1, \ldots, T-1\}.$$

We elect to implement the *average-cut*, rather than *multi-cut*, so we set $\beta_{t+1}^k = \mathbb{E}_{\omega \in \Omega_{t+1}}[\beta_{t+1}^{k\omega}]$, and $\alpha_{t+1}^k = \mathbb{E}_{\omega \in \Omega_{t+1}}[\theta_{t+1}^{k\omega}] - \beta_{t+1}^k \hat{x}_t^k$. This gives the cut:

$$\theta_{t+1} \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Background

Traditional SDDP stage-problem

For stage t, in iteration κ we approximate the stage-problem by:

$$
\begin{aligned}
V_t^\kappa(x_{t-1}, \omega) = \min_{x_t} \quad & c_t^{\omega\top} x_t + \theta_{t+1} \\
\text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\
& x_t \geq 0, \\
& \theta_{t+1} \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t, \quad \forall k \in \{1, \ldots, \kappa\}.
\end{aligned}
$$

Outline

Motivation

Stagewise Dependency (right-hand side)

Recall that one of the assumptions is the noise is stagewise independent. In fact, in the formulation we have allowed for this noise to appear in the objective coefficients, and the constraints (both the right-hand side and coefficients).

However, without altering the algorithm above, we can also model stagewise-dependent uncertainty in the right-hand side. To do this we simply need to store the noise in each stage into a state variable that is passed to the next stage.

The reason this works is that the right-hand side noise and the state variables appear linearly in the constraints.

Motivation

Stagewise Dependency (objective coefficients)

The same approach cannot be taken for stagewise-dependent uncertainty in the objective coefficients. The state variables do not appear linearly with the state variables (in fact, they are multiplied together).

Motivation

Stagewise Dependency (objective coefficients)

The same approach cannot be taken for stagewise-dependent uncertainty in the objective coefficients. The state variables do not appear linearly with the state variables (in fact, they are multiplied together).

The reason this is problematic is that the optimal objective value of convex optimization problems is

- convex for changes in the right-hand side, but

Motivation

Stagewise Dependency (objective coefficients)

The same approach cannot be taken for stagewise-dependent uncertainty in the objective coefficients. The state variables do not appear linearly with the state variables (in fact, they are multiplied together).

The reason this is problematic is that the optimal objective value of convex optimization problems is

- convex for changes in the right-hand side, but
- concave for changes in the objective.

Motivation

Traditional Modelling Choices

So if we wish to model stagewise-dependent objective coefficients as:

- auto-regressive (AR),
- auto-regressive moving average (ARMA), or
- log auto-regressive (log AR)

processes, then what are our options?

[1] Gjelsvik, A., Belsnes, M. & Haugstad, A., 1999. An algorithm for stochastic medium term hydro thermal scheduling under spot price uncertainty. 13$^{\mathrm{th}}$ Power Systems Computation Conference: Proceedings.

Motivation

Traditional Modelling Choices

So if we wish to model stagewise-dependent objective coefficients as:

- auto-regressive (AR),
- auto-regressive moving average (ARMA), or
- log auto-regressive (log AR)

processes, then what are our options?

1. Discretize the process and use a Markov chain

[1] Gjelsvik, A., Belsnes, M. & Haugstad, A., 1999. An algorithm for stochastic medium term hydro thermal scheduling under spot price uncertainty. 13$^{\text{th}}$ Power Systems Computation Conference: Proceedings.

Motivation

Traditional Modelling Choices

So if we wish to model stagewise-dependent objective coefficients as:

- auto-regressive (AR),
- auto-regressive moving average (ARMA), or
- log auto-regressive (log AR)

processes, then what are our options?

1. Discretize the process and use a Markov chain
2. ...?

[1] Gjelsvik, A., Belsnes, M. & Haugstad, A., 1999. An algorithm for stochastic medium term hydro thermal scheduling under spot price uncertainty. 13^{th} Power Systems Computation Conference: Proceedings.

Motivation

Traditional Modelling Choices

So if we wish to model stagewise-dependent objective coefficients as:

- auto-regressive (AR),
- auto-regressive moving average (ARMA), or
- log auto-regressive (log AR)

processes, then what are our options?

1. Discretize the process and use a Markov chain
2. ...?

What if you could interpolate between Markov states?[1]

[1] Gjelsvik, A., Belsnes, M. & Haugstad, A., 1999. An algorithm for stochastic medium term hydro thermal scheduling under spot price uncertainty. 13$^{\mathrm{th}}$ Power Systems Computation Conference: Proceedings.

Objective states

A new kind of state variable

To enable the modelling of objective coefficients using such stochastic processes we create a new type of state variable, denoted y_t. We will refer to these as *price-states*.

Below we show how these price-states evolve over time, and how they can be mapped to the cost coefficients c_t:

$$
\begin{aligned}
y_t &= B_t^{\omega} y_{t-1} + b_t^{\omega}, && \forall t \in \{2, \dots, T\}, \\
c_t &= f_t(y_t), && \forall t \in \{1, \dots, T\},
\end{aligned}
$$

where $f_t(\cdot)$ is a concave function.

Objective states

Updated stage-problem

The stage-problem therefore becomes:

$$\begin{aligned} V_t\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t} \quad & c_t^\top x_t + \mathcal{V}_{t+1}(x_t, y_t) \\ \text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\ & x_t \geq 0, \end{aligned}$$

Objective states

Updated stage-problem

The stage-problem therefore becomes:

$$
\begin{aligned}
V_t\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t} \quad & c_t^\top x_t + \mathcal{V}_{t+1}(x_t, y_t) \\
\text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\
& x_t \geq 0,
\end{aligned}
$$

where

$$
\begin{aligned}
y_t &= B_t^\omega y_{t-1} + b_t^\omega, \\
c_t &= f_t(y_t),
\end{aligned}
$$

Objective states

Updated stage-problem

The stage-problem therefore becomes:

$$\begin{aligned} V_t\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t} \quad & c_t^\top x_t + \mathcal{V}_{t+1}(x_t, y_t) \\ \text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\ & x_t \geq 0, \end{aligned}$$

where

$$\begin{aligned} y_t &= B_t^\omega y_{t-1} + b_t^\omega, \\ c_t &= f_t(y_t), \quad \text{and} \\ \mathcal{V}_{t+1}(x_t, y_t) &= \mathbb{E}_{\omega' \in \Omega_{t+1}}[V_{t+1}(x_t, y_t, \omega')]. \end{aligned}$$

Objective states

First observation

Lemma

If the terminal expected cost-to-go $\mathcal{V}_{T+1}(x_T, y_T)$ *is linear, then* $\mathcal{V}_{t+1}(x_t, y_t)$ *is a saddlefunction, which is convex with respect to* x_t *and concave with respect to* y_t*, for all* $t \in \{1, \ldots, T-1\}$*.*

Proof.

This can be proved inductively using the definition of convexity / concavity; see Downward et al. (2018)[2]. □

[2] Downward, A., Dowson O. & Baucke R., 2018. SDDP with stagewise dependent objective uncertainty. Optimization Online.

Outline

Example: hydroelectric generator

Details and assumptions

We wish to model the production from a single hydroelectric generator with a dedicated dam over a day.

- The hydroelectric dam has a maximum storage level (in MWh).
- There is a turbine with a maximum power output (in MW).
- Any electricity produced is sold on the spot-market, with a spot-price that can be modelled by a log auto-regressive process.
- The inflows each stage are uncertain, and occur (and are observed) at the beginning of each stage.

Example: hydroelectric generator

Stage-problem

$$
\begin{aligned}
V_t\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t, q_t} \quad & c_t q_t + \mathcal{V}_{t+1}(x_t, y_t) \\
\text{s.t.} \quad & x_t = x_{t-1} - q_t + I_t^{\omega} \\
& q_t \in [0, 100] \\
& x_t \in [0, 230],
\end{aligned}
$$

where

$$
\begin{aligned}
y_t &= y_{t-1} + b_t^{\omega}, \text{ and} \\
c_t &= -e^{y_t}.
\end{aligned}
$$

Example: hydroelectric generator

Requirements

We wish to maximize the overall profit (minimize the overall cost), however, because the cost-to-go is a saddlefunction, we need to check that the following conditions are met, if we want to solve this as a convex problem:

1. the price-state process evolves independently of our controls;
2. the price-state transition is linear; and
3. the objective coefficients are concave functions of the price-states.

Outline

Static interpolation

Piecewise linear interpolation

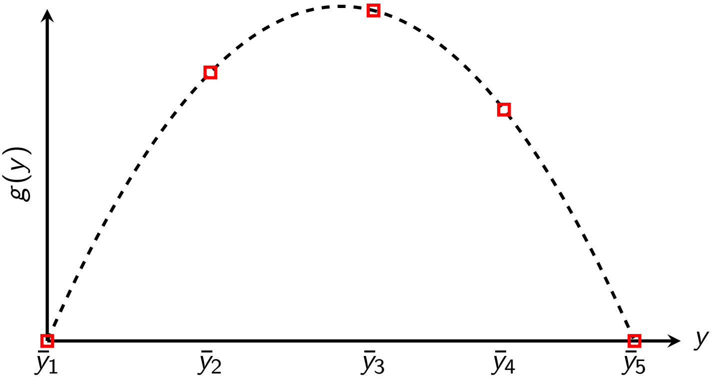


Figure: Piecewise linear interpolation of a one-dimensional function.

Static interpolation

Convex combination

$$
\begin{aligned}
\textbf{LIP:} \quad g(y) = \max_{\gamma} \quad & \sum_{i=1}^{N} \gamma_i g(\bar{y}_i) \\
\text{s.t.} \quad & \sum_{i=1}^{N} \gamma_i = 1 \\
& \sum_{i=1}^{N} \gamma_i \bar{y}_i = y \\
& \gamma_i \geq 0, \;\; i \in \{1, 2, \ldots, N\}.
\end{aligned}
$$

Static interpolation

Lower bound?

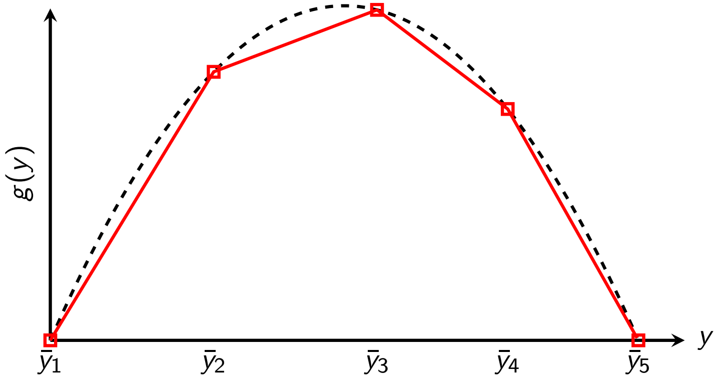
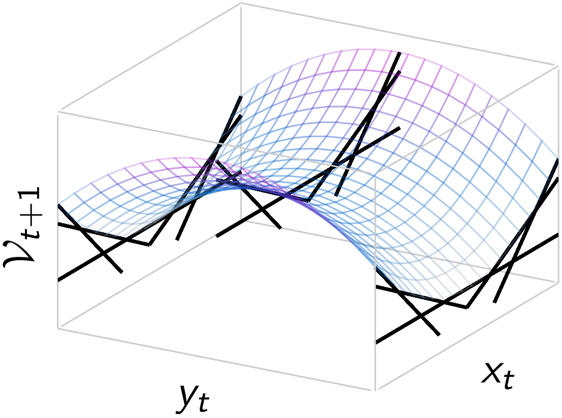


Figure: Piecewise linear interpolation of a one-dimensional function.

Static interpolation

Rib visualization

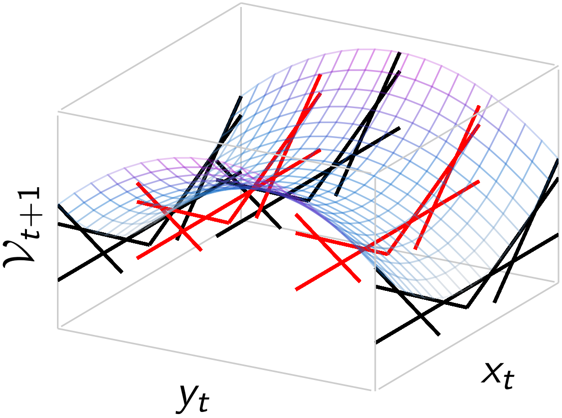
Each cost-to-go variable is a "rib"



Static interpolation

Rib visualization

Each cost-to-go variable is a “rib”



Static interpolation

Static stage-problem

SP-Static$_t^\kappa(x_{t-1}, y_{t-1}, \omega)$:
Calculate $y_t = B_t^\omega y_{t-1} + b_t^\omega$ and $c_t = f_t(y_t)$, compute γ_r, $\forall r \in \mathcal{R}_t$, fix as constants, then solve:

$$
\begin{aligned}
V_t^\kappa(x_{t-1}, y_{t-1}, \omega) = \min_{x_t, \theta} \quad & c_t^\top x_t + \sum_{r \in \mathcal{R}_t} \gamma_r \theta_{t+1}^r \\
\text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\
& x_t \geq 0 \\
& \theta_{t+1}^r \geq \alpha_{t+1}^{r,k} + \beta_{t+1}^{r,k} x_t, \ \forall r \in \mathcal{R}_t, \ k \in \{1, \ldots, \kappa\}.
\end{aligned}
$$

Static interpolation

SDDP algorithm

What changes in the forward and backward passes?

Static interpolation

SDDP algorithm

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.

Static interpolation

SDDP algorithm

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_t and y_t to the next stage.

Static interpolation

SDDP algorithm

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_t and y_t to the next stage.
3. On the backward pass, we add a cut for every rib in every stage.

Dynamic interpolation

Ribs removed

The key drawbacks of the static interpolation method, just outlined, is that we must define ribs in advance, and may never converge to the true solution, if the ribs are not close enough together.

[3] Baucke, R., Downward, A. & Zakeri G., 2018. A deterministic algorithm for solving stochastic minimax dynamic programmes, Optimization Online.

Dynamic interpolation

Ribs removed

The key drawbacks of the static interpolation method, just outlined, is that we must define ribs in advance, and may never converge to the true solution, if the ribs are not close enough together.

We will now describe a dynamic method of interpolating over the price-states. This utilizes the concept of *saddle-cuts* introduced by Baucke et al. (2018)[3].

[3]Baucke, R., Downward, A. & Zakeri G., 2018. A deterministic algorithm for solving stochastic minimax dynamic programmes, Optimization Online.

Dynamic interpolation

Ribs removed

The key drawbacks of the static interpolation method, just outlined, is that we must define ribs in advance, and may never converge to the true solution, if the ribs are not close enough together.

We will now describe a dynamic method of interpolating over the price-states. This utilizes the concept of *saddle-cuts* introduced by Baucke et al. (2018)[3].

We will briefly explain the concept here, but for the details of the method and some other interesting applications of this concept, see Regan Baucke's presentation on Friday at 9:30am in Amphi DENGIES.

[3] Baucke, R., Downward, A. & Zakeri G., 2018. A deterministic algorithm for solving stochastic minimax dynamic programmes, Optimization Online.

Dynamic interpolation

Convex combination duality

$$
\begin{aligned}
\mathbf{P}: \quad & \max_{\gamma} && \sum_{k=1}^{\kappa} \gamma_k \theta_{t+1}^k \\
& \text{s.t.} && \sum_{k=1}^{\kappa} \gamma_k \hat{y}_t^k = y_t && [\mu] \\
& && \sum_{k=1}^{\bar{k}} \gamma_k = 1 && [\varphi] \\
& && \gamma_k \geq 0 && \forall k \in \{1, \ldots, \kappa\},
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{D}: \quad & \min_{\mu,\varphi} && \mu^\top y_t + \varphi \\
& \text{s.t.} && \mu^\top \hat{y}_t^k + \varphi \geq \theta_{t+1}^k, \ k \in \{1, \ldots, \kappa\},
\end{aligned}
$$

Dynamic interpolation

Saddle-cuts

We will now derive the set of saddle-cuts for stage t, in the k^{th} iteration. From convexity with respect to x_t we have that

$$\theta_{t+1}^k \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Dynamic interpolation

Saddle-cuts

We will now derive the set of saddle-cuts for stage t, in the k^{th} iteration. From convexity with respect to x_t we have that

$$\theta_{t+1}^{k} \geq \alpha_{t+1}^{k} + \beta_{t+1}^{k} x_t.$$

Moreover, in **D** we have:

$$\mu^\top \hat{y}_t^k + \varphi \geq \theta_{t+1}^{k}.$$

Dynamic interpolation

Saddle-cuts

We will now derive the set of saddle-cuts for stage t, in the k^{th} iteration. From convexity with respect to x_t we have that

$$\theta_{t+1}^k \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Moreover, in **D** we have:

$$\mu^\top \hat{y}_t^k + \varphi \geq \theta_{t+1}^k.$$

Together these give our saddle-cut constraints:

$$\mu^\top \hat{y}_t^k + \varphi \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Dynamic interpolation

Saddle-cuts

We will now derive the set of saddle-cuts for stage t, in the k^{th} iteration. From convexity with respect to x_t we have that

$$\theta_{t+1}^k \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Moreover, in **D** we have:

$$\mu^\top \hat{y}_t^k + \varphi \geq \theta_{t+1}^k.$$

Together these give our saddle-cut constraints:

$$\mu^\top \hat{y}_t^k + \varphi \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t.$$

Theorem

The saddle-cut generated in iteration k forms a true lower bound for the cost-to-go function, and is equal to $\mathcal{V}_{t+1}^k(x_t, y_t)$ at the point $(\hat{x}_t^k, \hat{y}_t^k)$ that it was generated.

Dynamic interpolation

Dynamic stage-problem

SP-Dynamic$_t^\kappa\left(x_{t-1}, y_{t-1}, \omega\right)$:
Calculate $y_t = B_t^\omega y_{t-1} + b_t^\omega$ and $c_t = f_t(y_t)$, then solve:

$$
\begin{aligned}
V_t^\kappa\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t, q_t, \mu_t, \varphi_t} \quad & c_t^\top x_t + \mu_t y_t + \varphi_t \\
\text{s.t.} \quad & A_t^\omega x_t - x_{t-1} = a_t^\omega \\
& x_t \geq 0 \\
& \mu_t \hat{y}_t^k + \varphi_t \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t, \ \forall k \in \{1, \ldots, \kappa\}.
\end{aligned}
$$

Dynamic interpolation

SDDP Algorithm

What changes in the forward and backward passes?

Dynamic interpolation

SDDP Algorithm

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.

Dynamic interpolation

SDDP Algorithm

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_t and y_t to the next stage.

What changes in the forward and backward passes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_t and y_t to the next stage.
3. On the backward pass, we add a *saddle-cut* instead of a normal cut.

Dynamic interpolation

Convergence

Theorem

The dynamic interpolation method converges, almost surely, to an optimal policy in a finite number of stages.

Proof.

See Downward et al. (2018)[4] for the proof, which utilizes the fact that there are a finite number of cuts that can be produced. □

[4] Downward, A., Dowson O. & Baucke R., 2018. SDDP with stagewise-dependent objective uncertainty. Optimization Online.

Outline

Price-taking hydroelectric generator

Stage-problem

In the price-taking setting, we assume that there is a stochastic process for the price of electricity, and the action of the generator does not affect this price. This leads to the following stage-problem.

PT-Dynamic$_t^\kappa\left(x_{t-1}, y_{t-1}, \omega\right)$:
Calculate $y_t = B_t^\omega y_{t-1} + b_t^\omega$ and $c_t = -e^{y_t}$, then solve:

$$
\begin{aligned}
V_t^\kappa\left(x_{t-1}, y_{t-1}, \omega\right) = \min_{x_t, q_t, \mu_t, \varphi_t} \quad & c_t \times q_t + \mu_t y_t + \varphi_t \\
\text{s.t.} \quad & x_t = x_{t-1} - q_t + I_t^\omega \\
& q_t \in [0, 100] \\
& x_t \in [0, 230] \\
& \mu_t \hat{y}_t^k + \varphi_t \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t, \ \forall k \in \{1, \ldots, \kappa\}.
\end{aligned}
$$

Price-taking hydroelectric generator

Implementation

This algorithm has been implemented utilizing the `SDDP.jl`[5] Julia package. Oscar Dowson will be presenting about this package on Thursday at 5pm in salle 32.

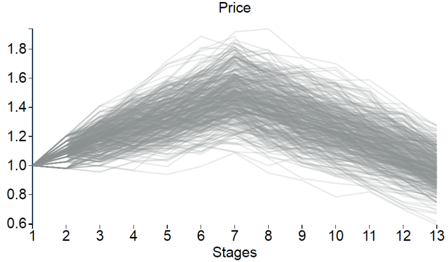
We have chosen a very simple stochastic process for price, a log AR-1 process, with mean reversion. This process is normalized so that the price is 1 at the beginning of the day.

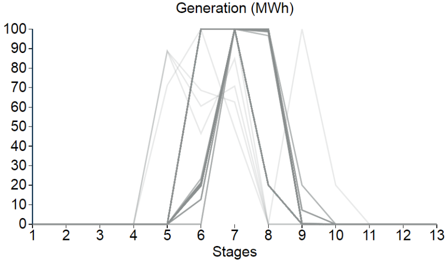
For simplicity in analyzing the results, we assume that there are no inflows over the 13 period time-horizon.

[5] Dowson O. & Kapelevich, L., 2017. SDDP.jl: a Julia package for Stochastic Dual Dynamic Programming. Optimization Online.

Price-taking hydroelectric generator

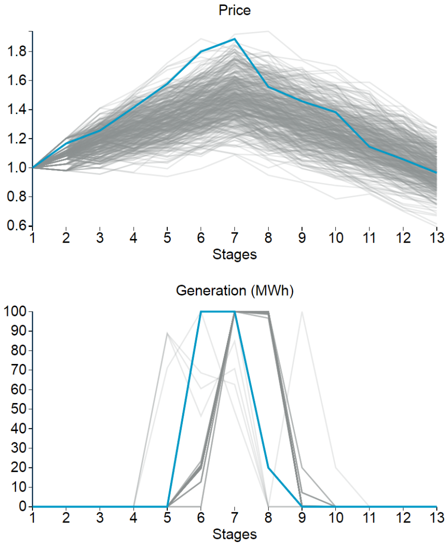
Results





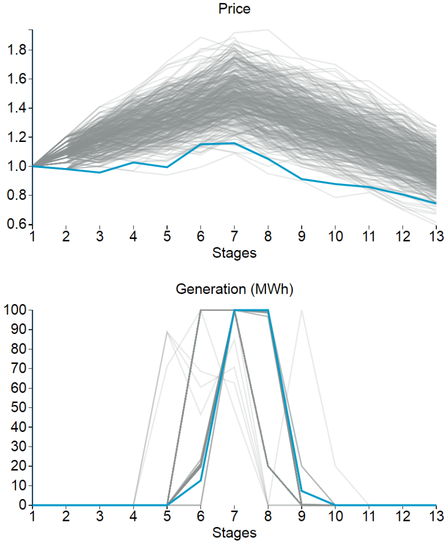
Price-taking hydroelectric generator

Results



Price-taking hydroelectric generator

Results



Price-making hydroelectric generator

Stage-problem

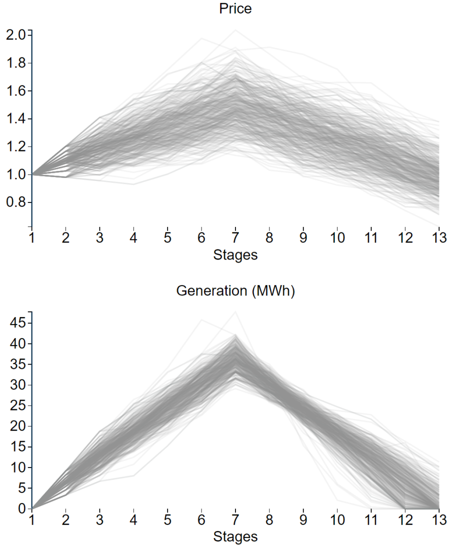
In the price-making setting, we assume there is an inverse residual demand curve of the form $p(q) = c^1 - c^2 q$. A simple example of a stochastic process for this is for c^1 (the intercept) to follow a log AR-1 process, and c^2 to be a constant.

PM-Dynamic$_t^\kappa\,(x_{t-1}, y_{t-1}, \omega)$:
Calculate $y_t = B_t^\omega y_{t-1} + b_t^\omega$, $c_t^1 = -e^{y_t^1}$ and $c_t^2 = -e^{y_t^2}$, then solve:

$$
\begin{aligned}
V_t^\kappa\,(x_{t-1}, y_{t-1}, \omega) = \min_{x_t, q_t, \mu_t, \varphi_t} \quad & (c_t^1 - c_t^2 q_t) \times q_t + \mu_t y_t + \varphi_t \\
\text{s.t.} \quad & x_t = x_{t-1} - q_t + I_t^\omega \\
& q_t \in [0, 100] \\
& x_t \in [0, 230] \\
& \mu_t \hat{y}_t^k + \varphi_t \geq \alpha_{t+1}^k + \beta_{t+1}^k x_t, \ \forall k \in \{1, \ldots, \kappa\}.
\end{aligned}
$$

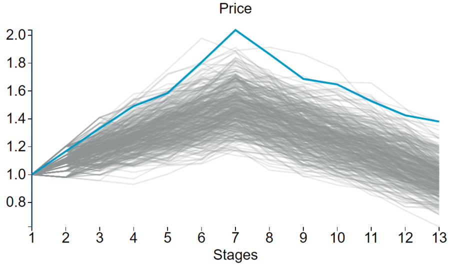
Price-making hydroelectric generator

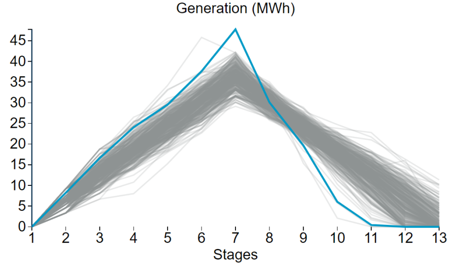
Results



Price-making hydroelectric generator

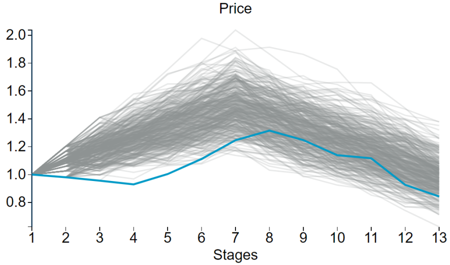
Results

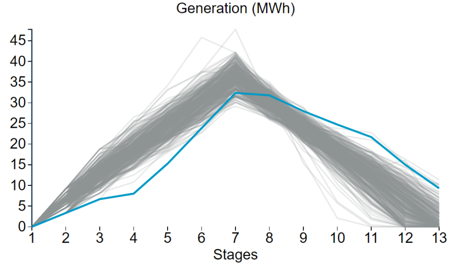




Price-making hydroelectric generator

Results





Outline

Remarks

Static Interpolation

Pros

- Good coverage over the state-space.
- Can use standard SDDP features e.g. cut selection.

Remarks

Static Interpolation

Pros

- Good coverage over the state-space.
- Can use standard SDDP features e.g. cut selection.

Cons

- How many ribs should be used?
- Where should they be placed?
- Low-dimensional problems.

Remarks

Dynamic Interpolation

Pros

- Can model multi-dimensional price processes.
- No need to choose the domain.

Remarks

Dynamic Interpolation

Pros

- Can model multi-dimensional price processes.
- No need to choose the domain.

Cons

- We sample the state-space randomly.
- Cut selection doesn't extend directly.

Remarks

Summary

This method permits many types of stochastic processes to be modelled in the objective function of multistage stochastic programs, so long the are following conditions are met.

1. The noises are stagewise-independent.
2. The price-state transition is linear.
3. The objective coefficient appears as a concave function of the price-state.

Merci de votre attention.

Des questions?

This presentation is based on the work in:
`http://www.optimization-online.org/DB_HTML/2018/02/6454.html`

`SDDP.jl` Julia Library `https://github.com/odow/SDDP.jl`

Contact me: `a.downward@auckland.ac.nz`