SDDP with stagewise-dependent objective uncertainty

a.k.a. SDDP with spot-prices

Oscar Dowson Tony Downward Regan Baucke

Department of Engineering Science
University of Auckland

June, 2018

Outline

Outline

Motivation

SDDP is great. But ...

I want to include a price process like:

$$p_{t+1} = \lambda p_t + (1-\lambda)\mu + \varepsilon_t.$$

Motivation

SDDP is great. But ...

$$
\begin{aligned}
V_t\left(x_t, p_t, \omega_t\right) = \min_{x_{t+1}, p_{t+1}} \quad & {p_{t+1}}^\top Q_t x_{t+1} + \mathcal{V}_{t+1}(x_{t+1}, p_{t+1}) \\
\text{s.t.} \quad & A_t^{\omega_t} x_{t+1} + a^{\omega_t} \geq {\color{red} x_t} \\
& p_{t+1} = B_t^{\omega_t} p_t + b^{\omega_t}
\end{aligned}
$$

Motivation

SDDP is great. But ...

$$
\begin{aligned}
V_t\left(x_t, p_t, \omega_t\right) = \min_{x_{t+1}, p_{t+1}} \quad & {\color{red} p_{t+1}}^\top Q_t x_{t+1} + \mathcal{V}_{t+1}(x_{t+1}, p_{t+1}) \\
\text{s.t.} \quad & A_t^{\omega_t} x_{t+1} + a^{\omega_t} \geq {\color{red} x_t} \\
& p_{t+1} = B_t^{\omega_t} p_t + b^{\omega_t}
\end{aligned}
$$

Motivation

SDDP is great. But ...

$$
\begin{aligned}
V_t\left(x_t, p_t, \omega_t\right) = \min_{x_{t+1}, p_{t+1}} \quad & {p_{t+1}}^\top Q_t x_{t+1} + \mathcal{V}_{t+1}(x_{t+1}, p_{t+1}) \\
\text{s.t.} \quad & A_t^{\omega_t} x_{t+1} + a^{\omega_t} \geq x_t \\
& p_{t+1} = B_t^{\omega_t} p_t + b^{\omega_t}
\end{aligned}
$$

Motivation

SDDP is great. But ...

What are our options?

Motivation

SDDP is great. But ...

What are our options?

1. Assume stagewise-independence

Motivation

SDDP is great. But ...

What are our options?

1. Assume stagewise-independence
2. Discretise the process and use a Markov chain

Motivation

SDDP is great. But ...

What are our options?

1. Assume stagewise-independence
2. Discretise the process and use a Markov chain

What if you could interpolate between Markov states?

Outline

An Example

The widget producer

- A company produces widgets
- Production is uncertain
- They sell on a spot-market
- The spot-price is multiplicative auto-regressive

An Example

The widget producer

$$
\begin{aligned}
V_t\left(x_t, p_t, \omega_t\right) = \min_{x_{t+1}, u_t} \quad & -p_{t+1} \times u_t + \mathcal{V}_{t+1}(x_{t+1}, p_{t+1}) \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t^w \\
& \log(p_{t+1}) = \log(p_t) + \omega_t^p \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350]
\end{aligned}
$$

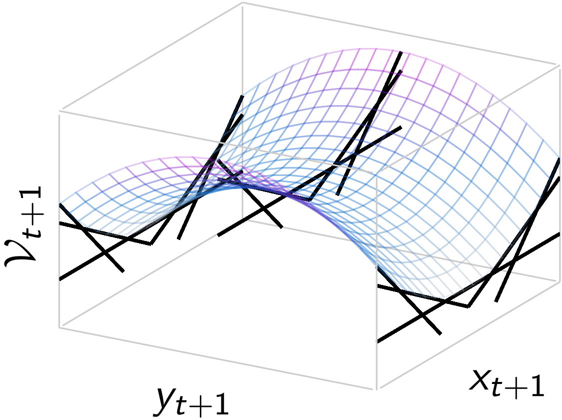
An Example

The widget producer

$$
\begin{aligned}
V_t\left(x_t, y_t, \omega_t\right) = \min_{x_{t+1}, u_t} \quad & -e^{y_{t+1}} \times u_t + \mathcal{V}_{t+1}(x_{t+1}, y_{t+1}) \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t^w \\
& y_{t+1} = y_t + \omega_t^p \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350]
\end{aligned}
$$

An Example

The widget producer



An Example

The widget producer

Requirements:

1. The price process evolves independently
2. The price transition is linear
3. The price appears as a concave function in objective

Outline

The Static Interpolation Method

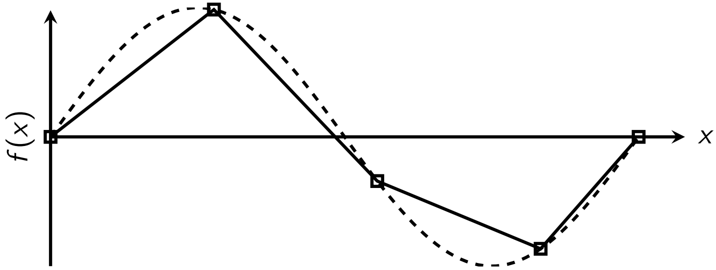


Figure: Piecewise linear interpolation of a one-dimensional function.

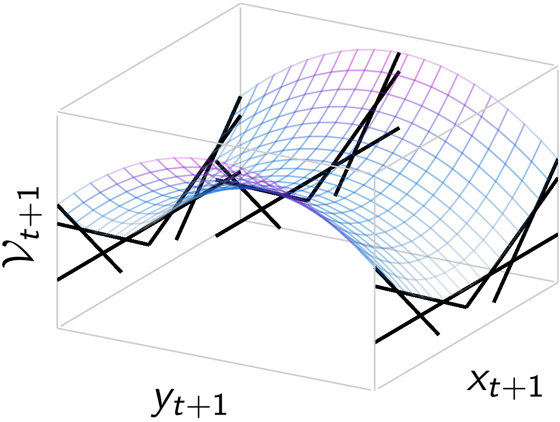
The Static Interpolation Method

$$
\textbf{LIP:} \quad f(\hat{x}) = \max_{\gamma} \quad \sum_{i=1}^{N} \gamma_i f(\bar{x}_i)
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{i=1}^{N} \gamma_i = 1 \\
& \sum_{i=1}^{N} \gamma_i \bar{x}_i = \hat{x} \\
& \gamma_i \geq 0, \;\; i \in \{1, 2, \ldots, N\}.
\end{aligned}
$$

The Static Interpolation Method

Each cost-to-go variable is a “rib”



The Static Interpolation Method

$$
\begin{aligned}
V_t\left(x_t, y_t, \omega_t\right) = \min_{x_{t+1}, u_t} \quad & -e^{y_{t+1}} \times u_t + \mathcal{V}_{t+1}(x_{t+1}, y_{t+1}) \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t^w \\
& y_{t+1} = y_t + \omega_t^p \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350]
\end{aligned}
$$

The Static Interpolation Method

$$
\begin{aligned}
V_t(x_t, y_t, \omega_t) = \min_{x_{t+1}, u_t, y_{t+1}, \theta} \ \max_{\gamma} \quad & -e^{y_{t+1}} \times u_t + \sum_{r \in \mathcal{R}_t} \gamma_r \theta_{t,r} \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t^w \\
& y_{t+1} = y_t + \omega_t^p \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350] \\
& \sum_{r \in \mathcal{R}_t} \gamma_r \hat{y}_r = y_{t+1} \\
& \sum_{r \in \mathcal{R}_t} \gamma_r = 1 \\
& \gamma_r \geq 0, \quad \forall r \in \mathcal{R}_t \\
& \theta_{t,r} \geq \alpha_{t,r}^k + \beta_{t,r}^k x_{t+1}, \ \forall r, \ k
\end{aligned}
$$

The Static Interpolation Method

SP-Static$_t^K\left(x_t, y_t, \omega_t\right)$:
Calculate $y_{t+1} = y_t + \omega_t^p$, compute γ, fix as constants, then solve:

$$\begin{aligned} V_t\left(x_t, y_t, \omega_t\right) = \min_{x_{t+1}, u_t, \theta} \quad & -e^{y_{t+1}} \times u_t + \sum_{r \in \mathcal{R}_t} \gamma_r \theta_{t,r} \\ \text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t \\ & u_t \in [0, 100] \\ & x_{t+1} \in [0, 350] \\ & \theta_{t,r} \geq \alpha_{t,r}^k + \beta_{t,r}^k x_{t+1}, \quad \forall r, \; k. \end{aligned}$$

The Static Interpolation Method

What changes?

The Static Interpolation Method

What changes?

1. The two-step solve of each subproblem.

The Static Interpolation Method

What changes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_{t+1} and y_{t+1} to the next stage.

The Static Interpolation Method

What changes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_{t+1} and y_{t+1} to the next stage.
3. On the backward pass, we add a cut for every rib in every stage.

Outline

The Dynamic Interpolation Method

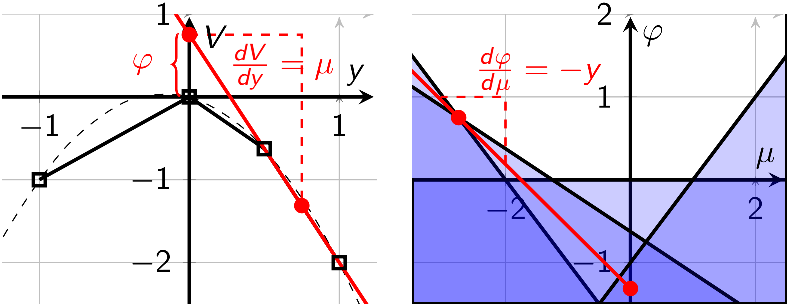
$$
\begin{aligned}
V_t(x_t, y_t, \omega_t) = \min_{x_{t+1}, u_t, y_{t+1}, \theta} \max_{\gamma} \quad & -e^{y_{t+1}} \times u_t + \sum_{r \in \mathcal{R}_t} \gamma_r \theta_{t,r} \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t^w \\
& y_{t+1} = y_t + \omega_t^p \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350] \\
& \sum_{r \in \mathcal{R}_t} \gamma_r \hat{y}_r = y_{t+1} \\
& \sum_{r \in \mathcal{R}_t} \gamma_r = 1 \\
& \gamma_r \geq 0, \quad \forall r \in \mathcal{R}_t \\
& \theta_{t,r} \geq \alpha_{t,r}^k + \beta_{t,r}^k x_{t+1}, \ \forall r, \ k
\end{aligned}
$$

The Dynamic Interpolation Method

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

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

The Dynamic Interpolation Method



(a) Primal Problem (b) Dual Problem

Figure: Geometric Interpretation.

The Dynamic Interpolation Method

SP-Dynamic$_t^K\left(x_t, y_t, \omega_t\right)$:
Calculate $y_{t+1} = y_t + \omega_t^p$ and set as constant, then solve:

$$
\begin{aligned}
V_t\left(x_t, y_t, \omega_t\right) = \min_{x_{t+1}, u_t, \, \mu_t, \varphi_t} \quad & -e^{y_{t+1}} \times u_t + \mu_t y_{t+1} + \varphi_t \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350] \\
& \mu_t y_{t+1}^k + \varphi_t \geq \alpha_t^k + \beta_t^k x_{t+1} \quad \forall k.
\end{aligned}
$$

The Dynamic Interpolation Method

SP-Dynamic$_t^K$ (x_t, y_t, ω_t) :
Calculate $y_{t+1} = y_t + \omega_t^p$ and set as constant, then solve:

$$
\begin{aligned}
V_t(x_t, y_t, \omega_t) = \min_{x_{t+1}, u_t, \mu_t, \varphi_t} \quad & -e^{y_{t+1}} \times u_t + \mu_t y_{t+1} + \varphi_t \\
\text{s.t.} \quad & x_{t+1} = x_t - u_t + \omega_t \\
& u_t \in [0, 100] \\
& x_{t+1} \in [0, 350] \\
& \mu_t y_{t+1}^k + \varphi_t \geq \alpha_t^k + \beta_t^k x_{t+1} \quad \forall k.
\end{aligned}
$$

Saddle Cut

$$
\mu_t y_{t+1}^k + \varphi_t \geq \alpha_t^k + \beta_t^k x_{t+1}
$$

The Dynamic Interpolation Method

What changes?

What changes?

1. The two-step solve of each subproblem.

The Dynamic Interpolation Method

What changes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_{t+1} and y_{t+1} to the next stage.

The Dynamic Interpolation Method

What changes?

1. The two-step solve of each subproblem.
2. On the forward pass, we carry x_{t+1} and y_{t+1} to the next stage.
3. On the backward pass, we add a *saddle-cut* instead of a normal cut.

Outline

Issues – Static Interpolation

Pros

- Good coverage over the state-space

Issues – Static Interpolation

Pros

- Good coverage over the state-space
- Can use cut selection etc.

Issues – Static Interpolation

Pros

- Good coverage over the state-space
- Can use cut selection etc.

Cons

- How many ribs?

Issues – Static Interpolation

Pros

- Good coverage over the state-space
- Can use cut selection etc.

Cons

- How many ribs?
- Where should I put them?

Issues – Static Interpolation

Pros

- Good coverage over the state-space
- Can use cut selection etc.

Cons

- How many ribs?
- Where should I put them?
- Low-dimensional problems

Issues – Dynamic Interpolation

Pros

- Multi-dimensional price process

Issues – Dynamic Interpolation

Pros

- Multi-dimensional price process
- No need to choose the domain

Issues – Dynamic Interpolation

Pros

- Multi-dimensional price process
- No need to choose the domain

Cons

- No cut selection (yet)

Issues – Dynamic Interpolation

Pros

- Multi-dimensional price process
- No need to choose the domain

Cons

- No cut selection (yet)
- We sample the state-space randomly

Issues – Dynamic Interpolation

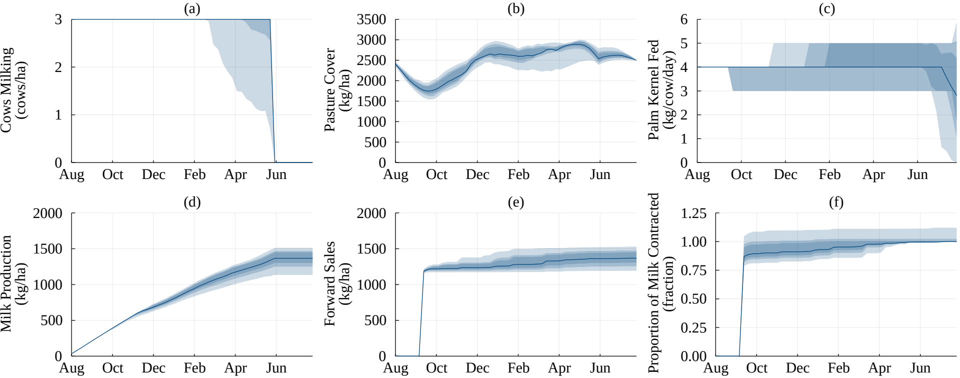
Pros

- Multi-dimensional price process
- No need to choose the domain

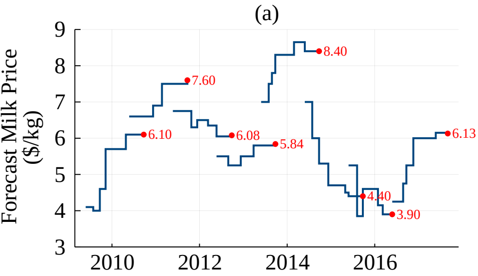
Cons

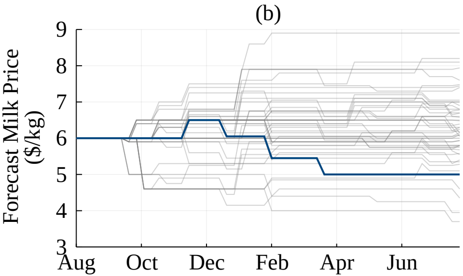
- No cut selection (yet)
- We sample the state-space randomly

"Temporal Drivers"



"Temporal Drivers"





Summary

Requirements:

1. The price process evolves independently;
2. The price transition is linear; and
3. The price appears as a concave function in objective.

Summary

Requirements:

1. The price process evolves independently;
2. The price transition is linear; and
3. The price appears as a concave function in objective.

Short-term hydrothermal scheduling with integrated contracting is now do-able!

Addendum

1. I'm interested in a file-format for Stochastic Programming;
2. Discrete time;
3. Finite discrete noise realizations;
4. Each subproblem stored as an LP/MPS/NL file;
5. Additional information to encode the noise parameters
6. and to describe the linkages between subproblems.

Questions

A link to our paper: http://www.optimization-online.org/DB_HTML/2018/02/6454.html

Thesis (preprint): https://odow.github.io/SDDP.jl/latest/assets/dowson_thesis.pdf

Contact me: o.dowson@auckland.ac.nz