Stochastic optimization
Stochastic dual dynamic programming

Fabian Bastin
bastin@iro.umontreal.ca
Université de Montréal – CIRRELT – IVADO – Fin-ML

Context

Based on Anthony Papavisiliou's slides.

Introduced by Perreia and Pinto, 1991. First designed to solve hydro-scheduling problems.

- Stochastic dynamic problem with finite horizon.
- Continuous, finite dimensional state and control.
- Convex cost, additive over stages – we focus here on linear costs.
- Discrete, independent noises

Julia implementation: SDDP.jl

The Nested L-Shaped Decomposition Subproblem

For each stage $t = 1, \ldots, H-1$, scenario $k = 1, \ldots, \mathcal{S}_t$, we consider the problem $NLSD(t, k)$

$$\begin{aligned}
\min_{x_{t,k}, \theta_{t,k}} \quad & c_{t,k}^T x_{t,k} + \theta_{t,k} \\
\text{s.t.} \quad & W_t x_{t,k} = h_{t,k} - T_{t-1,k} x_{t-1,a(k)}, (\pi_{t,k}) \\
& D_{t,k,j} x_{t,k} \geq d_{t,k,j}, \ j = 1, \ldots, r_{t,k}, (\rho_{t,k}) \qquad (1) \\
& E_{t,k,j} x_{t,k} + \theta_{t,k} \geq e_{t,k,j}, \ j = 1, \ldots, s_{t,k}, (\sigma_{t,k}) \qquad (2) \\
& x_{t,k} \geq 0.
\end{aligned}$$

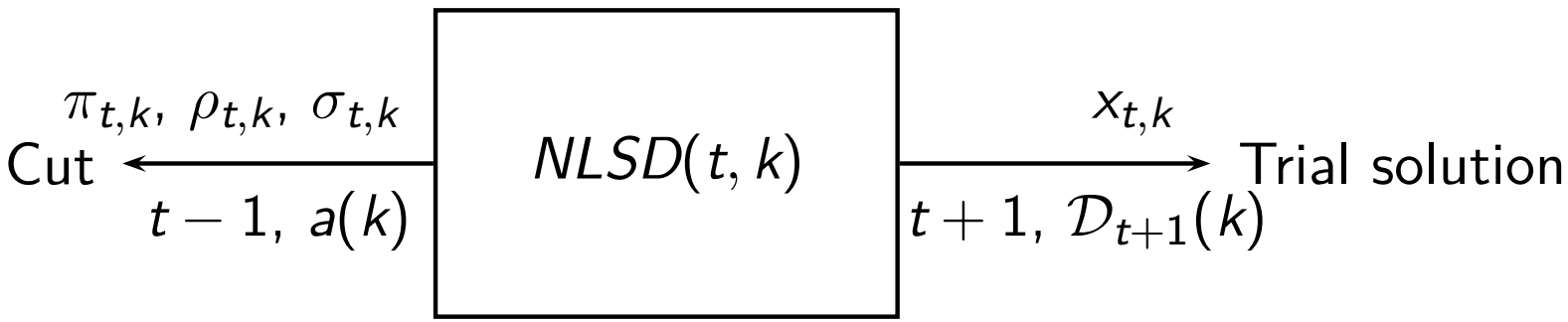
- $\mathcal{S}_t$: number of distinct scenarios at stage t,
- $a(k)$: ancestor of scenario k at stage $t-1$,
- $x_{t-1,a(k)}$: current solution from $a(k)$,
- Constraints (1): feasibility cuts,
- Constraints (2): optimality cuts

Nested L-Shaped Method

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

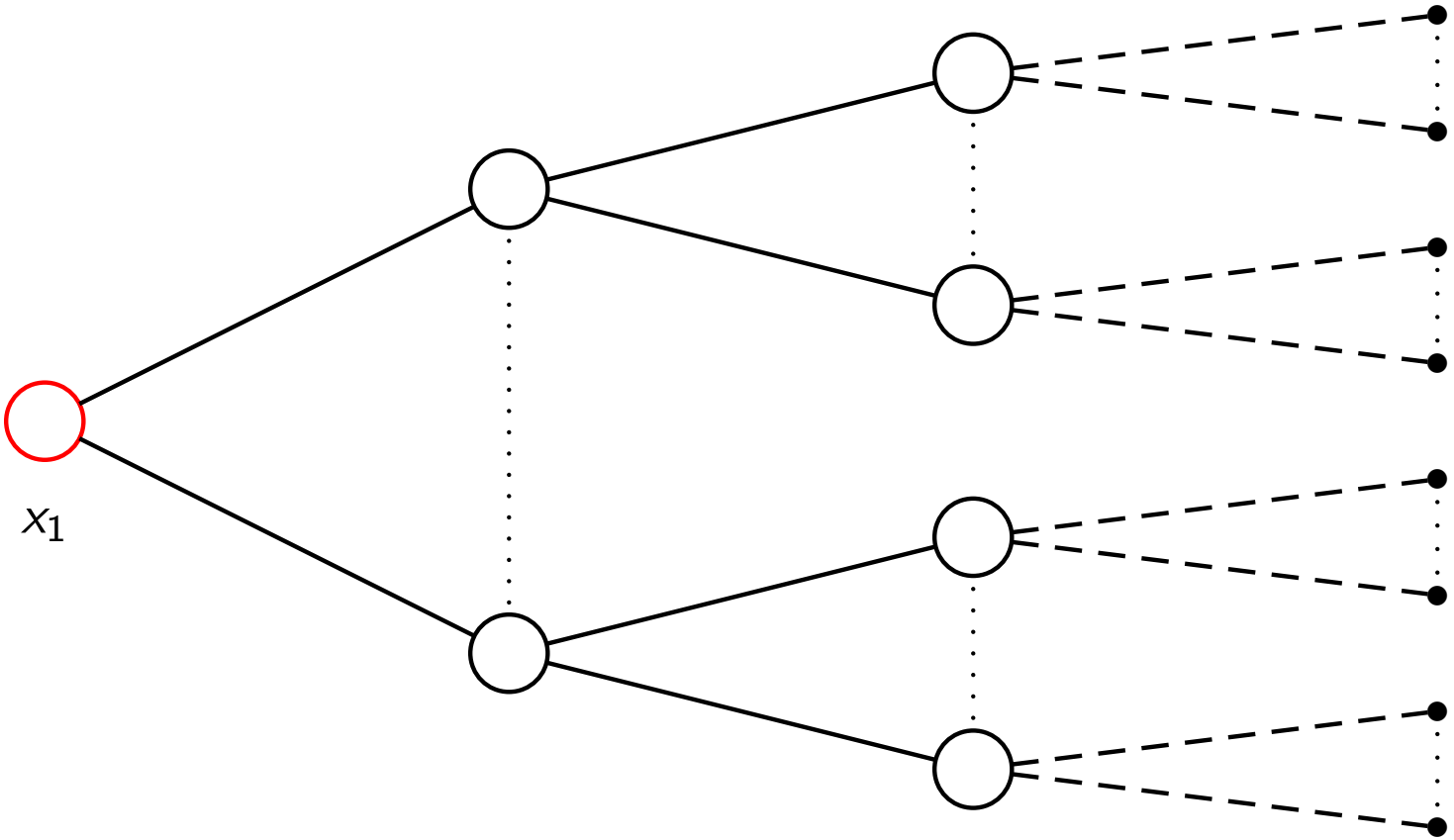
- Repeated application of the L-shaped method.
- Parent nodes send proposals for solutions to their children nodes
- Child nodes send cuts to their parents
- There are different sequence procedures that tell in which order the problems corresponding to different nodes in the scenario tree are solved.

Nested L-Shaped Method



- $a(k)$: ancestor of scenario k.
- $\mathcal{D}_{t+1}(k)$: descendants of scenario k in period $t+1$.

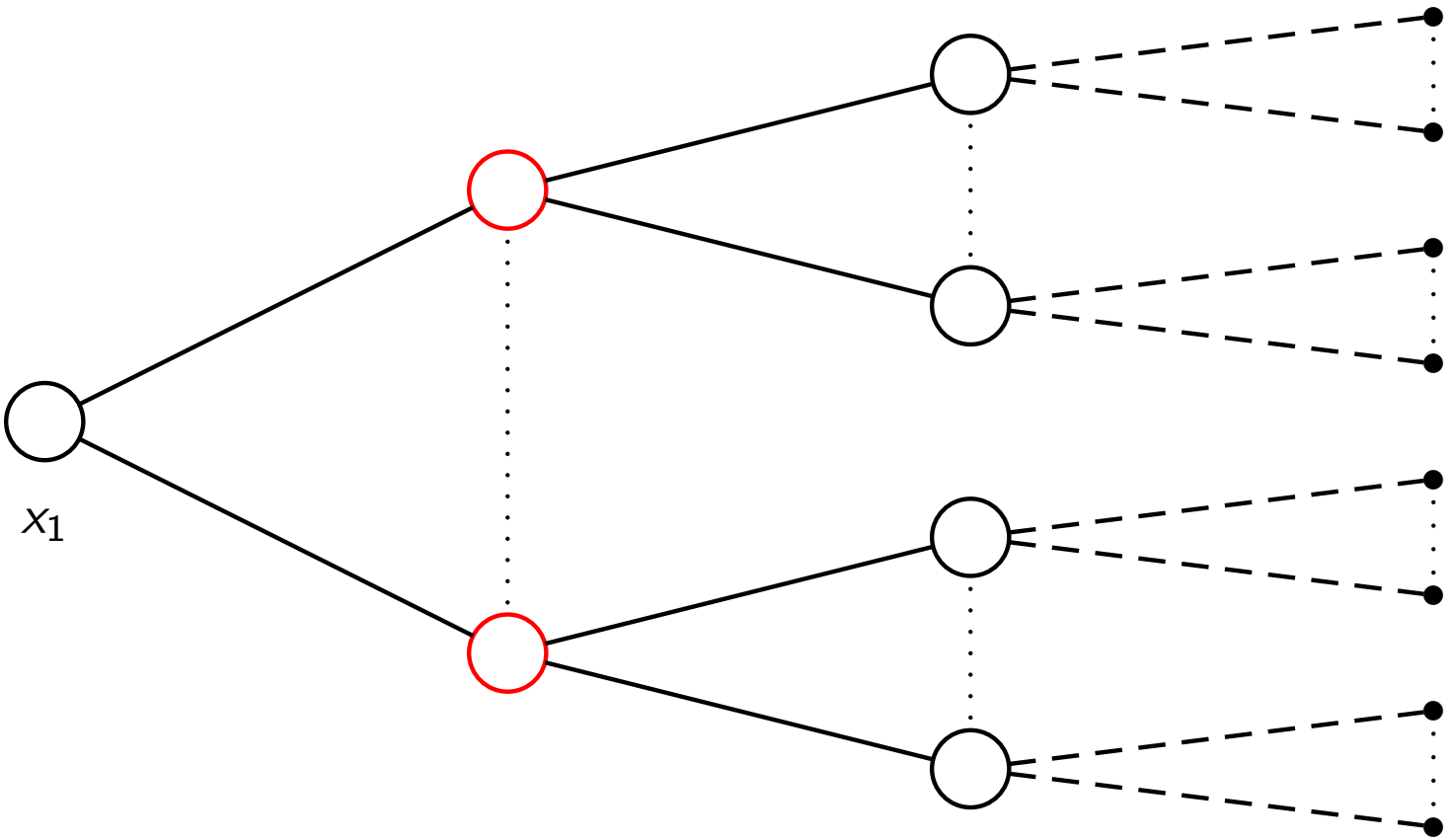
Illustration



First stage Second stage Third stage ...

- Node: $(t = 1, k = 1)$.
- Direction: forward.
- Output: x_1.

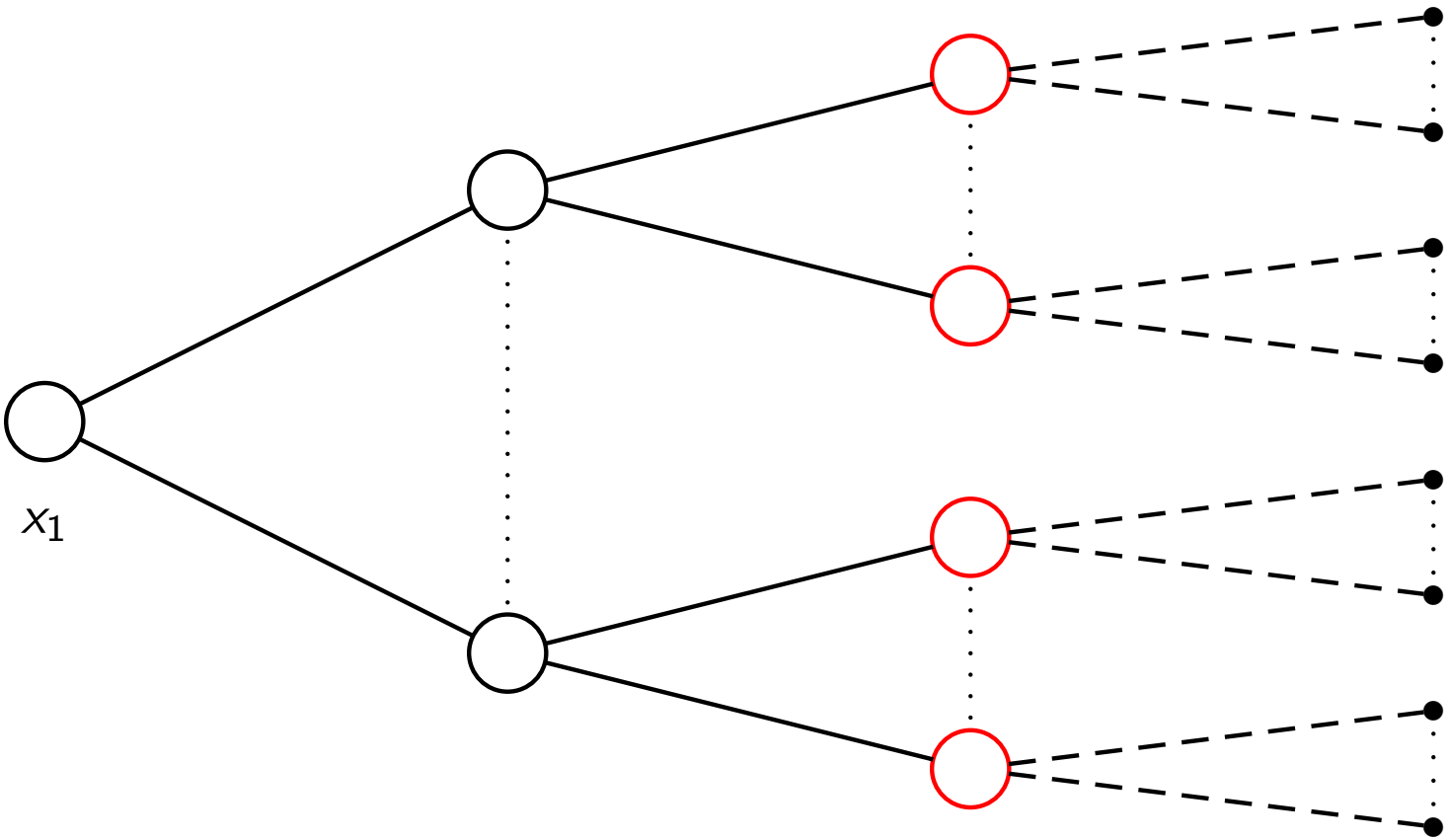
Illustration



First stage Second stage Third stage ...

- Node: $(t = 2, k)$, $k \in \{1, \ldots, k_2\}$.
- Direction: forward.
- Output: $x_{2,k}$, $k \in \{1, \ldots, k_2\}$.

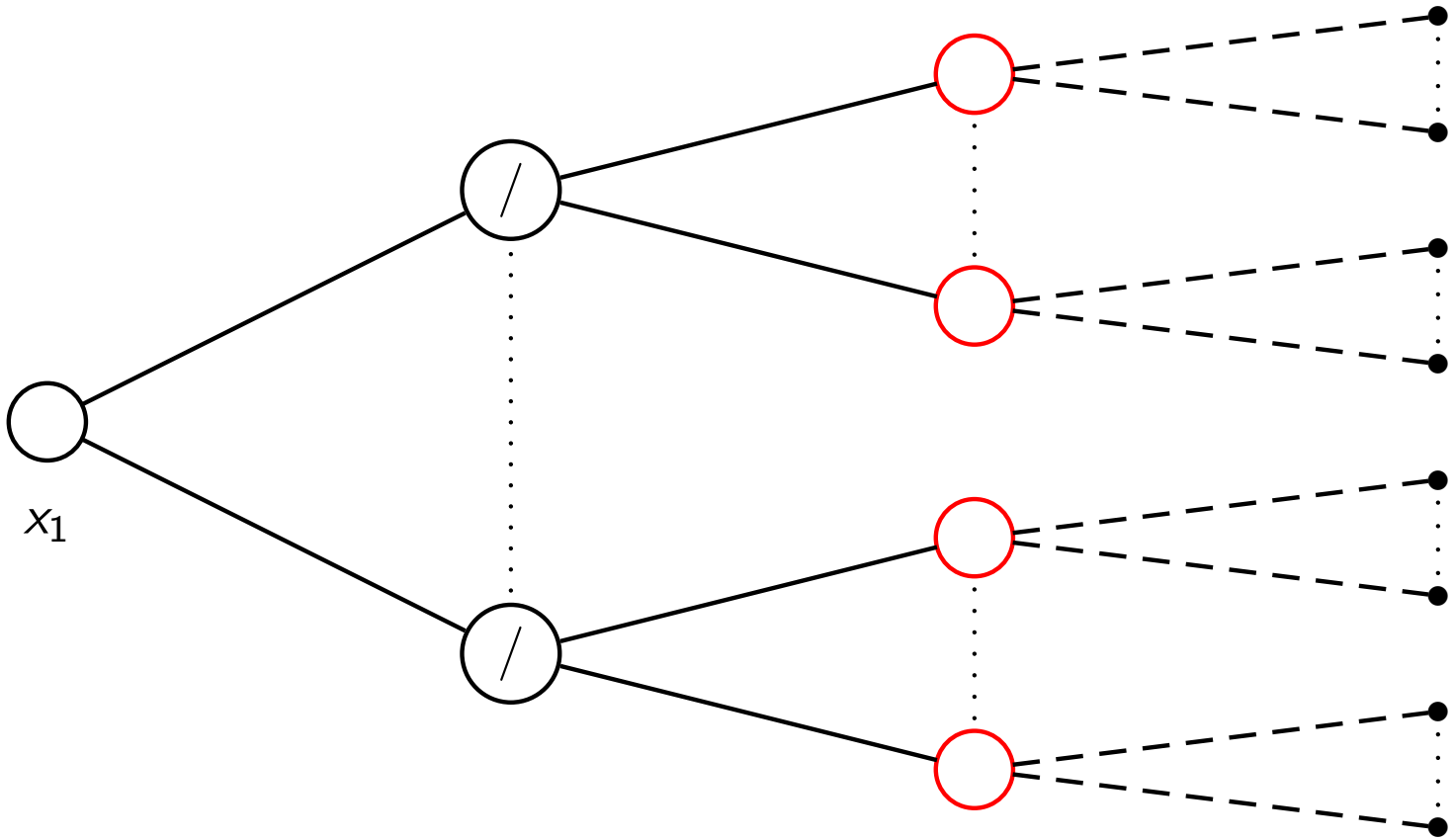
Illustration



First stage Second stage Third stage ...

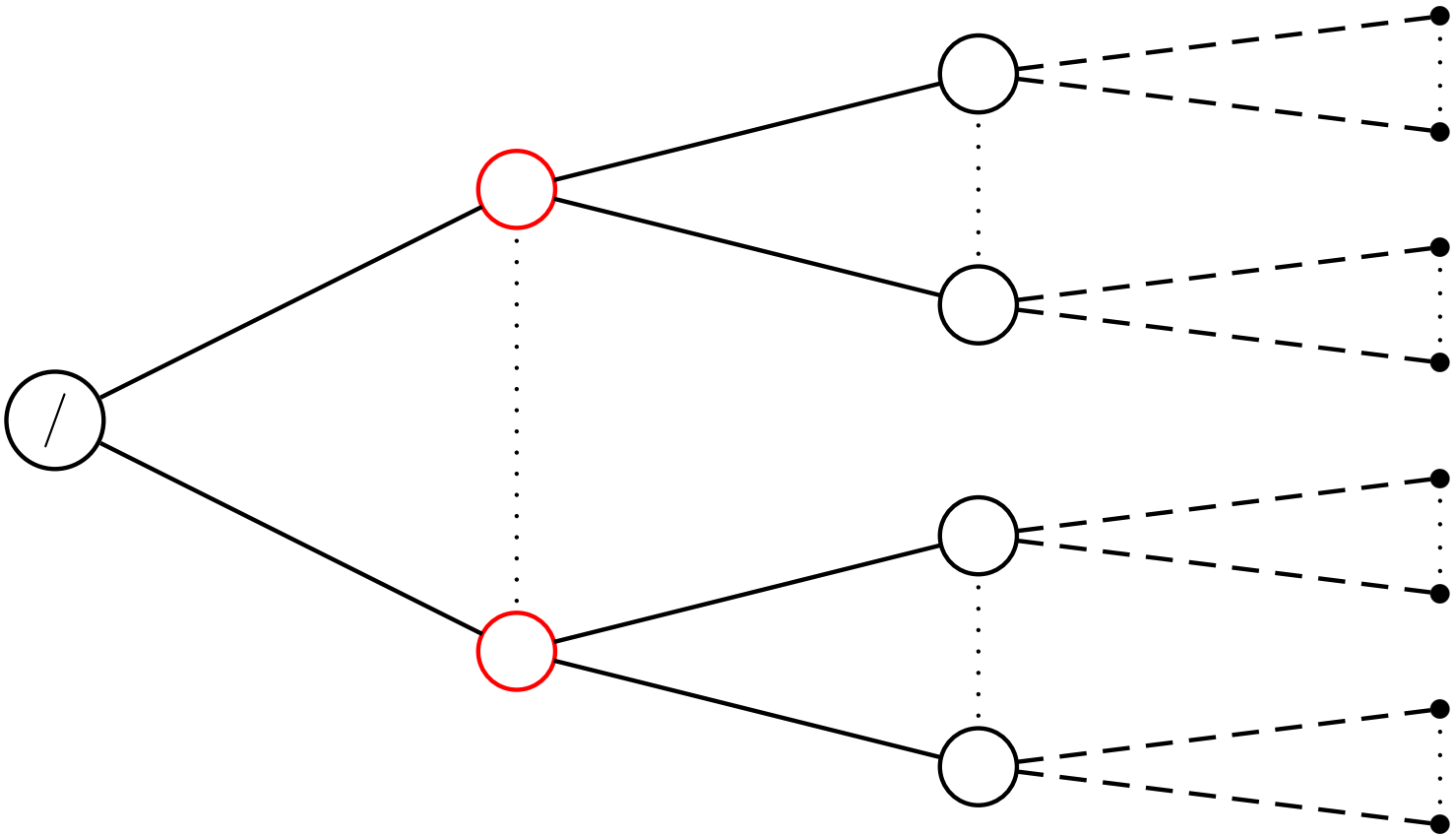
- Node: $(t = 3, k)$, $k \in \{1, \ldots, k_3\}$.
- Direction: forward.
- Output: $x_{3,k}$, $k \in \{1, \ldots, k_3\}$.

Illustration



- Node: $(t = 3, k)$, $k \in \{1, \ldots, k_3\}$.
- Direction: backward.
- Output: $(\pi_{3,k}, \rho_{3,k}, \sigma_{3,k})$, $k \in \{1, \ldots, k_3\}$.

Illustration



- Node: $(t = 2, k)$, $k \in \{1, \ldots, k_2\}$.
- Direction: backward.
- Output: $(\pi_{2,k}, \rho_{2,k}, \sigma_{2,k})$, $k \in \{1, \ldots, k_2\}$.

Feasibility cuts

If $NLSD(t,k)$ is infeasible, we compute $\pi_{t,k}$ and $\rho_{t,k} \leq 0$ such that

$$\pi_{t,k}^T \left(h_{t,k} - T_{t-1,k} x_{t-1,a(k)}\right) + \rho_{t,k}^T d_{t,k} > 0,$$
$$\pi_{t,k}^T W_t + \rho_{t,k}^T D_{t,k} \leq 0.$$

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

$$D_{t-1,a(k)} x_{t-1,a(k)} \geq d_{t-1,a(k)},$$

where

$$D_{t-1,a(k)} = \pi_{t,k}^T T_{t-1,k},$$
$$d_{t-1,a(k)} = \pi_{t,k}^T h_{t,k} + \rho_{t,k}^T d_{t,k}.$$

Optimality cuts

Solve $NLSD(t,k)$ for $j = 1, \ldots, \mathcal{S}_{t-1}$, and compute

$$E = \sum_{k \in \mathcal{D}_t(j)} p(k,t \,|\, j,t-1) \pi_{t,k}^T T_{t-1,k},$$

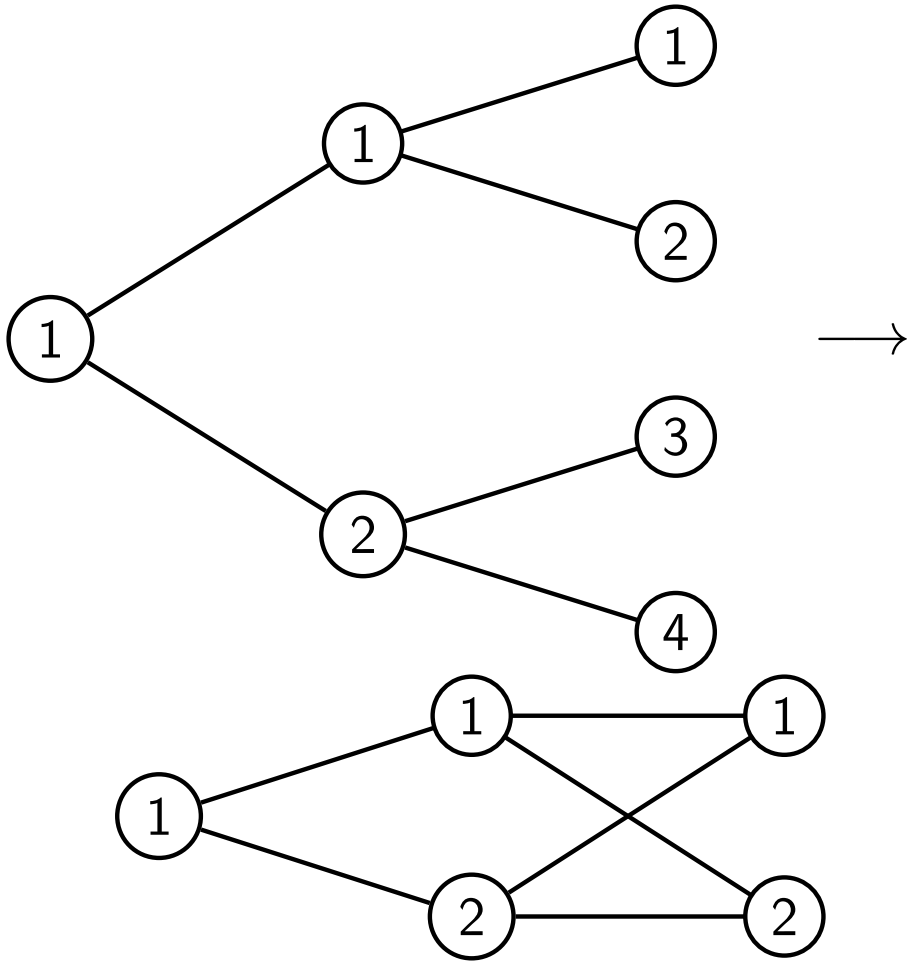
$$e = \sum_{k \in \mathcal{D}_t(j)} p(k,t \,|\, j,t-1) \left(\pi_{t,k}^T h_{t,k} + \sum_{i=1}^{r_{t,k}} \rho_{t,k,i} d_{t,k,i} + \sum_{i=1}^{s_{t,k}} \sigma_{t,k,i} e_{t,k,i} \right).$$

$\mathcal{D}_t(j)$: descendants at stage t of a scenario j at period $t-1$.

Note:

$$p(k,t \,|\, j,t-1) = \frac{p_{k,t}}{p_{j,t-1}}.$$

Recombined scenario tree

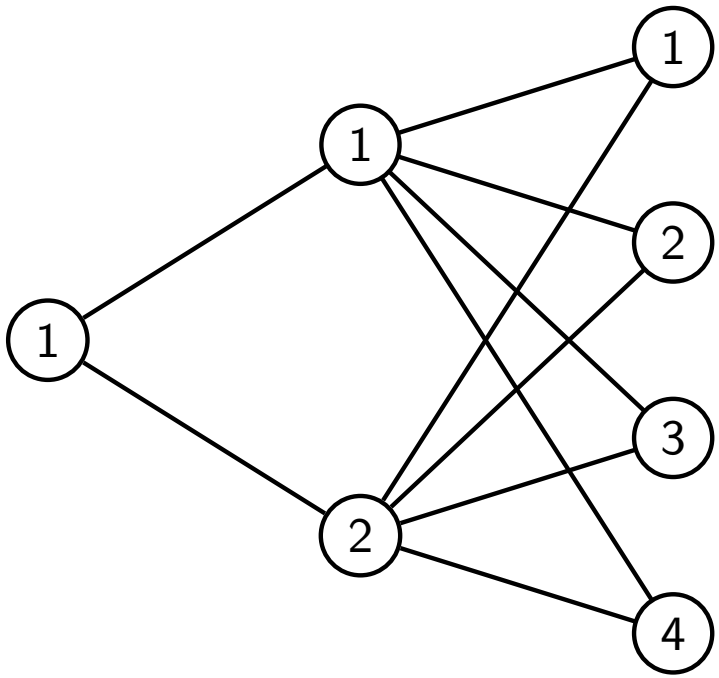


- When can we recombine nodes?
- When can we assign the same value function $V_{t+1}(x)$ to each node k of stage t?

Nested decomposition is non-scalable

Assume

- H time steps, M_t discrete outcomes in each stage.
- No infeasibility cuts.



$M_1 \qquad M_2 \qquad M_3$

- Forward pass: $M_1 + M_1 \times M_2 + \ldots = \sum_{t=1}^{H} \prod_{j=1}^{t} M_j$;
- Backward pass: $\sum_{t=2}^{H-1} \prod_{j=1}^{t} M_j$.

Nested decomposition vs extended form

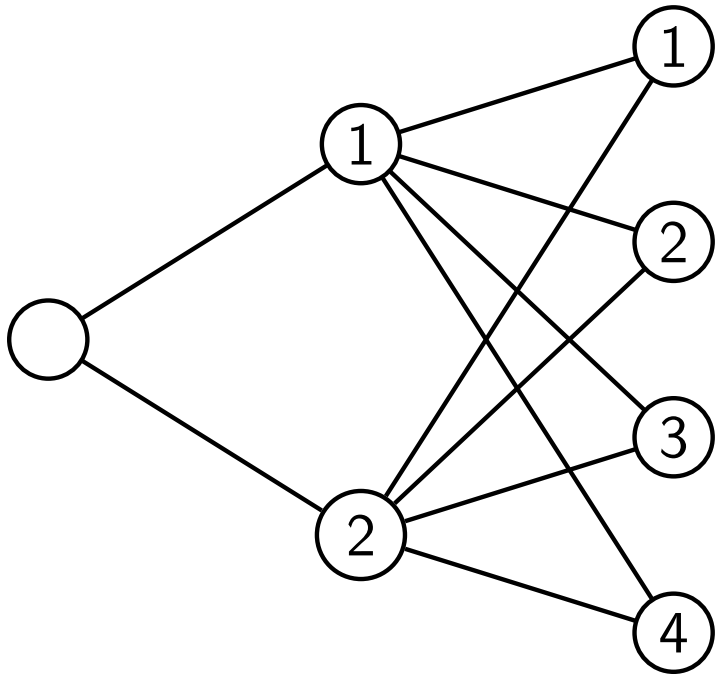
Alternative to nested decomposition is extended form.

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

Nested Decomposition lays the foundations for SDDP.

As the method name suggests, we are in the context of dynamic programming.

Enumerating Versus Simulating



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

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;

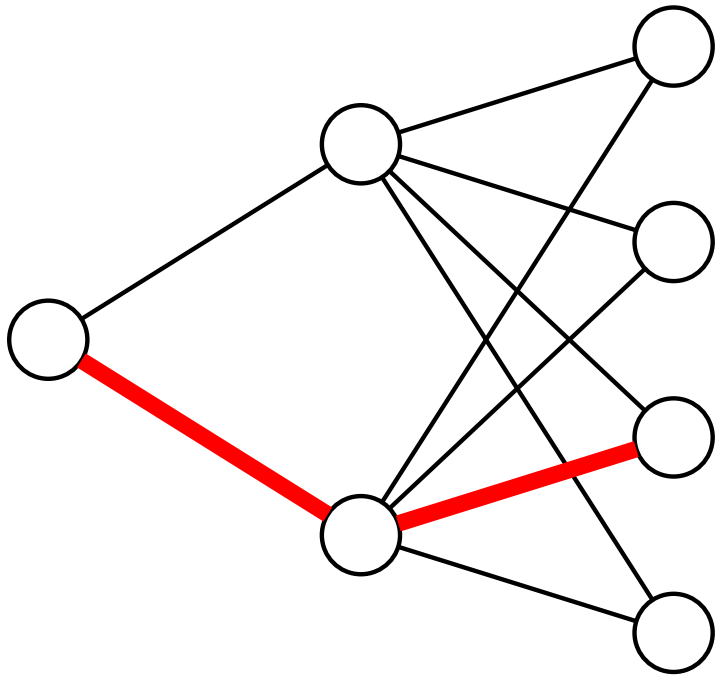
Solutions for backward pass

- In the backward pass, we share cuts among nodes of the same time period
- This requires an assumption of stagewise independence.

Stagewise independence: the random vector ξ_{t+1} is independent of $\xi_{[t]} = (\xi_1, \ldots, \xi_t)$ for $t = 1, \ldots, H-1$ (see for instance A. Shapiro, "Analysis of stochastic dual dynamic programming", European Journal of Operational Research, 209(1), 2011, pp. 63–72). Here, ξ_1 has a unique realization.

Implications for forward pass

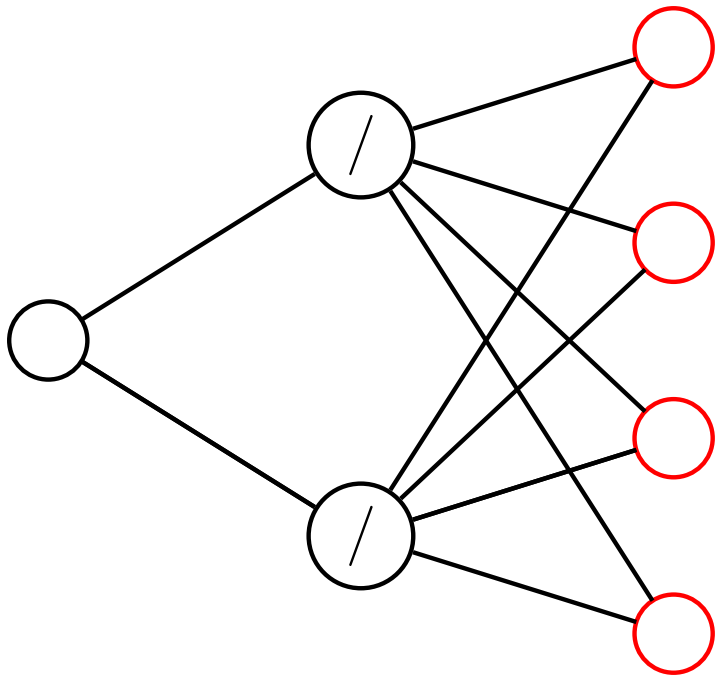
At each forward pass we solve $H-1$ NLSD problems



For K Monte Carlo simulations, we solve $1+K\times(H-1)$ linear programs.

Implications for backward pass

Serial independence implies same value function for all nodes of stage $t \Longrightarrow$ cut sharing .



For a given trial sequence $x_{t,k}$, we solve $\sum_{t=2}^{H} M_t$ linear programs.
For K trial sequences, we solve $K \sum_{t=2}^{H} M_t$ linear programs.

Stagewise independence is not necessary

We can use dual multipliers in stage $t+1$ for cuts in stage t even without stagewise independence.

However, each node in stage t has a different value function, so

- more memory consumption;
- more optimality cuts needed because we are approximating more value functions.

With stagewise independence, we can

- get rid of the scenario tree;
- work with continuous distribution of ξ_t.

SDDP forward pass

- Solve $NLSD(1,1)$. Let $x_{1,1}$ be the optimal solution. Initialize $\hat{x}_{1,i} = x_{1,1}$ for $i = 1, \ldots, K$.
- For $t = 2, \ldots, H$, $i = 1, \ldots, K$,
  - sample a vector $h_{t,i}$ from the set $h_{t,k}$, $k = 1, \ldots, M_t$;
  - solve the $NLSD(t,i)$ with trial decision $\hat{x}_{t-1,i}$;
  - store the optimal solution as $\hat{x}_{t,i}$.

SDDP backward pass

- For $t = H, H-1, \ldots, 2$
  - For $i = 1, \ldots, K$,
    - Repeat for $k = 1, \ldots, M_t$, solve $NLSD(t, k)$ with trial decision $\hat{x}_{t-1,i}$.
    - Compute

$$E_{t-1} = \sum_{k=1}^{M_t} p_{t,k} \pi_{t,k,i}^T T_{t-1,k},$$

$$e_{t-1} = \sum_{k=1}^{M_t} p_{t,k} \left(\pi_{t,k,i}^T h_{t,k} + \sigma_{t,k,i} e_{t,k} \right).$$

Add the optimality cut

$$E_{t-1} x_{t-1} + \theta_{t-1} \geq e_{t-1}$$

to every $NLSD(t-1, k)$, $k = 1, \ldots, M_{t-1}$.

Probabilistic upper bound

Suppose we draw a sample k of $(\xi_{t,k})$, $t = 1, \ldots, H$, and solve $NLSD(t, k)$ for $t = 1, \ldots, T$ (noting that for $t = 0$, the problem is deterministic).

- This gives $x_{t,k}$, $t = 1, \ldots, H$ with a cost $z_k = \sum_{t=1}^{H} c_k^T x_k$.
- Consider K independant samples, and the mean cost

$$\bar{z}_K = \frac{1}{K} \sum_{k=1}^{K} z_k.$$

By the Central Limit Theorem,

$$\bar{z}_K \xrightarrow{\mathcal{D}} N(\mu, \sigma^2)$$

- Each $(x_{t,k}, t = 1, \ldots, H)$ is feasible but not necessarily optimal, so $\bar{z}_K$ is an estimate of an upper bound.

Bounds and termination criterion

After solving $NLSD(1,1)$ in a forward pass we can compute a lower bound z_{LB} as the objective function value of $NLDS(1,1)$.

Terminates if $z_{LB} \in (\bar{z}_K - \Phi^{-1}(1-\alpha/2)\sigma, \bar{z}_K + \Phi^{-1}(1-\alpha/2)\sigma)$, where $\Phi(\cdot)$ is the cumulative distribution of a Normal(0,1), and α is a confidence level set but the user. Typically, $\alpha = 0.05$.

How to ensure 1% optimality gap with 95% confidence? Choose K such that $\Phi^{-1}(1-\alpha/2) \approx 0.01\bar{z}_K$. Using the empirical standard deviation

$$\hat{s} = \frac{1}{K-1}\sqrt{\sum_{k=1}^{K}(\bar{z}_K - z_k)^2},$$

this leads to

$$K = \left(\frac{\Phi^{-1}(1-\alpha/2)\hat{s}}{0.01\bar{z}_K}\right)^2.$$

SDDP algorithm

- Set K, $\bar{z}_K = \infty$, $\hat{s} = 0$.
- Forward pass, store z_{LB} and $\bar{z}_K$. Check termination.
- If the termination criterion is not met, do a backward pass.
- Go to forward pass.