Stochastic Dual Dynamic Programming

Operations Research

Anthony Papavasiliou

Contents

Table of Contents

The Nested L-Shaped Decomposition Subproblem

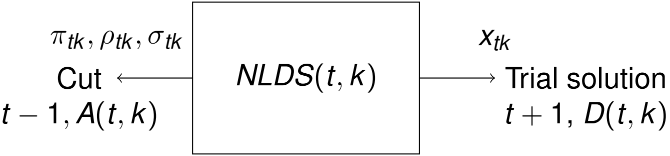
For each stage $t = 1, \ldots, H-1$, scenario $k = 1, \ldots, |\Xi_{[t]}|$

$$
\begin{aligned}
NLDS(t,k): \quad & \min (c_{t,k})^T x + \theta \\
(\pi): \quad & W_{t,k} x = h_{t,k} - T_{t-1,k} x_{t-1,A(t,k)} \\
(\rho_j): \quad & E_{t,k,j} x + \theta \geq e_{t,k,j}, j = 1, \ldots, r_{t,k} \qquad (1) \\
(\sigma_j): \quad & D_{t,k,j} x \geq d_{t,k,j}, j = 1, \ldots, s_{t,k} \qquad (2) \\
& x \geq 0
\end{aligned}
$$

- $\Xi_{[t]}$: support of $\xi_{[t]}$
- $A(t,k)$: ancestor of node k at stage t
- $x_{t-1,A(t,k)}$: current solution from $A(t,k)$
- Constraints (2): feasibility cuts
- Constraints (1): optimality cuts

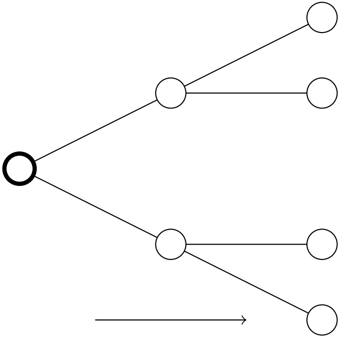
Nested L-Shaped Method

Building block: $NLDS(t, k)$: problem at stage t, scenario k



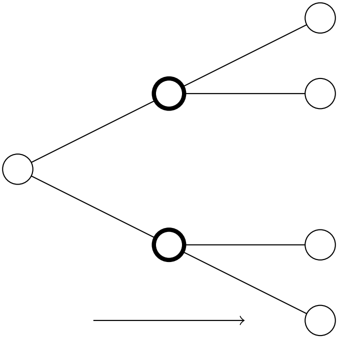
- $A(t, k)$: ancestor of outcome k in period t
- $D(t, k)$: descendants of outcome k in period t

Example



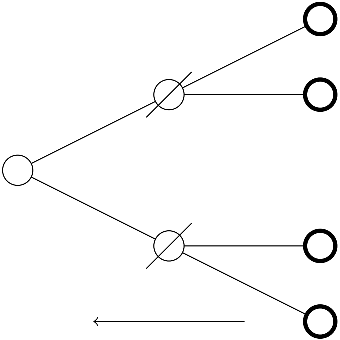
- Node: $(t = 1, k = 1)$
- Direction: forward
- Output: $x_{1,1}$

Example



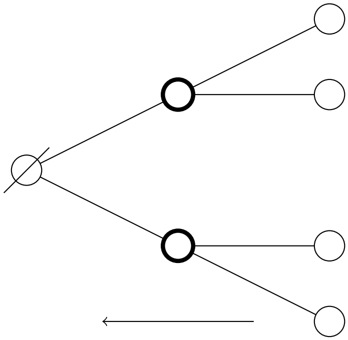
- Nodes: $(t = 2, k)$, $k \in \{1, 2\}$
- Direction: forward
- Output: $x_{2,k}$, $k \in \{1, 2\}$

Example



- Nodes: $(t = 3, k)$, $k \in \{1, 2, 3, 4\}$
- Direction: backward
- Output: $(\pi_{3,k}, \rho_{3,k}, \sigma_{3,k})$, $k \in \{1, 2, 3, 4\}$

Example



- Nodes: $(t = 2, k)$, $k \in \{1, 2\}$
- Direction: backward
- Output: $(\pi_{2,k}, \rho_{2,k}, \sigma_{2,k})$, $k \in \{1, 2\}$

Feasibility Cuts

If $NLDS(t,k)$ is infeasible, solver returns $(\pi, \sigma_1, \ldots, \sigma_{s_{t,k}})$ with $\sigma_j \geq 0, j = 1, \ldots, s_{t,k}$, such that:

- $\pi^T(h_{t,k} - T_{t-1,k}x_{t-1,A(t,k)}) + \sum_{j=1}^{s_{t,k}} \sigma_j^T d_{t,k,j} > 0$
- $\pi^T W_{t,k} + \sum_{j=1}^{s_{t,k}} \sigma_j^T D_{t,k,j} \leq 0$

The following is a valid feasibility cut for $NLDS(t-1, a(k))$:

$$(FC) : D_{t-1,A(t,k)}x \leq d_{t-1,A(t,k)}$$

where

$$\begin{aligned} D_{t-1,A(t,k)} &= \pi^T T_{t-1,k} \\ d_{t-1,A(t,k)} &= \pi^T h_{tk} + \sum_{j=1}^{s_{t,k}} \sigma_j^T d_{t,k,j} \end{aligned}$$

Optimality Cuts

For all $k \in D_{t-1,j}$, solve $NLDS(t,k)$, then compute

$$E_{t-1,j} = \sum_{k \in D(t-1,j)} p_t(k|j) \cdot (\pi_{t,k})^T T_{t-1,k}$$

$$e_{t-1,j} = \sum_{k \in D(t-1,j)} p_t(k|j) \cdot ((\pi_{t,k})^T h_{t,k} +$$

$$\sum_{i=1}^{r_{t,k}} \rho_{t,k,i}^T e_{t,k,i} + \sum_{i=1}^{s_{t,k}} \sigma_{t,k,i}^T d_{t,k,i})$$

The following is an optimality cut for $NLDS(t-1,j)$:

$$E_{t-1,j} x + \theta \geq e_{t-1,j}$$

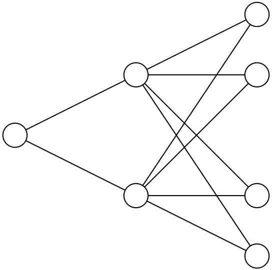
Table of Contents

Nested Decomposition Is Non-Scalable

Assume

- H time steps, $|S_t|$ discrete outcomes in each stage
- No infeasibility cuts

$|S_1| = 1 \quad |S_2| = 2 \quad |S_3| = 4$



- Forward pass: $|S_1| + |S_1| \cdot |S_2| + \ldots = \sum_{t=1}^{H-1} \Pi_{j=1}^{t} |S_j|$
- Backward pass: $\sum_{t=2}^{H} \Pi_{j=1}^{t} |S_j|$

Was Nested Decomposition any Good?

Alternative to nested decomposition is extended form

- Extended form will not even load in memory
- Nested decomposition will load in memory, but will not terminate (for large problems)

But: nested Decomposition lays the foundations for SDDP

Making Nested Decomposition Scalable

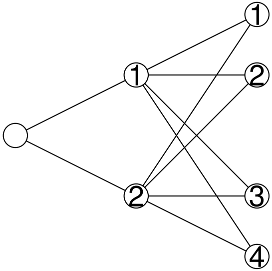
Solution for forward pass

- In the forward pass, we *simulate* instead of *enumerating*
- This results in a probabilistic upper bound / termination criterion

Solutions for backward pass

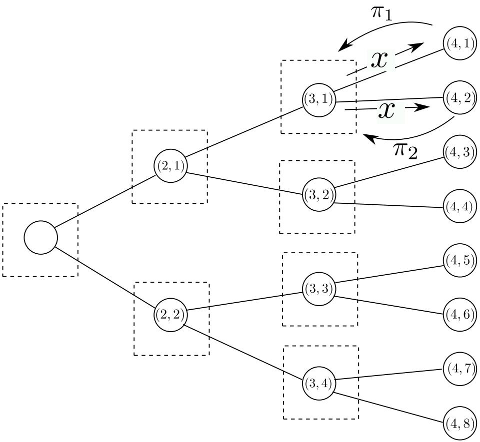
- In the backward pass, we share cuts among nodes of the same time period
- This can only be done on a lattice

Enumerating Versus Simulating



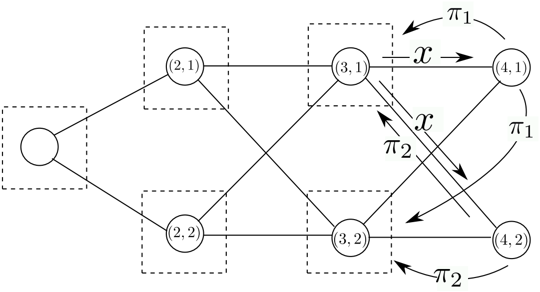
- Enumeration: {(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4))}
- Simulation (with 3 samples): {(1, 3), (2, 1), (1, 4)}

Scenario Tree without Cut Sharing



Dashed box represents storage of a different value function

Cut Sharing in a Lattice



Dashed box represents storage of a different value function

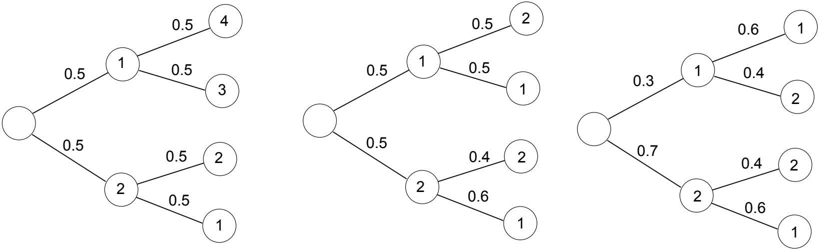
Serial Independence

A process satisfies **serial independence** if, for every stage t, ξ_t has a probability distribution that does not depend on the history of the process, i.e. one can define a probability measure $p_t(i)$ at each stage t, such that

$$\mathbb{P}[\xi_t(\omega) = i | \xi_{[t-1]}(\omega)] = p_t(i), \forall \xi_{[t-1]} \in \Xi_{[t-1]}, i \in \Xi_t$$

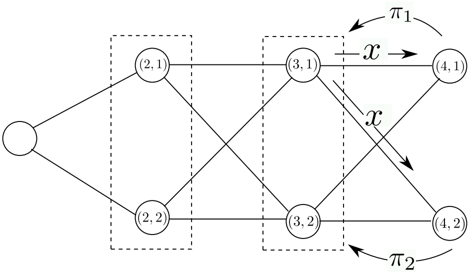
Checking for Serial Independence

Values on arcs indicate transition probabilities, values in nodes indicate realization of ξ_t



Which scenario tree(s) obey(s) serial independence

Cut Sharing with Serial Independence



Dashed box represents storage of a different value function

Intuition: problem is identical from *t* onwards, *independently* of node *k* in stage *t*

Table of Contents

Combining Sampling and Optimization in SDDP

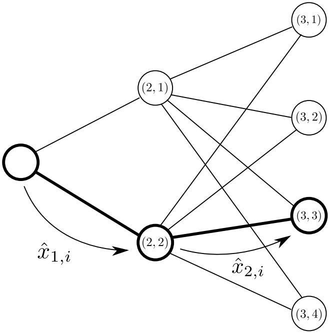
Sampling: Generate K samples of random process $(\xi_{1,i}, \ldots, \xi_{H,i})$, $i = 1, \ldots, K$

Optimization: Solve *NLDS* in order to generate trial decisions $\hat{x}_{t,i}$:

$$
\begin{aligned}
& \min c_{t,k}^T x + \theta \\
(\pi): \quad & T_{t-1,k}\hat{x}_{t-1,i} + W_{t,k}x = h_{t,k} \\
(\rho): \quad & E_{t,k}x + \theta \cdot 1 \geq e_{t,k} \\
& x \geq 0
\end{aligned}
$$

Implications for Forward Pass

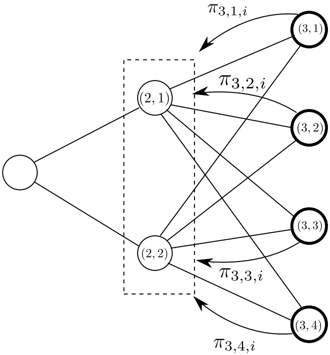
Denote $\hat{x}_{t,i}$ as trial decision



- At each forward pass, we solve $H - 1$ NLDS problems
- For K samples of $\xi_{[H]}$, we solve $1 + K \cdot (H - 2)$ linear programs

Implications for Backward Pass

Denote $(\pi_{t,k,i}, \rho_{t,k,i})$ as dual multipliers generated by trial i



- For a given trial sequence $x_{[H]}$, solve $\sum_{t=2}^{H} |\Xi_t|$ linear programs
- For K trial sequences, solve $K \sum_{t=2}^{H} |\Xi_t|$ linear programs

SDDP Forward Pass

- Solve $NLDS(1)$. Let x_1 be the optimal solution. Initialize $\hat{x}_{1,i} = x_1$ for $i = 1, \ldots, K$
- Repeat for $t = 2, \ldots, H$, $i = 1, \ldots, K$
  - Sample an outcome $\xi_{t,i}$ from the set Ξ_t
  - Solve $NLDS(t, i)$ with trial decision $\hat{x}_{t-1,i}$
  - Store the optimal solution as $\hat{x}_{t,i}$

SDDP Backward Pass

- Repeat for $t = H, H-1, \ldots, 2$
  - Repeat for $i = 1, \ldots, K$
    - Repeat for $k = 1, \ldots, |\Xi_t|$
      Solve $NLDS(t, k)$ with trial decision $\hat{x}_{t-1,i}$
    - For all $j = 1, \ldots, |\Xi_{t-1}|$, compute

$$E_{t-1,j,i} = \sum_{k=1}^{|\Xi_t|} p_t(k|j) \cdot \pi_{t,k,i}^T T_{t-1,k},$$

$$e_{t-1,i,j} = \sum_{k=1}^{|\Xi_t|} p_t(k|j) \cdot (\pi_{t,k,i}^T h_{t,k} + \rho_{t,k,i}^T e_{t,k})$$

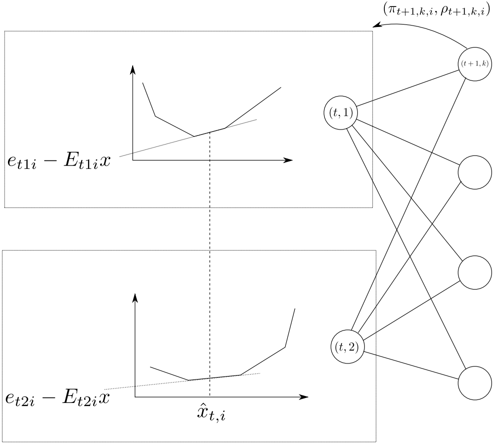
    - Add the optimality cut

$$E_{t-1,j,i} x + \theta \geq e_{t-1,j,i}$$

      to *every* $NLDS(t-1, j)$, $j = 1, \ldots, |\Xi_{t-1}|$

Reusing Multipliers

The propagation of cuts does not require serial independence, can also be done on a lattice



Number of Forward Samples K

Increasing K implies

- faster learning of value function (+)
- more LPs solved at each forward-backward pass (-)
- faster growth of *NLDS* (-)

Want to use large K in later forward passes (why?)

Table of Contents

Terminating SDDP

We have argued that terminating nested decomposition with an exact solution is impractical

Alternative: terminate when upper bound $\simeq$ upper bound

- Lower bound: objective function value of $NLDS(1)$, since $NLDS(1)$ solves for
  - underestimate of $V_2(x)$
  - in superset of dom $V_2(x)$

$$\underline{z} = \min_{x,\theta} c_1^T x + \theta$$

$$\text{s.t. } Ax = b$$

$$E_1 x + \theta \cdot 1 \geq e_1$$

$$x \geq 0$$

- Upper bound: <u>probabilistic</u>

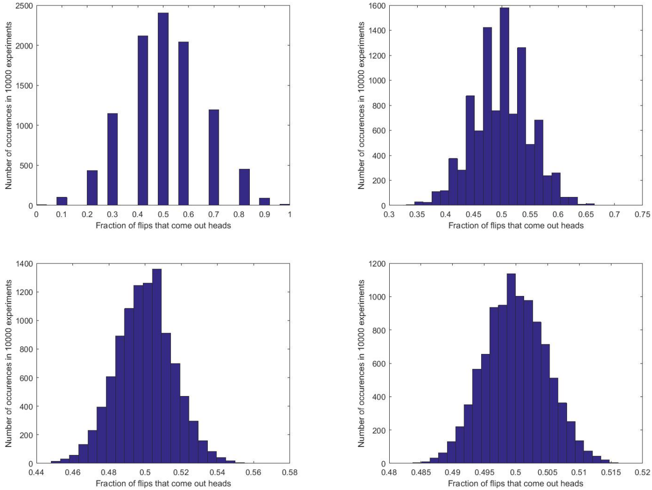
Central Limit Theorem

Suppose $\{X_1, X_2, ...\}$ is a sequence of independent identically distributed random variables with $\mathbb{E}[X_i] = \mu$ and $Var[X_i] = \sigma^2 < \infty$. Then

$$\sqrt{n}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu\right) \xrightarrow{d} N(0,\ \sigma^2).$$

where $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2

Example: Flipping Coins



- Flip a coin K times and count fraction of heads
- Repeat 10000 times, record histogram

Probabilistic Upper Bound

Suppose we draw a sample i of $\xi_{[H]}$ and perform a forward pass

- This gives us a vector $\hat{x}_{t,i}$, $t = 1, \ldots, H$
- We can compute a cost for this vector $z_k = \sum_{t=1}^{H} c_{t,i}\hat{x}_{t,i}$
- If we repeat this K times, we get a distribution of independent, identically distributed costs $z_i, i = 1, \ldots, K$
- By the Central Limit Theorem, $\bar{z} = \frac{1}{K}\sum_{i=1}^{K} z_i$ converges to a Gaussian with standard deviation estimated by

$$\sigma = \sqrt{\left(\frac{1}{K^2}\right)\sum_{k=1}^{K}(\bar{z} - z_i)^2}$$

- Each sequence $\hat{x}_{[H]}$ is feasible, but not necessarily optimal, so $\bar{z}$ is an estimate of an upper bound

Bounds and Pereira Termination Criterion

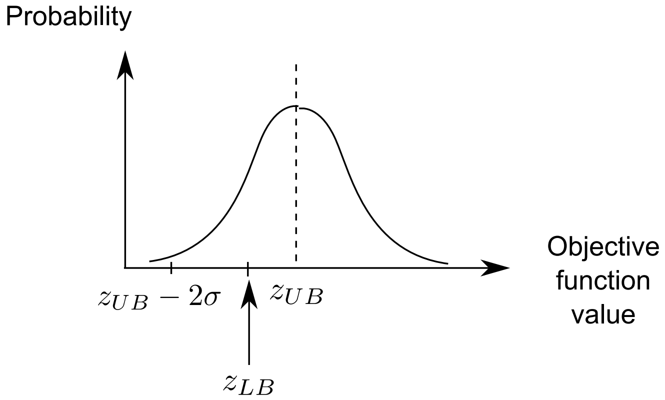
After solving NLDS(1) *in a forward pass*, we can compute a lower bound $\bar{z}$

After completing a forward pass, we can compute

$$
\begin{aligned}
z_i &= \sum_{t=1}^{H} c_{t,i} \hat{x}_{t,i} \\
\bar{z} &= \frac{1}{K} \sum_{i=1}^{K} z_i \\
\sigma &= \sqrt{\frac{1}{K^2} \sum_{i=1}^{K} (z_i - \bar{z})^2}
\end{aligned}
$$

Terminate if $\underline{z} \in (\bar{z} - 2\sigma, \bar{z} + 2\sigma)$, which is the 95.4% confidence interval of $\bar{z}$

Graphical Illustration of Pereira Termination Criterion



Size of Monte Carlo Sample

How can we ensure 1% optimality gap with 95.4% confidence?

- Choose K such that $2\sigma \simeq 0.01 \cdot \bar{z}$
- Mean $\bar{z}$ and variance s^2 depend (asymptotically) on the statistical properties of the process, not K

$$\bar{z} = \frac{1}{K}\sum_{i=1}^{K} z_i$$

$$s = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(z_i - \bar{z})^2} \Rightarrow \sigma = \frac{1}{\sqrt{K}} s$$

- Set

$$K \simeq \left(\frac{2 \cdot s}{0.01 \cdot \bar{z}}\right)^2$$

Full SDDP Algorithm

- Initialize: $\bar{z} = \infty$, $\sigma = 0$
- Forward pass
  - Store z^{LB} and $\bar{z}$
  - If $z^{LB} \in (\bar{z} - 2\sigma, \bar{z} + 2\sigma)$ terminate, else go to backward pass
- Backward pass
- Go to forward pass

Table of Contents

Lattice

We will use a lattice with

- 24 stages (1 month per stage)
- 20 nodes, random disturbance $w_{t,k}$ takes a value in $\{1, 2, \ldots, 20\}$
- uniformly distributed ($w_{t,k}$ equally likely to take any value in $\{1, 2, \ldots, 20\}$)

in order to develop three different rainfall models:

- Uniformly distributed rainfall
- Additive autoregressive rainfall
- Multiplicative autoregressive rainfall

Uniform Rainfall on a Lattice

Consider uniform i.i.d. rainfall over [0, 1000] MW

Discrete approximation for rainfall on node (t, k) of lattice:

$$R_{t,k} = 50 \cdot w_{t,k}$$

Additive Autoregressive Model for Rainfall

Consider additive autoregressive model for rainfall:

$$R_t = R_{t-1} + 100 \cdot \epsilon_t,$$

with ϵ_t standard normal independent identically distributed

Discrete approximation for rainfall on node (t, k) of the lattice:

$$R_{t,k} = R_{t-1} + 100 \cdot \Phi^{-1}(\frac{w_{t,k}}{20} - 0.025),$$

Multipicative Autoregressive Model for Rainfall

Consider multiplicative autoregressive model for rainfall:

$$R_t = R_{t-1} \cdot \eta_t,$$

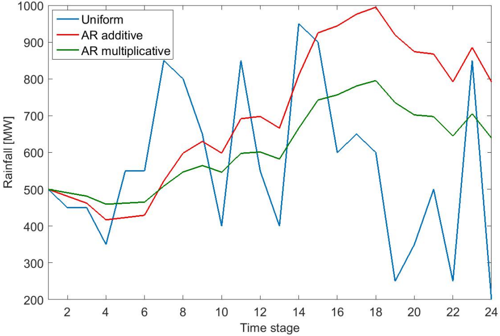
with η_t normal independent identically distributed, with mean 1, standard deviation 0.1

Discrete approximation for rainfall on node (t, k) of the lattice:

$$R_{t,k} = R_{t-1} \cdot (1 + 0.1 \cdot \Phi^{-1}(\frac{w_{t,k}}{20} - 0.025)),$$

Example: Lattice Model for Rainfall

Initial rainfall: 500 MW



- *Same* lattice for all models
- Rain trajectories correspond to *identical* trajectory of $w_{[t]}$ on the lattice

Problem Data

Consider the following hydrothermal planning problem:

- Horizon: 24 months
- Time step: 1 month
- Constant demand: 1000 MW
- Marginal cost of thermal production: 25 $/MWh
- Capacity of thermal units: 500 MW
- Value of lost load: 1000 $/MWh
- Reservoir storage capacity: 1000 MWh
- Initial reservoir level: 700 MWh

Notation

- p, q: thermal/hydro production
- l: unserved demand
- x_t: amount of stored hydro at the *beginning* of period t

The *NLDS* for Uniformly Distributed Rainfall

Assume i.i.d. rainfall, uniformly distributed in [0, 1000] MW

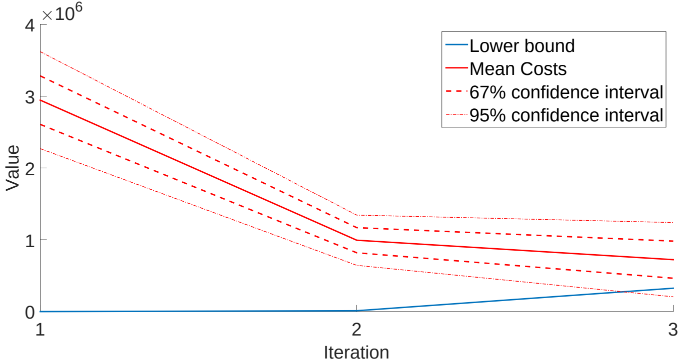
$$
\begin{aligned}
NLDS(t,k) = \quad & \min 1000 \cdot l + 25 \cdot p \\
& \text{s.t. } l + p + q \geq 1000 \\
& p \leq 500 \\
& q \leq x_{t-1} + R_{t,k} \\
& x = x_{t-1} + R_{t,k} - q \\
& x \leq 1000 \\
& l, p, q, x \geq 0
\end{aligned}
$$

where $R_{t,k}$ is the rainfall

Premature Convergence

Settings:

- Convergence for $K = 10$ samples in forward pass
- Confidence interval: $(\bar{z} - 2\sigma, \bar{z} + 2\sigma)$



Termination information:

- $\underline{z}$ = 324,594 \$
- $\bar{z}$ = 722,350 \$
- s = 817,923 \$

which satisfies criterion $\underline{z} = (\bar{z} - 2\frac{s}{\sqrt{K}}, \bar{z} + 2\frac{s}{\sqrt{K}})$

... however, running a forward pass with $K = 200$ samples *after* convergence gives very different estimates:

- $\underline{z}$ = 657,697 \$
- $\bar{z}$ = 770,440 \$
- s = 602,680 \$

which violates criterion $\underline{z} = (\bar{z} - 2\frac{s}{\sqrt{K}}, \bar{z} + 2\frac{s}{\sqrt{K}})$

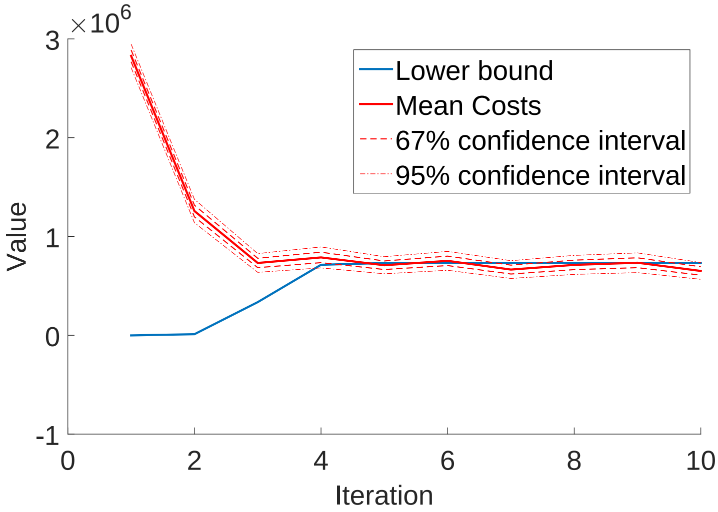
Conclusion: small K can lead to premature convergence due to large confidence interval, *not* $\underline{z} \simeq \bar{z}$

Selecting Appropriate K

Select K so as to achieve a 15% optimality criterion with 95% confidence:

$$K = (\frac{2 \cdot s}{0.15 \cdot \bar{z}})^2 = (\frac{2 \cdot 602,680}{0.15 \cdot 770,440})^2 \simeq 109$$

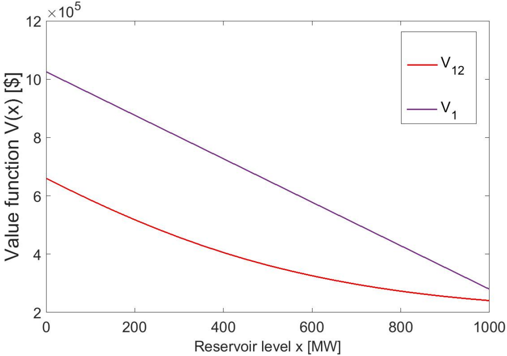
- Monte Carlo sample size: $K = 150$
- Convergence in 4 iterations, after which point $\bar{z}$ stabilizes



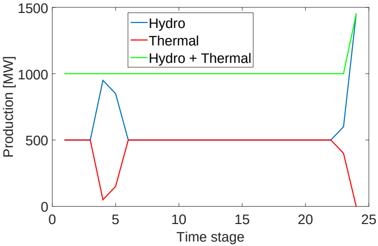
Value Functions

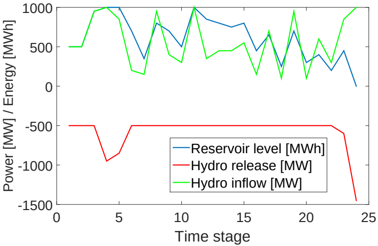
Note:

- $V_1(x) \geq V_{12}(x)$
- V_1 exhibits constant slope, V_{12} is more "interesting"



Dispatch Policy





Left panel: dispatch of hydro and thermal power

Right panel: management of reservoir level

The *NLDS* for Additive Autoregressive Rainfall

$$
\begin{aligned}
NLDS(t,k): \quad & \min 1000 \cdot l + 25 \cdot p \\
& \text{s.t. } l + p + q \geq 1000 \\
& p \leq 500 \\
& q \leq x_{t-1} + r \\
& x = x_{t-1} + r - q \\
& x \leq 1000 \\
& r \leq r_{t-1} + h_{t,k} \\
& l, p \geq 0
\end{aligned}
$$

Some Observations on the *NLDS*

- Stochastic disturbance: $h_{t,k} = 100\Phi^{-1}(\frac{w_{t,k}}{20} - 0.025)$
- New state variable r_t: rainfall in *beginning* of period t
- Non-negativity of x, q, r has been lifted in order to ensure feasibility

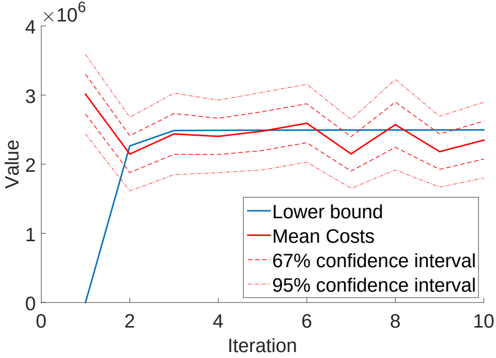
The model can capture temporal dependency of rainfall, but ...

- State vector dimension increases
- The rainfall may become negative!

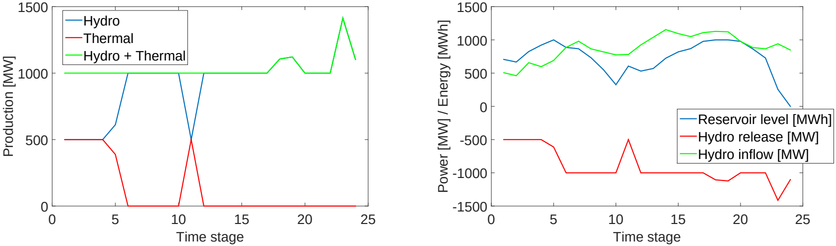
Convergence with Additive Autoregressive Rainfall

SDDP settings:

- $K = 150$
- Confidence interval: $(\bar{z} - 2\sigma, \bar{z} + 2\sigma)$



Dispatch Policy



Left panel: dispatch of hydro and thermal power

Right panel: management of reservoir level

Observations on Dispatch Policy

- Optimal policy attempts to balance minimization of thermal generator dispatch with risk of depleting reservoir capacity
- Autoregressive behavior of the rainfall results in more aggressive dispatch of hydro in periods of high rainfall (e.g. periods 2-5)
- Unfavorable rainfall outcomes may result in *negative* hydro dispatch q and rainfall r $\rightarrow$ can be corrected by multiplicative autoregressive models

The *NLDS* for Multiplicative Autoregressive Rainfall

$$
\begin{aligned}
NLDS(t,k) = \quad & \min 1000 \cdot l + 25 \cdot p \\
& \text{s.t. } l + p + q \geq 1000 \\
& p \leq 500 \\
& q \leq x_{t-1} + r \\
& x = x_{t-1} + r - q \\
& x \leq 1000 \\
& r = r_{t-1} \cdot h_{t,k} \\
& l, p, q, r, x \geq 0
\end{aligned}
$$

Some Observations on the *NLDS*

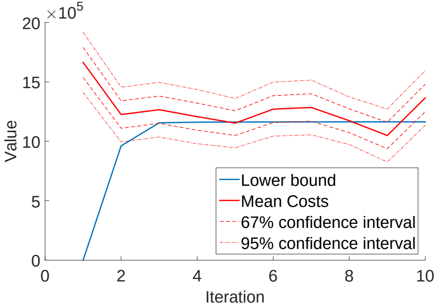
- Stochastic disturbance: $h_{t,k} = 1 + 0.1 \cdot \Phi^{-1}(\frac{w_{t,k}}{20} - 0.025)$
- State variable r_t: rainfall in *beginning* of period t
- The variables x, q, r are non-negative, without causing *NLDS* to be infeasible

The model can capture temporal dependency of rainfall, without rainfall becoming negative

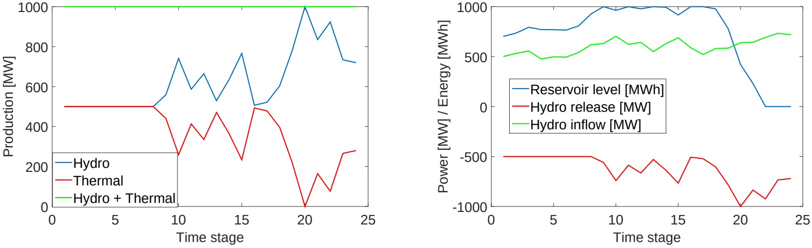
Convergence with Multiplicative Autoregressive Rainfall

SDDP settings:

- $K = 150$
- Confidence interval: $(\bar{z} - 2\sigma, \bar{z} + 2\sigma)$



Dispatch Policy



Left panel: dispatch of hydro and thermal power

Right panel: management of reservoir level