Decomposition in multistage stochastic integer programming

Andy Philpott
Electric Power Optimization Centre
University of Auckland.
www.epoc.org.nz

NTNU Winter School, Passo Tonale, January 15, 2017.

Plans

"Plans are nothing, planning is everything" Dwight Eisenhower

What did he mean by this?

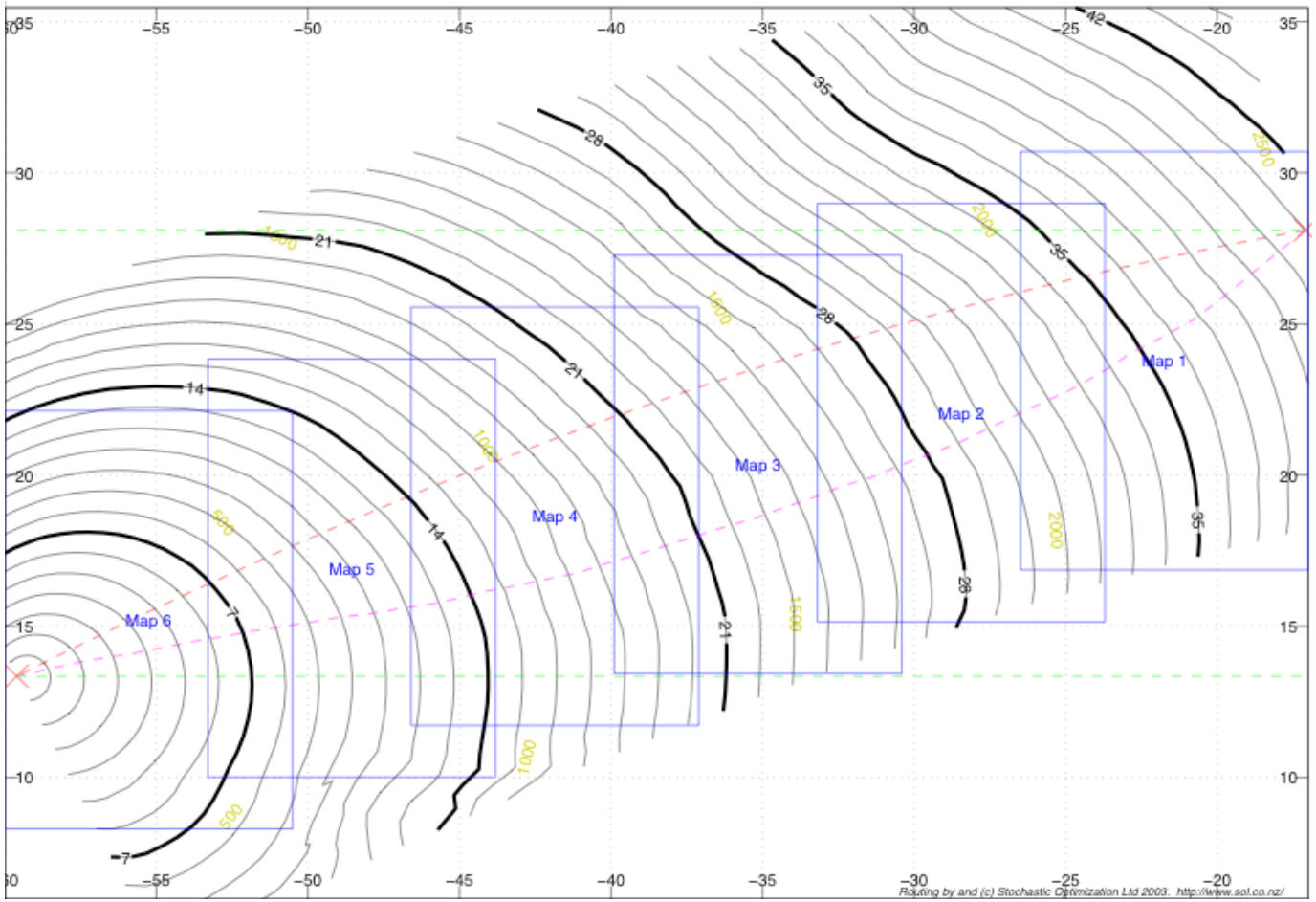
Policies and stochastic optimal control

"... the thought was finally forced upon me that the desired solution in a control process was a policy: 'Do thus-and-thus if you find yourself in this portion of state space with this amount of time left.' Conversely, once it was realized that the concept of policy was fundamental in control theory, the mathematicization of the basic engineering concept of 'feedback control', then the emphasis upon a state variable formulation became natural. We see then a very interesting interaction between dynamic programming and control theory."

Richard Bellman, 1984.

Isochrones for Holiday Shoppe Rowing Challenge 2003

(for details see http://homepages.engineering.auckland.ac.nz/~aphi018/)



Daily isochrones for rowing from La Gomera to Barbados.

Plans are nothing, planning is everything

- A plan tells us a fixed sequence of actions which we are pretty sure are unlikely all to be taken. "Plans are nothing".
- The process of planning itself organizes our thinking about the future, and indicates some possible contingent actions.
- If we plan in this way, then in Eisenhower's experience we usually do much better than being myopic. "Planning is everything".
- But planning does not necessarily need to define a policy.
- Recall that Dantzig's "programming in a linear space" originally meant planning, not operating.

Multistage stochastic programming

- Multistage stochastic programming formalizes the planning process, to give a mathematical description of a plan with contingencies, assuming a given scenario tree for uncertain parameters throughout. Warren Powell calls this a lookahead policy.
- To enact the plan, we take the first-stage action then recompute a new first-stage action after time has passed and new information (i.e. new data) becomes available. This typically involves a new scenario tree that can be quite different.
- If viewed as a policy, the solution to a multistage stochastic program is really just a heuristic.
- There are other (possibly better) heuristics: e.g. linear decison rules, adjustable robust counterpart, etc. The out-of-sample performance is the key performance indicator.

Stochastic optimal control seeks a policy

If we seek a policy then one thinks in terms of a stochastic control problem. We seek an optimal policy yielding Bellman function $C_t(\mathbf{x})$, where

$$\begin{aligned} C_t(\mathbf{x}) &= \mathbb{E}_{\xi_t}\left[\min_{u\in U(x)}\{r_t(\mathbf{x},\mathbf{u},\boldsymbol{\xi}_t)+C_{t+1}(f_t(\mathbf{x},\mathbf{u},\boldsymbol{\xi}_t))\}\right], \quad (1) \\ C_{T+1}(\mathbf{x}) &= R(\mathbf{x}). \end{aligned}$$

This is a dynamic programming recursion. Given a state $\mathbf{x}$ at stage t, the optimal action solves the stage problem (1). If we are maximizing then we use the same form but with value $V_t(\mathbf{x})$ replacing $V_t(\mathbf{x})$.

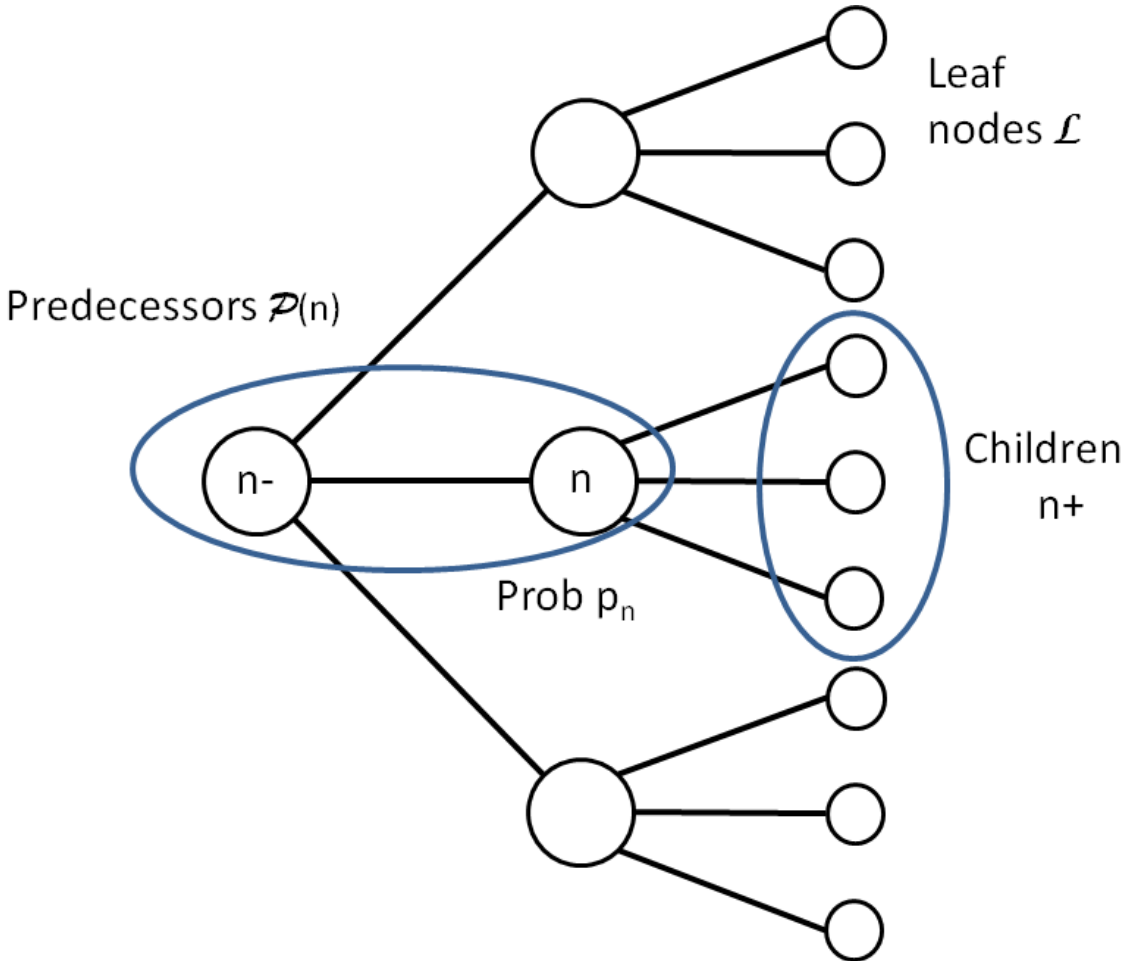
Summary: planning and operating under uncertainty

- Planning under uncertainty is most naturally treated as a multistage stochastic program - compute an "optimal" plan with contingencies.
- Operating under uncertainty is most naturally treated as a stochastic control problem - compute an "optimal" policy.
- To see which model you are using, ask the following: if one were to simulate the actions defined by the model, would one need to solve a new version of the model to define optimal actions at a later stage?
- In other words time-consistency is less important in planning models than operational models.

Summary

Summary

Multistage stochastic program uses a scenario tree



A scenario tree $\mathcal{N}$. The child nodes of n are denoted $n+$, and parent node $n-$.

Tree formulation of stochastic optimization problem

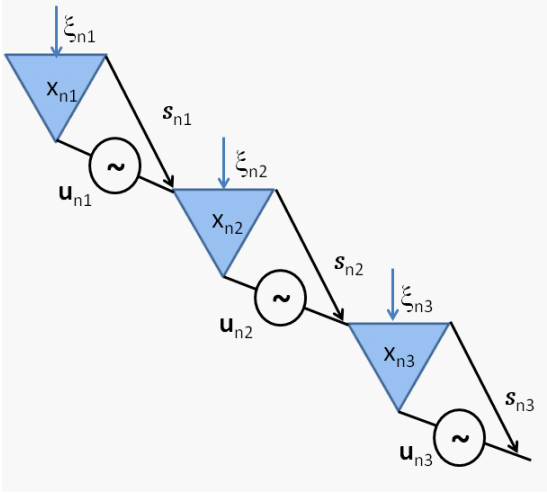
$$
\begin{array}{lll}
\text{MSPT:} & \min & \sum_{n\in\mathcal{N}\setminus\{0\}} p_n r_n(x_{n-}, u_n) + \sum_{n\in\mathcal{L}} p_n C(x_n) \\
& \text{s.t.} & x_n = f_{n-}(x_{n-}, u_n, \xi_n), \\
& & x_0 = \overline{x}, \\
& & u_n \in U(x_n), \\
& & x_n \in X_n.
\end{array}
$$

Recursive form is:

$$
\begin{aligned}
C_n(x_n) &= \sum_{m\in n+} \frac{p(m)}{p(n)} \min_{u\in U(x_n)} \left\{ r_m(x_n, u) + C_m\left(f_n(x_n, u, \xi_m)\right)\right\} \\
C_n(x_n) &= R(x_n),\ n \in \mathcal{L},
\end{aligned}
$$

where we seek u_n, $n \in \mathcal{N} \setminus \{0\}$, that minimizes $C_0(x_0)$.

Example: River chain optimization under uncertainty



The dynamics in node n of the scenario tree are

$$x_{n,i} = x_{n-,i} + u_{n,i-1} - u_{n,i} - s_{n,i} + \xi_{n,i}.$$

Given electricity prices π_n in stage n the generator arranges feasible releases $u_{n,i}$ and spill $s_{n,i}$ of water to maximize expected revenue

$$\sum_{n \in \mathcal{N} \setminus \{0\}} p_n \sum_t \pi_n \sum_i g_i(u_{n,i}).$$

Here g_i converts water flow through station i into energy.

Example: Capacity expansion under uncertainty

We have initial capacity $\mathbf{x}_0 \in \mathbb{R}^K$ that we increase with actions $\mathbf{u}_n \in \{0,1\}^Z$ in node n of $\mathcal{N}$. Investment in node n costs $\mathbf{c}_n^\top \mathbf{u}_n$ and contributes additional capacity $U_n \mathbf{u}_n \geq 0$ to the system. So the capacity in node n is $\mathbf{x}_0 + \sum_{h \in \mathcal{P}_n} U_h \mathbf{u}_h$. In each state corresponding to node n we operate our capacity choosing actions $\mathbf{y}_n \in \mathcal{Y}_n$ to minimize cost $\mathbf{q}_n^\top \mathbf{y}_n$.

$$
\begin{aligned}
\min \quad & \textstyle\sum_{n \in \mathcal{N}} p_n \left(\mathbf{c}_n^\top \mathbf{u}_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \\
\text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{x}_0 + \textstyle\sum_{h \in \mathcal{P}_n} U_h \mathbf{u}_h, \quad n \in \mathcal{N} \\
& \mathbf{y}_n \in \mathcal{Y}_n, \quad n \in \mathcal{N} \\
& \mathbf{u}_n \in \{0,1\}^Z, \quad n \in \mathcal{N}
\end{aligned}
$$

Summary

Stochastic optimal control

We seek an optimal policy in terms of a Bellman function $C_t(\mathbf{x})$, where

$$\begin{aligned} C_t(\mathbf{x}) &= \mathbb{E}_{\xi_t}\left[\min_{u \in U(x)} \{r_t(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}_t) + C_{t+1}(f_t(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}_t))\}\right], \\ C_{T+1}(\mathbf{x}) &= R(\mathbf{x}). \end{aligned}$$

This is a dynamic programming recursion.

Example: stochastic dual dynamic programming (SDDP)

[Pereira and Pinto, 1991] See the last talk by Vitor.

Assume $C_t(\mathbf{x})$ is convex.
For cut sharing assume scenario tree represents a stagewise independent process.
Represent an approximately optimal policy by an outer approximation of $C_t(\mathbf{x})$ using cutting planes.
Evaluate the approximation by simulating sample paths of the states under the policy by solving linear programming stage problems.
Improve the policy at the states visited by simulation by updating the outer approximation using subgradients of the stage optimal value function.

Examples of SDDP in practice

- Computing system electricity prices and dispatch (hydro-thermal sheduling);
- Assessing security of supply;
- Benchmarking competition;
- System optimization of battery storage;
- Price-taking agent optimization, e.g. river chain optimization.

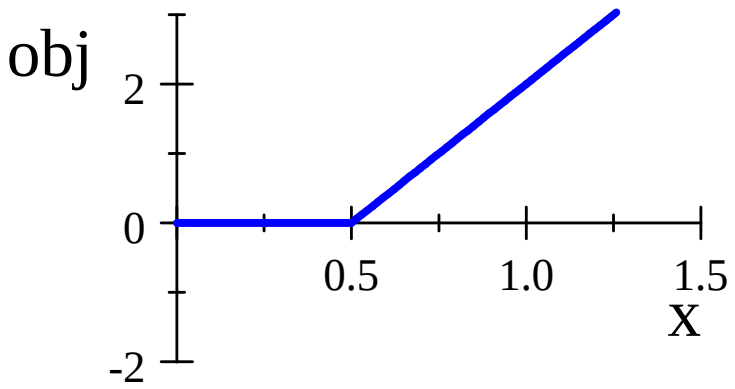
Is SDDP planning or operating?

- In one interpretation, SDDP gives a policy that is defined for every possible state and so it looks like operating. e.g. short-term river-chain optimization, battery storage.
- SDDP over longer term looks like planning.
- In practice, SDDP terminates with an action for the current observed state, and contingent actions defined by the approximately optimal policy for future stages. It does not define an optimal action for every possible state we could be in, since it only explores states that are reached from the initial state via sample paths.

Some curious behaviour

- Suppose we solve SDDP from an initial state $x(0)$ to give a set of cuts for each stage. The upper bound and lower bound appear to have converged.
- Now starting from $x(0)$ simulate the policy many times with new random seeds. One finds the policy performs worse than expected. Is this perhaps post-decision disappointment?
- It's not, as this still happens with a deterministic two-stage problem.

Two-stage example
(Dowson, 2016)



Plot of $f(x) = x + \min\{-x, 3x - 2\}$

Minimizing the function $f(x)$ over $[0, 1]$ is equivalent to

$$\begin{array}{lrl} \text{P: min} & x + y_2 & \\ \text{s.t.} & x - y_1 & = 0 \\ & y_1 + y_2 & \geq 0 \\ & -3y_1 + y_2 & \geq -2 \\ & x \in [0, 1] & \end{array}$$

This has optimal solution $(1 - \lambda)(0, 0, 0) + \lambda(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2})$.

Benders decomposition

Start at solution $(0, -1)$ to

$$\begin{aligned} \text{MP: min} \quad & x + \theta \\ \text{s.t.} \quad & x \leq 1 \\ & \theta \geq -1 \\ & x \geq 0 \end{aligned}$$

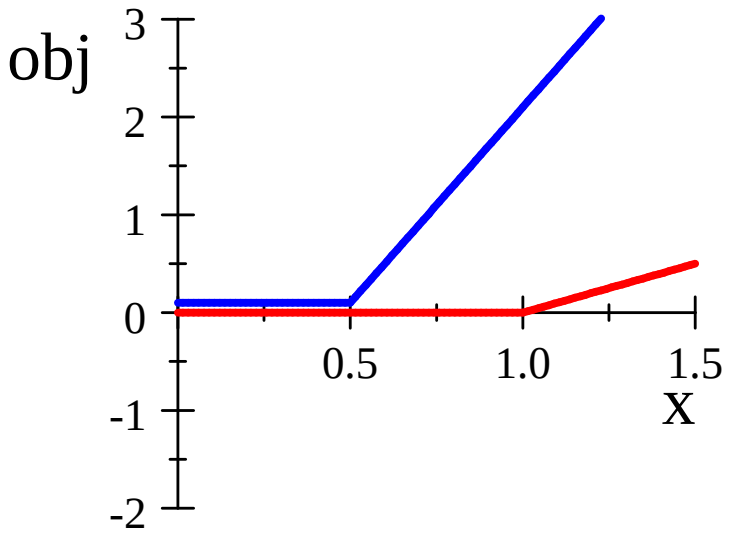
At $x = 0$, we get

$$\begin{aligned} \text{SP: min} \quad & y_2 \\ \text{s.t.} \quad & y_1 = x \qquad [\pi] \\ & y_1 + y_2 \geq 0 \\ & -3y_1 + y_2 \geq -2 \end{aligned}$$

has solution $y_1 = y_2 = 0$, $\pi = -1$, and cut $\theta \geq 0 + -1(x - 0)$.

Benders decomposition

$$
\begin{array}{lll}
\text{MP:} & \min & x + \theta \\
& \text{s.t.} & x \leq 1 \\
& & \theta \geq -1 \\
& & \theta \geq -x \\
& & x \geq 0
\end{array}
$$



True objective function is blue. Approximation of objective function (red) gives an optimal solution $x = 0$, $\theta = 0$.

Suppose we now simulate the policy

The policy is defined by

$$\begin{aligned} \text{MP: min} \quad & x + \theta \\ \text{s.t.} \quad & x \leq 1 \\ & \theta \geq -1 \\ & \theta \geq -x \\ & x \geq 0. \end{aligned}$$

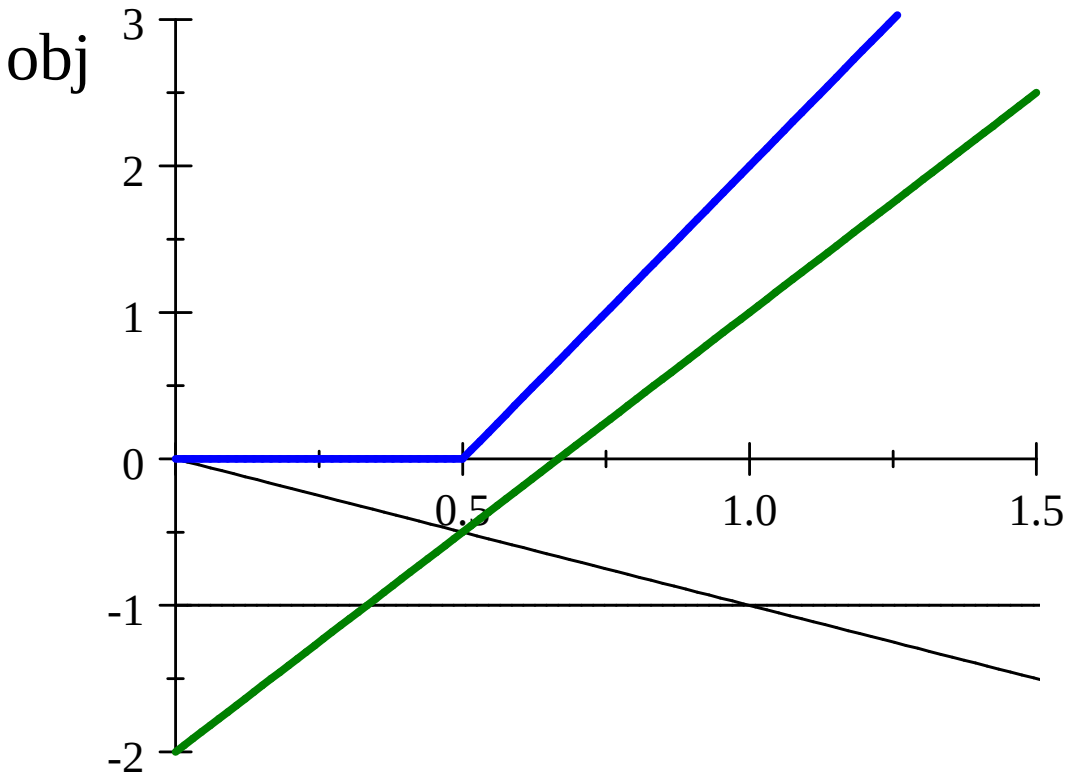
But MP also has solution $(1, -1)$. At $x = 1$, we get

$$\begin{aligned} \text{SP: min} \quad & y_2 \\ \text{s.t.} \quad & y_1 = x \quad [\pi] \\ & y_1 + y_2 \geq 0 \\ & -3y_1 + y_2 \geq -2 \end{aligned}$$

has solution $y_1 = 1, y_2 = 1$,and optimal value 1. So the cuts at termination do not define an optimal policy if stage problems have alternative solutions.

What is the missing cut?

We require enough cuts possibly at every optimal solution to a stage problem.



If we add the green cut then the overall objective of MP is the true objective of the problem (blue).

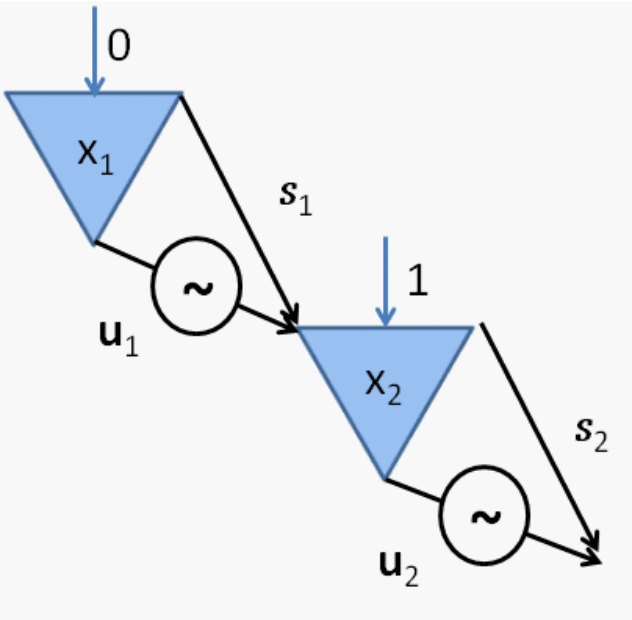
The takeaway message for SDDP

- SDDP terminates when the lower bound and upper bound on optimal value are sufficiently close.
- SDDP appears to give an optimal policy, but it is not optimal until cuts are computed for every possible alternative primal solution to each stage problem.
- SDDP has given an optimal plan only for the states it visits.
- In hydrothermal scheduling it can help to use stage problems with unique solutions (e.g. strictly concave production functions).

Summary

River chain model with integer releases

Suppose initial storage of a and b. Turbine 1 is twice as efficient as turbine 2. Maximum total release is constrained to 1.5 by environmental regulation. Stage problem is:



$$
\begin{aligned}
\varphi(a,b) = \quad & \min \quad && -2u_1 - u_2 + C(x_1, x_2) \\
& \text{s.t.} && x_1 = a - u_1 - s_1 \\
& && x_2 = b + u_1 - u_2 - s_2 + 1 \\
& && u_1 + u_2 \leq 1.5 \\
& && x_i \geq 0,\ u_i \in \mathbb{Z}_+,\ s \geq 0.
\end{aligned}
$$

Optimal value function

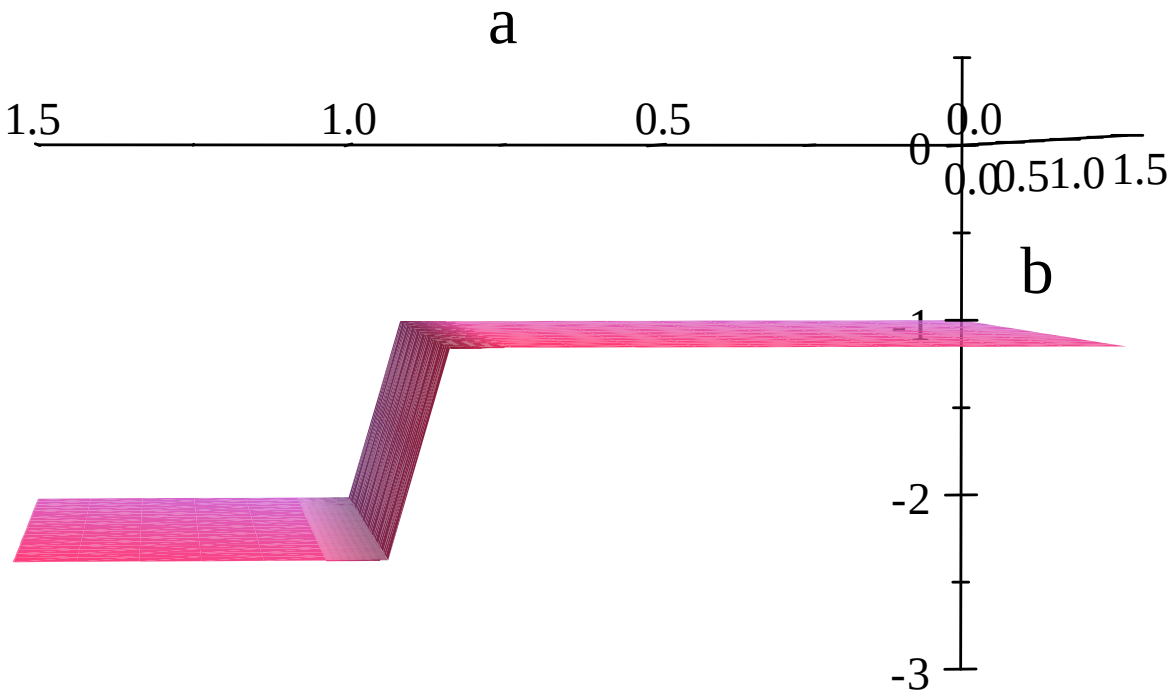
Assume $C(x_1, x_2)$ is 0.

$$\begin{aligned} \varphi(a,b) = \quad & \min \quad -2u_1 - u_2 \\ & \text{s.t.} \quad x_1 = a - u_1 - s_1 \\ & \qquad x_2 = b + u_1 - u_2 - s_2 + 1 \\ & \qquad u_1 + u_2 \leq 1.5 \\ & \qquad x_i \geq 0,\ u_i \in \mathbb{Z}_+,\ s \geq 0. \end{aligned}$$

The optimal solution is to make u_1 as large as possible then u_2. So if $a \geq 1$ then $u_1 = 1$, $u_2 = 0$, $\varphi(1, b) = -2$. If $a < 1$ then $u_1 = 0$, $u_2 = 1$, $\varphi(a, b) = -1$.

$\varphi(a,b)$	$b = 0$	$b = 1$
$a < 1$	-1	-1
$a \geq 1$	-2	-2

MIP value function



Plot of MIP value function $\varphi(a, b)$

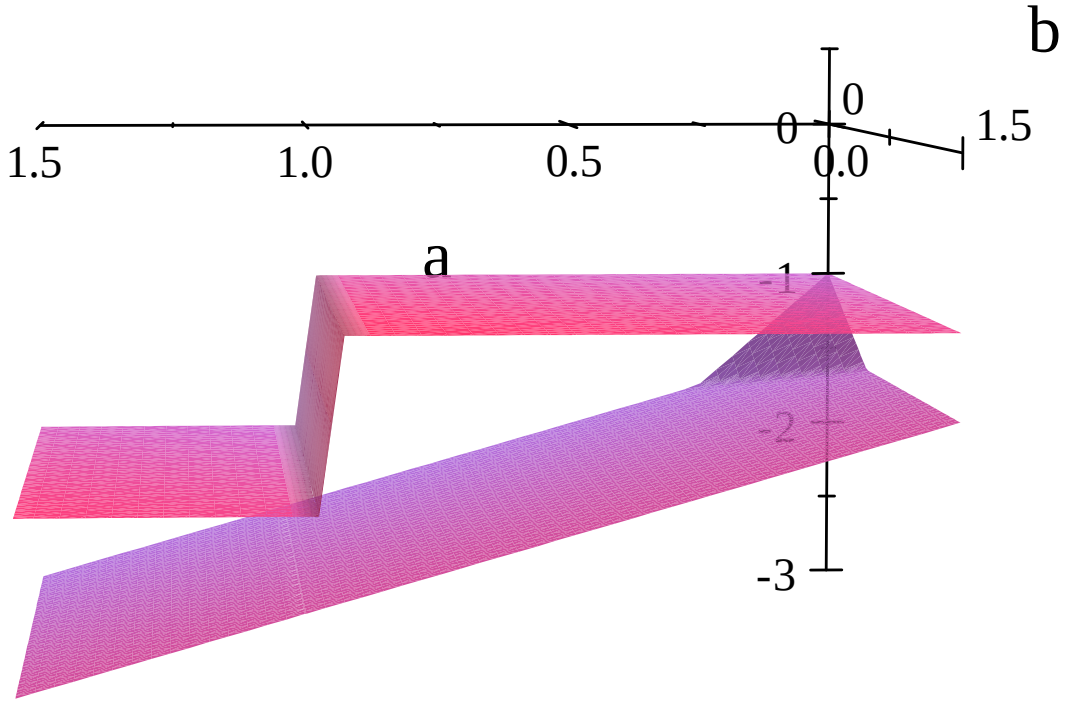
Linear relaxation

(Assume $C(x_1, x_2) = 0$.)

$$\begin{aligned} \varphi^r(a,b) = \quad & \min \quad -2u_1 - u_2 \\ & \text{s.t.} \quad x_1 = a - u_1 - s_1 \\ & \qquad\quad x_2 = b + u_1 - u_2 - s_2 + 1 \\ & \qquad\quad u_1 + u_2 \leq 1.5 \\ & \qquad\quad x_i \geq 0,\ u_i \geq 0,\ s \geq 0. \end{aligned}$$

Optimal solution is $u_1 = a$, $u_2 = \min\{a + b + 1, 1.5 - a\}$,

Plot of linear relaxation



Plot of $\varphi(a, b)$ and LP relaxation $\varphi^r(a, b)$.

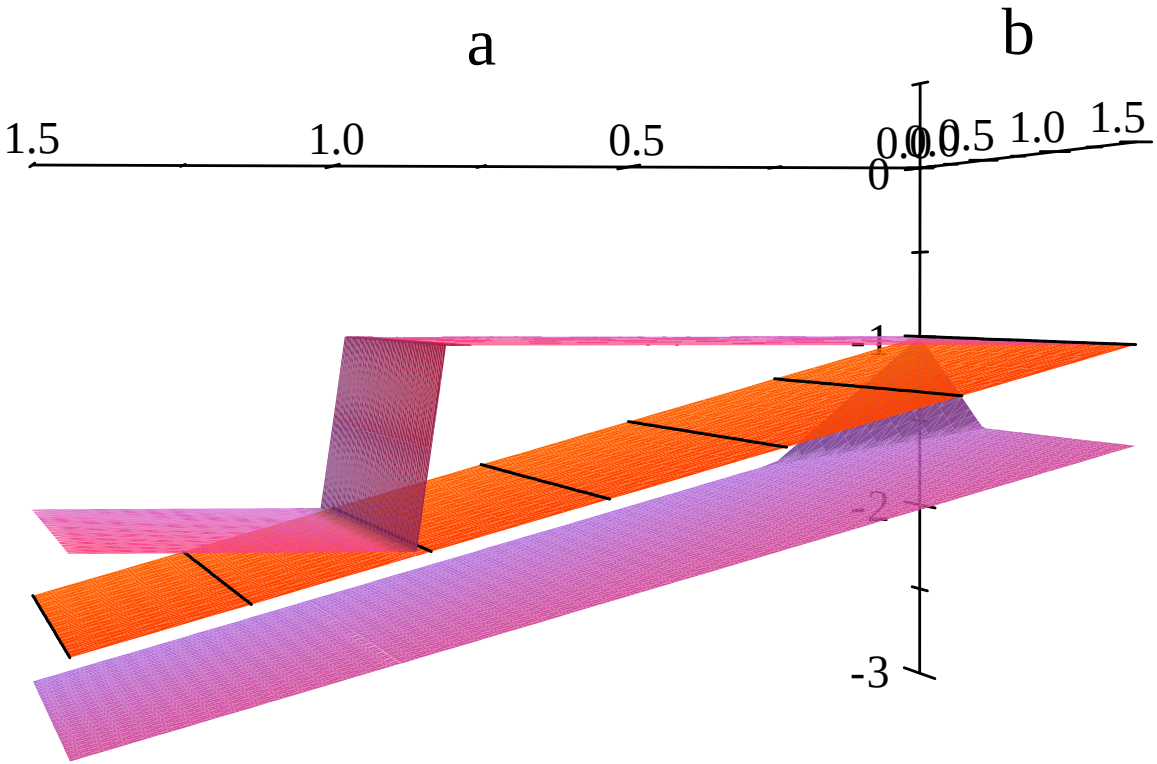
Lagrangean relaxation
(In SDDP see Thome et al, PSR, 2013)

$$
\begin{aligned}
\mathcal{L}(a,b,\pi) = \quad \min \quad & -2u_1 - u_2 \\
& +\pi_1(-a + u_1 + s_1 + x_1) \\
& +\pi_2(-b - u_1 + u_2 + s_2 - 1 + x_2) \\
\text{s.t.} \quad & u_1 + u_2 \leq 1.5 \\
& x_i \geq 0,\ u_i \in \mathbb{Z}_+,\ s \geq 0.
\end{aligned}
$$

$$\varphi^r(a,b) \leq \varphi^l(a,b) = \max_{\pi} \mathcal{L}(a,b,\pi) \leq \varphi(a,b)$$

$$\varphi^l(a,b) = \max_{\pi}\{\min(-\pi_2 + \pi_1 - 2, \pi_2 - 1, 0) - a\pi_1 - (b+1)\pi_2\}$$

Plot of relaxations



Plot of $\varphi(a, b)$, $\varphi^r(a, b)$ and $\varphi^l(a, b)$. Lagrangean relaxation $\varphi^l(a, b)$ gives better approximation to $\varphi(a, b)$ over all a and b.

SDDP with binary state variables

(Ahmed, Sun and Zhou, 2016)

In previous example, Lagrangean relaxation is exact at $a = 1$. Suppose stage problems need to have binary states. ASZ trick is to split variables in the stage problem, to get an SDDP method for binary state variables.

$$
\begin{aligned}
\varphi(a, b) = \quad & \min \quad -2u_1 - u_2 \\
& \text{s.t.} \quad x_1 = z_1 - u_1 - s_1 \\
& \qquad\quad x_2 = z_2 + u_1 - u_2 - s_2 + 1 \\
& \qquad\quad u_1 + u_2 \leq 1.5 \\
& \qquad\quad z_1 = a, \;\; z_2 = b \\
& \qquad\quad x_i \in \{0, 1\}, \; u_i \in \{0, 1\}, \; s \geq 0, \\
& \qquad\quad z_i \in [0, 1].
\end{aligned}
$$

Lagrangean relaxation of stage problem

$$
\begin{aligned}
\varphi^l(a,b) = \quad & \max_\pi \min && -2u_1 - u_2 + \pi_1(a - z_1) + \pi_2(b - z_2) \\
& \text{s.t.} && x_1 = z_1 - u_1 - s_1 \\
& && x_2 = z_2 + u_1 - u_2 - s_2 + 1 \\
& && u_1 + u_2 \leq 1.5 \\
& && x_i \in \{0,1\},\ u_i \in \{0,1\},\ s \geq 0, \\
& && z_i \in [0,1].
\end{aligned}
$$

It is easy to show that if $a, b \in \{0,1\}$ then $z_i \in \{0,1\}$.

SDDP with binary state variables
(Ahmed et al, 2016)

- All stage problems are solved as MIPs with cutting planes giving an outer approximation of $\varphi^l(\mathbf{x})$.
- Subgradients for $\varphi^l(\mathbf{x}_{n-})$ for stage problem define cutting planes.
- Evaluate the approximation by simulating sample paths of the states under the policy by solving MIP stage problems.
- Improve the policy at the states visited by simulation by updating the outer approximation using subgradients of $\varphi^l(\mathbf{x})$.
- SDDP can be proved to converge (a.s.) to an optimal solution. Reported computational results are impressive.

Mixed integer dynamic approximation scheme (MIDAS)

[P, Wahid et al, 2016]

Suppose actions u must be integer.

Warning: we are now maximizing revenue

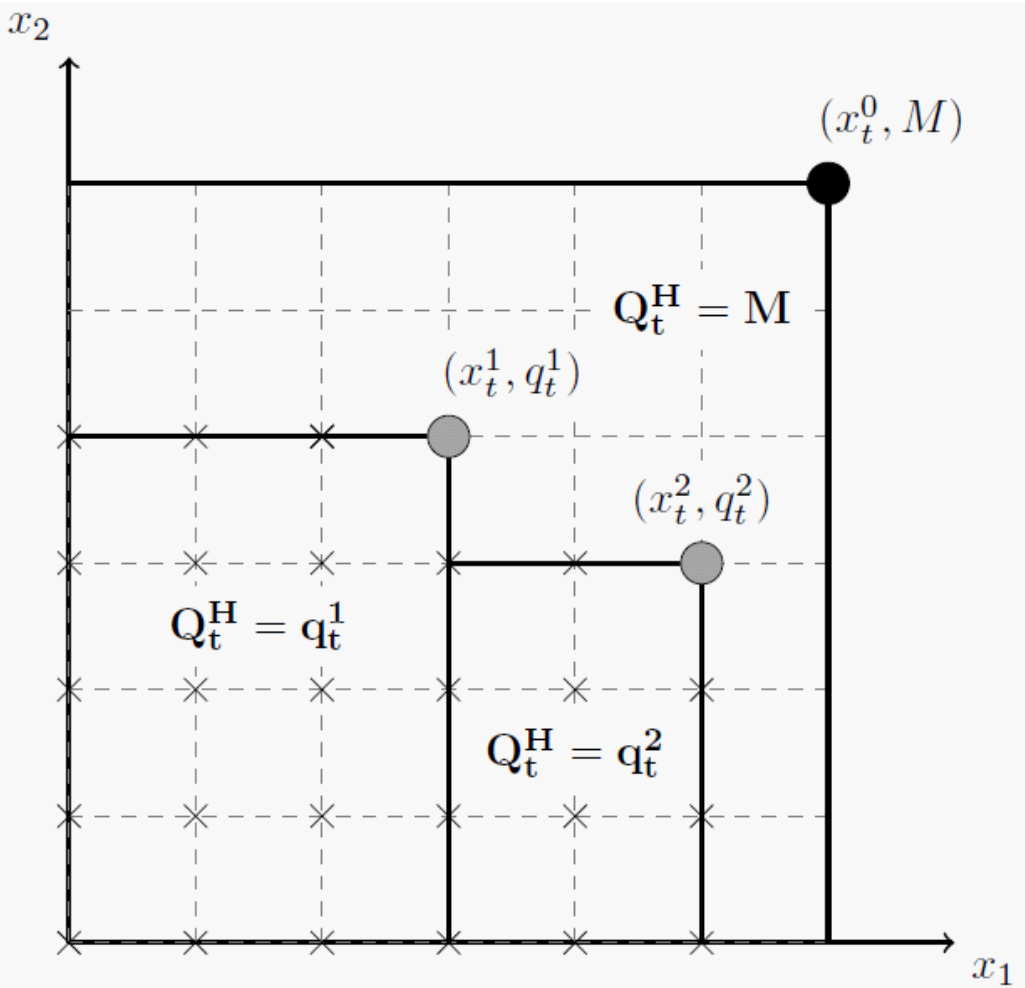
Assume $V_t(\mathbf{x})$ is nondecreasing. This amounts to an assumption of free disposal.

Represent an approximately optimal policy by an outer approximation of $V_t(\mathbf{x})$ using step functions.

Evaluate the approximation by simulating sample paths of the states under the policy by solving mixed-integer programming stage problems.

Improve the policy at the states visited by simulation using a step-function approximation defined by the optimal values of the MIP stage problems.

Approximation of value function



$V_t(x)$ is (outer) approximated by a piecewise constant function $Q^H(x)$. Here $q_t^1 < q_t^2$.

Approximation represented by a mixed integer program

$$
\begin{array}{rrcll}
Q^H(x) = \max & \varphi & & & \\
\text{s.t.} & \varphi & \leq & q^h + (M - q^h)y^h, & h = 1, 2, \ldots, H, \\
 & x_k & \geq & (x_k^h + 1)z_k^h, & k = 1, 2, \ldots, n, \\
 & & & & h = 1, 2, \ldots, H, \\
 & \sum_{k=1}^n z_k^h & \geq & y^h, & h = 1, 2, \ldots, H, \\
 & y^h & \in & \{0, 1\}, & h = 1, 2, \ldots, H, \\
 & z_k^h & \in & \{0, 1\}, & k = 1, 2, \ldots, n, \\
 & & & & h = 1, 2, \ldots, H.
\end{array}
$$

MIDAS algorithm for stochastic dynamic programming

1. Set $Q_n^1(x) = M$, for every $n \in \mathcal{N} \setminus \mathcal{L}$;
2. For $H = 1, 2, \ldots,$
   - set $Q_n^H(x) = R(x)$, for every $n \in \mathcal{L}$;
   - perform a forward pass then a backward pass.

Forward pass

Set $x_0^H = \overline{x}$, and $n = 0$. While $n \notin \mathcal{L}$:

1. Sample $m \in n+$ to give ξ_m^H;
2. Solve $\left\{\max_{u \in U(x_n^H)} \left\{r_m(x_n^H, u) + Q_m^H(f_n(x_n^H, u, \xi_m^H))\right\}\right\}$ to give u_m^H;
3. Set $n = m$.

Backward pass

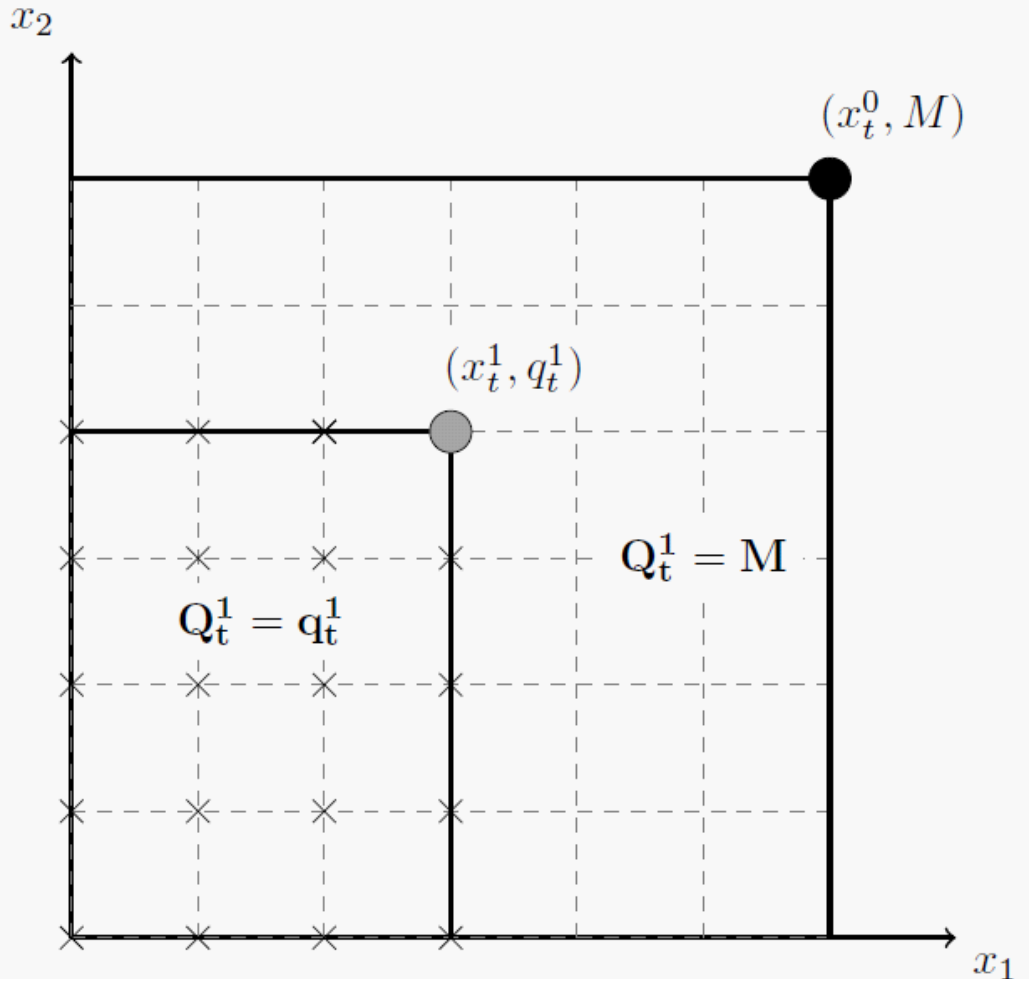
For the particular node $n \in \mathcal{L}$ at the end of step 1 update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = R(x_n^H)$ at point x_n^{H+1}. While $n > 0$

1. Set $n = n-$;
2. Compute

$$\varphi = \sum_{m \in n+} \frac{p(m)}{p(n)} \max_{u \in U(x_n^H)} \{r_m(x_n^{H+1}, u) + Q_m^{H+1}(f_n(x_n^{H+1}, u, \xi_m))\}$$

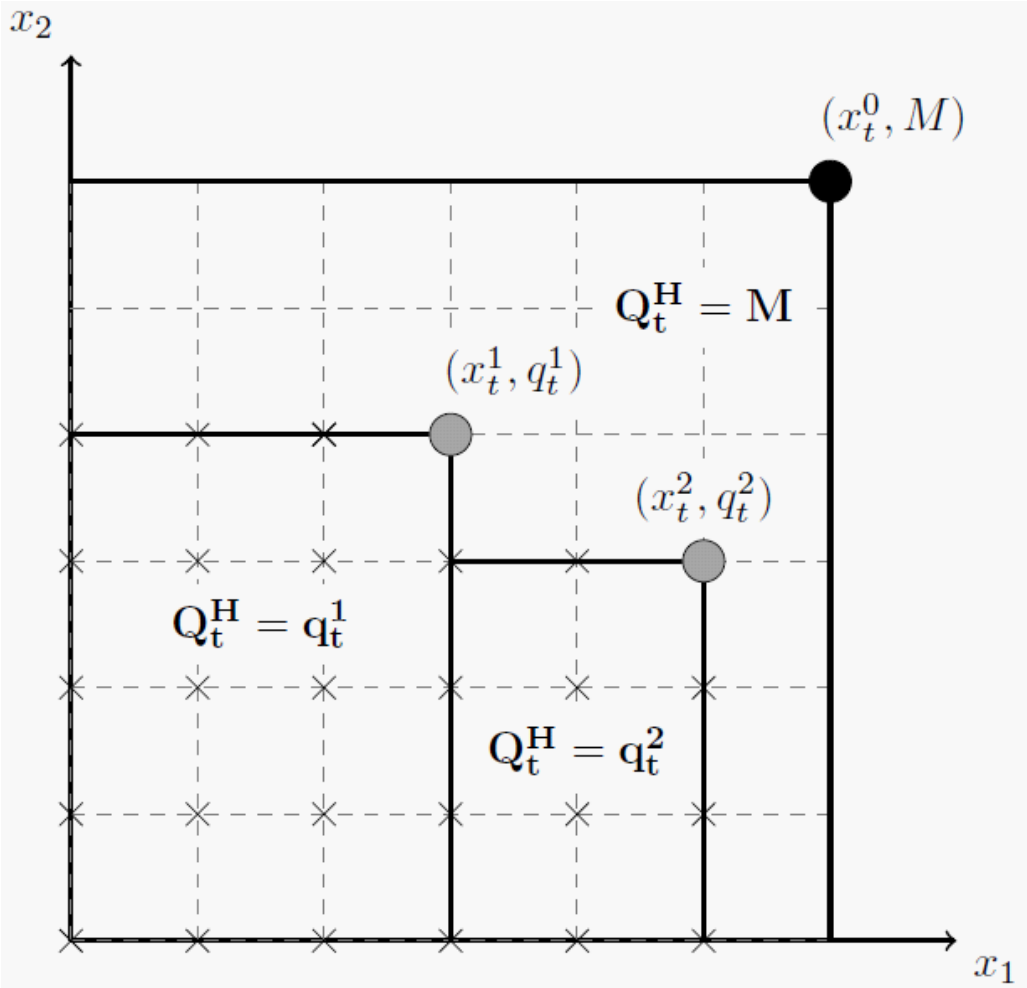
3. Update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = \varphi$ at point x_n^{H+1};
4. Increase H by 1 and begin a new forward pass.

Example in two dimensions



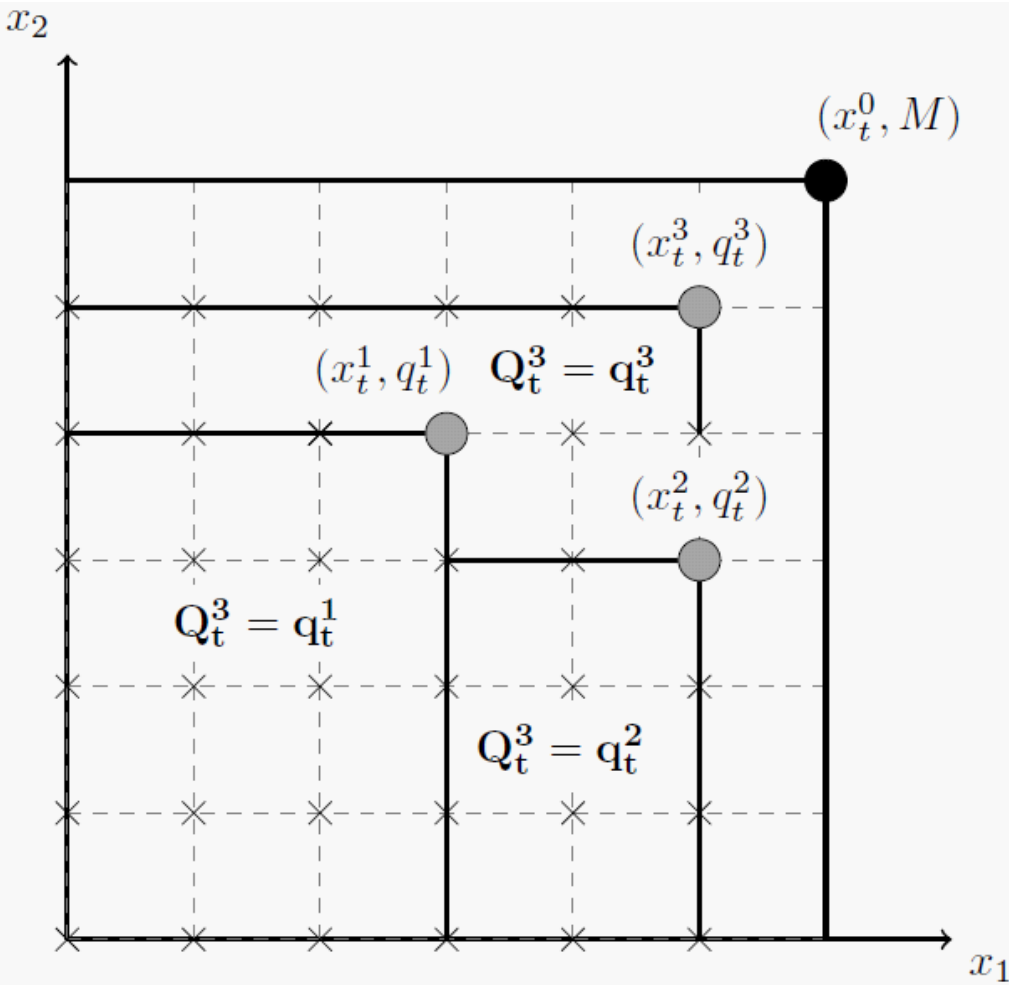
New point added at x_t^1 with value q_t^1.

Example in two dimensions



New point added at x_t^2 with value q_t^2 (Here $q_t^2 > q_t^1$).

Example in two dimensions



New point added at x_t^3 with value q_t^3 (Here $q_t^3 > q_t^2 > q_t^1$).

Sampling property

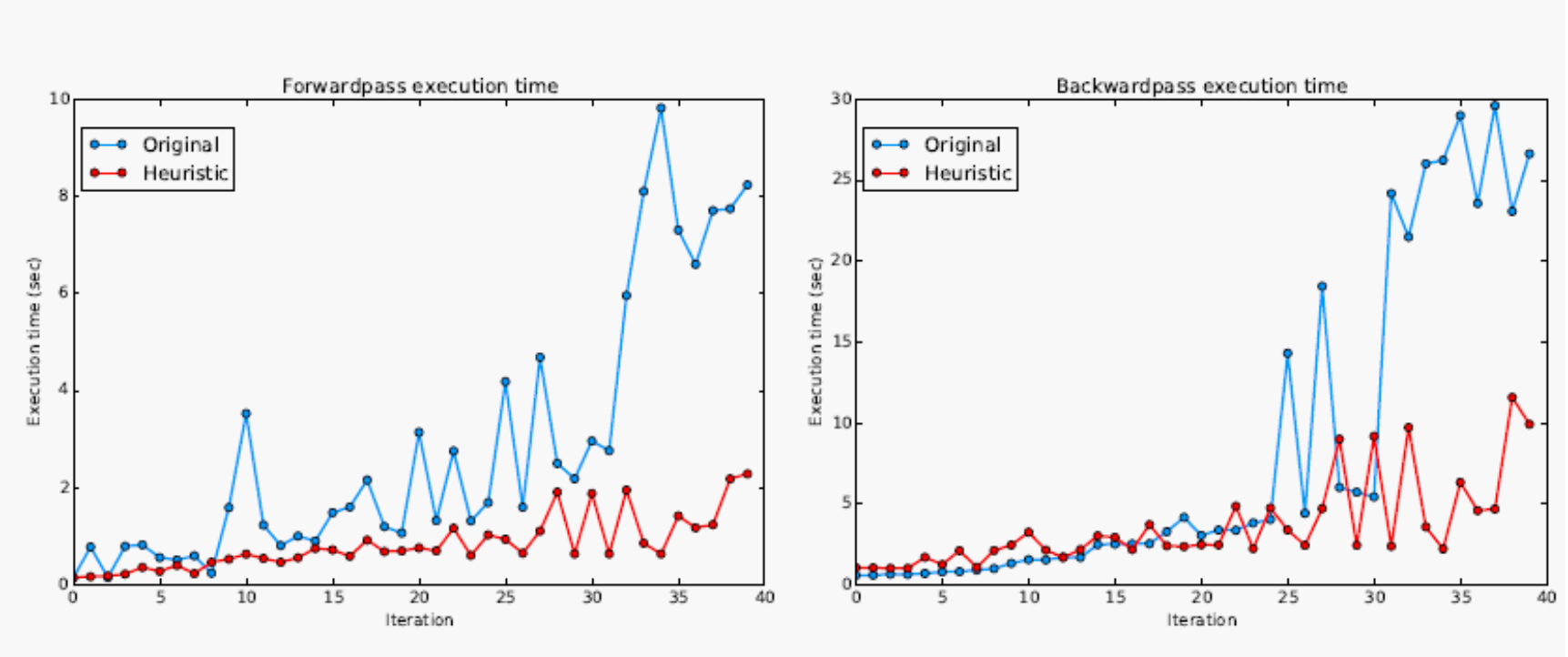
FPSP: For each $n \in \mathcal{L}$, with probability 1

$$\left|\left\{H : \xi_n^H = \xi_n\right\}\right| = \infty.$$

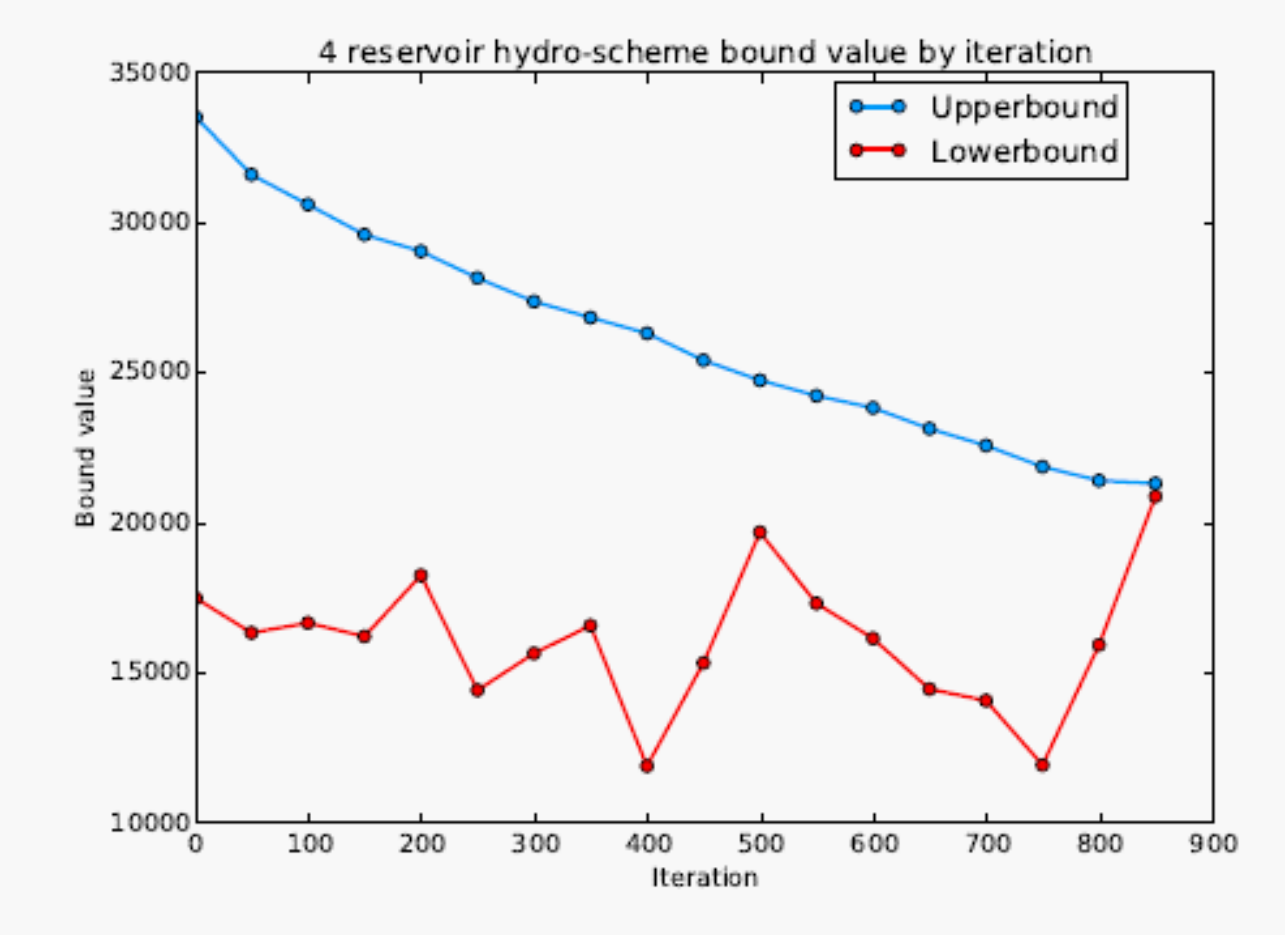
Theorem

If step 1 of forward pass satisfies FPSP then sampled MIDAS converges almost surely to an optimal solution.

Computational results (4 reservoir river chain)



MIPS become expensive as points added. "Heuristic" eliminates dominated points and uses integer L-shaped method to solve subproblems.



MIDAS bounds with 4 periods, 4 reservoirs with levels 0-200, and 3 price states.

Summary

Stochastic optimal control with random prices

The control u seeks to maximize revenue $r_t(x, u, \boldsymbol{\xi}_t) = \pi g(u)$ where π is a random price with AR1 dynamics

$$\pi_{t+1} = \alpha_t \pi_t + (1 - \alpha_t) b_t + \eta_t.$$

We seek an optimal Bellman function $V_t(s_t, \pi_t)$ satisfying

$$\begin{aligned} V_t(s_t, \pi_t) &= \mathbb{E}_{\xi_t}\left[\max_{u \in U(x)} \{r_t(s, u, \boldsymbol{\xi}_t) + V_{t+1}(s_{t+1}\pi_{t+1})\}\right], \\ \begin{bmatrix} s_{t+1} \\ \pi_{t+1} \end{bmatrix} &= \begin{bmatrix} f_{t+1}(s_t, v_t, \omega_t) \\ \alpha_t \pi_t + (1 - \alpha_t) b_t + \eta_t \end{bmatrix} \\ V_{T+1}(s, \pi) &= R(s, \pi). \end{aligned}$$

Outer approximation of continuous monotonic functions

Given a continuous nondecreasing function $Q(x) \leq M$, and a finite set of values

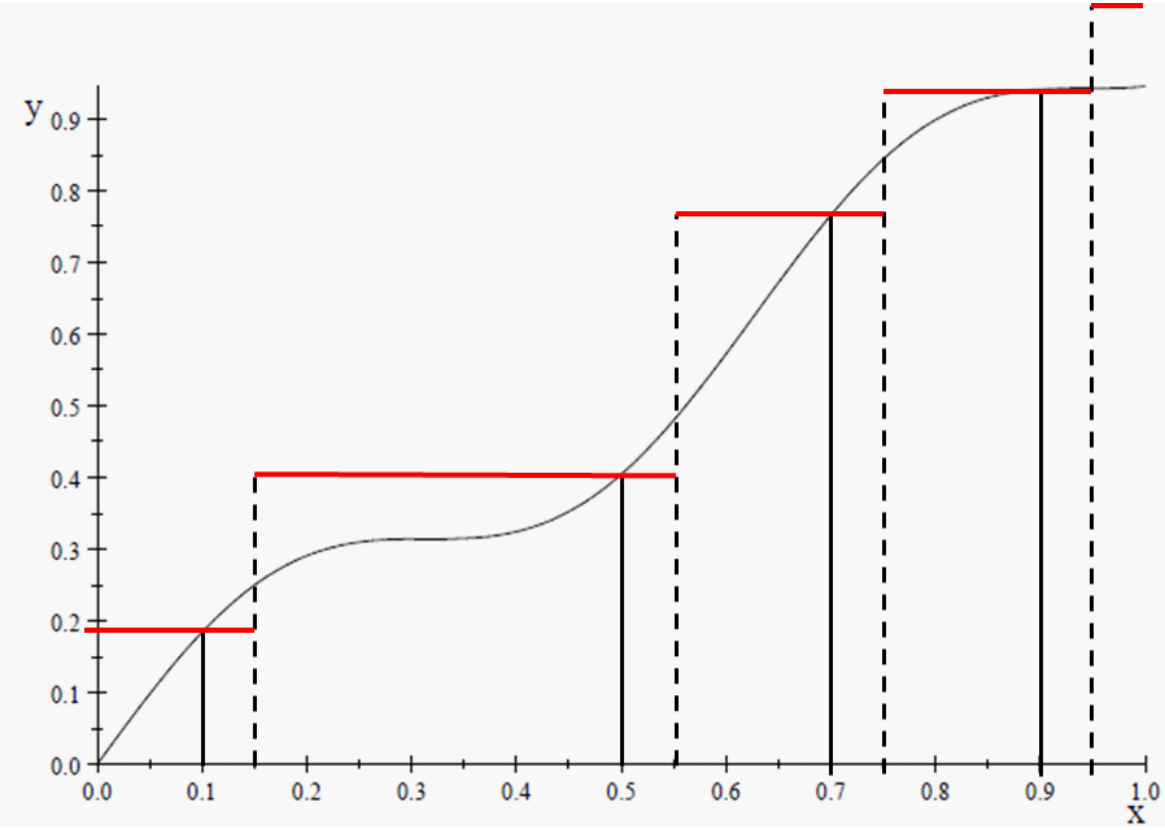
$$Q(x^h) = q^h,\ h = 1, 2, \ldots, H,$$

approximate $Q(x)$ by a piecewise constant function $Q^H(x)$ so that for every x

$$Q(x) \leq Q^H(x) + \varepsilon$$

- q^h is a real number and Q^H is a function;
- $Q(x)$ is assumed monotonic to guarantee that $Q(x) \leq Q^H(x) + \varepsilon$ for every x.

Example



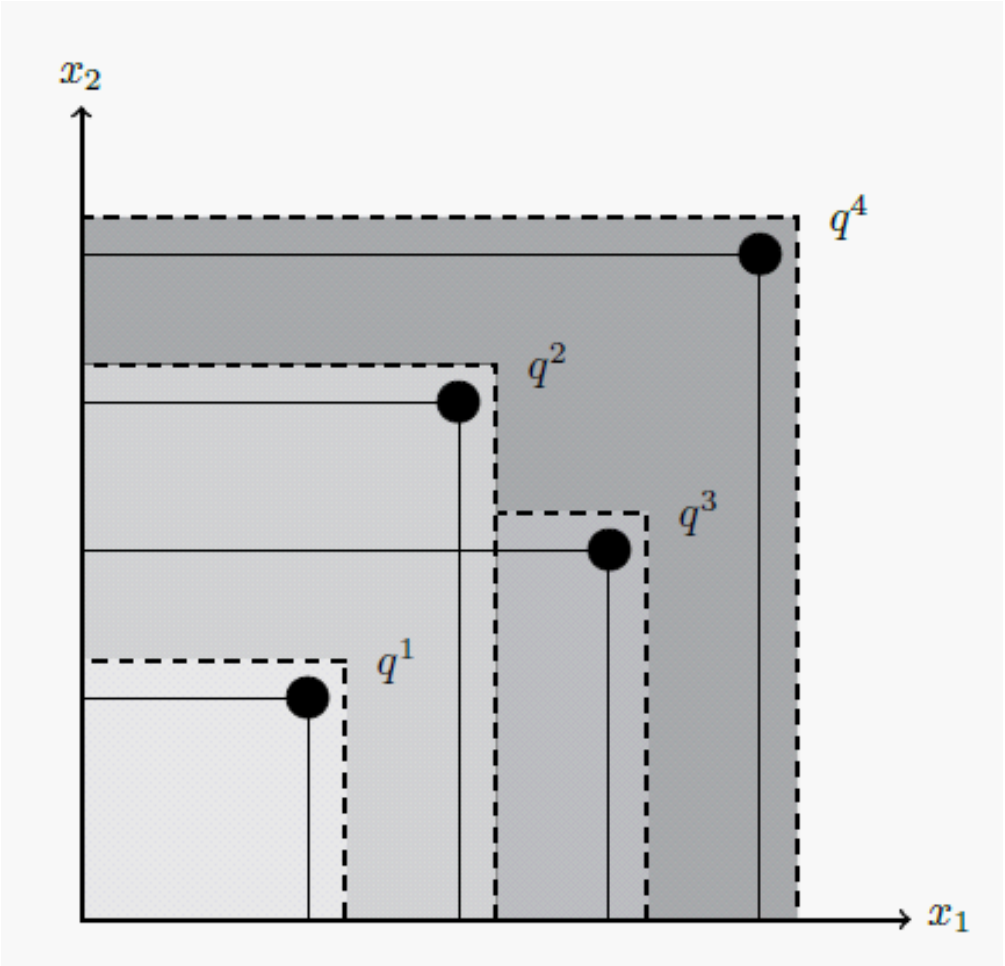
Approximation of $Q(x)$ at points $x^h = 0.1, 0.5, 0.7, 0.9$, and $\delta = 0.05$. $Q^H(x)$ shown in red is upper semicontinuous, and is an upper bound on $Q(x) - \varepsilon$.

MIP approximates a continuous monotonic function

Assume that

$$
\begin{array}{rrcll}
Q^H(x) = \max & \varphi & & & \\
\text{s.t.} & \varphi & \leq & q^h + (M - q^h)y^h, & h = 1, 2, \ldots, H, \\
 & x_k & \geq & (x_k^h + \delta)z_k^h, & k = 1, 2, \ldots, n, \\
 & & & & h = 1, 2, \ldots, H, \\
 & \sum_{k=1}^n z_k^h & \geq & y^h, & h = 1, 2, \ldots, H, \\
 & y^h & \in & \{0, 1\}, & h = 1, 2, \ldots, H, \\
 & z_k^h & \in & \{0, 1\}, & k = 1, 2, \ldots, n, \\
 & & & & h = 1, 2, \ldots, H.
\end{array}
$$

Example in two dimensions



Contour plot of $Q^H(x)$ when $H = 4$. Circled points are x^h, $h = 1, 2, 3, 4$. Darker shading indicates increasing values of $Q^H(x)$ that equals $Q(x^h)$ in each region containing x^h, $h = 1, 2, 3, 4$.

Forward pass

Set $x_0^H = x_0$, and $n = 0$. While $n \notin \mathcal{L}$:

1. Sample $m \in n+$ to give ξ_m^H;
2. Solve $\max_{u \in U(x_n^H)} \left\{ r_m(x_n^H, u) + Q_m^H(f_n(x_n^H, u, \xi_m^H)) \right\}$ to give u_m^H;
3. If $\left\| f_n(x_n^H, u_m^H, \xi_m^H) - x_m^h \right\|_\infty < \delta$ for $h < H$ then set $x_m^{H+1} = x_m^h$, else set $x_m^{H+1} = f_n(x_n^H, u_m^H, \xi_m^H)$;
4. Set $n = m$.

Backward pass

For the particular node $n \in \mathcal{L}$ at the end of forward pass update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = R(x_n^{H+1})$ at point x_n^{H+1}.
While $n > 0$

1. Set $n = n-$;
2. Compute

$$\varphi_n = \sum_{m \in n+} \frac{p(m)}{p(n)} \max_{u \in U(x_n^H)} \left\{ r_m(x_n^{H+1}, u) + Q_m^{H+1}(f_n(x_n^{H+1}, u, \xi_m)) \right\}$$

3. Update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = \varphi_n$ at point x_n^{H+1};

Convergence

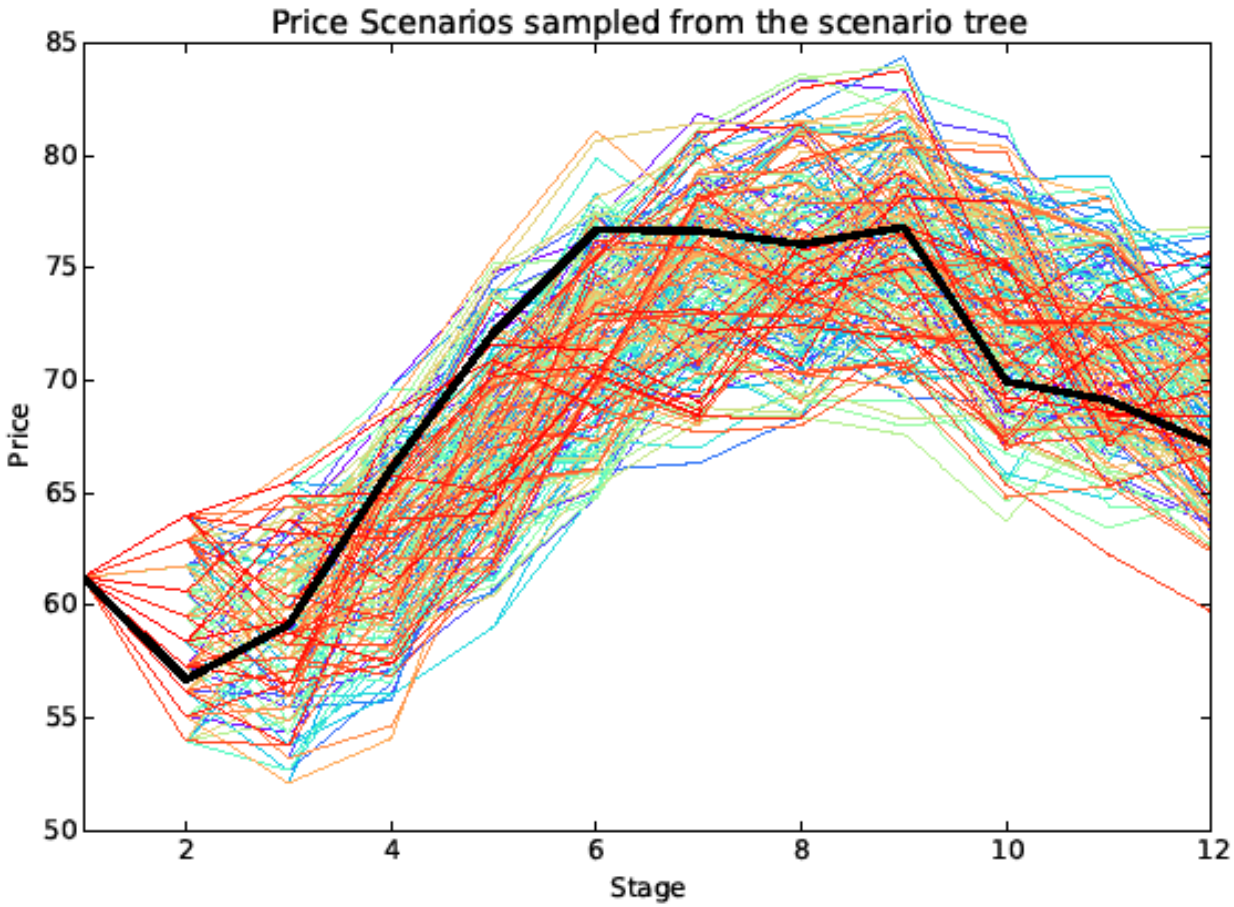
FPSP: For each $n \in \mathcal{L}$, with probability 1

$$\left|\left\{H : \xi_n^H = \xi_n\right\}\right| = \infty.$$

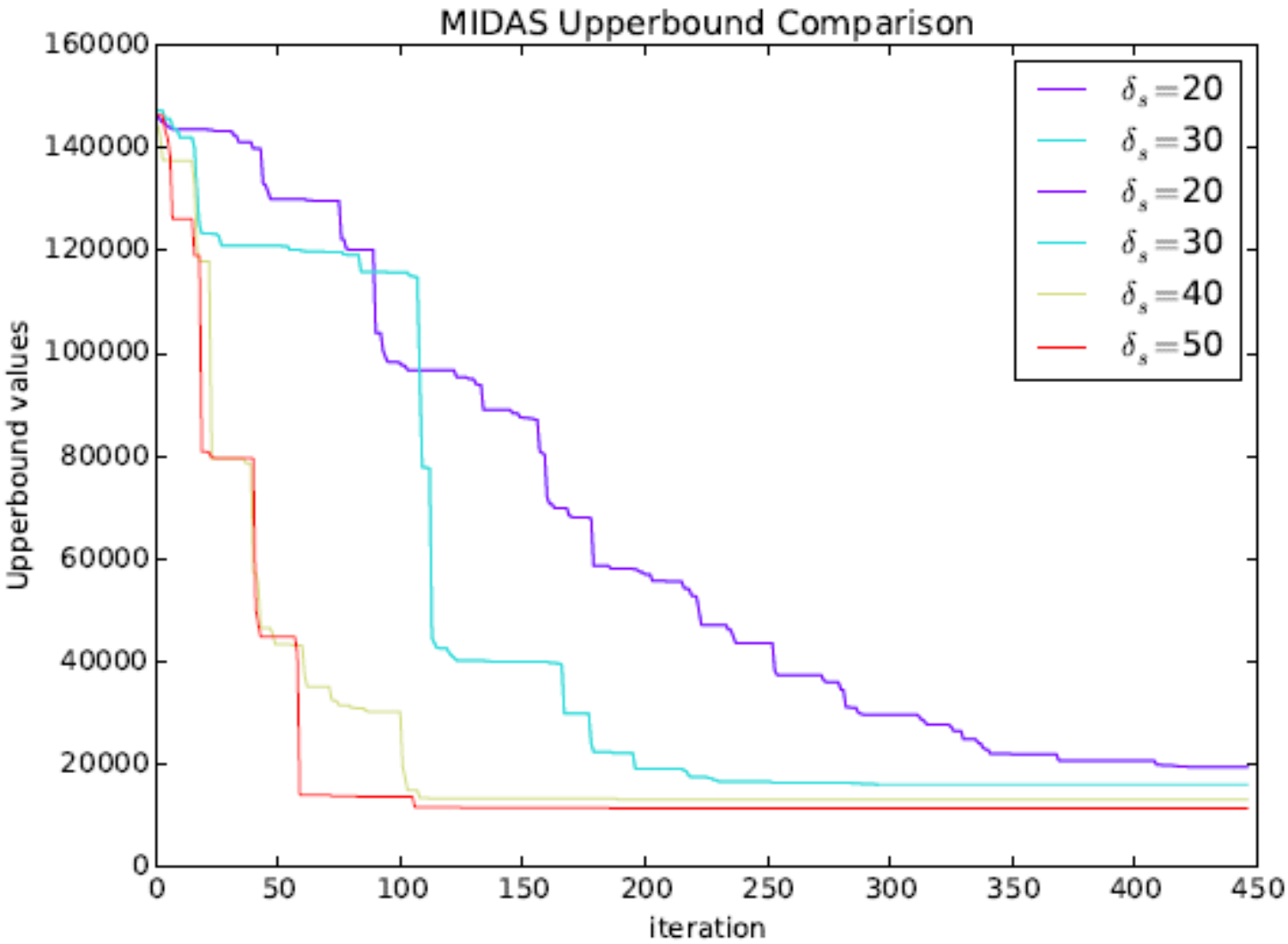
Theorem

If step 1 of forward pass satisfies FPSP then sampled MIDAS converges almost surely to a $(T+1)\varepsilon$-optimal solution.

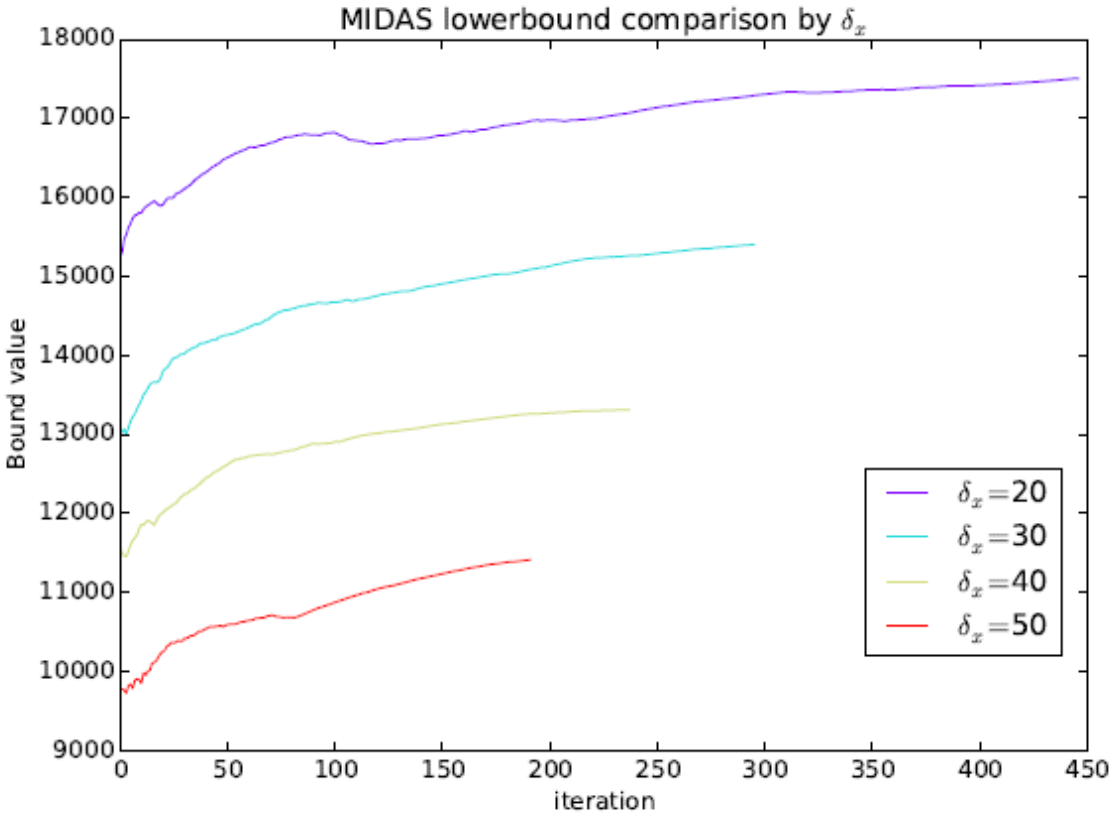
Price scenarios sampled from AR1 model



Epsilon upper bounds for two-reservoir problem

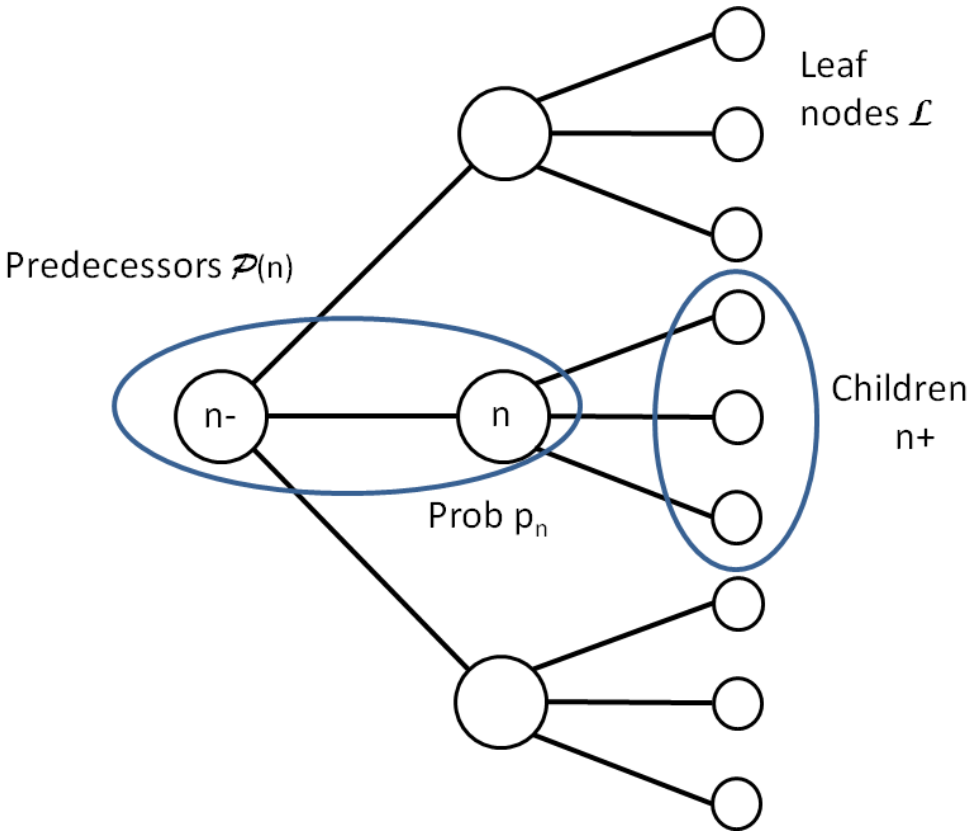


Epsilon lower bounds for two-reservoir problem



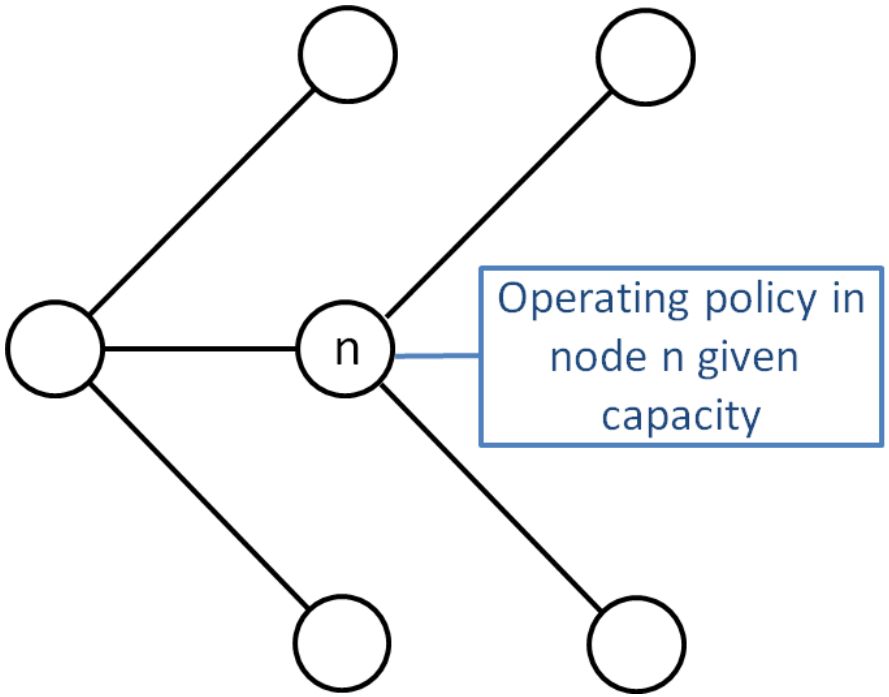
Summary

Scenario tree represents contingent events



Construct a plan of investments in the scenario tree to minimize expected cost.

In each node of the tree we have an operating problem



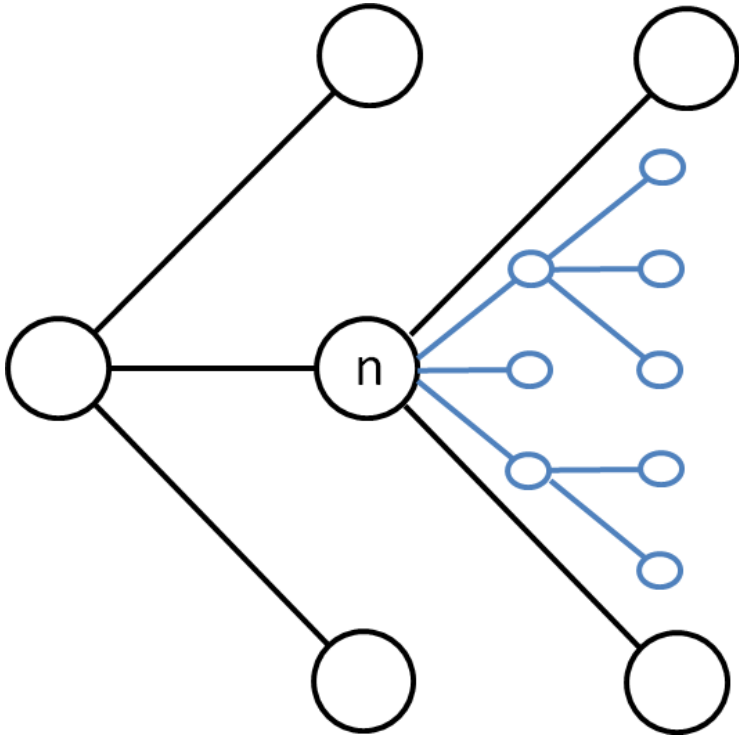
In each node n we solve an operational problem that uses capacity from the history of investments made in n and its predecessor nodes.

Examples of this type of problem

- Investment in newsprint capacity in a paper mill (Everett, P., Cook, 2000).
- Investment in new lines in distribution networks (Singh et al, 2008, 2009)
- Investment in flexible plant to deal with wind turbines (Wu, P and Zakeri, 2016)
- Investment in transmission lines to deal with uncertain demand growth (Florez-Quiroz et al, 2016)
- Investment in transmission switches to deal with intermittent wind power(Villumsen and P. 2012)
- Investment in batteries in distribution networks lines to backup line failures (Dominguez-Martin and P. 2017)

Model as multi-horizon scenario trees

(Kaut et al, 2013; Skar et al, 2016)



In each node n we solve another operating stochastic program that uses capacity from the history of investments made in n and its predecessor nodes.

Dantzig-Wolfe decomposition (general form)

(Singh, P., Wood, 2009)

We have initial capacity $\mathbf{x}_0 \in \mathbb{R}^K$ that we increase with actions $\mathbf{u}_n \in \{0,1\}^Z$ in node n of $\mathcal{N}$. Investment in node n costs $\mathbf{c}_n^\top \mathbf{u}_n$ and contributes additional capacity $U_n \mathbf{u}_n \geq 0$ to the system. So the capacity in node n is $\mathbf{x}_0 + \sum_{h \in \mathcal{P}_n} U_h \mathbf{u}_h$. In each state corresponding to node n we operate our capacity choosing actions $\mathbf{y}_n \in \mathcal{Y}_n$ to minimize cost $\mathbf{q}_n^\top \mathbf{y}_n$.

$$
\begin{aligned}
\min \quad & \textstyle\sum_{n \in \mathcal{N}} p(n) \left(\mathbf{c}_n^\top \mathbf{u}_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \\
\text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{x}_0 + \textstyle\sum_{h \in \mathcal{P}_n} U_h \mathbf{u}_h, \quad n \in \mathcal{N}, \\
& \mathbf{y}_n \in \mathcal{Y}_n, \quad n \in \mathcal{N}, \\
& \mathbf{u}_n \in \{0,1\}^Z, \quad n \in \mathcal{N},
\end{aligned}
$$

Remarks

1. V_n and U_h are matrices of order $K \times Y$ and $K \times Z$
2. The actions $\mathbf{y}_n$ are not affected by any decisions in other nodes apart from capacity actions $\mathbf{u}_h$, $h \in \mathcal{P}_n$. This precludes e.g. inventory being transferred from stage to stage.
3. The constraints $\mathbf{y}_n \in \mathcal{Y}_n$ can be quite general, e.g. include binary variables, nonlinearities.
4. Our approach can solve this problem to quite large scale, as long as we have an efficient method of solving the (almost) single node problem:

$$\begin{aligned} \text{SP}(n)\text{: min} \quad & \phi_n \mathbf{q}_n^\top \mathbf{y}_n - \textstyle\sum_{h \in \mathcal{P}_n} \boldsymbol{\pi}_{hn}^\top \mathbf{u}_h - \boldsymbol{\mu}_n \\ \text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{x}_0 + \textstyle\sum_{h \in \mathcal{P}_n} U_h \mathbf{u}_h, \\ & \mathbf{y}_n \in \mathcal{Y}_n, \quad \mathbf{u}_h \in \{0,1\}^Z, \ h \in \mathcal{P}_n \end{aligned}$$

SP(n) gives a column

Suppose we can solve SP(n) fast to give $\mathbf{u}_h \in \{0,1\}^Z, \quad h \in \mathcal{P}_n$. Then populate $\widehat{\mathbf{u}}_{hn} \in \mathbb{R}^{NZ}$, a column of capacity decisions (one for each node $h \in \mathcal{N}$) with $\mathbf{0}$ if $h \notin \mathcal{P}_n$ and $\mathbf{u}_h$ if $h \in \mathcal{P}_n$. So

$$\widehat{\mathbf{u}}_{hn} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Dantzig-Wolfe master problem

The columns $\widehat{\mathbf{u}}_{hn} \in \mathbb{R}^{NZ}$ form the columns of a Dantzig-Wolfe master problem, and can be generated dynamically with different $\boldsymbol{\pi}_{hn}$ and $\boldsymbol{\mu}_n$.

$$
\begin{array}{lll}
\text{MP:min} & \sum_{n\in\mathcal{N}} \phi_n \mathbf{c}_n^\top \mathbf{u}_n + \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}_n} \phi_n \mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j w_n^j & \\
\text{s.t.} & \sum_{j\in\mathcal{J}_n} \widehat{\mathbf{u}}_{hn}^j w_n^j \leq \mathbf{u}_h \quad h \in \mathcal{P}_n, \quad n \in \mathcal{N}, & [\boldsymbol{\pi}_{hn}] \\
& \sum_{j\in\mathcal{J}_n} w_n^j = 1 \quad n \in \mathcal{N}, & [\boldsymbol{\mu}_n] \\
& w_n^j \in \{0,1\} \quad n \in \mathcal{N},\ j \in \mathcal{J}_n, & \\
& \mathbf{u}_n \in \{0,1\}^Z \quad n \in \mathcal{N}. &
\end{array}
$$

A restricted model

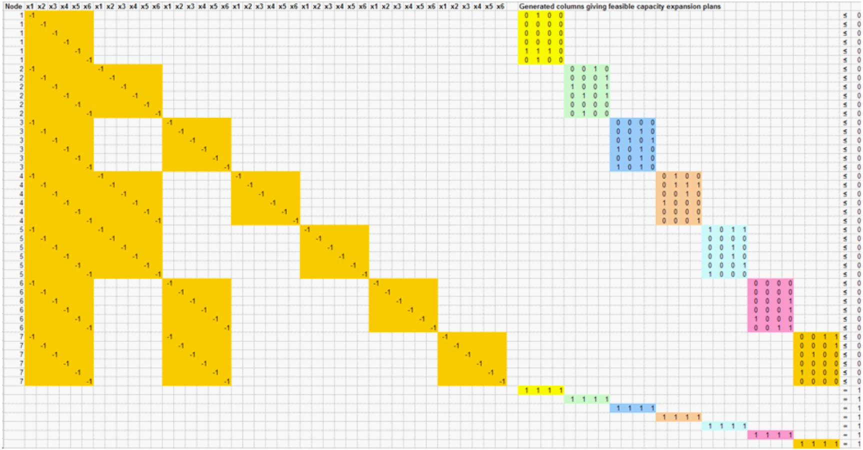
- Suppose only one expansion per capital item is allowed over the time horizon.
- In addition suppose $U_{hn} = U$, so capacity increment matrix is deterministic and independent of time.
- Then the subproblem simplifies to:

$$\begin{aligned} \text{SP}(n)\text{: min} \quad & \phi_n \mathbf{q}_n^\top \mathbf{y}_n - \boldsymbol{\pi}_n^\top \mathbf{u}_n - \boldsymbol{\mu}_n \\ \text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{x}_0 + U \mathbf{u}_n, \\ & \mathbf{y}_n \in \mathcal{Y}_n, \quad \mathbf{u}_n \in \{0,1\}^Z. \end{aligned}$$

Dantzig Wolfe master problem (at most one expansion)

$$
\begin{aligned}
\text{MP:min} \quad & \textstyle\sum_{n\in\mathcal{N}} \phi_n \mathbf{c}_n^\top \mathbf{u}_n + \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}_n} \phi_n \mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j w_n^j \\
\text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{J}_n} \widehat{\mathbf{u}}_n^j w_n^j \leq \sum_{h\in\mathcal{P}_n} \mathbf{u}_n \quad n \in \mathcal{N} && [\boldsymbol{\pi}_n] \\
& \textstyle\sum_{j\in\mathcal{J}_n} w_n^j = 1 \quad n \in \mathcal{N}, && [\boldsymbol{\mu}_n] \\
& w_n^j \in \{0,1\} \quad j \in \mathcal{J}_n, \quad n \in \mathcal{N}, \\
& \mathbf{u}_n \in \{0,1\}^F \quad n \in \mathcal{N}.
\end{aligned}
$$

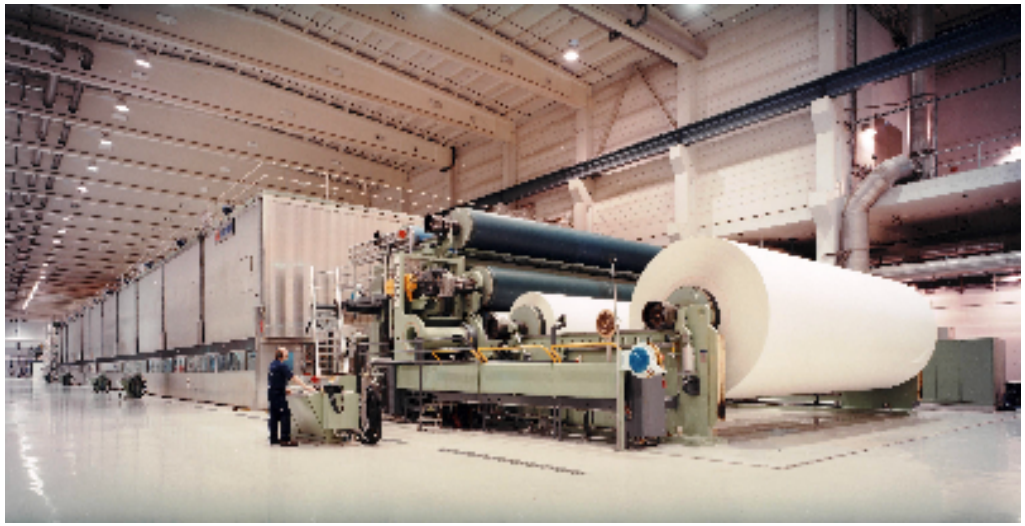
Master problem matrix for 7 nodes (one expansion)



Node	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6
1	-1																																									
1		-1																																								
1			-1																																							
1				-1																																						
1					-1																																					
1						-1																																				
2	-1						-1																																			
2		-1						-1																																		
2			-1						-1																																	
2				-1						-1																																
2					-1						-1																															
2						-1						-1																														
3	-1												-1																													
3		-1												-1																												
3			-1												-1																											
3				-1												-1																										
3					-1												-1																									
3						-1												-1																								
4	-1						-1												-1																							
4		-1						-1												-1																						
4			-1						-1												-1																					
4				-1						-1												-1																				
4					-1						-1												-1																			
4						-1						-1												-1																		
5	-1						-1																		-1																	
5		-1						-1																		-1																
5			-1						-1																		-1															
5				-1						-1																		-1														
5					-1						-1																		-1													
5						-1						-1																		-1												
6	-1												-1																		-1											
6		-1												-1																		-1										
6			-1												-1																		-1									
6				-1												-1																		-1								
6					-1												-1																		-1							
6						-1												-1																		-1						
7	-1												-1																								-1					
7		-1												-1																								-1				
7			-1												-1																								-1			
7				-1												-1																								-1		
7					-1												-1																								-1	
7						-1												-1																								-1

Node	Generated columns giving feasible capacity expansion plans																													
1	0	1	0	0																									≤	0
1	0	0	0	0																									≤	0
1	0	0	0	0																									≤	0
1	0	0	0	0																									≤	0
1	1	1	1	0																									≤	0
1	0	1	0	0																									≤	0
2					0	0	1	0																					≤	0
2					0	0	0	1																					≤	0
2					1	0	0	1																					≤	0
2					0	1	0	1																					≤	0
2					0	0	0	0																					≤	0
2					0	1	0	0																					≤	0
3									0	0	0	0																	≤	0
3									0	0	1	0																	≤	0
3									0	1	0	1																	≤	0
3									1	0	1	0																	≤	0
3									0	0	1	0																	≤	0
3									1	0	1	0																	≤	0
4													0	1	0	0													≤	0
4													0	1	1	1													≤	0
4													0	0	1	0													≤	0
4													1	0	0	0													≤	0
4													0	0	0	0													≤	0
4													0	0	0	1													≤	0
5																	1	0	1	1									≤	0
5																	0	0	0	0									≤	0
5																	0	0	1	0									≤	0
5																	0	0	1	0									≤	0
5																	0	0	0	1									≤	0
5																	1	0	0	0									≤	0
6																					0	0	0	0					≤	0
6																					0	0	0	0					≤	0
6																					0	0	0	1					≤	0
6																					0	0	0	1					≤	0
6																					1	0	0	0					≤	0
6																					0	0	1	1					≤	0
7																									0	0	1	1	≤	0
7																									0	0	0	1	≤	0
7																									0	1	0	0	≤	0
7																									0	0	0	0	≤	0
7																									1	0	0	0	≤	0
7																									0	0	0	0	≤	0
	1	1	1	1																									=	1
					1	1	1	1																					=	1
									1	1	1	1																	=	1
													1	1	1	1													=	1
																	1	1	1	1									=	1
																					1	1	1	1					=	1
																									1	1	1	1	=	1

Paper mill capacity expansion with random demand



- Based on SOCRATES model [Everett, P. Cook, 2001].
- Two paper mills in British Columbia with various capacity expansion options.
- Scheduling the plants over a year is a MIP - part of the PIVOT model [Everett *et al*, 2010].

Some products require new capital items

	1	2	4	5	6	7	8	9	10	11	12	14	15	16	18
CSC6	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1
DW0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DW3	0	1	1	1	1	0	0	0	0	0	0	1	0	1	1
DYCINT0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DYCINT3	0	1	1	1	1	0	0	0	0	0	0	1	0	1	1
DYCNA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DYCNA3	0	1	1	1	1	0	0	0	0	0	0	1	0	1	1
DYHINT0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
DYHINT6	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
DYHNA0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
DYHNA6	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
HB0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HBL0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1
HBL3	0	1	1	1	1	0	1	0	1	0	1	0	0	1	1
HBL6	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
HBLNA6	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
LWC3	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1
LWC6	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1
M0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M3	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1
M6	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1
MR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MR3	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1
MR6	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1
SN0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SN3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SN6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
SNC0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
SNC3	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
SNC6	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
SNL0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1
SNL3	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
SNR0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
SNR3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SNR6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sample product type-capital item matrix for one of the six paper machines.

Problem size

- $M = 6$ machines,
- $J = 35$ products,
- $K = 6$ markets:
- $\mathbf{u}_n$ has 114 binary variables per time period
- $\mathbf{y}_n$ has 210 binary variables per time period
- In 2000, CPLEX 6.5 took 8 hours to solve a single scenario, i.e. $N = 10$.
- In 2009 for three stages and nine scenarios ($N = 13$) CPLEX 9.0 took 43 hours to get a 6% MIP gap.

Computational results of column generation
[Kirch, 2009]

Problem	stages	scenarios	branch and bound (s)	column generation (s)
A2	2	3	3	44
A3	3	9	240	177
A4	4	27	1172	630
A5	5	81	10812	1747
B2	2	2	2	36
B3	3	4	173	153
B6	6	32	5691	872
B8	8	128	-	4361

CPU times from application of column-generation technique to SOCRATES problem with at most one capacity expansion option chosen per scenario, and no capacity shutdowns. [Kirch, 2009].

Capacity expansion in electricity networks with switches
[Villumsen & P. 2011]

- In electricity transmission networks switching out transmission lines can lower dispatch cost.
- Deterministic MIP models for this have been developed [Fisher *et al*, 2008], [Hedman, 2010]
- Two-stage model
  - first stage invests in transmission switches on existing lines;
  - second stage switches to decrease cost in each scenario;
  - minimize overall annual capital cost minus expected annual saving from switching.
- To resolve any fractions we use the COIN DIP branch-and-price code. [Galati, 2009]

Computational results of column generation

Instance		Branch-and-price-and-cut					Branch-and-bound		
		time (s)		price-	cut-	gap	time (s)	gap	lower
$\|\omega\|$	k	total	master	passes	passes				bound
2	3	96	0	10	0	0	547	0	1351.22
2	5	1427	1	20	0	0	330	0	1338.55
4	3	257	4	16	0	0	2310	0	1036.93
4	5	3172	6	38	0	0	7133	0	1033.95
4	10	22444	30	98	0	0	3055	0	1009.60
8	3	978	8	26	1	0	2070 †	2.26	846.17
16	3	3213	11	36	0	0	3722 †	0	775.01
32	3	25477	279	129	2	0	4968 †	9.68	690.81
64	3	11126	72	38	0	0	8589 †	28.84	678.44

Installing transmission switches with random demand on IEEE118-bus network (185 lines) in a two-stage model (Villumsen and P., 2011)

Summary

Conclusions

- Integer stochastic programming is hard. To solve at scale it is better to solve many small MIPs than one huge MIP.
- Progressive hedging seems to be the method of choice for stochastic integer programming. SDDP and Dantzig-Wolfe are attractive alternatives.
- Open questions
  - What is the best stochastic optimization model to use for multistage optimization as measured by out-of-sample performance?
  - How does one reconcile stochastic optimization with incomplete, imperfectly competitive markets? Risk plays a key role here, so connections with social planning are tenuous.
  - Industry are starting to use stochastic operational models. How do we get industry to use stochastic planning models?

References

- Everett, G., Philpott, A.B. and Cook, G. Capital planning under uncertainty at Fletcher Challenge Canada, Proceedings of the 35th Annual Conference of ORSNZ, 2000, 231-240.
- Everett, G., Philpott, A.B., Vatn, K. and Gjessing, R., Norske Skog improves global profitability using OR, *Interfaces*, 40, 58-70, 2010.
- Flores-Quiroz, A., Palma-Behnke, R., Zakeri, G. and Moreno, R. A column generation approach for solving generation expansion planning problems with high re-newable energy penetration, *Electric Power Systems Research*, 136, 232–241, 2016.
- Kaut, M., Midthun, K.T., Werner, A.S., Tomasgard, A., Hellemo, L, and Fodstad, M. Multi-horizon stochastic programming, *Computational Management Science*, 2013.
- Kirch, J.O. Capacity Planning For Process Industries, Proceedings or ORSNZ, 2009.

References

- Pereira M.V.F. and Pinto L.M.V.G. Multi-stage stochastic optimization applied to energy planning. *Math Prog*, 52(1):359–375, 1991.
- Philpott, A.B., Wahid, F. and Bonnans, J.F. MIDAS: A Mixed Integer Dynamic Approximation Scheme. Under review *Mathematical Programming A*, 2016.
- Singh, K., Philpott, A.B. and Wood, K., Column generation for design of survivable networks. Downloadable from www.epoc.org.nz, 2008.
- Singh, K., Philpott, A.B. and Wood, K., Dantzig-Wolfe decomposition for solving multi-stage stochastic capacity planning problems, *Operations Research*, 57, 1271-1286, 2009.
- Skar, C., Doorman, G., Pérez-Valdés,G.A, and Tomasgard,A., A multi-horizon stochastic programming model for the European power system, CenSES working paper 2/2016 March 15, 2016.

References

- Villumsen, J.C. and Philpott, A.B., Investment in electricity networks with transmission switching, *European Journal of Operational Research*, 222, 377–385, 2012.
- Wu, A., Philpott, A. and Zakeri, G., Investment and generation optimization in electricity systems with intermittent supply. *Energy Systems*, 1-21, 2016.
- Zhou, J. Ahmed, S. and Sun, A. Nested decomposition of multistage stochastic integer programs with binary state variables, Optimization Online May 2016.