Introduction to Decomposition Methods in Stochastic Optimization

V. Leclère

2017, March 23

École des Ponts ParisTech

UNIVERSITÉ PARIS-EST

Presentation Outline

Presentation Outline

An optimization problem

A standard optimization problem

$$\begin{aligned} \min_{u_0} \quad & L(u_0) \\ s.t. \quad & g(u_0) \leq 0 \end{aligned}$$

An optimization problem with uncertainty

Adding uncertainty ξ in the mix

$$\min_{u_0} \quad L(u_0, \xi)$$
$$s.t. \quad g(u_0, \xi) \leq 0$$

Remarks:

- ξ is unknown. Two main way of modelling it:
  - $\xi \in \Xi$ with a known uncertainty set Ξ, and a pessimistic approach. This is the robust optimization approach (RO).
  - ξ is a random variable with known probability law. This is the Stochastic Programming approach (SP).
- Cost is not well defined.
  - RO : $\max_{\xi \in \Xi} L(u, \xi)$.
  - SP : $\mathbb{E}\big[L(u, \xi)\big]$.
- Constraints are not well defined.
  - RO : $g(u, \xi) \leq 0, \qquad \forall \xi \in \Xi$.
  - SP : $g(u, \xi) \leq 0, \qquad \mathbb{P} - a.s.$.

An optimization problem with uncertainty

Adding uncertainty ξ in the mix

$$\begin{aligned} \min_{u_0} \quad & L(u_0, \xi) \\ s.t. \quad & g(u_0, \xi) \leq 0 \end{aligned}$$

Remarks:

- ξ is unknown. Two main way of modelling it:
  - $\xi \in \Xi$ with a known uncertainty set Ξ, and a pessimistic approach. This is the robust optimization approach (RO).
  - ξ is a random variable with known probability law. This is the Stochastic Programming approach (SP).
- Cost is not well defined.
  - RO : $\max_{\xi \in \Xi} L(u, \xi)$.
  - SP : $\mathbb{E}\big[L(u, \boldsymbol{\xi})\big]$.
- Constraints are not well defined.
  - RO : $g(u, \xi) \leq 0, \qquad \forall \xi \in \Xi$.
  - SP : $g(u, \boldsymbol{\xi}) \leq 0, \qquad \mathbb{P} - a.s.$.

An optimization problem with uncertainty

Adding uncertainty ξ in the mix

$$\min_{u_0} \quad L(u_0, \xi)$$
$$s.t. \quad g(u_0, \xi) \leq 0$$

Remarks:

- ξ is unknown. Two main way of modelling it:
  - $\xi \in \Xi$ with a known uncertainty set Ξ, and a pessimistic approach. This is the robust optimization approach (RO).
  - ξ is a random variable with known probability law. This is the Stochastic Programming approach (SP).
- Cost is not well defined.
  - RO : $\max_{\xi \in \Xi} L(u, \xi)$.
  - SP : $\mathbb{E}\big[L(u, \boldsymbol{\xi})\big]$.
- Constraints are not well defined.
  - RO : $g(u, \xi) \leq 0, \qquad \forall \xi \in \Xi$.
  - SP : $g(u, \boldsymbol{\xi}) \leq 0, \qquad \mathbb{P} - a.s.$.

Alternative cost functions I

- When the cost $L(u, \xi)$ is random it might be natural to want to minimize its expectation $\mathbb{E}\big[L(u, \xi)\big]$.
- This is even justified if the same problem is solved a large number of time (Law of Large Number).
- In some cases the expectation is not really representative of your risk attitude. Lets consider two examples:
  - Are you ready to pay \$1000 to have one chance over ten to win \$10000 ?
  - You need to be at the airport in 1 hour or you miss your flight, you have the choice between two mean of transport, one of them take surely 50', the other take 40' four times out of five, and 70' one time out of five.

Alternative cost functions II

Here are some cost functions you might consider

- Probability of reaching a given level of cost : $\mathbb{P}(L(u,\boldsymbol{\xi}) \leq 0)$
- Value-at-Risk of costs $V@R_\alpha(L(u,\boldsymbol{\xi}))$, where for any real valued random variable $\boldsymbol{X}$,

$$V@R_\alpha(\boldsymbol{X}) := \inf_{t \in \mathbb{R}} \Big\{ \mathbb{P}(\boldsymbol{X} \geq t) \leq \alpha \Big\}.$$

  In other word there is only a probability of α of obtaining a cost worse than $V@R_\alpha(\boldsymbol{X})$.
- Average Value-at-Risk of costs $AV@R_\alpha(L(u,\boldsymbol{\xi}))$, which is the expected cost over the α worst outcomes.

Alternative constraints I

- The natural extension of the deterministic constraint $g(u, \xi) \leq 0$ to $g(u, \boldsymbol{\xi}) \leq 0 \ \mathbb{P} - as$ can be extremely conservative, and even often without any admissible solutions.
- For example, if u is a level of production that need to be greated than the demand. In a deterministic setting the realized demand is equal to the forecast. In a stochastic setting we add an error to the forecast. If the error is unbouded (e.g. Gaussian) no control u is admissible.

Alternative constraints II

Here are a few possible constraints

- $\mathbb{E}\big[g(u,\xi)\big] \leq 0$, for quality of service like constraint.
- $\mathbb{P}(g(u,\xi) \leq 0) \geq 1-\alpha$ for chance constraint. Chance constraint is easy to present, but might lead to misconception as nothing is said on the event where the constraint is not satisfied.
- $AV@R_\alpha(g(u,\xi)) \leq 0$

Presentation Outline

One-Stage Problem

Assume that $\boldsymbol{\Xi}$ as a discrete distribution[1], with $\mathbb{P}(\boldsymbol{\xi} = \xi_i) = p_i > 0$ for $i \in [\![1, n]\!]$. Then, the one-stage problem

$$\begin{aligned} \min_{u_0} \quad & \mathbb{E}\Big[L(u_0, \boldsymbol{\xi})\Big] \\ s.t. \quad & g(u_0, \boldsymbol{\xi}) \leq 0, \qquad \mathbb{P} - a.s \end{aligned}$$

can be written

$$\begin{aligned} \min_{u_0} \quad & \sum_{i=1}^{n} p_i L(u_0, \xi_i) \\ s.t \quad & g(u_0, \xi_i) \leq 0, \qquad \forall i \in [\![1, n]\!]. \end{aligned}$$

[1] If the distribution is continuous we can sample and work on the sampled distribution, this is called the Sample Average Approximation approach with lots of guarantee and results

Recourse Variable

In most problem we can make a correction u_1 once the uncertainty is known:

$$u_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow u_1 .$$

As the recourse control u_1 is a function of $\boldsymbol{\xi}$ it is a random variable. The two-stage optimization problem then reads

$$\begin{aligned} \min_{u_0} \quad & \mathbb{E}\Big[L(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1) \Big] \\ s.t. \quad & g(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1) \leq 0, \qquad \mathbb{P} - a.s \\ & \sigma(\boldsymbol{u}_1) \subset \sigma(\boldsymbol{\xi}) \end{aligned}$$

Two-stage Problem

The extensive formulation of

$$\begin{aligned} \min_{u_0, \boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1) \leq 0, \qquad \mathbb{P} - a.s \end{aligned}$$

is

$$\begin{aligned} \min_{u_0, \{u_1^i\}_{i \in [\![1,n]\!]}} \quad & \sum_{i=1}^{n} p_i L(u_0, \xi_i, u_1^i) \\ s.t \quad & g(u_0, \xi_i, u_1^i) \leq 0, \qquad \forall i \in [\![1, n]\!]. \end{aligned}$$

Recourse assumptions

- We say that we are in a complete recourse framework, if for all u_0, and all possible outcome $\boldsymbol{\xi}$, every control $\boldsymbol{u}_1$ is admissible.
- We say that we are in a relatively complete recourse framework, if for all u_0, and all possible outcome ξ, there exists a control $\boldsymbol{u}_1$ that is admissible.
- For a lot of algorithm relatively complete recourse is a condition of convergence. It means that there is no induced constraints.

Presentation Outline

Linear 2-stage stochastic program

Consider the following problem

$$\begin{aligned}
\min \quad & \mathbb{E}\Big[c^T x + \boldsymbol{q}^T \boldsymbol{y}\Big] \\
s.t. \quad & Ax = b, \quad x \geq 0 \\
& \boldsymbol{T}x + \boldsymbol{W}\boldsymbol{y} = \boldsymbol{h}, \quad \boldsymbol{y} \geq 0, \qquad \mathbb{P} - a.s. \\
& x \in \mathbb{R}^n, \quad \sigma(\boldsymbol{y}) \subset \sigma(\underbrace{\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{W}, \boldsymbol{h}}_{\boldsymbol{\xi}})
\end{aligned}$$

With associated Extended Formulation

$$\begin{aligned}
\min \quad & c^T x + \sum_{i=1}^{N} \pi_i q_i^T y_i \\
s.t. \quad & Ax = b, \quad x \geq 0 \\
& T_i x + W_i y_i = h_i, \quad y_i \geq 0, \forall i
\end{aligned}$$

Decomposition of linear 2-stage stochastic program

We rewrite the extended formulation as

$$\begin{aligned} \min \quad & c^T x + \theta \\ s.t. \quad & Ax = b, \quad x \geq 0 \\ & \theta \geq Q(x) \qquad\qquad x \in \mathbb{R}^n \end{aligned}$$

where $Q(x) = \sum_{i=1}^{N} \pi_i Q_i(x)$ with

$$\begin{aligned} Q_i(x) := \min_{y_i \in \mathbb{R}^m} \quad & q_i^T y_i \\ s.t. \quad & T_i x + W_i y_i = h_i, \quad y_i \geq 0 \end{aligned}$$

Note that $Q(x)$ is a polyhedral function of x, hence $\theta \geq Q(x)$ can be rewritten $\theta \geq \alpha_k^T x + \beta_k, \forall k$.

Obtaining cuts

Recall that

$$Q_i(x) := \min_{y_i \in \mathbb{R}^m} \quad q_i^T y_i \qquad s.t. \quad T_i x + W_i y_i = h_i, \quad y_i \geq 0$$

can also be written (through strong duality)

$$Q_i(x) := \max_{\lambda_i \in \mathbb{R}^m} \quad \lambda_i^T (h_i - T_i x) \qquad s.t. \quad W_i^T \lambda_i \leq q_i$$

In particular we have, for the optimal solution $\lambda_i^\sharp$,

$$Q_i(x) \geq \underbrace{h_i^T \lambda_i^\sharp}_{\beta_i^k} \underbrace{-(\lambda_i^\sharp)^T T_i}_{\alpha_i^k} x.$$

L-shaped method

1. We have a collection of K cuts, such that $Q(x) \geq \alpha^k x + \beta^k$.
2. Solve the master problem, with optimal primal solution x^k.

$$\begin{aligned} \min_{Ax=b, x\geq 0} \quad & c^T x + \theta \\ s.t. \quad & \theta \geq \alpha^k x + \beta^k, \forall k = 1, .., K \end{aligned}$$

3. Solve N slave dual problems, with optimal dual solution $\lambda_i^\sharp$

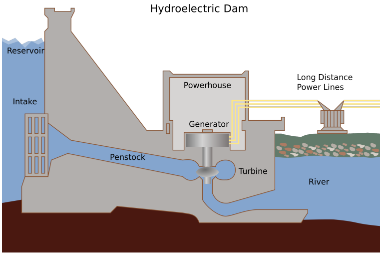
$$\begin{aligned} \max_{\lambda_i \in \mathbb{R}^m} \quad & \lambda_i^T (h_i - T_i x^k) \\ s.t. \quad & W_i^T \lambda_i \leq q_i \end{aligned}$$

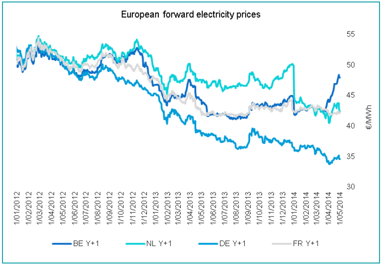
4. construct new cut with

$$\alpha^{K+1} := \sum_{i=1}^N -\pi_i (\lambda_i^\sharp)^T T_i, \qquad \beta^{K+1} := \sum_{i=1}^N \pi_i h_i^T \lambda_i^\sharp.$$

Presentation Outline

Managing a dam





A dam can be seen as a battery, with random inflow of free electricity to be used at the best time.

A multistage problem

Let formulate this as a mathematical problem

$$\begin{aligned} \min_{u_1,\dots,u_{T-1}} \quad & \sum_{t=1}^{N} L_t(x_t, u_t) \\ s.t \quad & x_{t+1} = f_t(x_t, u_t), \quad x_0 \text{ fixed} && t = 1, \dots, T-1 \\ & u_t \in U_t, \quad x_t \in X_t && t = 1, \dots, T-1 \end{aligned}$$

- x_t is the state of the system at time t (e.g. the stock of water)
- u_t is the control applied at time t (e.g. the water turbined)
- f_t is the dynamic of the system, i.e. the rule describing the evolution of the system (e.g. $f_t(x_t, u_t) = x_t - u_t + W_t$)
- U_t (resp X_t) are constraints set on the control u_t (resp the state x_t)

Open-loop VS closed-loop solution

$$\min_{u_1,\dots,u_{T-1}} \quad \sum_{t=1}^{N} L_t(x_t,u_t)$$
$$s.t \quad x_{t+1} = f_t(x_t,u_t), \quad x_0 \text{ fixed} \qquad t=1,\dots,T-1$$
$$u_t \in U_t, \quad x_t \in X_t \qquad t=1,\dots,T-1$$

- An open-loop solution to the problem is a planning $(u_1,\dots,u_{T-1})$.
- A closed-loop solution to the problem is a policy, i.e. a function π take into argument the current state x_t and the current time t and return a control u_t.
- In a deterministic setting a closed loop solution can be reduced to an open-loop solution.

Open-loop VS closed-loop solution

$$\min_{u_1,\dots,u_{T-1}} \quad \sum_{t=1}^{N} L_t(x_t, u_t)$$
$$s.t \quad x_{t+1} = f_t(x_t, u_t), \quad x_0 \text{ fixed} \qquad t = 1, \dots, T-1$$
$$u_t \in U_t, \quad x_t \in X_t \qquad t = 1, \dots, T-1$$

- An open-loop solution to the problem is a planning $(u_1, \dots, u_{T-1})$.
- A closed-loop solution to the problem is a policy, i.e. a function π take into argument the current state x_t and the current time t and return a control u_t.
- In a deterministic setting a closed loop solution can be reduced to an open-loop solution.

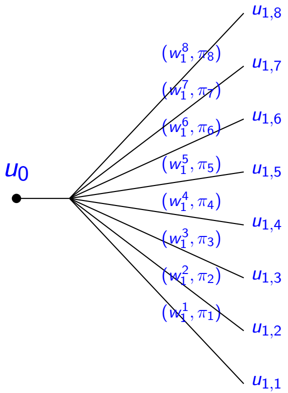
What happen with stochasticity ?

- Assume now that the dynamic is not deterministic anymore (e.g. the inflow are random).
- In this case an open-loop solution is a solution where you decide your production beforehand and stick to it, whatever the actual current state.
- Whereas a closed-loop solution will look at the current state before choosing the control.
- Even if you look for an open-loop solution, replacing the random vector by its expectation is not optimal. It can even give wrong indication.

What happen with stochasticity ?

- Assume now that the dynamic is not deterministic anymore (e.g. the inflow are random).
- In this case an open-loop solution is a solution where you decide your production beforehand and stick to it, whatever the actual current state.
- Whereas a closed-loop solution will look at the current state before choosing the control.
- Even if you look for an open-loop solution, replacing the random vector by its expectation is not optimal. It can even give wrong indication.

Where do we come from: two-stage programming



- We take decisions in two stages

$$u_0 \rightsquigarrow \boldsymbol{W}_1 \rightsquigarrow \boldsymbol{u}_1 \ ,$$

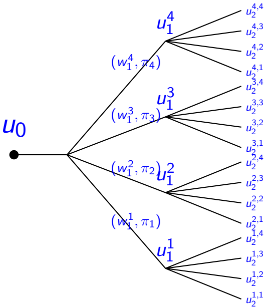
with $\boldsymbol{u}_1$: recourse decision .

- On a tree, it means solving the extensive formulation:

$$\min_{u_0, u_{1,s}} \sum_{s \in \mathbb{S}} \pi_s \Big[\langle c_s \ , u_0 \rangle + \langle p_s \ , u_{1,s} \rangle \Big] \ .$$

We have as many $u_{1,s}$ as scenarios!

Extending two-stage to multistage programming



$$\min_{\boldsymbol{u}} \ \mathbb{E}\big(j(\boldsymbol{u}, \boldsymbol{W})\big)$$
$$\boldsymbol{U} = (\boldsymbol{u}_0, \cdots, \boldsymbol{U}_T)$$
$$\boldsymbol{W} = (\boldsymbol{w}_1, \cdots, \boldsymbol{W}_T)$$

We take decisions in T stages

$$\boldsymbol{W}_0 \rightsquigarrow \boldsymbol{u}_0 \rightsquigarrow \boldsymbol{W}_1 \rightsquigarrow \boldsymbol{u}_1 \rightsquigarrow \cdots \rightsquigarrow \boldsymbol{w}_T \rightsquigarrow \boldsymbol{u}_T \ .$$

Introducing the non-anticipativity constraint

We do not know what holds behind the door.

Non-anticipativity

At time t, decisions are taken sequentially, only knowing the past realizations of the perturbations.

Mathematically, this is equivalent to say that at time t, the decision $\boldsymbol{u}_t$ is

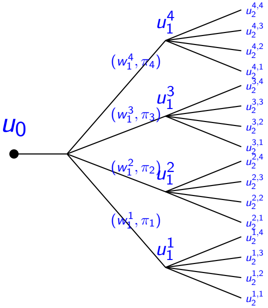
1. a function of past noises

$$\boldsymbol{u}_t = \pi_t(\boldsymbol{W}_0, \cdots, \boldsymbol{W}_t) ,$$

2. taken knowing the available information,

$$\sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{W}_0, \cdots, \boldsymbol{w}_t) .$$

Multistage extensive formulation approach



Assume that $w_t \in \mathbb{R}^{n_w}$ can take n_w values and that $U_t(x)$ can take n_u values.

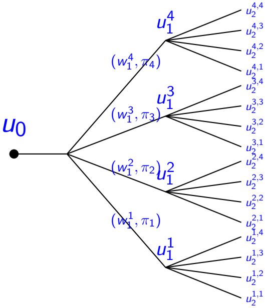
Then, considering the extensive formulation approach, we have

- n_w^T scenarios.
- $(n_w^{T+1} - 1)/(n_w - 1)$ nodes in the tree.
- Number of variables in the optimization problem is roughly $n_u \times (n_w^{T+1} - 1)/(n_w - 1) \approx n_u n_w^T$.

The complexity grows exponentially with the number of stage. :-(

A way to overcome this issue is to compress information!

Multistage extensive formulation approach



Assume that $w_t \in \mathbb{R}^{n_w}$ can take n_w values and that $U_t(x)$ can take n_u values.

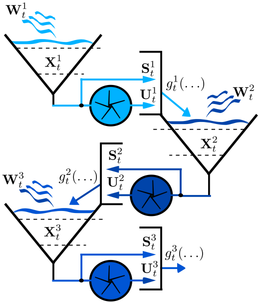
Then, considering the extensive formulation approach, we have

- n_w^T scenarios.
- $(n_w^{T+1} - 1)/(n_w - 1)$ nodes in the tree.
- Number of variables in the optimization problem is roughly $n_u \times (n_w^{T+1} - 1)/(n_w - 1) \approx n_u n_w^T$.

The complexity grows exponentially with the number of stage. :-(

A way to overcome this issue is to compress information!

Illustrating extensive formulation with the damsvalley example



- 5 interconnected dams
- 5 controls per timesteps
- 52 timesteps (one per week, over one year)
- $n_w = 10$ noises for each timestep

We obtain 10^{52} scenarios, and $\approx 5.10^{52}$ constraints in the extensive formulation ... Estimated storage capacity of the Internet: 10^{24} bytes.

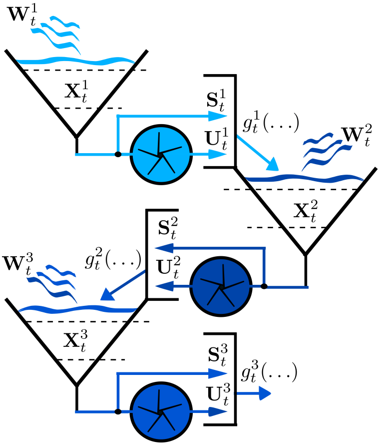
Mulststage Stochastic Optimization: an Example

Objective function:

$$\mathbb{E}\Big[\sum_{i=1}^{N}\sum_{t=0}^{T-1} L_t^i(\underbrace{\boldsymbol{x}_t^i}_{\text{state}}, \underbrace{\boldsymbol{u}_t^i}_{\text{control}}, \underbrace{\boldsymbol{w}_{t+1}}_{\text{noise}})\Big]$$

Constraints:

- dynamics:
  $\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_{t+1})$,
- nonanticipativity:
  $\boldsymbol{u}_t \preceq \mathcal{F}_t$,
- spatial coupling:
  $\boldsymbol{z}_t^{i+1} = g_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}^i)$.



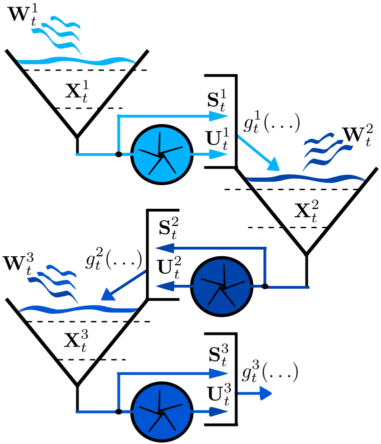
Mulstistage Stochastic Optimization: an Example

Objective function:

$$\mathbb{E}\Big[\sum_{i=1}^{N}\sum_{t=0}^{T-1} L_t^i(\underbrace{\boldsymbol{x}_t^i}_{\text{state}}, \underbrace{\boldsymbol{u}_t^i}_{\text{control}}, \underbrace{\boldsymbol{w}_{t+1}}_{\text{noise}})\Big]$$

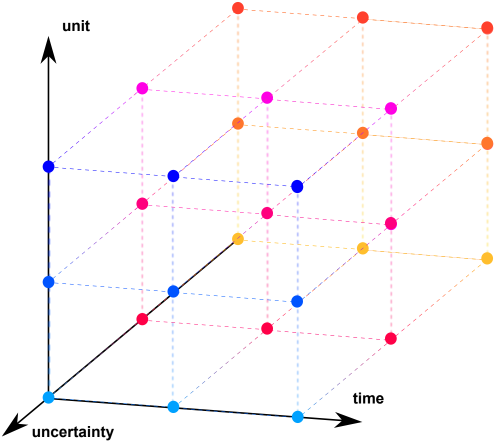
Constraints:

- dynamics:
  $\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_{t+1})$,
- nonanticipativity:
  $\boldsymbol{u}_t \preceq \mathcal{F}_t$,
- spatial coupling:
  $\boldsymbol{z}_t^{i+1} = g_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}^i)$.



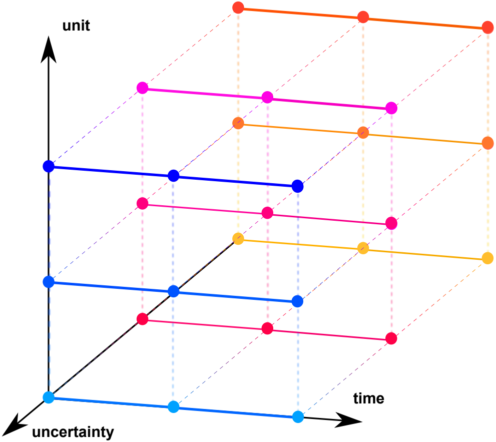
Presentation Outline

Couplings for Stochastic Problems



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

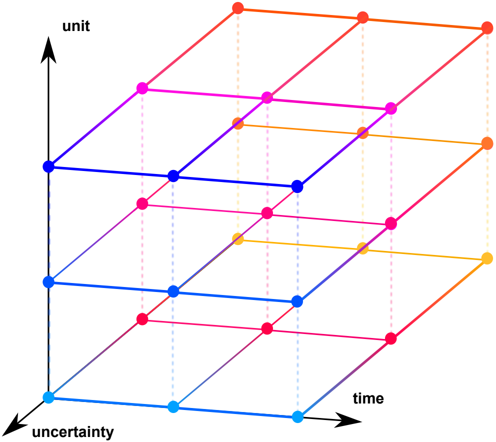
Couplings for Stochastic Problems: in Time



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_{\omega} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

Couplings for Stochastic Problems: in Uncertainty

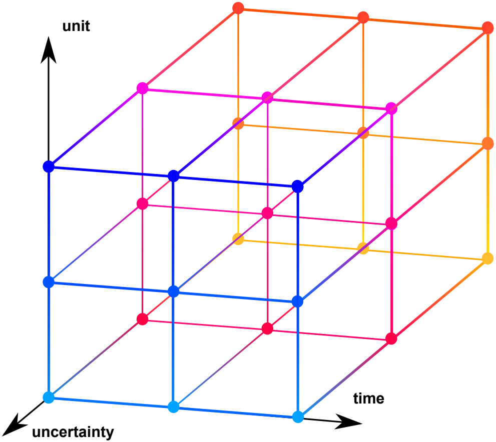


$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma\big(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t\big)$$

Couplings for Stochastic Problems: in Space



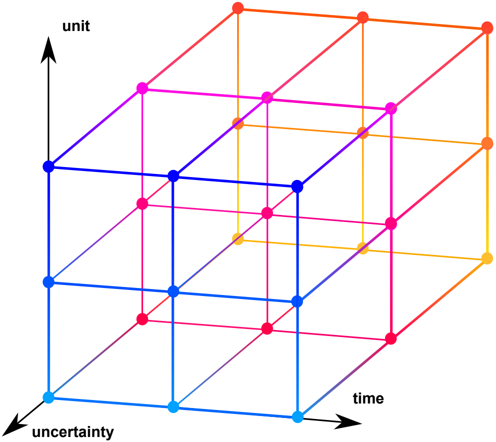
$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t)$$

$$\sum_i \Theta_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i) = 0$$

Couplings for Stochastic Problems: a Complex Problem



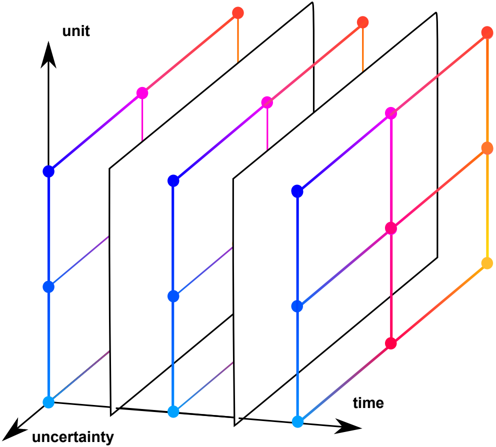
$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma\big(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t\big)$$

$$\sum_i \Theta_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i) = 0$$

Decompositions for Stochastic Problems: in Time



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

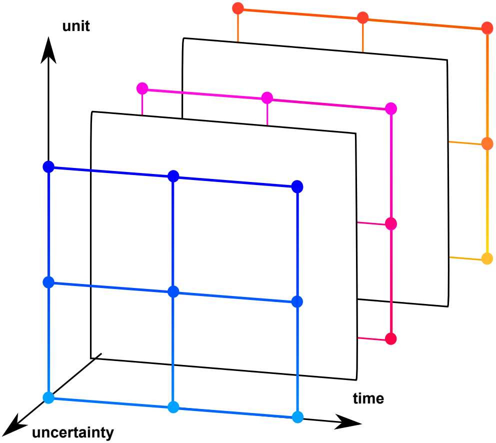
$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma\big(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t\big)$$

$$\sum_i \Theta_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i) = 0$$

Dynamic Programming

Bellman (56)

Decompositions for Stochastic Problems: in Uncertainty



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

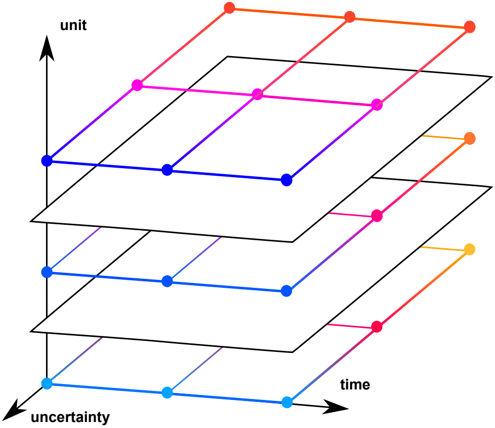
$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma\big(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t\big)$$

$$\sum_i \Theta_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i) = 0$$

Progressive Hedging

Rockafellar - Wets (91)

Decompositions for Stochastic Problems: in Space



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1})$$

$$\boldsymbol{u}_t^i \preceq \mathcal{F}_t = \sigma(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t)$$

$$\sum_i \Theta_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i) = 0$$

Dual Approximate Dynamic Programming

Presentation Outline

Stochastic Controlled Dynamic System

A stochastic controlled dynamic system is defined by its dynamic

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1})$$

and initial state

$$\boldsymbol{x}_0 = x_0$$

The variables

- $\boldsymbol{x}_t$ is the state of the system,
- $\boldsymbol{u}_t$ is the control applied to the system at time t,
- $\boldsymbol{\xi}_t$ is an exogeneous noise.

Examples

- Stock of water in a dam:
  - $\boldsymbol{x}_t$ is the amount of water in the dam at time t,
  - $\boldsymbol{u}_t$ is the amount of water turbined at time t,
  - $\boldsymbol{\xi}_t$ is the inflow of water at time t.
- Boat in the ocean:
  - $\boldsymbol{x}_t$ is the position of the boat at time t,
  - $\boldsymbol{u}_t$ is the direction and speed chosen at time t,
  - $\boldsymbol{\xi}_t$ is the wind and current at time t.
- Subway network:
  - $\boldsymbol{x}_t$ is the position and speed of each train at time t,
  - $\boldsymbol{u}_t$ is the acceleration chosen at time t,
  - $\boldsymbol{\xi}_t$ is the delay due to passengers and incident on the network at time t.

Optimization Problem

We want to solve the following optimization problem

$$\min \quad \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \tag{1a}$$

$$s.t. \quad \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_0 = x_0 \tag{1b}$$

$$\boldsymbol{u}_t \in U_t(\boldsymbol{x}_t) \tag{1c}$$

$$\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t := \sigma(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_t) \tag{1d}$$

Where

- constraint (1b) is the dynamic of the system ;
- constraint (1c) refer to the constraint on the controls;
- constraint (1d) is the information constraint : $\boldsymbol{u}_t$ is choosen knowing the realisation of the noises $\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_t$ but without knowing the realisation of the noises $\boldsymbol{\xi}_{t+1}, \dots, \boldsymbol{\xi}_{T-1}$.

Dynamic Programming Principle

Theorem

Assume that the noises $\boldsymbol{\xi}_t$ are independent and exogeneous. Then, there exists an optimal solution, called a strategy, of the form $\boldsymbol{u}_t = \pi_t(\boldsymbol{x}_t)$.

We have

$$\pi_t(x) \in \arg\min_{u \in U_t(x)} \mathbb{E}\Big[\underbrace{L_t(x, u, \boldsymbol{\xi}_{t+1})}_{\text{current cost}} + \underbrace{V_{t+1} \circ f_t(x, u, \boldsymbol{\xi}_{t+1})}_{\text{future costs}}\Big] ,$$

where (Dynamic Programming Equation)

$$\begin{cases} V_T(x) &= K(x) \\ V_t(x) &= \min_{u \in U_t(x)} \mathbb{E}\Big[L_t(x, u, \boldsymbol{\xi}_{t+1}) + V_{t+1} \circ \underbrace{f_t(x, u, \boldsymbol{\xi}_{t+1})}_{\text{"}\boldsymbol{X}_{t+1}\text{"}}\Big] \end{cases}$$

Dynamic Programming Principle

Theorem

Assume that the noises $\boldsymbol{\xi}_t$ are independent and exogeneous. Then, there exists an optimal solution, called a strategy, of the form $\boldsymbol{u}_t = \pi_t(\boldsymbol{x}_t)$.
We have

$$\pi_t(x) \in \underset{u \in U_t(x)}{\arg\min}\, \mathbb{E}\Big[\underbrace{L_t(x,u,\boldsymbol{\xi}_{t+1})}_{\text{current cost}} + \underbrace{V_{t+1} \circ f_t(x,u,\boldsymbol{\xi}_{t+1})}_{\text{future costs}}\Big] ,$$

where (Dynamic Programming Equation)

$$\begin{cases} V_T(x) &= K(x) \\ V_t(x) &= \min_{u \in U_t(x)} \mathbb{E}\Big[L_t(x,u,\boldsymbol{\xi}_{t+1}) + V_{t+1} \circ \underbrace{f_t(x,u,\boldsymbol{\xi}_{t+1})}_{\text{"}\boldsymbol{X}_{t+1}\text{"}}\Big] \end{cases}$$

Interpretation of Bellman Value

The Bellman's value function $V_{t_0}(x)$ can be interpreted as the value of the problem starting at time t_0 from the state x. More precisely we have

$$
\begin{aligned}
V_{t_0}(x) = \min \quad & \mathbb{E}\Big[\sum_{t=t_0}^{T-1} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \\
s.t. \quad & \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_{t_0} = x \\
& \boldsymbol{u}_t \in U_t(\boldsymbol{x}_t) \\
& \sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_t)
\end{aligned}
$$

Information structure

In Problem (1), constraint (1d) is the information constraint. There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure

In Problem (1), constraint (1d) is the information constraint. There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing ξ_{t+1}.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of ξ_{t+1}.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure

In Problem (1), constraint (1d) is the information constraint. There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure

In Problem (1), constraint (1d) is the information constraint.
There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure

In Problem (1), constraint (1d) is the information constraint.
There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure II

Be careful when modeling your information structure:

- Open-loop information structure might happen in practice (you have to decide on a planning and stick to it). If the problem does not require an open-loop solution then it might be largely suboptimal (imagine driving a car eyes closed...). In any case it yields an upper-bound of the problem.
- In some cases decision-hazard and hazard-decision are both approximation of the reality. Hazard-decision yield a lower value then decision-hazard.
- Anticipative structure is never an accurate modelization of the reality. However it can yield a lower-bound of your optimization problem relying on deterministic optimization and Monte-Carlo.

Information structure II

Be careful when modeling your information structure:

- Open-loop information structure might happen in practice (you have to decide on a planning and stick to it). If the problem does not require an open-loop solution then it might be largely suboptimal (imagine driving a car eyes closed...). In any case it yields an upper-bound of the problem.
- In some cases decision-hazard and hazard-decision are both approximation of the reality. Hazard-decision yield a lower value then decision-hazard.
- Anticipative structure is never an accurate modelization of the reality. However it can yield a lower-bound of your optimization problem relying on deterministic optimization and Monte-Carlo.

Information structure II

Be careful when modeling your information structure:

- Open-loop information structure might happen in practice (you have to decide on a planning and stick to it). If the problem does not require an open-loop solution then it might be largely suboptimal (imagine driving a car eyes closed...). In any case it yields an upper-bound of the problem.
- In some cases decision-hazard and hazard-decision are both approximation of the reality. Hazard-decision yield a lower value then decision-hazard.
- Anticipative structure is never an accurate modelization of the reality. However it can yield a lower-bound of your optimization problem relying on deterministic optimization and Monte-Carlo.

Non-independence of noise in DP

- The Dynamic Programming equation requires only the time-independence of noises.
- This can be relaxed if we consider an extended state.
- Consider a dynamic system driven by an equation

$$\boldsymbol{y}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\varepsilon}_{t+1})$$

where the random noise $\boldsymbol{\varepsilon}_t$ is an AR1 process :

$$\boldsymbol{\varepsilon}_t = \alpha_t \boldsymbol{\varepsilon}_{t-1} + \beta_t + \boldsymbol{\xi}_t,$$

$\{\boldsymbol{\xi}_t\}_{t\in\mathbb{Z}}$ being independent.
- Then $\boldsymbol{y}_t$ is called the physical state of the system and DP can be used with the information state $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{\varepsilon}_{t-1})$.
- Generically speaking, if the noise $\boldsymbol{\xi}_t$ is exogeneous (not affected by decisions $\boldsymbol{u}_t$), then we can always apply Dynamic Programming with the state

Presentation Outline

Dynamic Programming Algorithm - Discrete Case

Data: Problem parameters
Result: optimal control and value;
$V_T \equiv K$;
for $t : T-1 \to 0$ **do**
  for $x \in \mathbb{X}_t$ **do**

$$V_t(x) = \min_{u \in U_t(x)} \mathbb{E}\Big(L_t(x, u, \boldsymbol{W}_{t+1}) + V_t(f_t(x, u, \boldsymbol{W}_{t+1}))\Big)$$

  end
end

Algorithm 1: We iterate over the discretized state space

Dynamic Programming Algorithm - Discrete Case

Data: Problem parameters
Result: optimal control and value;
$V_T \equiv K$;
for $t : T-1 \to 0$ **do**
    for $x \in \mathbb{X}_t$ **do**
        $V_t(x) = \infty$;
        for $u \in U_t(x)$ **do**
            $v_u = \mathbb{E}\Big(L_t(x, u, \boldsymbol{W}_{t+1}) + V_t(f_t(x, u, \boldsymbol{W}_{t+1}))\Big)$ **if** $v_u < V_t(x)$ **then**
                $V_t(x) = v_u$;
                $\pi_t(x) = u$;
            end
        end
    end
end

Dynamic Programming Algorithm - Discrete Case

Data: Problem parameters
Result: optimal control and value;
$V_T \equiv K$;
for $t : T-1 \to 0$ **do**
    for $x \in \mathbb{X}_t$ **do**
        $V_t(x) = \infty$;
        for $u \in U_t(x)$ **do**
            $v_u = 0$;
            for $w \in \mathbb{W}_t$ **do**
                $v_u = v_u + \mathbb{P}\{w\}\big(L_t(x,u,w) + V_{t+1}(f_t(x,u,w))\big)$;
            end
            if $v_u < V_t(x)$ **then**
                $V_t(x) = v_u$;
                $\pi_t(x) = u$;
            end
        end

3 curses of dimensionality

Complexity = $O(T \times |\mathbb{X}_t| \times |\mathbb{U}_t| \times |\mathbb{W}_t|)$

Linear in the number of time steps, but we have 3 curses of dimensionality :

1. State. Complexity is exponential in the dimension of $\mathbb{X}_t$
   e.g. 3 independent states each taking 10 values leads to a loop over 1000 points.
2. Decision. Complexity is exponential in the dimension of $\mathbb{U}_t$.
   $\rightsquigarrow$ due to exhaustive minimization of inner problem. Can be accelerated using faster method (e.g. MILP solver).
3. Expectation. Complexity is exponential in the dimension of $\mathbb{W}_t$.
   $\rightsquigarrow$ due to expectation computation. Can be accelerated through Monte-Carlo approximation (still at least 1000 points)

In practice DP is not used for state of dimension more than 5.

3 curses of dimensionality

Complexity $= O(T \times |\mathbb{X}_t| \times |\mathbb{U}_t| \times |\mathbb{W}_t|)$

Linear in the number of time steps, but we have 3 curses of dimensionality :

1. State. Complexity is exponential in the dimension of $\mathbb{X}_t$
   e.g. 3 independent states each taking 10 values leads to a loop over 1000 points.
2. Decision. Complexity is exponential in the dimension of $\mathbb{U}_t$.
   $\rightsquigarrow$ due to exhaustive minimization of inner problem. Can be accelerated using faster method (e.g. MILP solver).
3. Expectation. Complexity is exponential in the dimension of $\mathbb{W}_t$.
   $\rightsquigarrow$ due to expectation computation. Can be accelerated through Monte-Carlo approximation (still at least 1000 points)

In practice DP is not used for state of dimension more than 5.

Illustrating the curse of dimensionality

We are in dimension 5 (not so high in the world of big data!) with 52 timesteps (common in energy management) plus 5 controls and 5 independent noises.

1. We discretize each state's dimension in 100 values: $|\mathbb{X}_t| = 100^5 = 10^{10}$
2. We discretize each control's dimension in 100 values: $|\mathbb{U}_t| = 100^5 = 10^{10}$
3. We use optimal quantization to discretize the noises' space in 10 values: $|\mathbb{W}_t| = 10$

Number of flops: $\mathcal{O}(52 \times 10^{10} \times 10^{10} \times 10) \approx \mathcal{O}(10^{23})$.
In the TOP500, the best computer computes 10^{17} flops/s.
Even with the most powerful computer, it takes at least 12 days to solve this problem.

Numerical considerations

- The DP equation holds in (almost) any case.
- The algorithm shown before compute a look-up table of control for every possible state offline. It is impossible to do if the state is (partly) continuous.
- Alternatively, we can focus on computing offline an approximation of the value function V_t and derive the optimal control online by solving a one-step problem, solved only at the current state :

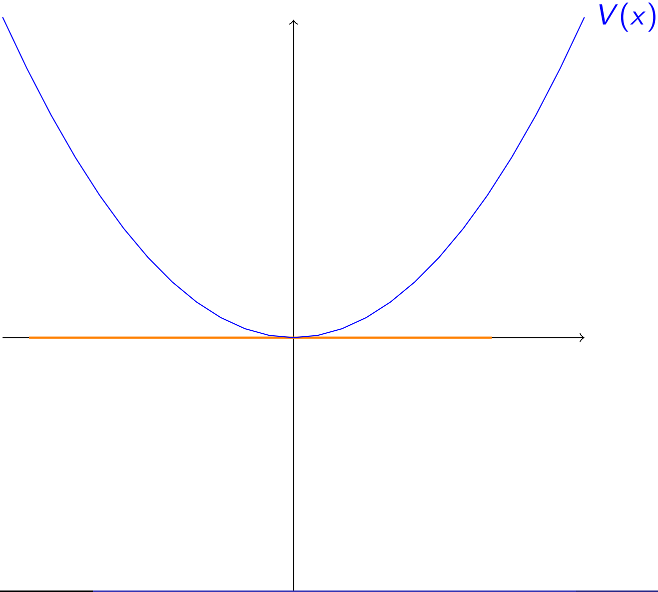
$$\pi_t(x) \in \arg\min_{u \in U_t(x)} \mathbb{E}\Big[L_t(x, u, \xi_{t+1}) + V_{t+1} \circ f_t(x, u, \xi_{t+1})\Big]$$

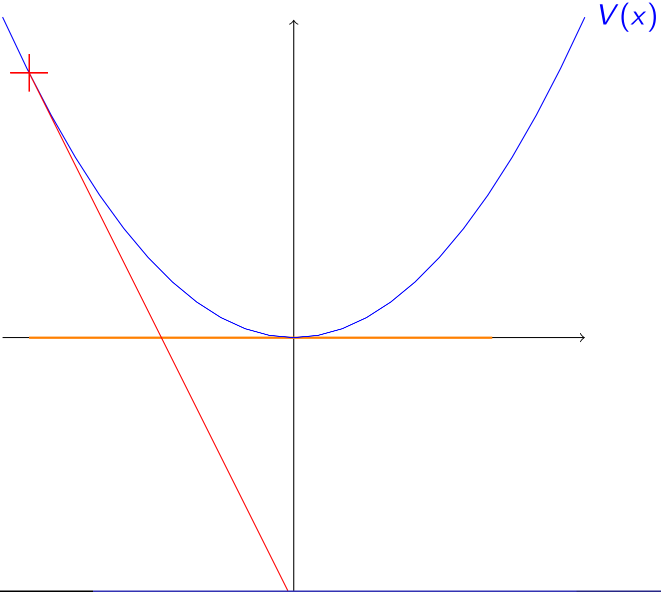
- The field of Approximate DP gives methods for computing those approximate value function.
- The simpler one consisting in discretizing the state, and then interpolating the value function.

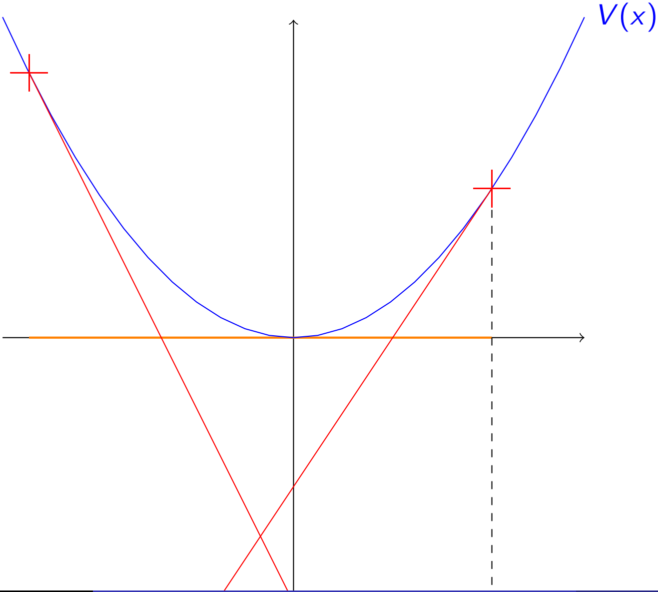
Presentation Outline

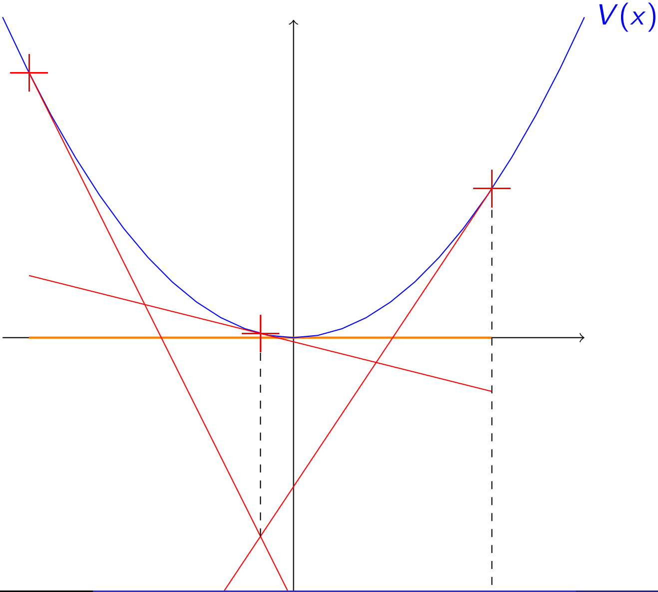
Dynamic Programming : continuous and convex case

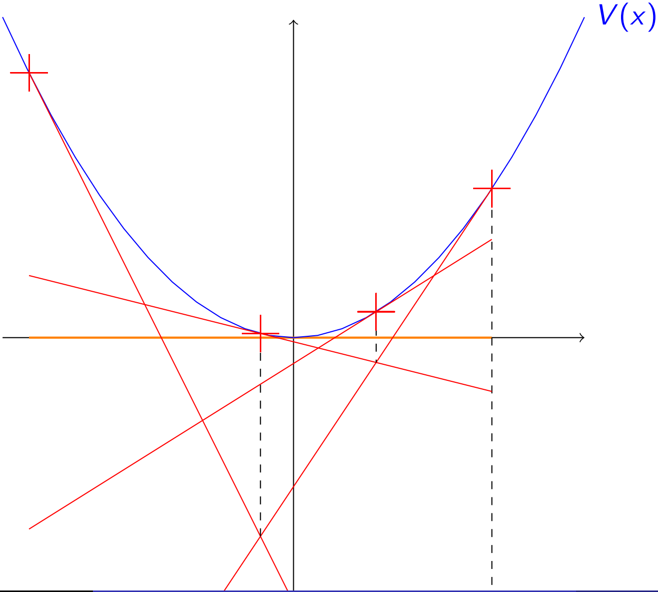
- If the problem has continuous states and control the classical approach consists in discretizing.
- With further assumption on the problem (convexity, linearity) we can look at a dual approach:
  - Instead of discretizing and interpolating the Bellman function we choose to do a polyhedral approximation.
  - Indeed we choose a "smart state" in which we compute the value of the function and its marginal value (tangeant).
  - Knowing that the problem is convex and using the power of linear solver we can efficiently approximate the Bellman function.
- This approach is known as SDDP in the electricity community and widely used in practice.


V(x)


V(x)


V(x)

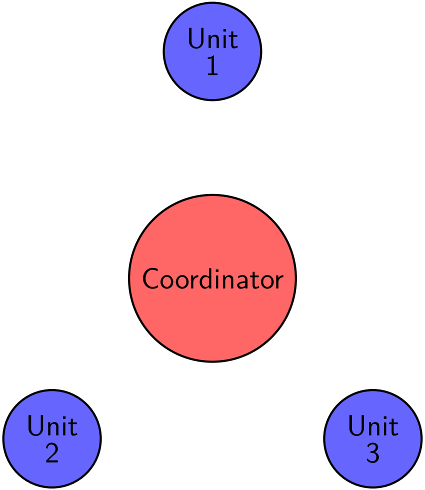

$V(x)$


$V(x)$

Presentation Outline

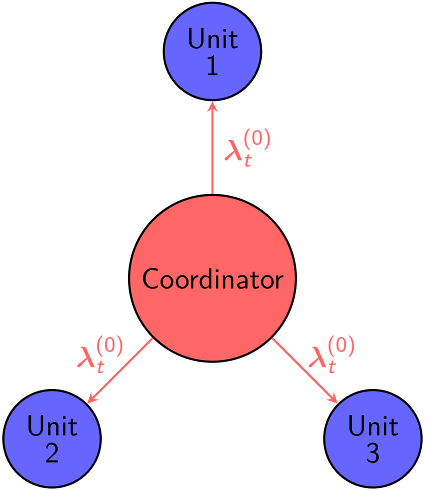
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



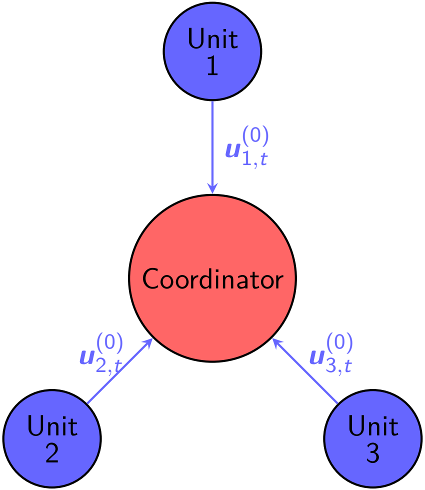
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



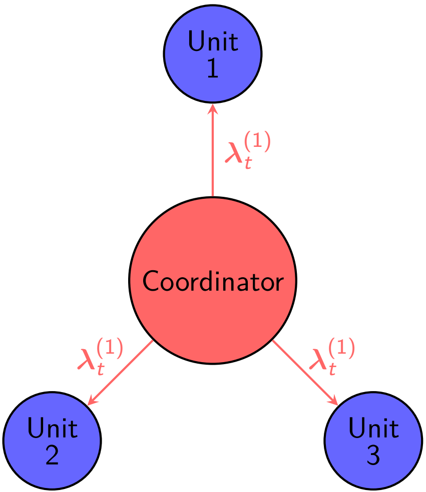
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



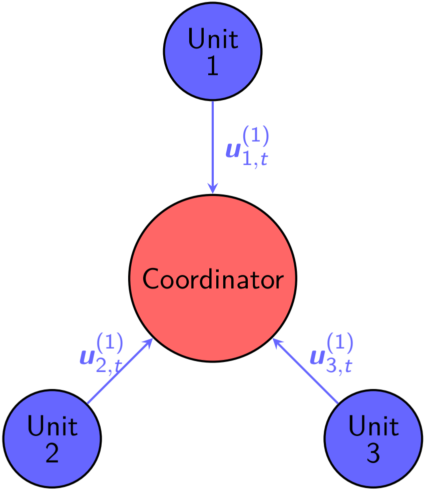
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



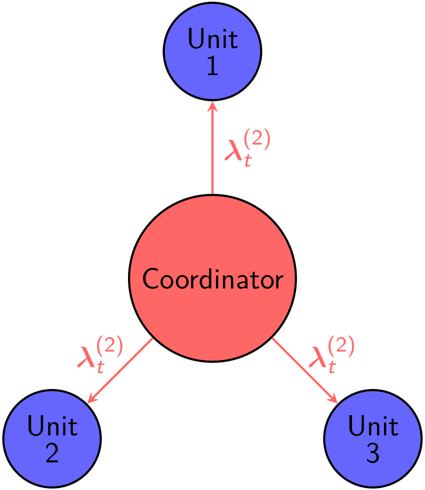
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



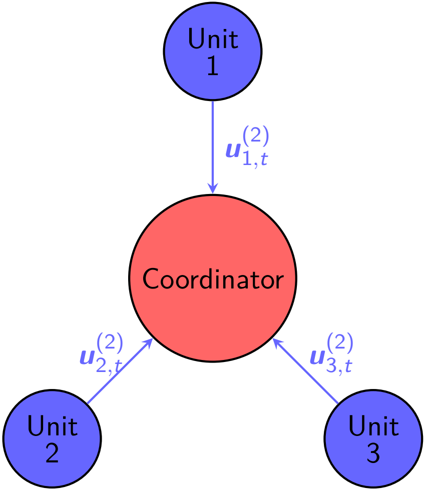
Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...

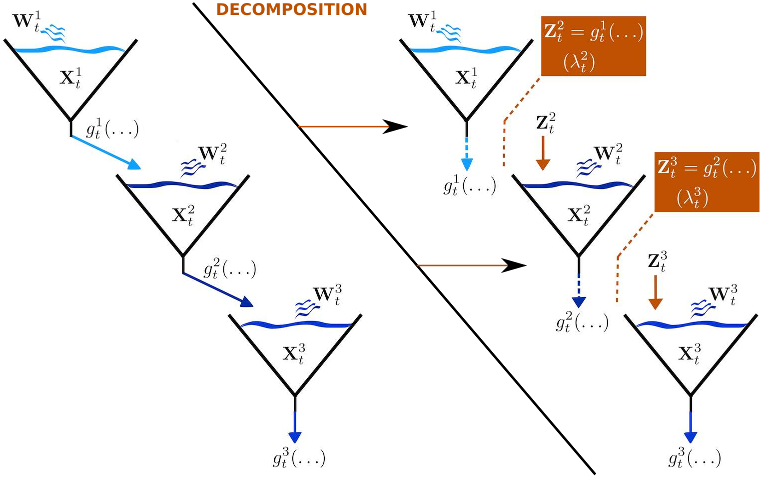


Intuition of Spatial Decomposition

- Satisfy a demand (over T time step) with N units of production at minimal cost.
- Price decomposition:
  - the coordinator sets a sequence of price λ_t,
  - the units send their production planning $\boldsymbol{u}_t^{(i)}$,
  - the coordinator compares total production and demand and updates the price,
  - and so on...



Application to dam management



Presentation Outline

Primal Problem

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=1}^{N} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\forall\, i, \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\forall\, i, \quad \boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

$$\sum_{i=1}^{N} \theta_t^i(\boldsymbol{u}_t^i) = 0$$

Solvable by DP with state $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$

Primal Problem

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=1}^{N} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\forall\, i, \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\forall\, i, \quad \boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

$$\sum_{i=1}^{N} \theta_t^i(\boldsymbol{u}_t^i) = 0 \qquad \rightsquigarrow \boldsymbol{\lambda}_t \quad \text{multiplier}$$

Solvable by DP with state $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$

Primal Problem with Dualized Constraint

$$\min_{\boldsymbol{x},\boldsymbol{u}} \max_{\boldsymbol{\lambda}} \sum_{i=1}^{N} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + \langle \boldsymbol{\lambda}_t, \theta_t^i(\boldsymbol{u}_t^i) \rangle + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\forall\, i, \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\forall\, i, \quad \boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

Coupling constraint dualized $\Longrightarrow$ all constraints are unit by unit

Dual Problem

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=1}^{N} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + \langle \boldsymbol{\lambda}_t, \theta_t^i(\boldsymbol{u}_t^i)\rangle + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\forall\, i, \quad \boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\forall\, i, \quad \boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

Exchange operator min and max to obtain a new problem

Decomposed Dual Problem

$$\max_{\boldsymbol{\lambda}} \sum_{i=1}^{N} \min_{\boldsymbol{x}^i, \boldsymbol{u}^i} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + \langle \boldsymbol{\lambda}_t, \theta_t^i(\boldsymbol{u}_t^i)\rangle + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

For a given λ, minimum of sum is sum of minima

Inner Minimization Problem

$$\min_{\boldsymbol{x}^i,\boldsymbol{u}^i} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + \langle \boldsymbol{\lambda}_t, \theta_t^i(\boldsymbol{u}_t^i)\rangle + K^i(\boldsymbol{x}_T^i)\Big]$$
$$\boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$
$$\boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

We have N smaller subproblems. Can they be solved by DP ?

Inner Minimization Problem

$$\min_{\boldsymbol{x}^i,\boldsymbol{u}^i} \mathbb{E}\Big[\sum_{t=0}^{T} L_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}) + \langle \boldsymbol{\lambda}_t, \theta_t^i(\boldsymbol{u}_t^i)\rangle + K^i(\boldsymbol{x}_T^i)\Big]$$

$$\boldsymbol{x}_{t+1}^i = f_t^i(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, \boldsymbol{w}_{t+1}), \quad \boldsymbol{x}_0^i = x_0^i,$$

$$\boldsymbol{u}_t^i \in \mathcal{U}_{t,i}^{ad}, \quad \boldsymbol{u}_t^i \preceq \mathcal{F}_t,$$

No : $\boldsymbol{\lambda}$ is a time-dependent noise $\rightsquigarrow$ state $\big(\boldsymbol{w}_1, \dots, \boldsymbol{w}_t\big)$

Presentation Outline

Dual approximation as constraint relaxation

The original problem is (abstract form)

$$\min_{\boldsymbol{u}\in\mathcal{U}} \quad J(\boldsymbol{u})$$
$$s.t. \quad \Theta(\boldsymbol{u}) = 0$$

written as

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\big[\langle \boldsymbol{\lambda}, \Theta(\boldsymbol{u})\rangle\big]$$

Subsituting $\boldsymbol{\lambda}$ by $\mathbb{E}(\boldsymbol{\lambda} | \boldsymbol{y})$ gives

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\Big[\big\langle \mathbb{E}(\boldsymbol{\lambda} | \boldsymbol{y}), \Theta(\boldsymbol{u})\big\rangle\Big]$$

Dual approximation as constraint relaxation

The original problem is (abstract form)

$$\min_{\boldsymbol{u}\in\mathcal{U}} \quad J(\boldsymbol{u})$$
$$s.t. \quad \Theta(\boldsymbol{u}) = 0$$

written as

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\big[\langle \boldsymbol{\lambda}, \Theta(\boldsymbol{u})\rangle\big]$$

Subsituting $\boldsymbol{\lambda}$ by $\mathbb{E}\big(\boldsymbol{\lambda}\big|\boldsymbol{y}\big)$ gives

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\Big[\big\langle \boldsymbol{\lambda}, \mathbb{E}\big(\Theta(\boldsymbol{u})\big|\boldsymbol{y}\big)\big\rangle\Big]$$

Dual approximation as constraint relaxation

The original problem is (abstract form)

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathcal{U}} \quad & J(\boldsymbol{u}) \\ s.t. \quad & \Theta(\boldsymbol{u}) = 0 \end{aligned}$$

written as

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\big[\langle \boldsymbol{\lambda}, \Theta(\boldsymbol{u})\rangle\big]$$

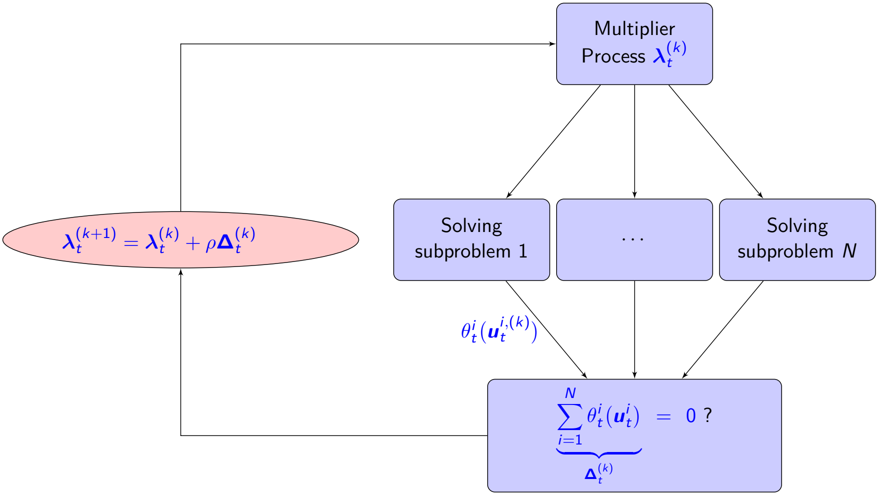
Subsituting $\boldsymbol{\lambda}$ by $\mathbb{E}\big(\boldsymbol{\lambda}\big|\boldsymbol{y}\big)$ gives

$$\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{\lambda}} \quad J(\boldsymbol{u}) + \mathbb{E}\Big[\Big\langle \boldsymbol{\lambda}, \mathbb{E}\big(\Theta(\boldsymbol{u})\big|\boldsymbol{y}\big)\Big\rangle\Big]$$

equivalent to

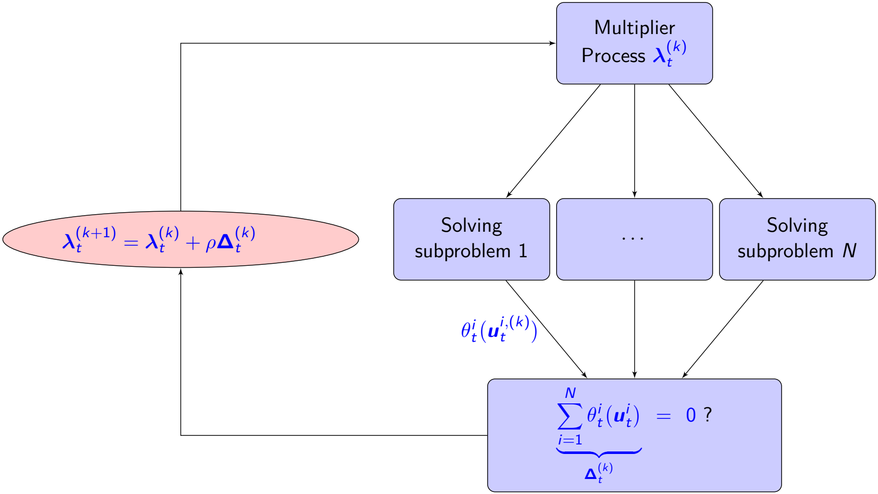
$$\begin{aligned} \min_{\boldsymbol{u}\in\mathcal{U}} \quad & J(\boldsymbol{u}) \\ s.t. \quad & \mathbb{E}\big(\Theta(\boldsymbol{u})\big|\boldsymbol{y}\big) = 0 \end{aligned}$$

Stochastic spatial decomposition scheme



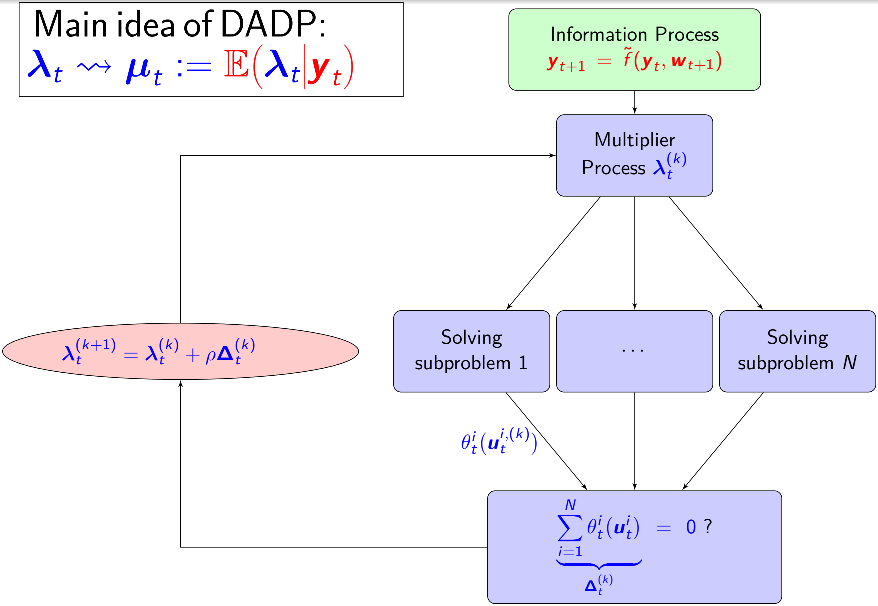
Main idea of DADP:

$$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$$



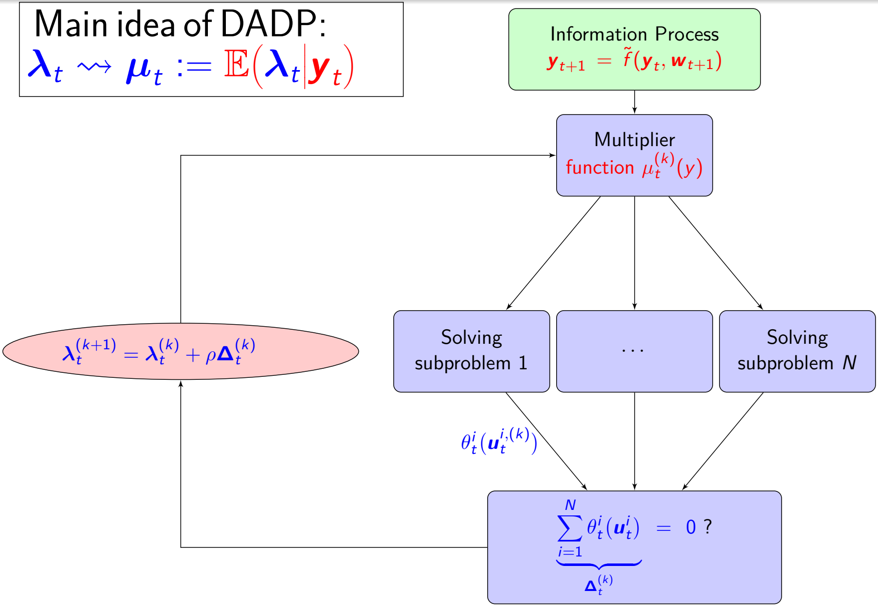
Main idea of DADP:

$$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$$

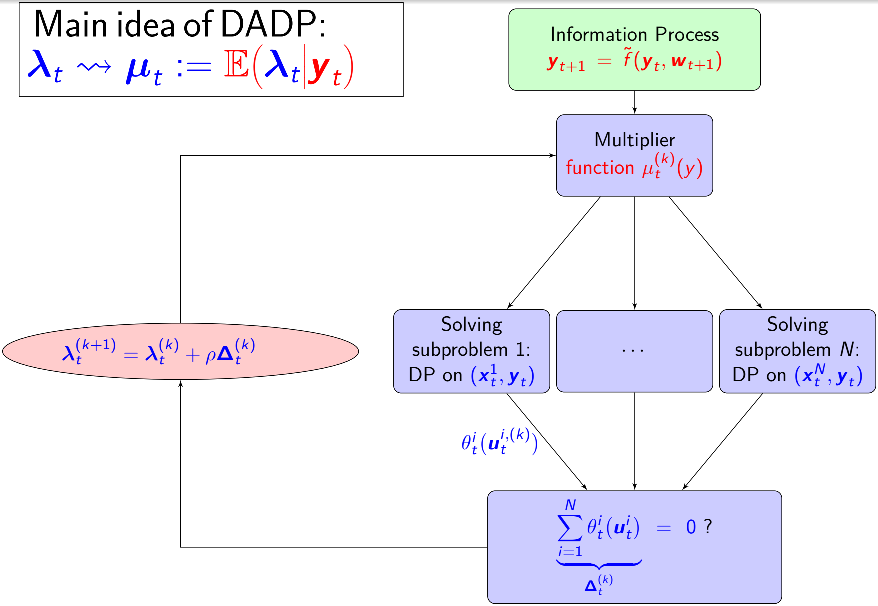


Main idea of DADP:

$$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$$

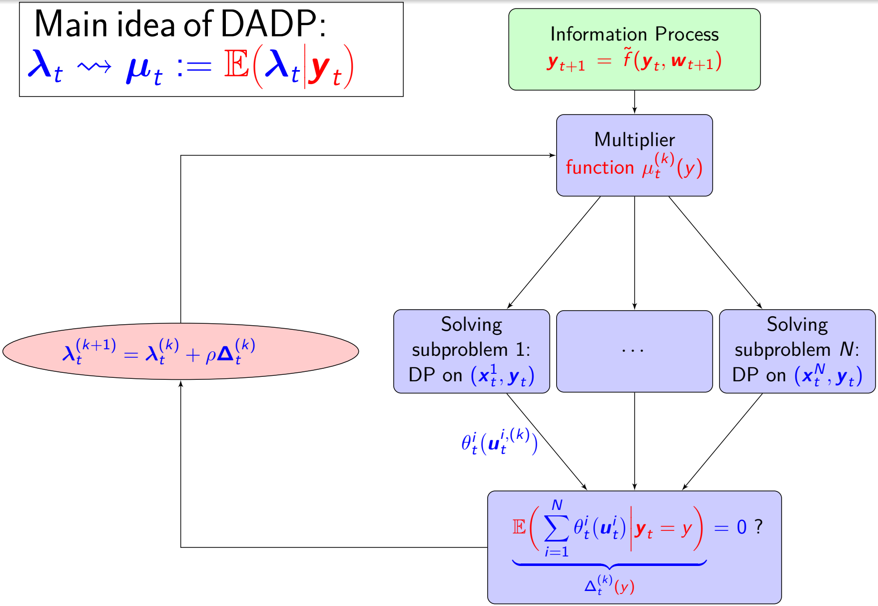


Main idea of DADP:
$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$



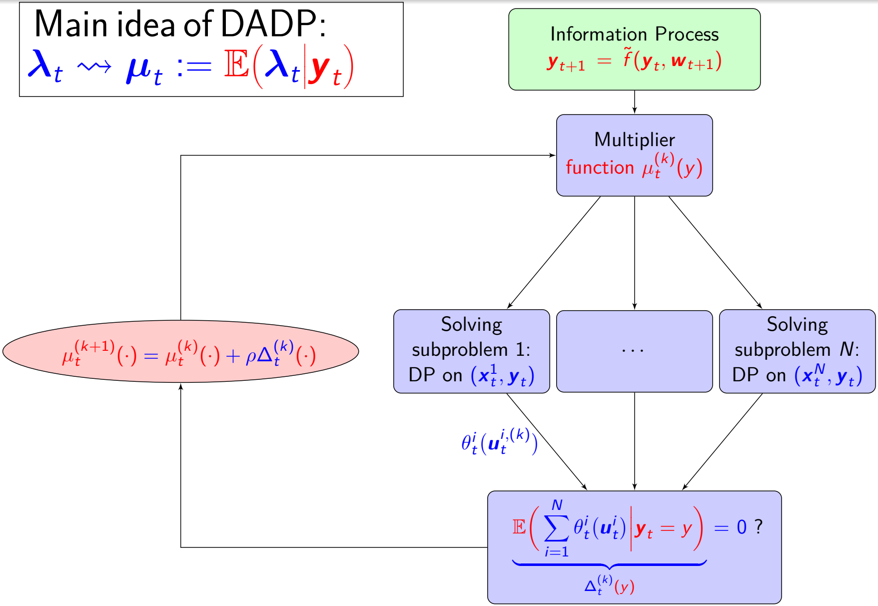
Main idea of DADP:

$$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$$

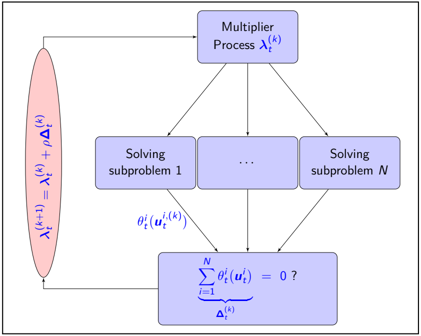


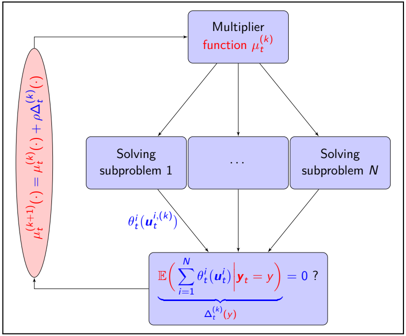
Main idea of DADP:

$$\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big)$$



Main idea of DADP: $\boldsymbol{\lambda}_t \rightsquigarrow \boldsymbol{\mu}_t := \mathbb{E}\left(\boldsymbol{\lambda}_t \middle| \boldsymbol{y}_t\right)$





Main problems:

- Subproblems not easily solvable by DP
- $\boldsymbol{\lambda}^{(k)}$ live in a huge space

Advantages:

- Subproblems solvable by DP with state $\left(\boldsymbol{x}_t^i, \boldsymbol{y}_t\right)$
- $\mu^{(k)}$ live in a smaller space

Three Interpretations of DADP

- DADP as an approximation of the optimal multiplier

$$\boldsymbol{\lambda}_t \quad \rightsquigarrow \quad \mathbb{E}\big(\boldsymbol{\lambda}_t \big| \boldsymbol{y}_t\big) \ .$$

- DADP as a decision-rule approach in the dual

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{u}} L(\boldsymbol{\lambda}, \boldsymbol{u}) \quad \rightsquigarrow \quad \max_{\boldsymbol{\lambda}_t \preceq \boldsymbol{y}_t} \min_{\boldsymbol{u}} L(\boldsymbol{\lambda}, \boldsymbol{u}) \ .$$

- DADP as a constraint relaxation in the primal

$$\sum_{i=1}^{n} \theta_t^i(\boldsymbol{u}_t^i) = 0 \quad \rightsquigarrow \quad \mathbb{E}\bigg(\sum_{i=1}^{n} \theta_t^i(\boldsymbol{u}_t^i) \bigg| \boldsymbol{y}_t\bigg) = 0 \ .$$

Presentation Outline

Conclusion

- Large multistage stochastic program are numerically difficult.
- To tackle such problems one can use decomposition methods.
- If the number of stages is small enough, decomposition per scenario (like Progressive-Hedging) is numerically efficient, and use special deterministic methods.
- If the noises are time-independent Dynamic Programming equations are available.
  - If the state dimension is small enough direct discretized dynamic programming is available.
  - If dynamics is linear and cost are convex SDDP approach allow for larger states
  - Finally we can also spatially decompose problems, and with an approximation recover Dynamic Programming equations for the subproblems.