Decomposition in multistage stochastic integer programming

Andy Philpott
Electric Power Optimization Centre
University of Auckland.
www.epoc.org.nz
“Plans are nothing, planning is everything” Dwight Eisenhower

What did he mean by this?
“... 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.”

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 decision 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 $C_t(x)$, where

$$
C_t(x) = \mathbb{E}_{\xi_t} \left[ \min_{u \in U(x)} \left\{ r_t(x, u, \xi_t) + C_{t+1}(f_t(x, u, \xi_t)) \right\} \right] \quad (1)
$$

$$
C_{T+1}(x) = R(x).
$$

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

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
Summary

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
A scenario tree $\mathcal{N}$. The child nodes of $n$ are denoted $n+$, and parent node $n-$. 

**Leaf nodes $\mathcal{L}$**

**Predecessors $\mathcal{P}(n)$**

**Prob $p_n$**

**Children $n+$**
Tree formulation of stochastic optimization problem

MSPT: \[ \min \sum_{n \in \mathcal{N} \setminus \{0\}} p_n r_n(x_{n-}, u_n) + \sum_{n \in \mathcal{L}} p_n C(x_n) \]
\[ \text{s.t.} \]
\[ x_n = f_n(x_{n-}, u_n, \xi_n), \]
\[ x_0 = \bar{x}, \]
\[ u_n \in U(x_n), \]
\[ x_n \in X_n. \]

Recursive form is:
\[ C_n(x_n) = \sum_{m \in n^+} \frac{p(m)}{p(n)} \min_{u \in U(x_n)} \{ r_m(x_n, u) + C_m(f_n(x_n, u, \xi_m)) \} \]
\[ C_n(x_n) = R(x_n), \quad n \in \mathcal{L}, \]

where we seek \( u_n, \quad n \in \mathcal{N} \setminus \{0\}, \) that minimizes \( C_0(x_0). \)
Example: River chain optimization under uncertainty

The dynamics in node \( n \) of the scenario tree are

\[
x_{n,i} = x_{n-1,i} + u_{n,i-1} - u_{n,i} - s_{n,i} + \zeta_{n,i}.
\]

Given electricity prices \( \pi_n \) 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 $x_0 \in \mathbb{R}^K$ that we increase with actions $u_n \in \{0, 1\}^Z$ in node $n$ of $\mathcal{N}$. Investment in node $n$ costs $c_n^T u_n$ and contributes additional capacity $U_n u_n \geq 0$ to the system. So the capacity in node $n$ is $x_0 + \sum_{h \in \mathcal{P}_n} U_h u_h$. In each state corresponding to node $n$ we operate our capacity choosing actions $y_n \in \mathcal{Y}_n$ to minimize cost $q_n^T y_n$.

$$\min \sum_{n \in \mathcal{N}} \rho_n \left( c_n^T u_n + q_n^T y_n \right)$$

s.t. $V_n y_n \leq x_0 + \sum_{h \in \mathcal{P}_n} U_h u_h$, $n \in \mathcal{N}$

$y_n \in \mathcal{Y}_n$, $n \in \mathcal{N}$

$u_n \in \{0, 1\}^Z$, $n \in \mathcal{N}$
Summary

1. Introduction

2. Multistage stochastic programs

3. **Stochastic dual dynamic programming**

4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS

5. Continuous nonconvex SDDP

6. Capacity expansion planning models
   - Decomposition
   - Applications

7. Conclusions
Stochastic optimal control

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

$$C_t(x) = \mathbb{E}_{\xi_t} \left[ \min_{u \in U(x)} \left\{ r_t(x, u, \xi_t) + C_{t+1}(f_t(x, u, \xi_t)) \right\} \right] ,$$

$$C_{T+1}(x) = R(x).$$

This is a dynamic programming recursion.
Example: stochastic dual dynamic programming (SDDP)

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

Assume $C_t(x)$ is convex.
For cut sharing assume scenario tree represents a stagewise independent process.
Represent an approximately optimal policy by an outer approximation of $C_t(x)$ 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 scheduling);
- 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

\[
P: \min \quad x + y_2 \\
\text{s.t.} \quad x - y_1 = 0 \\
\quad y_1 + y_2 \geq 0 \\
\quad -3y_1 + y_2 \geq -2 \\
\quad x \in [0, 1]
\]

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

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

\[
\begin{align*}
\text{MP: } & \min & x + \theta \\
\text{s.t.} & & x \leq 1 \\
& & \theta \geq -1 \\
& & x \geq 0
\end{align*}
\]

At \(x = 0\), we get

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

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

MP: \( \min \ x + \theta \)
\[
\text{s.t.} \quad x \leq 1 \\
\theta \geq -1 \\
\theta \geq -x \\
x \geq 0
\]

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

$$\text{MP: min } x + \theta$$

s.t.  
$$x \leq 1$$
$$\theta \geq -1$$
$$\theta \geq -x$$
$$x \geq 0.$$  

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

$$\text{SP: min } y_2$$

s.t.  
$$y_1 = x \ [\pi]$$
$$y_1 + y_2 \geq 0$$
$$-3y_1 + y_2 \geq -2$$

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

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
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:

$$
\varphi(a, b) = \min \quad -2u_1 - u_2 + C(x_1, x_2)
$$

s.t.  
$$
x_1 = a - u_1 - s_1 \\
x_2 = b + u_1 - u_2 - s_2 + 1 \\
u_1 + u_2 \leq 1.5 \\
x_i \geq 0, \ u_i \in \mathbb{Z}_+, \ s \geq 0.
$$
Optimal value function

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

$$\phi(a, b) = \min -2u_1 - u_2$$

s.t.

$$x_1 = a - u_1 - s_1$$

$$x_2 = b + u_1 - u_2 - s_2 + 1$$

$$u_1 + u_2 \leq 1.5$$

$$x_i \geq 0, u_i \in \mathbb{Z}_+, s \geq 0.$$

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

<table>
<thead>
<tr>
<th>$\phi(a, b)$</th>
<th>$b = 0$</th>
<th>$b = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a &lt; 1$</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>$a \geq 1$</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>
MIP value function

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

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

$$\varphi^r(a, b) = \min \quad -2u_1 - u_2$$

s.t. $\begin{align*}
x_1 &= a - u_1 - s_1 \\
x_2 &= b + u_1 - u_2 - s_2 + 1 \\
u_1 + u_2 &\leq 1.5 \\
x_i &\geq 0, \ u_i \geq 0, \ s \geq 0.
\end{align*}$

Optimal solution is $u_1 = a, \ u_2 = \min\{a + b + 1, 1.5 - a\}$,
Plot of $\varphi(a, b)$ and LP relaxation $\varphi^r(a, b)$. 
Lagrangian relaxation
(In SDDP see Thome et al, PSR, 2013)

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

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

\[
\varphi^l(a, b) = \max_{\pi} \{ \min(-\pi_2 + \pi_1 - 2, \pi_2 - 1, 0) - a\pi_1 - (b + 1)\pi_2 \}.
\]
Plot of $\varphi(a, b)$, $\varphi_r(a, b)$ and $\varphi_l(a, b)$. Lagrangean relaxation $\varphi_l(a, b)$ gives better approximation to $\varphi(a, b)$ over all $a$ and $b$. 

Plot of relaxations
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.

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

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 $\varphi^l(x)$.
- Subgradients for $\varphi^l(x_n)$ 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 $\varphi^l(x)$.
- SDDP can be proved to converge (a.s.) to an optimal solution. Reported computational results are impressive.
Suppose actions $u$ must be integer.

**Warning:** we are now maximizing revenue

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

Represent an approximately optimal policy by an outer approximation of $V_t(x)$ 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**

\[ V_t(x) \text{ is (outer) approximated by a piecewise constant function } Q^H(x). \text{ Here } q^1_t < q^2_t. \]
Approximation represented by a mixed integer program

\[ Q^H(x) = \max \quad \varphi \quad \text{s.t.} \]
\[ \begin{align*}
\varphi & \leq q^h + (M - q^h)y^h, & h = 1, 2, \ldots, H, \\
x_k & \geq (x^h_k + 1)z^h_k, & k = 1, 2, \ldots, n, \\
\sum_{k=1}^n z^h_k & \geq y^h, & h = 1, 2, \ldots, H, \\
y^h & \in \{0, 1\}, & h = 1, 2, \ldots, H, \\
z^h_k & \in \{0, 1\}, & k = 1, 2, \ldots, n, \\
& & h = 1, 2, \ldots, H.
\end{align*} \]
MIDAS algorithm for stochastic dynamic programming

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

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

1. Sample $m \in n^+$ to give $\xi_m^H$;
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 $Q_{n+1}^H(x)$ by adding $q_{n+1}^H = R(x_n^H)$ at point $x_{n+1}^H$. While $n > 0$

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

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

3. Update $Q_n^H(x)$ to $Q_{n+1}^H(x)$ by adding $q_{n+1}^H = \varphi$ at point $x_{n+1}^H$;
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.*
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

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
The control $u$ seeks to maximize revenue $r_t(x, u, \xi_t) = \pi g(u)$ where $\pi$ 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

$$(s_{t+1}, \pi_{t+1}) = \begin{bmatrix} f_{t+1}(s_t, v_t, \omega_t) \\ \alpha_t \pi_t + (1 - \alpha_t) b_t + \eta_t \end{bmatrix}$$

and $V_{T+1}(s, \pi) = R(s, \pi)$. 

The control $u$ seeks to maximize revenue $r_t(x, u, \xi_t) = \pi g(u)$ where $\pi$ 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

$$(s_{t+1}, \pi_{t+1}) = \begin{bmatrix} f_{t+1}(s_t, v_t, \omega_t) \\ \alpha_t \pi_t + (1 - \alpha_t) b_t + \eta_t \end{bmatrix}$$

and $V_{T+1}(s, \pi) = R(s, \pi)$.
Outer approximation of continuous monotonic functions

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

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

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

\[ Q^H(x) = \max \varphi \]

s.t.

\[ \begin{align*}
\varphi & \leq q^h + (M - q^h)y^h, & h = 1, 2, \ldots, H, \\
x_k & \geq (x_k^h + \delta)z_k^h, & k = 1, 2, \ldots, n, \\
\sum_{k=1}^{n} z_k^h & \geq y^h, & h = 1, 2, \ldots, H, \\
y^h & \in \{0, 1\}, & h = 1, 2, \ldots, H, \\
z_k^h & \in \{0, 1\}, & k = 1, 2, \ldots, n, \\
& & h = 1, 2, \ldots, H.
\]
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 $\xi_m^H$;
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 $x_{m+1}^H = x_m^h$, else set $x_{m+1}^H = f_n(x_n^H, u_m^H, \xi_m^H)$;
4. Set $n = m$. 
Backward pass

For the particular node $n \in \mathcal{L}$ at the end of forward pass update $Q^H_n(x)$ to $Q^{H+1}_n(x)$ by adding $q^{H+1}_n = R(x^{H+1}_n)$ at point $x^{H+1}_n$.

While $n > 0$

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

$$\varphi_n = \sum_{m \in n^+} \frac{p(m)}{p(n)} \max_{u \in U(x^H_n)} \left\{ r_m(x^{H+1}_n, u) + Q^{H+1}_m(f_n(x^{H+1}_n, u, \xi_m)) \right\}$$

3. Update $Q^H_n(x)$ to $Q^{H+1}_n(x)$ by adding $q^{H+1}_n = \varphi_n$ at point $x^{H+1}_n$. 
Convergence

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

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

**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

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
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 $x_0 \in \mathbb{R}^K$ that we increase with actions $u_n \in \{0, 1\}^Z$ in node $n$ of $\mathcal{N}$. Investment in node $n$ costs $c_n^T u_n$ and contributes additional capacity $U_n u_n \geq 0$ to the system. So the capacity in node $n$ is $x_0 + \sum_{h \in \mathcal{P}_n} U_h u_h$. In each state corresponding to node $n$ we operate our capacity choosing actions $y_n \in \mathcal{Y}_n$ to minimize cost $q_n^T y_n$.

$$\begin{align*}
\min & \quad \sum_{n \in \mathcal{N}} p(n) \left( c_n^T u_n + q_n^T y_n \right) \\
\text{s.t.} & \quad V_n y_n \leq x_0 + \sum_{h \in \mathcal{P}_n} U_h u_h, \quad n \in \mathcal{N}, \\
& \quad y_n \in \mathcal{Y}_n, \quad n \in \mathcal{N}, \\
& \quad u_n \in \{0, 1\}^Z, \quad n \in \mathcal{N},
\end{align*}$$
Remarks

1. $V_n$ and $U_h$ are matrices of order $K \times Y$ and $K \times Z$
2. The actions $y_n$ are not affected by any decisions in other nodes apart from capacity actions $u_h$, $h \in P_n$. This precludes e.g. inventory being transferred from stage to stage.
3. The constraints $y_n \in Y_n$ 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:

$$\text{SP}(n): \min \quad \phi_n q_n^\top y_n - \sum_{h \in P_n} \pi_{hn}^\top u_h - \mu_n$$

s.t. \hspace{1cm} $V_n y_n \leq x_0 + \sum_{h \in P_n} U_h u_h,$

$y_n \in Y_n,$ \hspace{0.5cm} $u_h \in \{0,1\}^Z,$ \hspace{0.5cm} $h \in P_n$
SP(n) gives a column

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

\[
\hat{u}_{hn} = \begin{bmatrix}
u_1 \\
\vdots \\
u_n \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
Dantzig-Wolfe master problem

The columns $\hat{u}_{hn} \in \mathbb{R}^{NZ}$ form the columns of a Dantzig-Wolfe master problem, and can be generated dynamically with different $\pi_{hn}$ and $\mu_{n}$.

\[\text{MP: min} \quad \sum_{n \in N} \phi_n \mathbf{c}^\top \mathbf{u}_n + \sum_{n \in N} \sum_{j \in J_n} \phi_n \mathbf{q}_n^\top \hat{y}_n^j w_n^j\]

s.t. \quad \sum_{j \in J_n} \hat{u}_{hn}^j w_n^j \leq u_h \quad h \in P_n, \quad n \in N, \quad [\pi_{hn}] \]

\[\sum_{j \in J_n} w_n^j = 1 \quad n \in N, \quad [\mu_n]\]

$w_n^j \in \{0, 1\} \quad n \in N, \quad j \in J_n,$

$u_n \in \{0, 1\}^Z \quad n \in N.$
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:

$$\text{SP}(n): \min \quad \phi_n q_n^T y_n - \pi_n^T u_n - \mu_n$$

s.t.

$$V_n y_n \leq x_0 + U u_n,$$

$$y_n \in \mathcal{Y}_n, \quad u_n \in \{0,1\}^Z.$$
Dantzig Wolfe master problem (at most one expansion)

\[
\text{MP:} \quad \min \quad \sum_{n \in \mathcal{N}} \phi_n c_n^T u_n + \sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{J}_n} \phi_n q_n^T \hat{y}_n^j w_n^j \\
\text{s.t.} \quad \sum_{j \in \mathcal{J}_n} \hat{u}_n^j w_n^j \leq \sum_{h \in \mathcal{P}_n} u_n^h, \quad n \in \mathcal{N}, \quad [\pi_n] \\
\sum_{j \in \mathcal{J}_n} w_n^j = 1, \quad n \in \mathcal{N}, \quad [\mu_n] \\
w_n^j \in \{0, 1\}, \quad j \in \mathcal{J}_n, \quad n \in \mathcal{N}, \\
u_n \in \{0, 1\}^F, \quad n \in \mathcal{N}.
\]
Master problem matrix for 7 nodes (one expansion)
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

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

- $M = 6$ machines,
- $J = 35$ products,
- $K = 6$ markets:
  - $u_n$ has 114 binary variables per time period
  - $y_n$ 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]

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

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

<table>
<thead>
<tr>
<th>Instance</th>
<th>Branch-and-price-and-cut</th>
<th>Branch-and-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time (s)</td>
<td>master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\omega</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>96</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1427</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>257</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3172</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>22444</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>978</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>3213</td>
</tr>
<tr>
<td>32</td>
<td>3</td>
<td>2547</td>
</tr>
<tr>
<td>64</td>
<td>3</td>
<td>11126</td>
</tr>
</tbody>
</table>

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

1. Introduction
2. Multistage stochastic programs
3. Stochastic dual dynamic programming
4. Integer SDDP
   - Lagrangian relaxation
   - MIDAS
5. Continuous nonconvex SDDP
6. Capacity expansion planning models
   - Decomposition
   - Applications
7. Conclusions
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

References


References

