Hydrothermal Planning

Anthony Papavasiliou

Source: chapter 8, Papavasiliou [1]

Outline

- Multi-stage stochastic linear programming
  - Two-stage stochastic linear programs
  - Modeling multi-period uncertainty
  - Multi-stage stochastic linear programs
- The hydrothermal planning problem
  - Model formulation
  - Value function
- Performance of stochastic programs

Multi-stage stochastic linear programming

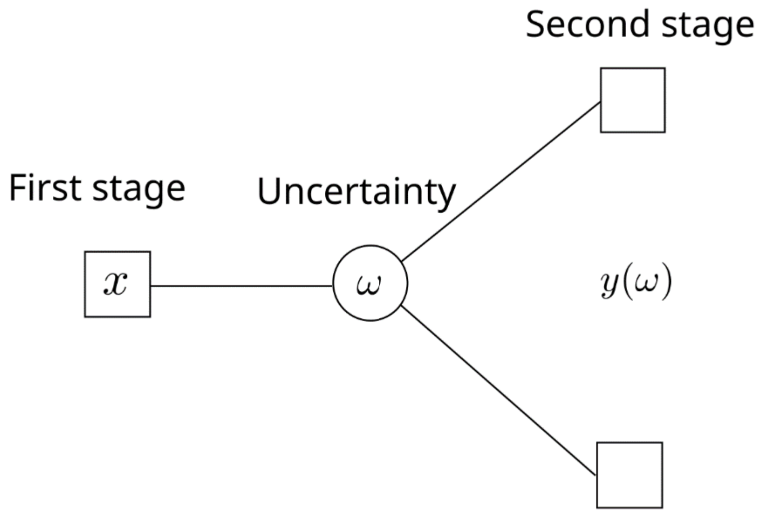
Two-stage stochastic linear programs

Modeling multi-period uncertainty

Multi-stage stochastic linear programs

Sequence of events

1. **First-stage decisions**: decisions taken before uncertainty is revealed
2. **Second-stage decisions**: decisions taken after uncertainty is revealed
3. Sequence of events: $x \rightarrow \omega \rightarrow y(\omega)$



Mathematical formulation

$$(TSLP): \min_{x,y} c^T x + \mathbb{E}[q(\omega)^T y]$$
$$\text{s.t. } Ax = b$$
$$T(\omega)x + W(\omega)y(\omega) = h(\omega), \omega \in \Omega$$
$$x \geq 0, y \geq 0$$

- First-stage decisions $x \in \mathbb{R}^{n_1}$, second-stage decisions $y(\omega) \in \mathbb{R}^{n_2}$
- First-stage parameters: $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$, $A \in \mathbb{R}^{m_1 \times n_1}$
- Second-stage parameters: $q(\omega) \in \mathbb{R}^{n_2}$, $h(\omega) \in \mathbb{R}^{m_2}$, $T(\omega) \in \mathbb{R}^{m_2 \times n_1}$, $W(\omega) \in \mathbb{R}^{m_2 \times n_2}$

Example 8.1: deterministic two-stage model

- Hydrothermal system
  - Unit G1: 60 MW at 10 $/MWh
  - Unit G2: 100 MW at 50 $/MWh
  - Hydroelectric unit: efficiency $\eta = 0.8$
- Load:
  - Period 1: 50 MW
  - Period 2: 100 MW
- Express the model in the generic two-stage format
  - Hint: introduce a decision variable that corresponds to the amount of energy that is stored in the reservoir

Mathematical formulation of hydrothermal scheduling

$$\min_{p,dH,pH,e} \sum_{g\in G} MC_g \cdot p_{g1} + \sum_{g\in G} MC_g \cdot p_{g2}$$
$$p_{g1} \leq P_g, g \in G$$
$$e_1 = dH_1 - \frac{pH_1}{\eta}$$
$$D_1 + dH_1 - \sum_{g\in G} p_{g1} - pH_1 = 0$$
$$p_{g2} \leq P_g, g \in G$$
$$e_2 = dH_2 - \frac{pH_2}{\eta} + e_1$$
$$D_2 + dH_2 - \sum_{g\in G} p_{g2} - pH_2 = 0$$
$$e_2 = 0$$
$$p, dH, pH, e \geq 0$$

Notation and assumptions

- Decision variables:
  - p_{gt}: production from thermal unit $g \in G$ at period t
  - dH_t: electricity demand from hydrothermal unit at period t
  - pH_t: production from hydrothermal unit at period t
  - e_t: amount of hydro energy in the reservoir at the end of period t

- We assume that the amount of energy in the reservoir at the beginning of period 1 is zero
- We require $e_2 = 0$
- Load must be fully served

First and second-stage decisions

- First-stage decisions: p_{g1}, pH_1, dH_1, e_1
- Second-stage decisions: p_{g2}, pH_2, dH_2, e_2

- Second-stage decisions are connected to first-stage decisions through the fifth constraint
- All other constraints are either first- or second-stage

Optimal solution

- Unit G1 produces at its capacity in both periods
- The excess energy of period 1 is stored in the reservoir
  - This is beneficial even if 20% of the stored energy is lost due to losses
  - Because unit G2 is overly costly

General formulation of hydrothermal programming

$$\max_{p\geq 0,d\geq 0,dH\geq 0,pH\geq 0,e\geq 0} V\cdot d_1 - \sum_{g\in G} MC_{g1}\cdot p_{g1} + \sum_{\omega\in\Omega} P_\omega \cdot (V\cdot d_2(\omega) - \sum_{g\in G} MC_{g2}\cdot p_{g2}(\omega))$$

$$p_{g1} \leq P_g, g\in G$$
$$e_1 = R_1 + dH_1 - \frac{pH_1}{\eta}$$
$$e_1 \leq E, d_1 \leq D_1$$
$$d_1 + dH_1 - \sum_{g\in G} p_{g1} - pH_1 = 0$$

$$p_{g2}(\omega) \leq P_g, g\in G, \omega\in\Omega$$
$$e_2(\omega) = R_2(\omega) + dH_2(\omega) - \frac{pH_2(\omega)}{\eta} + e_1, \omega\in\Omega$$
$$d_2(\omega) + dH_2(\omega) - \sum_{g\in G} p_{g2}(\omega) - pH_2(\omega) = 0, \omega\in\Omega$$
$$e_2(\omega) \leq E, d_2(\omega) \leq D_2, \omega\in\Omega$$

Characteristics of the general model

- Monthly time steps
- Representation of rainfall
- Uncertainty: e.g. in rainfall
- Hydro storage limits
- Temporal variations of fuel cost
- Load shedding

Notation of the hydrothermal scheduling problem

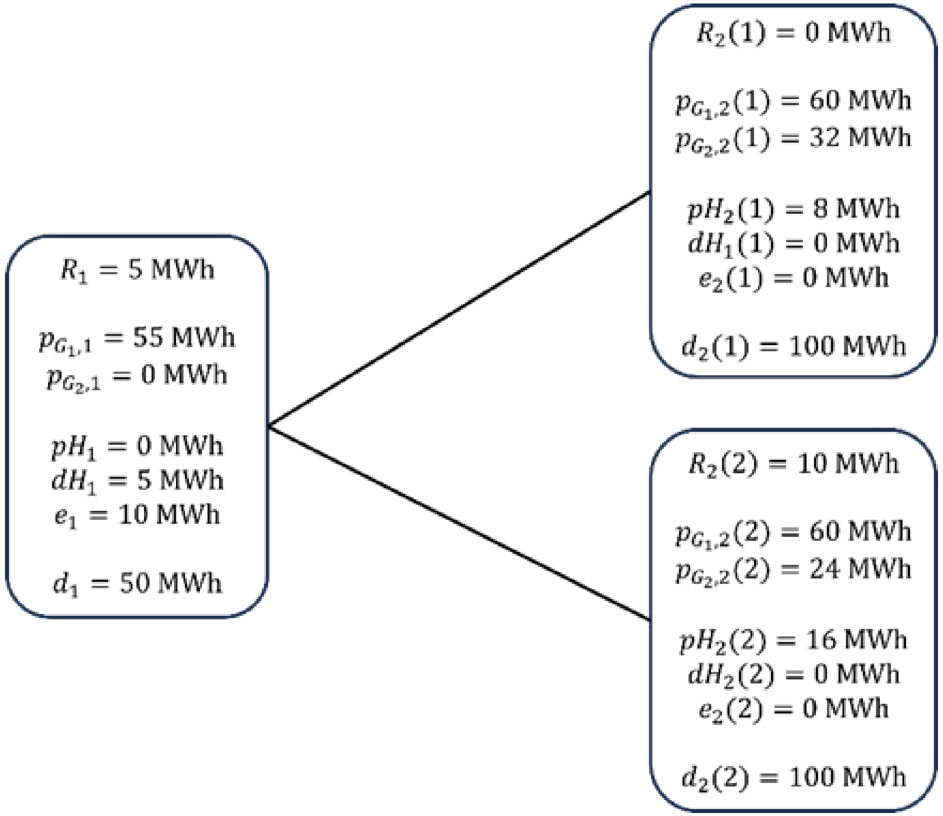
- P_ω: probability of outcome ω
- Possibility to represent uncertainty in:
  - Rainfall, $R_2(\omega)$
  - Load, $D_2(\omega)$
  - Availability of generation units, $P_g(\omega)$
  - Fuel cost, $MC_{gt}(\omega)$: computationally hard
- R_1: rainfall in period 1
- $R_2(\omega)$: uncertain rainfall in period 2
- V: consumer valuation
- E: hydro reservoir limit
- MC_{gt}: marginal cost of technology g in period t

Example 8.2: stochastic two-stage hydrothermal scheduling model

- Consumer valuation: $V = 1000$ \$/MWh
- Storage limit: $E = 10$ MWh
- Rainfall of period 1: $R_1 = 5$ MWh
- Two scenarios of period 2, $\Omega = \{1,2\}$:
  - $R_2(1) = 0$ MWh, $P_1 = 0.5$
  - $R_2(2) = 10$ MWh, $P_2 = 0.5$

Optimal solution

- In period 1, unit G1 is used to the greatest possible extent (until the reservoir is full)
- In period 2, unit G2 is used for covering the demand that is not covered by G1 and the hydro unit



Multi-stage stochastic linear programming

Two-stage stochastic linear programs

Modeling multi-period uncertainty

Multi-stage stochastic linear programs

Scenario trees

- A **scenario tree** is a graphical representation of a stochastic process $\{\xi_t\}, t = 1, \dots, H$
- A graph that includes a set of nodes N and a set of directed edged E, where:
  - Each node of the tree corresponds to a history of outcomes up to a stage t: $\xi_{[t]} = \{\xi_1, \dots, \xi_t\}$
  - Each edge of the tree corresponds to transitions from $\xi_{[t]}$ to $\xi_{[t+1]}$

Root, ancestor and descendants

- The root of the tree corresponds to the first stage, $t = 1$
- The **ancestor** of a node $\xi_{[t]}$, $A(\xi_{[t]})$, is the *unique* neighboring node in the scenario tree that precedes $\xi_{[t]}$:

$$A\left(\xi_{[t]}\right) = \left\{\xi_{[t-1]} : (\xi_{[t-1]}, \xi_{[t]}) \in E\right\}$$

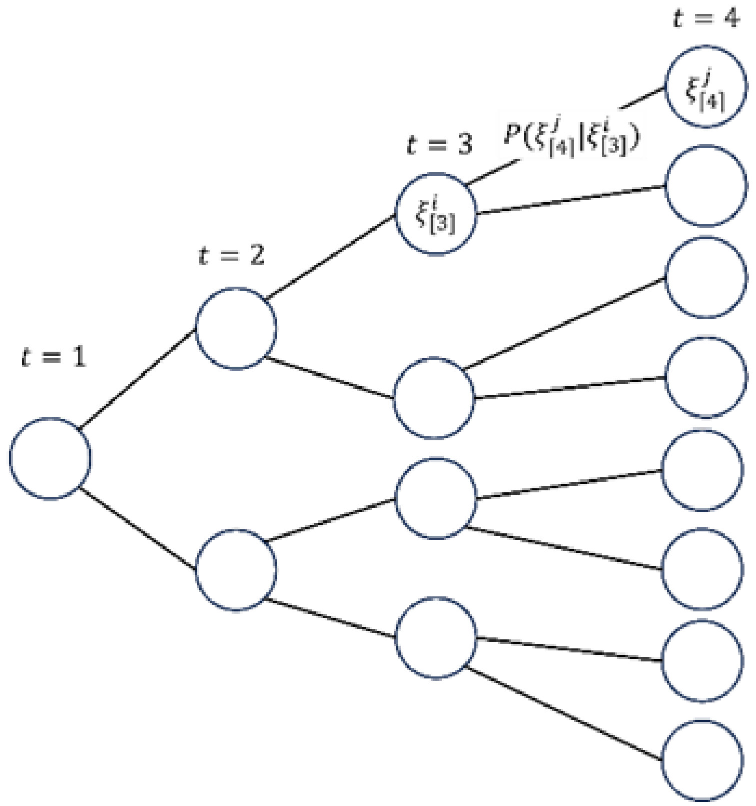
- The **descendants of a node**, $C\left(\xi_{[t]}\right)$, is the set of nodes that are adjacent to $\xi_{[t]}$ and occur at stage $t + 1$:

$$C\left(\xi_{[t]}\right) = \left\{\xi_{[t+1]} : \left\{\xi_{[t]}, \xi_{[t+1]}\right\} \in E\right\}$$

- Each node of the scenario tree has a time label

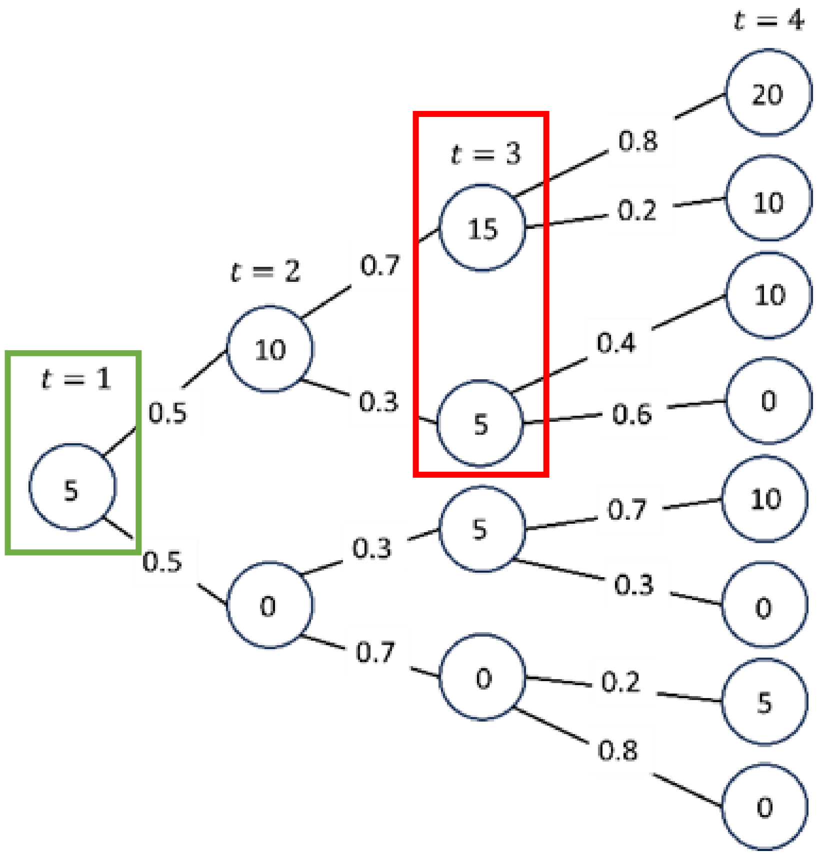
Structure of a scenario tree

- To each node of the scenario tree corresponds a history $\xi_{[t]}$
- To each edge of the tree corresponds a transition probability $P(\xi_{[t+1]}|\xi_{[t]})$



Example 8.3: four-stage rainfall scenario tree

- Same setup as example 8.2, but two more stages
- Numbers within nodes: rainfall at the given node of the given stage
- Numbers on edges: transition probabilities
- Red indicates descendants of node (5,10)
- Green indicates ancestor of node (5,10)

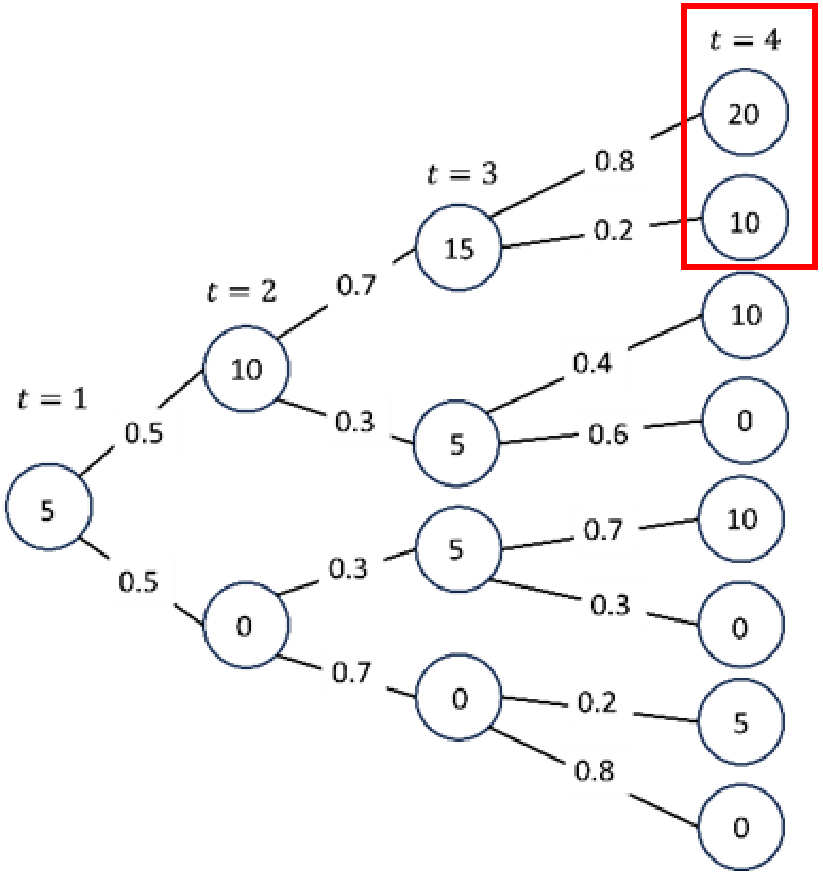


Scenarios and events

- **Scenarios** in stochastic programming are synonymous to **outcomes** in probability theory
- **Events** are subsets of the set of events
- Set of events Ω_H of a scenario tree:
  - Set of trajectories from period 1 to the end of the horizon H
  - Set of nodes of the scenario tree at stage H
- Set Ω_t: set of scenarios from period 1 until period t

Events encode information

- The set in the red box is an event in Ω_4
- It is the event where there was plenty of rain in periods 1, 2, 3
- Events encode information in the sense that we can only tell them apart if we have enough information
- Event (5,10,15,20) (plenty of rain in all periods) is an event in Ω_4, but not in Ω_3
- Even if we know that we are in outcome (5,10,15) of Ω_3, this does not guarantee which outcome of Ω_4 we are in



Probability of a node

Probability of being at a node of the scenario tree = probability of trajectory that lands on the node:

$$P\left(\xi_{[t]}^{i}\right) = P\left(\xi_{[1]}^{i}\right) \cdot P\left(\xi_{2}^{i} \middle| \xi_{[1]}^{i}\right) \cdot \cdots \cdot P\left(\xi_{t}^{i} \middle| \xi_{[t-1]}^{i}\right) \quad (8.1)$$

Markov processes

- **Markov processes** are stochastic processes where the conditional distribution of ξ_t depend sonly on the last outcome of the process, ξ_{t-1}, and not the history of the process, $\xi_{[t-1]} = (\xi_1, \ldots, \xi_{t-1})$:

$$P\left(\xi_t \middle| \xi_{[t-1]}\right) = P(\xi_t | \xi_{t-1})$$

for all stages t and all possible trajectories $\xi_{[t]}$

- Can be represented as scenario trees if transition probabilities obey the Markov properties
- Can alternatively be represented as lattices

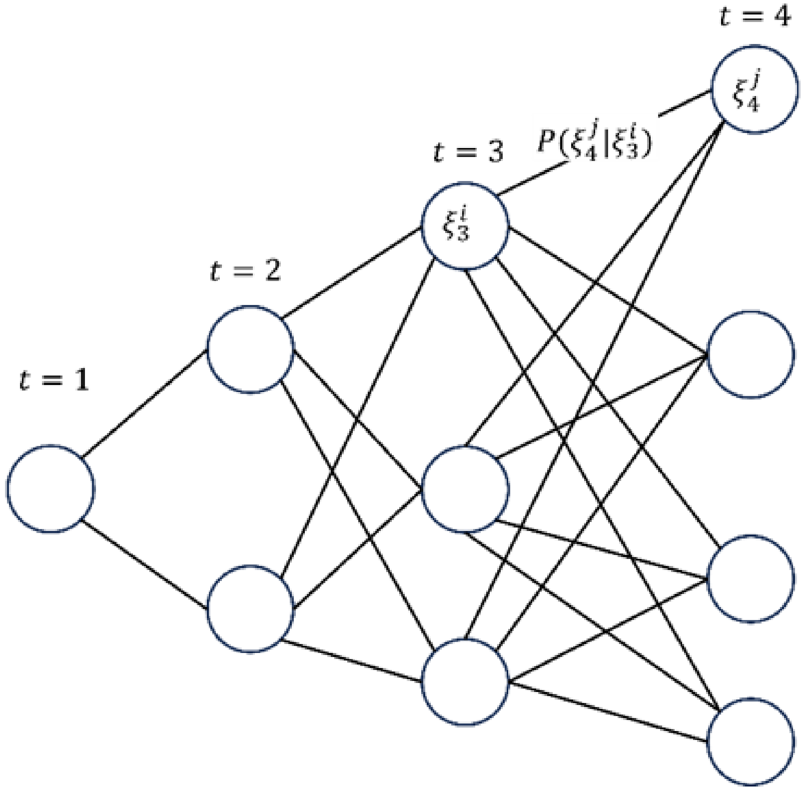
Lattices

A **lattice** is a special case of scenario tree, which can be used for representing a Markov process $\xi_{[t]}, t = 1, \ldots, H$, where

- Each node of the lattice corresponds to an outcome ξ_t, and
- Each edge corresponds to a transition from ξ_t to ξ_{t+1} and a transition probability $P(\xi_{t+1}|\xi_t)$

Graphical representation of a lattice

- Each node $\xi_t \in N$ corresponds to a realization of the stochastic parameter
- Each edge corresponds to a transition probability $P(\xi_{t+1}|\xi_t)$



Probability of each node of a lattice

- The probability of landing on a given node of a lattice can be computed recursively

- The root in period 1 has probability 1:

$$P\left(\xi_1 = \xi_1^1\right) = 1$$

- For each ξ_t in the set Ξ_t at period t:

$$P(\xi_2 = n) = P(\xi_2 = n|\xi_1^1), n \in \Xi_2$$

$$\vdots$$

$$P(\xi_t = n) = \sum_{m \in \Xi_{t-1}} P(\xi_t = n|\xi_{t-1} = m) \cdot P(\xi_{t-1} = m), n \in \Xi_t$$

$$\vdots$$

$$P(\xi_H = n) = \sum_{m \in \Xi_{H-1}} P(\xi_H = n|\xi_{H-1} = m) \cdot P(\xi_{H-1} = m), n \in \Xi_H$$

Scenario trees and lattices

- We can unfold lattices to scenario trees
  - Meaning we can represent Markov processes using scenario tress
- But we cannot always fold a scenario tree to a lattice
  - Meaning that not all scenario trees are Markov processes

Example 8.4: a scenario tree that is not a Markov process

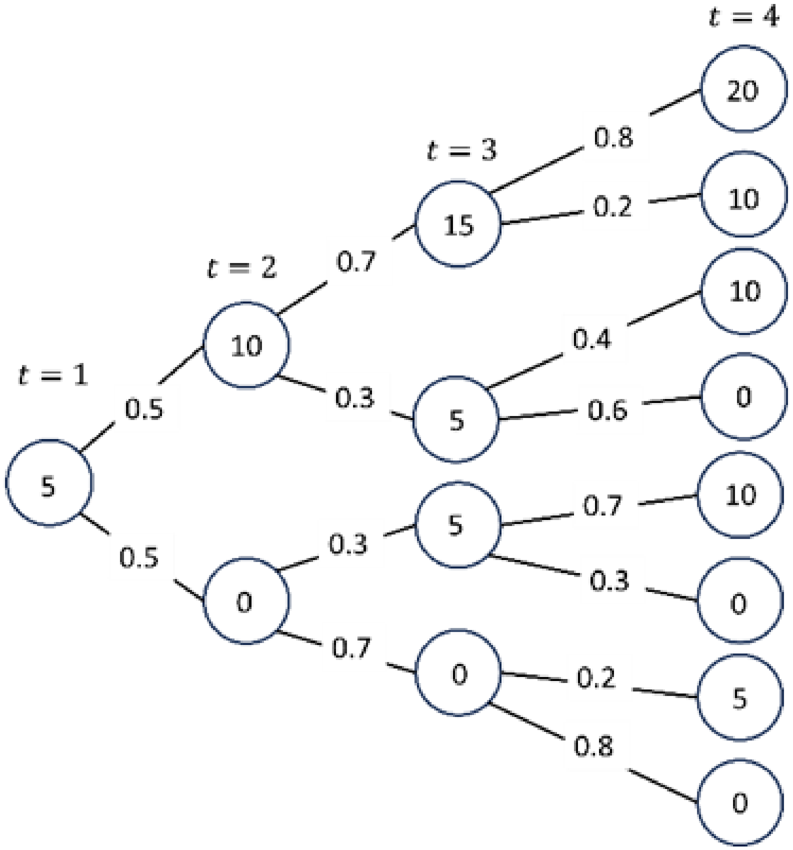
- Consider the scenario tree of example 8.3
- Probability of $\xi_4 = 10$ given history (5,10,5):

$$P\big(\xi_4 = 10 | \xi_{[3]} = (5,10,5)\big) = 0.4$$

- Probability that $\xi_4 = 10$ given last outcome of history (5,10,5):

$$P(\xi_4 = 10 | \xi_3 = 5) = \frac{P(\xi_4 = 10, \xi_3 = 5)}{P(\xi_3 = 5)}$$

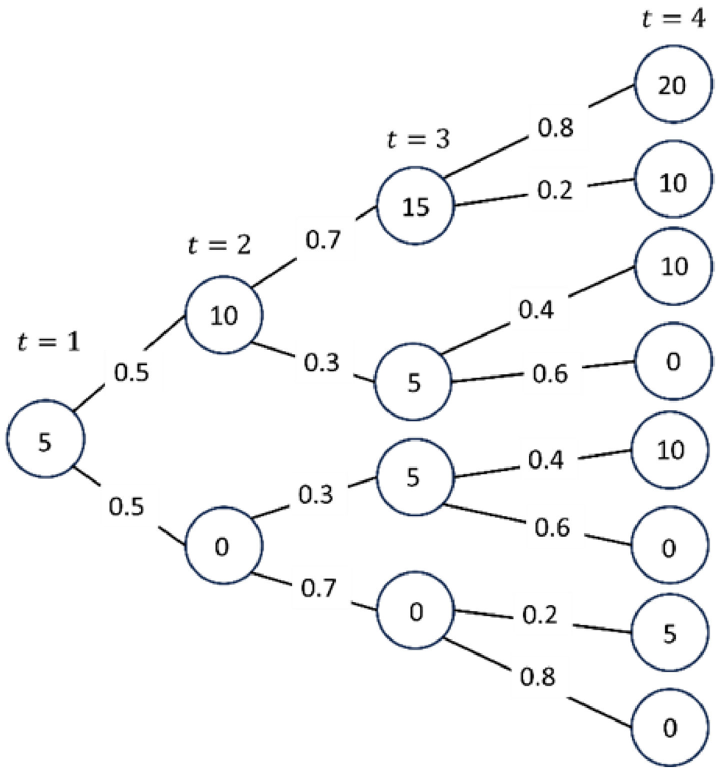
$$= \frac{(0.5 \cdot 0.3) \cdot 0.4 + (0.5 \cdot 0.3) \cdot 0.7}{0.5 \cdot 0.3 + 0.5 \cdot 0.3} = 0.55$$

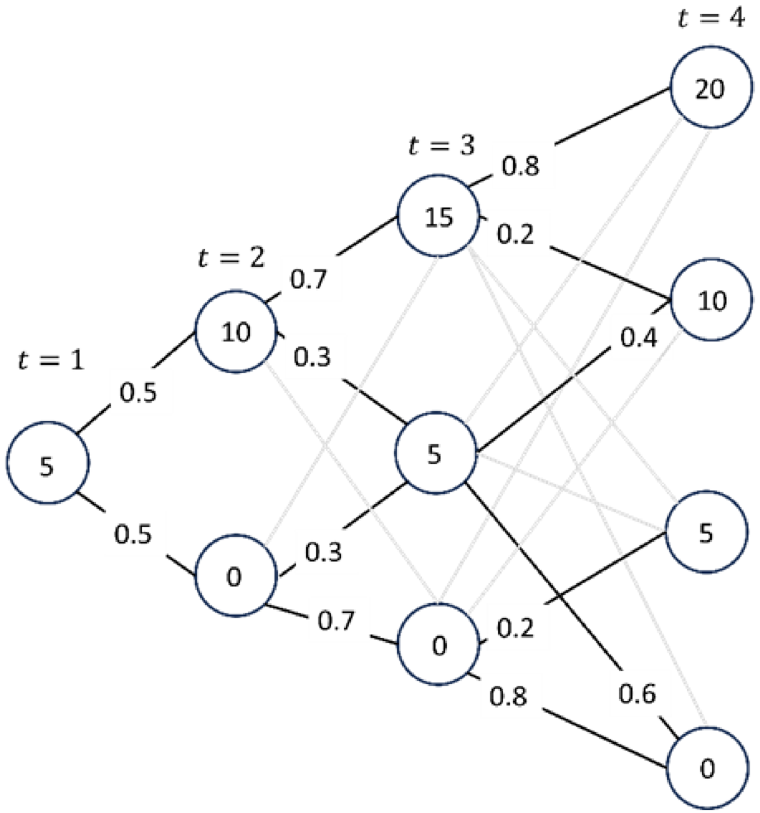


Example 8.5: a four-stage scenario tree that is Markov

- Condition for a scenario tree to be Markov: whenever $\xi_t = \xi_t^i$ in two different nodes of the same stage, the transition probabilities from that node to every node of the next stage must be equal
- Let's slightly modify the scenario tree of the previous slide
- At stage 3 of the next slide, we have two possible paths that lead to outcome $\xi_3 = 5$
- In order for the Markov property to hold, the transition probabilities from $\xi_3 = 5$ to all nodes of stage 4 must be equal

Example 8.5: unfolding a scenario tree into a lattice





Gray edges: zero probability

Stagewise independence

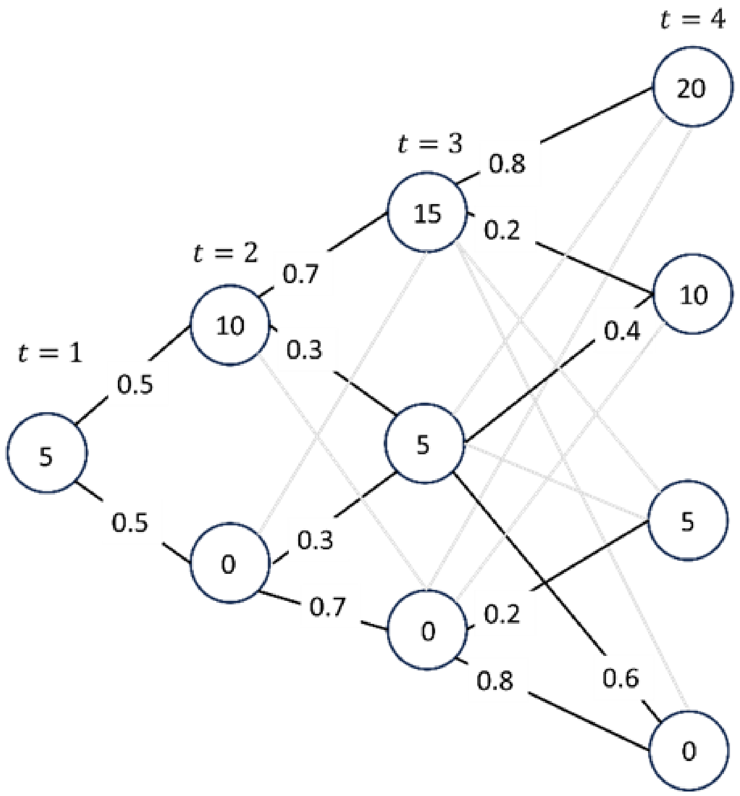
- **Stagewise independent processes** are stochastic processes $\{\xi_t, t = 1, \dots, H\}$ where the probability distribution of ξ_t depends only on stage t, meaning ξ_t is independent of $\xi_{[t-1]}$
- They are Markov processes
- Therefore they can be represented both as scenario trees as well as lattices
- In the same way that scenario trees $\nRightarrow$ lattices, also lattices $\nRightarrow$ stagewise independent

Example 8.6: lattice that is not stagewise independent

- Returning to the lattice of slide 31, not stagewise independent
- For example, the probability of $\xi_4 = 10$ depends on ξ_3:

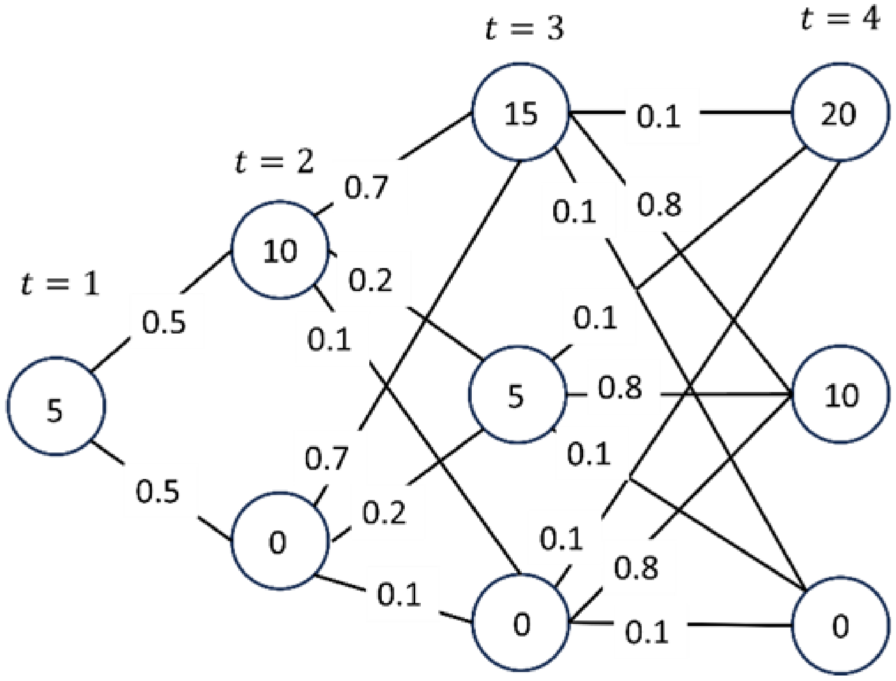
$$P(\xi_4 = 10|\xi_3 = 5) = 0.4$$

$$P(\xi_4 = 10|\xi_3 = 15) = 0.2$$



Example 8.7: four-stage lattice that is stagewise independent

- We require that the probability leading to each ξ_t should be equal, regardless of the adjacent node in stage $t-1$
- For example, $\xi_4 = 20$ with probability 0.1, regardless if $\xi_3 = 15$, 5, 10



Multi-stage stochastic linear programming

Two-stage stochastic linear programs

Modeling multi-period uncertainty

Multi-stage stochastic linear programs

Multi-stage stochastic linear programs

- Linear programs
  - Linear constraints
  - Linear objective function
- Unfolding in multiple stages
- We decide at every stage

Multi-stage stochastic linear program in extended form

$$(MSLP-ST): \min_x \sum_{t=1}^{H} \sum_{n \in \Omega_t} P(n) \cdot c_t(n)^T x_t(n)$$

$$\text{s.t. } W_1 x_1 = h_1$$

$$T_1(n) x_1 + W_2(n) x_2(n) = h_2(n), n \in \Omega_2$$

$$\vdots$$

$$T_{t-1}(n) x_{t-1}(A(n)) + W_t(n) x_t(n) = h_t(n), n \in \Omega_t$$

$$\vdots$$

$$T_{H-1}(n) x_{H-1}(A(n)) + W_H(n) x_H(n) = h_H(n), n \in \Omega_H$$

$$x_1 \geq 0, x_t(n) \geq 0, t = 2, \ldots, H, n \in \Omega_t$$

Model formulation in extended form

- The **extended form** of the model describes the overall decision problem as a large-scale monolithic linear program
- Impossible to solve in practical scale
- But the model has interesting
  - Mathematical properties that inspire the development of dynamic programming algorithms
  - Economic interpretations

Model notation and observations

- $N = \cup_{t=1}^{H} \Omega_t$: set of nodes of scenario tree
- $P(n)$: probability of node $n \in N$
- c_t, T_t, W_t: sets of random variables (indexed by scenario $n \in \Omega_t$)
- x_t: decisions <u>also</u> indexed by $n \in \Omega_t$, therefore corresponding to a **policy**
- This means that decision $x_t(\xi_{[t]})$ depends on stage t and the information $\xi_{[t]} \in \Omega_t$ revealed so far
- Constraints connecting decisions of stage t to decisions of stage $t-1$ only, therefore the decision variables must summarize the state of the system
  - Example: in hydrothermal scheduling, water levels in period t are expressed as a function of water levels in period $t-1$, and not decisions before period $t-1$ that led to this state

Non-anticipativity

- **Non-anticipativity**: decisions in stage t do not have access to future information
- So they can depend on information $n \in \Omega_t$ revealed so far (and not future information $m \in \Omega_\tau, \tau > t$)
- Example: in a two-stage model, first-stage decisions must be identical for all scenarios

Η αρχή του δυναμικού προγραμματισμού

- **The dynamic programming principle**: the optimal solution of a dynamic decision making problem under uncertainty is such that the optimal policy is also optimal the sub-problem in which we only consider a part of the horizon
- Intuition: what’s done is done, and we do the best we can from now on
- This principle leads to the dynamic programming algorithm, which (often) works from the end of the horizon backwards

Value function for stage H

- The **value function** of stage H is defined as:

$$Q_H(x_{H-1}, \xi_{[H]}) = \min_{x_H} c_H(\xi_{[H]})^T x_H$$
$$W_H(\xi_{[H]}) x_H = h_H(\xi_{[H]}) - T_{H-1}(\xi_{[H]}) x_{H-1}$$
$$x_H \geq 0$$

- It is a function of x_{H-1} and $\xi_{[H]}$ (which is all we need to know in order to decide what to do in the last stage H)

Value function for stage $H - 1$

- Value function for stage $H - 1$:

$$Q_{H-1}\left(x_{H-2}, \xi_{[H-1]}\right)$$

$$= \min_{x_{H-1}} c_{H-1}\left(\xi_{[H-1]}\right)^T x_{H-1} + \sum_{\xi_{[H]} \in \Omega_H} P(\xi_{[H]} | \xi_{[H-1]}) \cdot Q_H\left(x_{H-1}, \xi_{[H]}\right)$$

$$W_{H-1}\left(\xi_{[H-1]}\right) x_{H-1} = h_{H-1}\left(\xi_{[H-1]}\right) - T_{H-2}\left(\xi_{[H-1]}\right) x_{H-2}$$

$$x_{H-1} \geq 0$$

- Balances
  - cost of current period $c_{H-1}\left(\xi_{[H-1]}\right)^T x_{H-1}$
  - with expected future cost $\sum_{\xi_{[H]} \in \Omega_H} P(\xi_{[H]} | \xi_{[H-1]}) \cdot Q_H\left(x_{H-1}, \xi_{[H]}\right)$

Dynamic programming equation

- **Dynamic programming equation** for any stage $t = 1, \ldots, H$:

$$Q_t\left(x_{t-1}, \xi_{[t]}\right) = \min_{x_t} c_t\left(\xi_{[t]}\right)^T x_t + \sum_{\xi_{[t+1]} \in \Omega_{t+1}} P(\xi_{[t+1]} | \xi_{[t]}) \cdot Q_{t+1}\left(x_t, \xi_{[t+1]}\right)$$

$$W_t\left(\xi_{[t]}\right) x_t = h_t\left(\xi_{[t]}\right) - T_{t-1}\left(\xi_{[t]}\right) x_{t-1}$$

$$x_t \geq 0$$

- Intuitively, the function $Q_t\left(x_{t-1}, \xi_{[t]}\right)$ quantifies
  - the expected future cost of deciding x_{t-1} at stage $t-1$
  - given that the information available at stage t is $\xi_{[t]}$
  - assuming that we will act optimally from stage t onwards

The hydrothermal planning problem

Model formulation

Value functions

Formulation of hydrothermal planning model

$$(Hydro - ST): \max_{p,d,pH,dH} \sum_{t=1}^{H} \sum_{n \in \Omega_t} P(n) \cdot (V \cdot d_t(n) - \sum_{g \in G} MC_{gt} \cdot p_{gt}(n))$$

$$p_{gt}(n) \leq P_g, g \in G, t = 1, \dots, H, n \in \Omega_t$$

$$e_t(n) = R_t(n) + dH_t(n) - \frac{pH_t(n)}{\eta} + e_{t-1}(A(n)), t = 1, \dots, H, n \in \Omega_t$$

$$d_t(n) + dH_t(n) - \sum_{g \in G} p_{gt}(n) - pH_t(n) = 0, t = 1, \dots, H, n \in \Omega_t$$

$$e_t(n) \leq E, t = 1, \dots, H, n \in \Omega_t$$

$$d_t(n) \leq D_t, t = 1, \dots, H, n \in \Omega_t$$

$$p, d, pH, dH, e \geq 0$$

- Input data: initially stored energy e_0

Stochastic dual dynamic programming algorith,

- The **stochastic dual dynamic programming (SDDP)** algorithm is the most broadly used method for solving the problem in practical applications
- The algorithm combines ideas from Monte Carlo simulation with dynamic programming

The real problem

- Generalizations in real applications (that can be approximated linearly):
  - Complex representation of hydroelectric production as a function of outflow and head
  - Wide geographical coverage: river networks, where water management in certain dams affects water flow in the same river system
- Horizon: a few years
  - Monthly time steps $\Rightarrow$ 120 time steps for a planning horizon of 10 years
  - Practical applications: horizon $\gg$ 120 stages
- The problem is formulated on a lattice in practice (Markov process)
  - States of the art in research and industry applications: 100 outcomes of uncertainty per time stage
- Dimension of state vector
  - Academic research: 50
  - Practical applications: much higher (but without performance guarantees)
  - The state vector includes the water level of hydro dams, and rainfall in previous months

Risk neutrality and risk aversion

- One can prove the equivalence between $(Hydro - ST)$ and a decentralized economic equilibrium, where risk-neutral agents maximize expected profit
- The result can be generalized to agents with risk aversion, under certain (optimistic) assumptions about the availability of financial instruments in the market

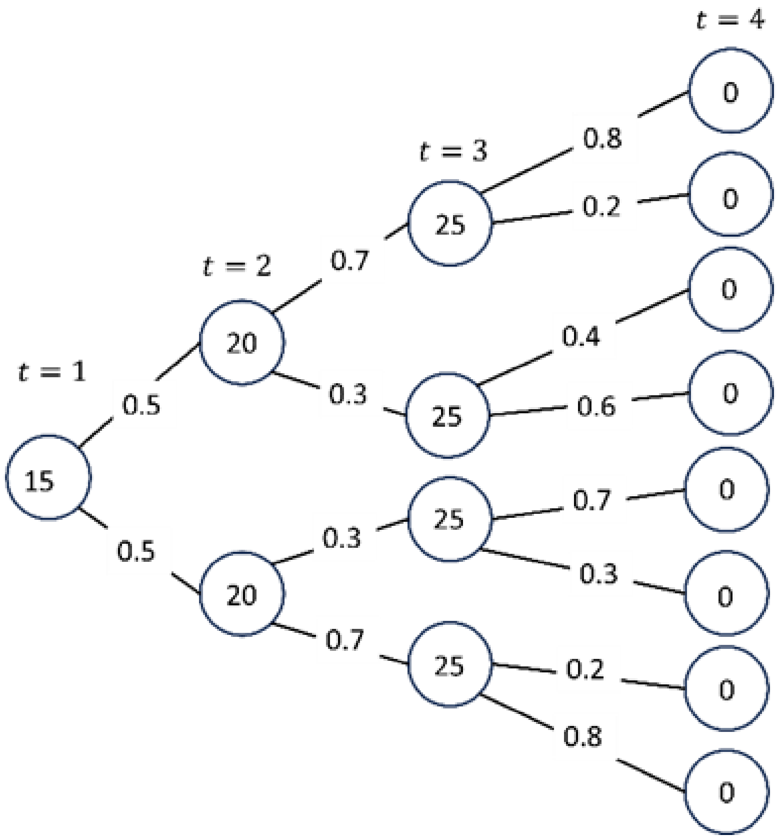
Representation in code

- Define a set of nodes N of the scenario tree
- Uncertain parameters indexed by node of the scenario tree
- Probability of a node is the probability corresponding to the history of outcomes leading to this node
- Each node of the tree corresponds to a time period
- Redundant to define stochastic parameters per node and per time stage, because time stage implied by node of the scenario tree
- Sufficient to define constraints per node of scenario tree, because notation $t = 1, \ldots, H$ and $n \in \Omega_t$ is equivalent to $n \in N$

Example 8.8: four-stage hydrothermal planning on a scenario tree

- We return to the scenario tree of example 8.3
- The hydro reservoir is initially empty
- Two thermal units:
  - G1: 60 MW at 10 $/MWh
  - G2: 100 MW at 50 $/MWh
  - Load:
    - Curtailment cost 1000 $/MWh
    - Load in period 3: 120 MW
    - Load in period 4: 180 MW
  - Reservoir:
    - Energy storage capacity 50 MWh
    - Efficiency 0.8

Example 8.8: hydro storage level



The hydrothermal planning problem

Model formulation

Value functions

Dynamic programming and the value of water

- The application of dynamic programming to hydrothermal scheduling has interesting connections to duality
- The **value of water** is one of the outputs of the SDDP algorithm
  - Computationally: slope of the value functions of the dynamic programming algorithm
  - Intuitively: opportunity cost of using water in hydro units
- Storing, let alone solving, $(Hydro - ST)$ is impossible
  - For a problem with 121 stage and 5 outcomes per stage, the number of nodes in the last stage is $5^{120} >$ number of atoms in the universe
- The dynamic programming algorithm decomposes the problem per time stage and uncertainty outcome

Value function it the last stage

The value function of the last stage Q_H depends on the rainfall outcome $\xi_{[H]}$ and the level of stored hydro energy e_{H-1}

$$Q_H(e_{H-1}, \xi_{[H]}) = \max_{p,d,pH,dH,e} V \cdot d - \sum_{g \in G} MC_{gH} \cdot p_g$$

$$(\mu_g): p_g \leq P_g, g \in G$$

$$(\lambda H): e = R_H(\xi_{[H]}) + dH - \frac{pH}{\eta} + e_{H-1}$$

$$(\lambda): d + dH - \sum_{g \in G} p_g - pH = 0$$

$$(\delta): e \leq E$$

$$(\nu): d \leq D_H$$

$$p, d, pH, dH, e \geq 0$$

What the value function does <u>not</u> depend on

- The value function does ***not*** depend directly on p, pH, dH, d in period $H-1$
- The only thing that matters as far as decisions in stage H are concerned is how these decisions in stage $H-1$ affect the water level e_{H-1}

Proposition 8.1: analytical characterization of the value function

The value function $Q_H\left(e_{H-1}, \xi_{[H]}\right)$ can be expressed as:

- Case 1 (load curtailment): if

$$D_H > \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + \sum_{g \in G} P_g$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right) = V \cdot \left(\sum_{g \in G} P_g + \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right)\right) - \sum_{g \in G} MC_{gH} \cdot P_g$$

Proposition 8.1: analytical characterization of the value function

- Denote the unit before $\bar{g}$ in the merit order as $\bar{g}^-$
- Case 2 (using thermal units): if

$$\eta \cdot \left(R_H(\xi_{[H]}) + e_{H-1}\right) + \sum_{g \in G: MC_{gH} < MC_{\bar{g}^-H}} P_g \leq D_H \leq \eta \cdot \left(R_H(\xi_{[H]}) + e_{H-1}\right) + \sum_{g \in G: MC_{gH} < MC_{\bar{g}H}} P_g$$

then

$$Q_H(e_{H-1}, \xi_{[H]}) = V \cdot D_H - \sum_{g \in G: MC_{gH} < MC_{\bar{g}H}} MC_{gH} \cdot P_g$$
$$-MC_{\bar{g}H} \cdot \left(D_H - \sum_{g \in G: MC_{gH} < MC_{\bar{g}H}} P_g - \eta \cdot \left(R_H(\xi_{[H]}) + e_{H-1}\right) \right)$$

Proposition 8.1: analytical characterization of the value function

- Case 3 (use of water): if

$$D_H < \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + \sum_{g \in G} P_g$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right) = V \cdot D_H$$

Proof: dual of the last-stage problem

The dual of the linear program that defines $Q_H\left(e_{H-1}, \xi_{[H]}\right)$ is:

$$\min_{\mu,\lambda H,\lambda,\delta,\nu} \sum_{g \in G} \mu_g \cdot P_g - \lambda H \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + \delta \cdot E + \nu \cdot D_H$$

$$(p_g): \mu_g - \lambda \geq -MC_{gH}, g \in G$$

$$(d): \nu + \lambda \geq V$$

$$(pH): -\frac{\lambda H}{\eta} - \lambda \geq 0$$

$$(dH): \lambda H + \lambda \geq 0$$

$$(e): -\lambda H + \delta \geq 0$$

$$\mu \geq 0, \nu \geq 0, \delta \geq 0$$

Proof: sign of λH and λ

- Suppose that $\lambda H > 0$
- Then $\lambda \geq -\lambda H$ and $\lambda \leq -\lambda H / \eta$
- Which is impossible, because $0 < \eta < 1$
- Thus $\lambda H \leq 0$, and $\lambda \geq 0$

Proof: optimal values of dual multipliers

- Arguing by contradiction we can prove that $\delta = 0$ at the optimal solution
- Similarly, we show that $\mu_g = \max(\lambda - MC_{gH}, 0)$ at the optimal solution
- And $\nu = \max(V - \lambda, 0)$ at the optimal solution
- Finally, $\lambda H = \lambda \cdot \eta$ at the optimal solution
- Thus the optimal objective value of the dual problem is

$$d^* = \sum_{g \in G} \max(\lambda - MC_{gH}, 0) \cdot P_g + \lambda \cdot \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right)$$
$$+ \max(V - \lambda, 0) \cdot D_H$$

Proof: optimal solution in case of water oversupply

- For $\lambda = 0$, we have
$$d^* = V \cdot D_H$$
- None of the thermal units is producing, and demand is covered by hydro units
  - We can show this using KKT conditions
- The value of water is zero, because additional water is not useful

Proof: optimal solution when thermal units are used

- For $\lambda = MC_{\bar{g}H}$, where $\bar{g}$ the marginal thermal unit, we have

$$d^* = V \cdot D_H - \sum_{g \in G : MC_{gH} < MC_{\bar{g}H}} MC_{gH} \cdot P_g$$
$$- MC_{\bar{g}H} \cdot \left(D_H - \sum_{g \in G : MC_{gH} < MC_{\bar{g}H}} P_g - \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1} \right) \right)$$

- All demand is satisfied, units cheaper than $\bar{g}$ produce at technical maximum, and $\bar{g}$ produces the rest
- The value of water is $\eta \cdot MC_{\bar{g}H}$, because 1 MWh of additional water results in producing η MWh less energy from unit $\bar{g}$

Proof: optimal solution in case of load shedding

- For $\lambda = V$, we have

$$d^* = V \cdot \left(\sum_{g \in G} P_g + \eta \cdot \left(R_H(\xi_{[H]}) + e_{H-1} \right) \right) - \sum_{g \in G} MC_{gH} \cdot P_g$$

- Demand only partially satisfied, all units producing at full capacity
- The value of water is $\eta \cdot V$, because 1 MWh of additional water reduces load shedding by η MWh

Structure of value function

- The value function is piecewise linear concave
- This is already foreseen by the theory (of appendix A.10 [1])
- From proposition 2.8 of [1] we know that the dual multiplier λH is the value of water (and the slope of Q_{H-1} with respect to e_{H-1})
- Structure of the value function:
  - Geometric intuition: change of optimal basis as we change e_{H-1}, so the slope of Q_{H-1} changes
  - Physical intuition: changes in e_{H-1} result in “phase changes”: hydro only → hydro thermal → load shedding

Value function at stage t

- For any stage t, the value function is

$$Q_t\left(e_{t-1}, \xi_{[t]}\right) = \max_{p,d,pH,dH,e} V \cdot d - \sum_{g \in G} MC_{gt} \cdot p_g + \sum_{n \in \Omega_{t+1}} P(\xi_{t+1} = n | \xi_{[t]}) \cdot Q_{t+1}\left(e, \xi_{[t+1]}\right)$$

$$\left(\mu_g\right): p_g \leq P_g, g \in G$$

$$(\lambda H): e = R_t\left(\xi_{[t]}\right) + dH - \frac{pH}{\eta} + e_{t-1}$$

$$(\lambda): d + dH - \sum_{g \in G} p_g - pH = 0$$

$$(\delta): e \leq E$$

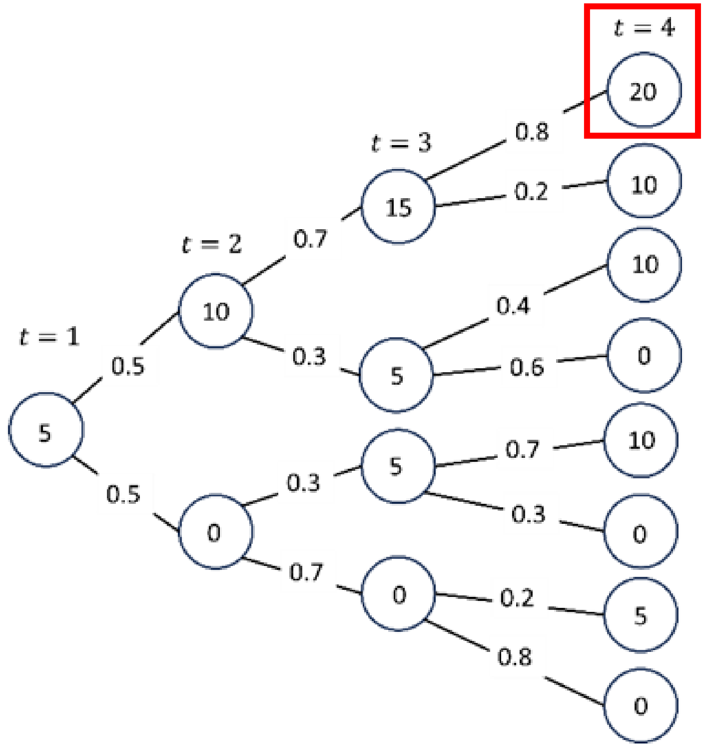
$$(\nu): d \leq D_t$$

$$p, d, pH, dH, e \geq 0$$

- Like Q_H, Q_t is piecewise linear concave

Example 8.9: value functions on a scenario tree

- Recall the scenario tree of example 8.8 (see figure)
- We use proposition 8.1 to compute the value function
- We focus on the node in the red box



Example 8.9: load shedding

- If

$$D_H > \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + \sum_{g \in G} P_g \Rightarrow 180 > 0.8 \cdot (20 + e_3) + (60 + 100)$$

$$\Rightarrow e_3 < 5$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right) = V \cdot \left(\sum_{g \in G} P_g + \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right)\right) - \sum_{g \in G} MC_{gH} \cdot P_g$$

$$= 1000 \cdot \left(60 + 100 + 0.8 \cdot (20 + e_3)\right) - 10 \cdot 60 - 50 \cdot 100$$

$$= 170400 + 800 \cdot e_3$$

- For $e_3 = 5$, the value function is \$174400

Example 8.9: the expensive thermal unit G2 is marginal

- If

$$\eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + P_{G_1} \leq D_H \leq \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + P_{G_1} + P_{G_2} \Rightarrow$$
$$0.8 \cdot (20 + e_3) + 60 \leq 180 \leq 0.8 \cdot (20 + e_3) + 60 + 100 \Rightarrow$$
$$5 \leq e_3 \leq 130$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right)$$
$$= V \cdot D_H - MC_{G_1 H} \cdot P_{G_1} - MC_{G_2 H} \cdot \left(D_H - P_{G_1} - \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right)\right)$$
$$= 1000 \cdot 180 - 10 \cdot 60 - 50 \cdot \left(180 - 60 - 0.8 \cdot (20 + e_3)\right) = 174200 + 40 \cdot e_3$$

- For $e_3 = 5$, the value function is \$174400 (so indeed continuous)
- For $e_3 = 130$, the value function is \$179400

Example 8.9: cheap thermal unit G1 is marginal

- If

$$\eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) \le D_H \le \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) + P_{G_1} \Rightarrow$$
$$0.8 \cdot (20 + e_3) \le 180 \le 0.8 \cdot (20 + e_3) + 60 \Rightarrow$$
$$130 \le e_3 \le 205$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right) = V \cdot D_H - MC_{G_1 H} \cdot \left(D_H - \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right)\right)$$
$$= 1000 \cdot 180 - 10 \cdot \left(180 - 0.8 \cdot (20 + e_3)\right) = 178360 + 8 \cdot e_3$$

- For $e_3 = 130$, the value function is \$179400 (so indeed continuous)
- For $e_3 = 205$, the value function is \$180000

Example 8.9: the hydro unit is marginal

- If

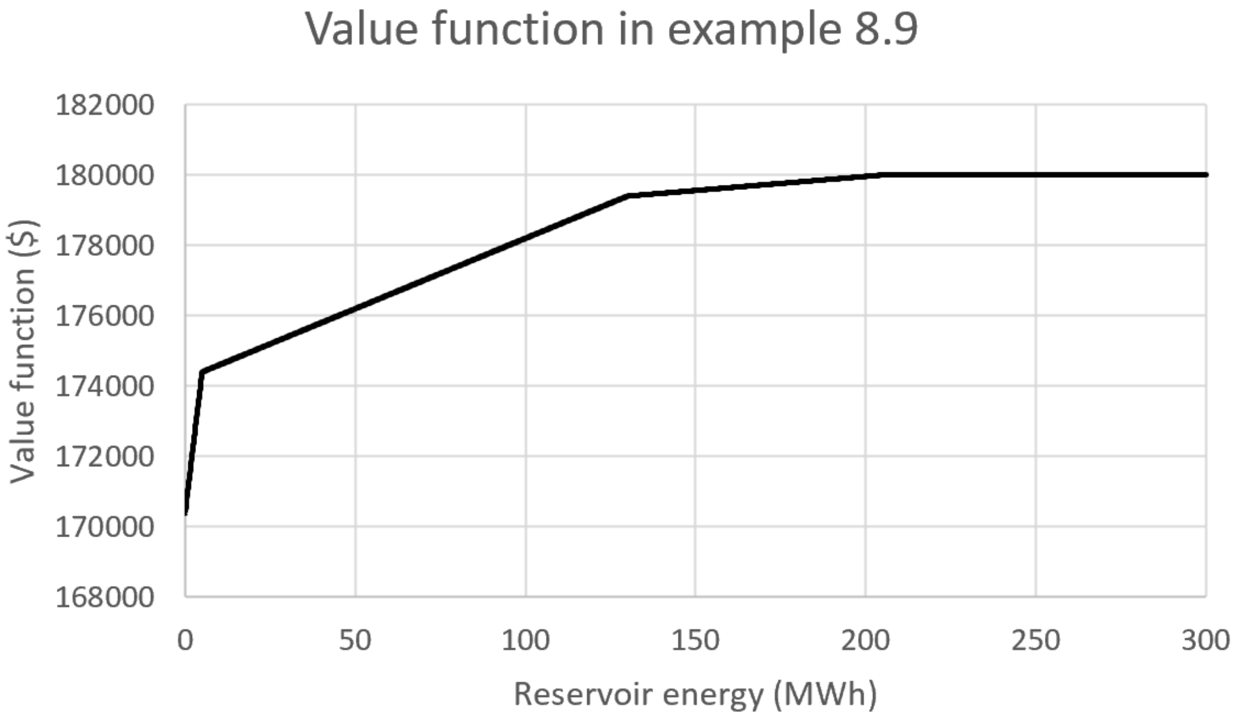
$$D_H < \eta \cdot \left(R_H\left(\xi_{[H]}\right) + e_{H-1}\right) \Rightarrow$$
$$180 < 0.8 \cdot (20 + e_3) \Rightarrow e_3 > 205$$

then

$$Q_H\left(e_{H-1}, \xi_{[H]}\right) = V \cdot D_H = 1000 \cdot 180 = 180000$$

- For $e_3 = 205$, the value function is \$180000 (so indeed continuous)

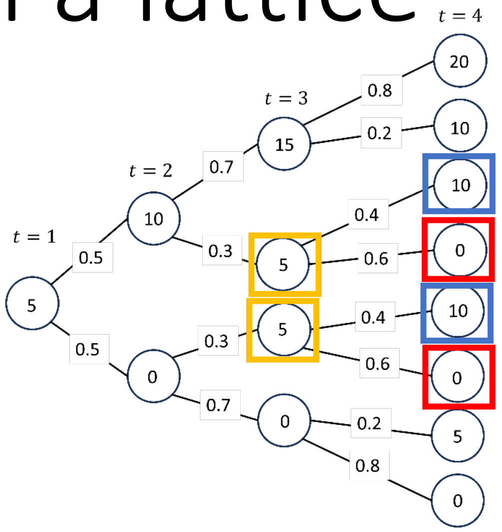
Example 8.9: graphical representation of the value function



Value functions for Markov processes

- If uncertainty is a Markov process, the value functions are identical when the value of ξ_t is the same in a given time stage
- Therefore, value functions do <u>not</u> depend on how we go there ($\xi_{[t]}$), but only where we are (ξ_t)
  - Intuitively consistent with behavior of Markov process, where what happens in the future depends only on where we are, not how we got there
- Important computational savings: algorithms like SDDP estimate value functions
  - Makes a big difference if these functions need to be estimate for every node of a scenario tree or lattice

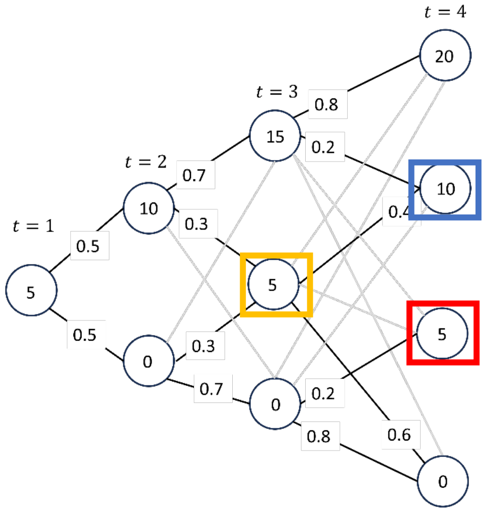
Example 8.10: value functions on a lattice



- We return to a process that can be described on a lattice
- We use the process of example 8.9 to compute the value function
- The value functions in the blue/red nodes of the scenario tree are identical, since
  - $R_4 = 10$ in both blue nodes
  - $R_4 = 0$ in both red nodes
- Therefore

$$0.4 \cdot Q_4\big(e_3, (5{,}10{,}5{,}10)\big) + 0.6 \cdot Q_4\big(e_3, (5{,}10{,}5{,}0)\big)$$
$$= 0.4 \cdot Q_4\big(e_3, (5{,}0{,}5{,}10)\big) + 0.6 \cdot Q_4\big(e_3, (5{,}0{,}5{,}0)\big)$$

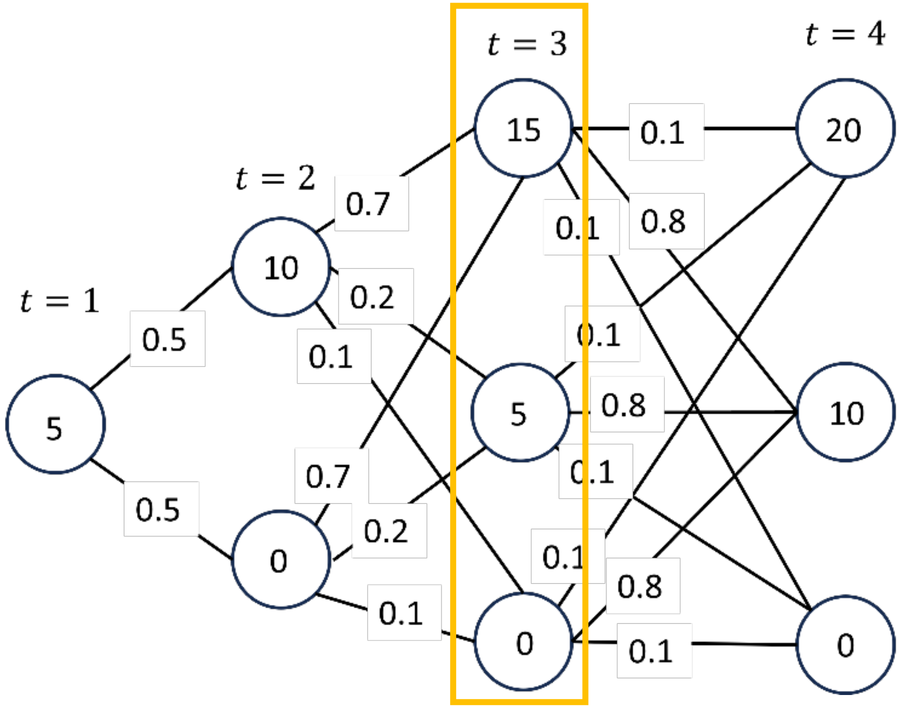
- Thus, based on the dynamic programming equation, the value functions in the orange nodes are identical
- Thus on a lattice it is enough to compute a value function per node of the lattice



Value functions for stagewise independent processes

- For stagewise independent processes, value functions are indeitical for all nodes of the same stage t
- So they are only functions of water level, $Q_t(e_{t-1})$, and not amount of rainfall, $Q_t(e_{t-1}, \xi_t)$

Example 8.11: value functions for stagewise independent processes



All orange nodes of the lattice have the same value function

Performance of stochastic programs

Alternatives for analyzing uncertainty

- Stochastic programs are computationally “heavy”
- We can analyze models under uncertainty with less computationally demanding techniques (which are approximations):
  - Performance when we have perfect foresight
  - Performance when uncertain parameters are replaced by their expected value

The function $z(x, \xi)$

- We focus on two-stage stochastic programs
- We define the function $z(x, \xi)$ as:

$$z(x, \xi) = c^T x + Q(x, \xi) + I(Ax = b, x \geq 0)$$

where

$$Q(x, \xi) = \min_y \{q(\omega)^T y | W(\omega) y = h(\omega) - T(\omega) x\}$$

and $I(x|K)$ equal to 0 for $x \in K$ and $+\infty$ for $x \notin K$

- Interpretation of $z(x, \xi)$: cost given
  - That we have decided x in the first stage
  - Outcome ξ occurs in the second stage
  - We react optimally in the second stage
- Easy to compute for given (x, ξ) (small linear program)

Wait-and-see value

- The **wait-and-see value** is the expected value of reacting with perfect foresight $x^*(\xi)$ to ξ
$$WS = \mathbb{E}[\min_x z(x,\xi)] = \mathbb{E}[z(x^*(\xi),\xi)]$$
- The **here-and-now value** is the expected value of the stochastic program:
$$SP = \min_x \mathbb{E}[z(x,\xi)]$$
- We have $WS \leq SP$, because we act with prior knowledge of what will happen
- **Expected value of perfect information**:
$$EVPI = SP - WS$$
- Interpretation of EVPI: how much we are willing to pay for a perfect forecast

Computational requirements

- Computing SP requires solving a potentially massive scale linear program (hard)
- Computing WS requires solving may small independent linear programs (computationally easier)

Example 8.12: expected value of perfect information in hydrothermal planning

- Return to example 8.2
- Difference: rainfall in period 2 under scenario 2: $R_2(2) = 55$
- Here and now value:

$$SP = \$148110$$

- Wait and see value:

$$WS = \$148115$$

- Expected value of perfect information:

$$EVPI = WS - SP = 148115 - 148110 = \$5$$

- The difference is that with perfect foresight we transfer less water to the reservoir in scenario 2
  - The policy without perfect foresight produces at full capacity in period 1
  - Slightly inefficient due to efficiency losses ($\eta = 0.8$) of hydro plant
  - But we protect ourselves from load shedding in the unfavorable scenario

Expected value problem and expected value solution

- In the **expected/mean value problem**, we replace uncertain parameters with their expected value, $\bar{\xi} = \mathbb{E}[\xi]$
- The **expected value solution** $x^*(\bar{\xi})$ is the optimal reaction to expected uncertainty
- The expected value of using the expected value solution $x^*(\bar{\xi})$ is:
$$EEV = \mathbb{E}\left[z\left(x^*(\bar{\xi}), \xi\right)\right]$$
- The **value of the stochastic solution** is
$$VSS = EEV - SP$$

Computational aspects

- Computing $x^*(\bar{\xi})$ is relatively easy (small linear program)
- Computation of EEV sets first-stage decision to $x^*(\bar{\xi})$, and computes the optimal decision of the second stage for every $\omega \in \Omega$
  - Computationally easy for a reasonable number of scenarios, $\omega \in \Omega$ (set of small linear programs)

Example 8.13: value of stochastic solution

- We return to example 8.12, with mean rainfall in period 2 equal to
$$\bar{\xi} = 0.5 \cdot 0 + 0.5 \cdot 55 = 27.5$$
- Optimal first-stage decision for $\xi = \bar{\xi}$: store 10 MWh of hydro
- Expected value of using the expected value solution:
$$EEV \ = \$\, 148110$$
- Value of stochastic solution:
$$VSS = SP - EEV = 148110 - 148110 = \$0$$
- In other models, VSS is typically (very) positive

Sample average approximation

When computing the expected value is computationally hard (e.g. for the case of a continuous random parameter ξ), we can estimate WS and EEV using **sample average approximation**:

- For $i = 1, \dots, K$
  - Sample ξ_i
  - Compute $x^*(\bar{\xi})$
  - Compute $WS_i = z(x^*(\xi_i), \xi_i)$ and $EEV_i = c^T x^*(\bar{\xi}) + Q(x^*(\bar{\xi}), \xi_i)$
- Estimate $\overline{WS} = \frac{1}{K}\sum_{i=1}^{K} WS_i$ and $\overline{EEV} = \frac{1}{K}\sum_{i=1}^{K} EEV_i$

The central limit theorem

- Intuition: the more samples K, the more accurate the estimation of WS and EEV
- This intuition can be made mathematically precise with the **central limit theorem**

Consider a sequence of independent, identically distributed random variables $X_1, X_2, \ldots$, with $\mathbb{E}[X_i] = \mu$ and $Var(X_i) = \sigma^2 < \infty$. Then, as n goes to infinity, the random variable $\sqrt{n} \cdot (\frac{1}{n}\sum_{i=1}^{n} X_i - \mu)$ converges in distribution to a normal random variable, $N(0, \sigma^2)$:

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

Importance sampling

- Sample average approximation can be “slow”, because it may take a long time to observe rare samples with a large impact on the mean
- This issue can be mitigated with **importance sampling**:

- Suppose that we want to estimate $\mathbb{E}[C]$, where C is distributed according to a density function f
- Sample average approximation samples C_i based on the distribution f and estimates $\mathbb{E}[C]$ ως $\frac{1}{N}\sum_{i=1}^{N} C_i$
- In importance sampling we sample C_i based on the distribution

$$g(x) = \frac{f(x) \cdot x}{\mathbb{E}[C]}$$

and estimate $\mathbb{E}[C]$ as

$$\frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i) \cdot x_i}{g(x_i)}$$

where we use a reasonable estimation of $\mathbb{E}[C]$ in the denominator $g(x)$

References

[1] A. Papavasiliou, Optimization Models in Electricity Markets, Cambridge University Press

https://www.cambridge.org/highereducation/books/optimization-models-in-electricity-markets/0D2D36891FB5EB6AAC3A4EFC78A8F1D3#overview