

# COURS RTE 2022

## OPTIMISATION STOCHASTIQUE



### Méthodes de décomposition spatiale II

“Dual Approximate Dynamic Programming”  
et méthodes connexes

P. Carpentier, J.-Ph. Chancelier, M. De Lara , V. Leclère



Vendredi 22 avril 2022



## Ultimate goal of the lecture

How to obtain “good” **strategies** for a **large scale** stochastic optimal control problem, for example a problem corresponding to the optimal management over a given time horizon of a system involving a large amount of dynamical production units.

- In order to obtain **decision strategies** (closed-loop controls), we have to use **Dynamic Programming** or related methods.
  - **Assumption**: Markovian case,
  - **Difficulty**: **curse of dimensionality**.
- In order to take into account the size of the system, we have to use **decomposition/coordination** techniques.
  - **Assumption**: convexity,
  - **Difficulty**: **information pattern** of the problem.

**Mixture of spatial and temporal decompositions**

# Lecture outline

- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

- 1 **Mixing spatial and temporal decompositions**
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 **Application to dams management problems**
  - Hydro valley modeling
  - Numerical experiments
- 3 **Application to microgrids management problems**
  - Urban microgrid modeling
  - Numerical experiments

# Optimization problem

We recall the SOC problem under consideration:

$$\min_{\mathbf{U}, \mathbf{X}} \mathbb{E} \left( \sum_{i=1}^N \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}) + K^i(\mathbf{X}_T^i) \right) \right), \quad (\mathcal{P})$$

subject to **dynamics** constraints:

$$\begin{aligned} \mathbf{X}_0^i &= f_{-1}^i(\mathbf{W}_0), \\ \mathbf{X}_{t+1}^i &= f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}), \end{aligned}$$

to **measurability** constraints:

$$\sigma(\mathbf{U}_t^i) \subset \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t), \quad \text{Decision-Hazard setting}$$

and to **instantaneous coupling** constraints

$$\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) = 0. \quad \text{Feasible constraints}$$

# Assumptions

## Assumption 1 (White noise)

Noises  $\mathbf{W}_0, \dots, \mathbf{W}_T$  are independent over time.

We have also assumed **full noise observation**:

$$\sigma(\mathbf{U}_t^i) \subset \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t).$$

As a consequence of these assumptions, there is **no optimality loss** to seek the control  $\mathbf{U}_t^i$  as a function of the state at time  $t$  rather than a function of the past noises:

$$\sigma(\mathbf{U}_t^i) \subset \sigma(\mathbf{x}_t^1, \dots, \mathbf{x}_t^N).$$

We are in the **Markovian** case, and **Dynamic Programming** applies.

*But DP faces the **curse of dimensionality** when  $N$  is large. . .*

# Lagrangian formulation

We dualize the **coupling constraints** and obtain the **Lagrangian**:

$$\mathcal{L}(\mathbf{X}, \mathbf{U}, \boldsymbol{\Lambda}) = \mathbb{E} \left( \sum_{i=1}^N \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}) + K^i(\mathbf{X}_T^i) + \sum_{t=0}^{T-1} \boldsymbol{\Lambda}_t \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) \right) \right),$$

where the  $\boldsymbol{\Lambda}_t$ 's are  $\sigma(\mathbf{W}_0, \dots, \mathbf{W}_t)$ -measurable **random variables**.

We assume that a **saddle point** of  $\mathcal{L}$  exists,<sup>1</sup> so that

$$\min_{\mathbf{U}, \mathbf{X}} \max_{\boldsymbol{\Lambda}} \mathcal{L}(\mathbf{X}, \mathbf{U}, \boldsymbol{\Lambda}) = \max_{\boldsymbol{\Lambda}} \min_{\mathbf{U}, \mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{U}, \boldsymbol{\Lambda}).$$

<sup>1</sup>Such an assumption is **highly non-trivial** for the considered problem. . .

# Uzawa algorithm

At **iteration**  $k$  of the algorithm,

- 1 **Solve** subproblem  $i$ ,  $i = 1, \dots, N$ , with  $\Lambda^{(k)}$  fixed:

$$\min_{\mathbf{u}^i, \mathbf{x}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}) + \Lambda_t^{(k)} \cdot \Theta_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i) \right) + K^i(\mathbf{x}_T^i) \right),$$

subject to

$$\mathbf{x}_{t+1}^i = f_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}),$$

$$\sigma(\mathbf{u}_t^i) \subset \sigma(\mathbf{w}_0, \dots, \mathbf{w}_t),$$

whose solution is denoted  $(\mathbf{u}^{i,(k+1)}, \mathbf{x}^{i,(k+1)})$ .

- 2 **Update** the multipliers  $\Lambda_t$ :

$$\Lambda_t^{(k+1)} = \Lambda_t^{(k)} + \rho_t \left( \sum_{i=1}^N \Theta_t^i(\mathbf{x}_t^{i,(k+1)}, \mathbf{u}_t^{i,(k+1)}) \right).$$



- 1 **Mixing spatial and temporal decompositions**
  - Problem formulation and price decomposition
  - **Dual approximate dynamic programming (DADP)**
  - Upper and lower bounds for large scale SOC problems
- 2 **Application to dams management problems**
  - Hydro valley modeling
  - Numerical experiments
- 3 **Application to microgrids management problems**
  - Urban microgrid modeling
  - Numerical experiments

## Main idea of the approximation

As already pointed out,  $\Lambda_t^{(k)}$  depends on  $(W_0, \dots, W_t)$ , so that solving a subproblem is as complex as solving the initial problem.

In order to overcome the difficulty, we **choose** at each time  $t$  and for each  $i$  a random variable  $Y_t^i$  which is measurable w.r.t. the past noises  $(W_0, \dots, W_t)$ . We call  $Y^i = (Y_0^i, \dots, Y_{T-1}^i)$  the **information process** for subsystem  $i$ .

The **core idea** of DADP is to replace the multiplier  $\Lambda_t^{(k)}$  by its **conditional expectation** w.r.t.  $Y_t^i$ , that is,  $\mathbb{E}(\Lambda_t^{(k)} | Y_t^i)$ . From an intuitive point of view, this leads to a good approximation if

$Y_t^i$  is **(highly) correlated** to the random variable  $\Lambda_t$ .

*Note that we require that the information process is not influenced by controls.*

## Subproblem approximation

Following this idea, we **replace** subproblem  $i$  in Uzawa algorithm by:

$$\min_{\mathbf{u}^i, \mathbf{x}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}) + \mathbb{E}(\Lambda_t^{(k)} \mid \mathbf{Y}_t^i) \cdot \Theta_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i) \right) + K^i(\mathbf{x}_T^i) \right),$$

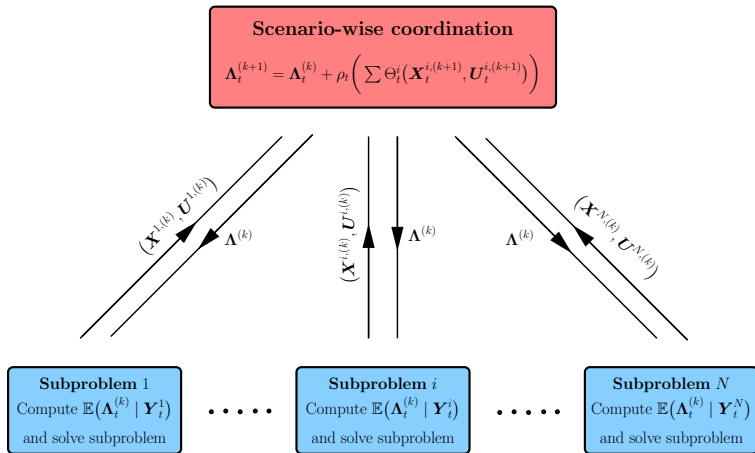
subject to

$$\begin{aligned} \mathbf{x}_{t+1}^i &= f_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}), \\ \sigma(\mathbf{U}_t^i) &\subset \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t). \end{aligned}$$

The **conditional expectation**  $\mathbb{E}(\Lambda_t^{(k)} \mid \mathbf{Y}_t^i)$  corresponds to a given function of the variable  $\mathbf{Y}_t^i$ , so that subproblem  $i$  now involves 2 **exogenous** random processes, that is,  $\mathbf{W}$  and  $\mathbf{Y}^i$ .

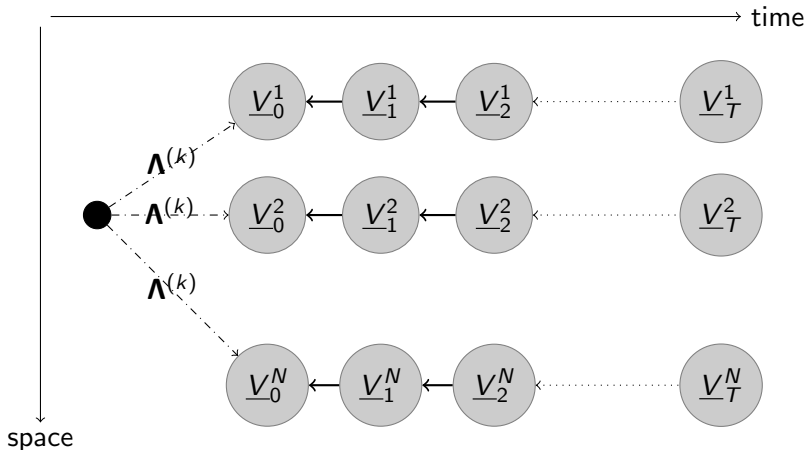
If  $\mathbf{Y}^i$  is a **short memory** process, DP applies effectively.

# DADP as a spatial decomposition (price) algorithm



Each subproblem is solved by DP: **temporal decomposition**.

# Mix of spatial and temporal decompositions in DADP



# Possible choices for the information process

- ① **Perfect memory:**  $\mathbf{Y}_t^i = (\mathbf{W}_0, \dots, \mathbf{W}_t)$ .
  - $\mathbb{E}(\boldsymbol{\Lambda}_t^{(k)} \mid \mathbf{Y}_t^i) = \boldsymbol{\Lambda}_t^{(k)}$ : **no approximation!**
  - The **state size** of the subproblem **increases** with time...
- ② **Minimal information:**  $\mathbf{Y}_t^i \equiv \text{cste}$ .
  - $\boldsymbol{\Lambda}_t^{(k)}$  is approximated by its **expectation**  $\mathbb{E}(\boldsymbol{\Lambda}_t^{(k)})$ .
  - The information variable does not deliver any information...
- ③ **Static information:**  $\mathbf{Y}_t^i = h_t^i(\mathbf{W}_t)$ .
  - Such a choice is guided by the intuition that a part of  $\mathbf{W}_t$  mostly “explains” the optimal multiplier.
- ④ **Dynamic information:**  $\mathbf{Y}_{t+1}^i = h_t^i(\mathbf{Y}_t^i, \mathbf{W}_{t+1})$ .
  - In the Dynamic Programming equation, the state vector is **augmented** by embedding  $\mathbf{Y}_t^i$ , that is, the **necessary memory** to compute the **information variable** at the next time step.

# Dynamic Programming equation

In the last case (**dynamic information**), the DP equation writes:

$$\underline{V}_T^i(x, y) = K^i(x),$$

$$\underline{V}_t^i(x, y) = \min_u \mathbb{E} \left( \left( L_t^i(x, u, \mathbf{W}_{t+1}) \right. \right. \\ \left. \left. + \mathbb{E}(\boldsymbol{\Lambda}_t^{(k)} \mid \mathbf{Y}_t^i = y) \cdot \Theta_t^i(x, u) \right. \right. \\ \left. \left. + \underline{V}_{t+1}^i(\mathbf{X}_{t+1}^i, \mathbf{Y}_{t+1}^i) \right) \right),$$

subject to the dynamics:

$$\mathbf{X}_{t+1}^i = f_t^i(x, u, \mathbf{W}_{t+1}),$$

$$\mathbf{Y}_{t+1}^i = h_t^i(y, \mathbf{W}_{t+1}).$$

## About the coordination

The task of coordination is performed in a **scenario-wise** manner.

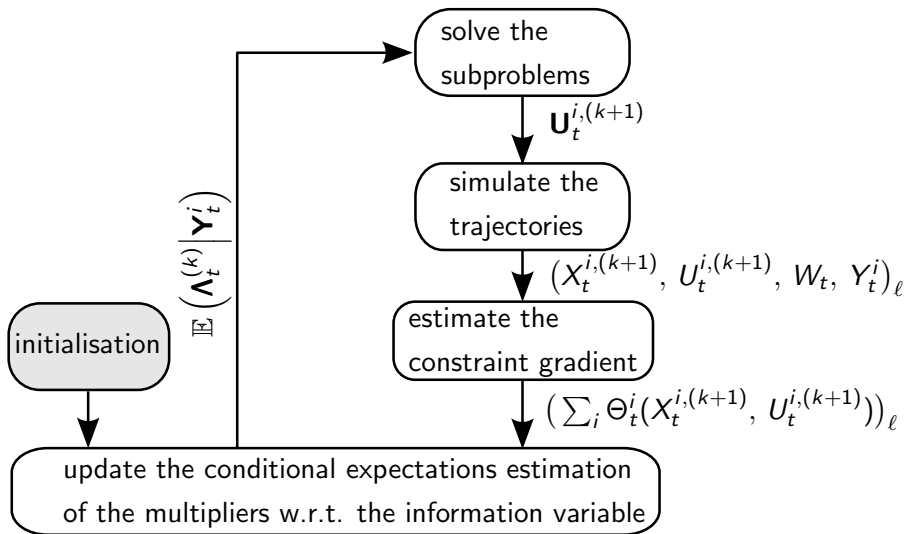
- A set of **noise scenarios** is drawn once for all. **Trajectories** of the information process  $\mathbf{Y}^i$  are **simulated** along the scenarios.
- At iteration  $k$ , the **optimal trajectories** of the state process  $\mathbf{X}^{i,(k+1)}$  and of the control process  $\mathbf{U}^{i,(k+1)}$  are **simulated** along the noise scenarios, for all subsystems.
- The **dual multipliers** are **updated** along the noise scenarios according to the formula:

$$\boldsymbol{\Lambda}_t^{(k+1)} = \boldsymbol{\Lambda}_t^{(k)} + \rho_t \left( \sum_{i=1}^N \Theta_t^i(\mathbf{x}_t^{i,(k+1)}, \mathbf{u}_t^{i,(k+1)}) \right).$$

- The **conditional expectations**  $\mathbb{E}(\boldsymbol{\Lambda}_t^{(k+1)} \mid \mathbf{Y}_t^i)$  are obtained by **regression** of the trajectories of  $\boldsymbol{\Lambda}_t^{(k+1)}$  on those of  $\mathbf{Y}_t^i$ .



# DADP flowchart (based on scenarios)



# Interpretation of DADP

The **approximation** made on the **dual process** allows to obtain a tractable way for solving the subsystems. It also provides an interpretation of what has been made in terms of **constraints**.

From now on, assume that the information variable  $\mathbf{Y}_t$  is **the same for all subsystems**. We consider a **new** problem derived from  $(\mathcal{P})$ :

$$\min_{\mathbf{U}, \mathbf{X}} \mathbb{E} \left( \sum_{i=1}^N \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}) + K^i(\mathbf{x}_T^i) \right) \right), \quad (\mathcal{P}_r)$$

subject to the **modified coupling** constraints:

$$\mathbb{E} \left( \sum_{i=1}^N \Theta_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i) \mid \mathbf{Y}_t \right) = 0.$$

# Interpretation of DADP



## Proposition 1

Assume that the Lagrangian associated with Problem  $(\mathcal{P}_r)$  has a *saddle point*. Then the *DADP* algorithm can be interpreted as the *Uzawa* algorithm *applied* to Problem  $(\mathcal{P}_r)$ .

**Proof.** Since the term  $\mathbb{E}(\mathbb{E}(\boldsymbol{\Lambda}_t^{(k)} \mid \mathbf{Y}_t) \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i))$  which appears in the cost function of subproblem  $i$  in DADP can be written:

$$\mathbb{E}(\mathbb{E}(\boldsymbol{\Lambda}_t^{(k)} \mid \mathbf{Y}_t) \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i)) = \mathbb{E}(\boldsymbol{\Lambda}_t^{(k)} \cdot \mathbb{E}(\Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) \mid \mathbf{Y}_t)) ,$$

the global constraint *really* handled by DADP is:

$$\mathbb{E}\left(\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) \mid \mathbf{Y}_t\right) = 0 . \quad \square$$

DADP thus consists in replacing an *almost-sure* constraint by its *conditional expectation* w.r.t. the *information variable*  $\mathbf{Y}_t$ .

# Summary

To summarize, DADP leads to solve the approximated problem:

$$\min_{\mathbf{U}, \mathbf{X}} \mathbb{E} \left( \sum_{i=1}^N \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) + K^i(\mathbf{X}_T^i) \right) \right) \quad \text{s.t.} \quad \mathbb{E} \left( \sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) \mid \mathbf{Y}_t \right) = 0,$$

whereas the true problem is:

$$\min_{\mathbf{U}, \mathbf{X}} \mathbb{E} \left( \sum_{i=1}^N \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) + K^i(\mathbf{X}_T^i) \right) \right) \quad \text{s.t.} \quad \sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i) = 0.$$

The conditional expectation constraint handled by DADP is a **relaxed version** of the almost sure constraint of the true problem.

**An immediate consequence is that the DADP optimal value is an **exact lower bound** of the true problem optimal value.**

## Some questions

### ★ What is the suitable theoretical framework of the algorithm?

The convergence of Uzawa's algorithm is granted provided that:

- the problem is posed in Hilbert spaces,
- and it exists a saddle point.

It thus seems natural to place ourselves in a Hilbert space. But it is known (works by Rockafellar and Wets) that a saddle point doesn't exist in Hilbert spaces for such problems. . .

### ★ Does the approximate solution converge to the true solution?

Epiconvergence results are available w.r.t. the information delivered by  $Y_t$ . But epiconvergence raises difficult technical problems when addressed to stochastic optimization problems.

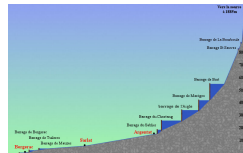
### ★ How to obtain a feasible solution from the approximate solution?

Use an appropriate heuristic (to be explained later on)!

## Progress status

- First, we have obtained a **lower** bound for a global optimization problem with coupling constraints thanks to a price decomposition and coordination scheme (**spatial decomposition**).
- Second, we have computed the lower bound by dynamic programming (**temporal decomposition**)
- Using the price Bellman value functions, we have an heuristic procedure to devise an **online policy** for the **global** problem

We will apply this decomposition scheme to **dams management problems**



We now investigate **two decomposition schemes** (price and resource) to obtain **lower and upper bounds** for a global optimization problem.

- 1 **Mixing spatial and temporal decompositions**
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 **Application to dams management problems**
  - Hydro valley modeling
  - Numerical experiments
- 3 **Application to microgrids management problems**
  - Urban microgrid modeling
  - Numerical experiments

# An abstract optimization problem

We consider the following **optimization problem**

$$V_0^\# = \min_{u^1 \in \mathcal{U}_{\text{ad}}^1, \dots, u^N \in \mathcal{U}_{\text{ad}}^N} \sum_{i=1}^N J^i(u^i)$$

$$\text{s.t. } \underbrace{(\Theta^1(u^1), \dots, \Theta^N(u^N))}_{\text{coupling constraint}} \in -S$$

with

- $u^i \in \mathcal{U}^i$  be a local decision variable
- $J^i : \mathcal{U}^i \rightarrow \mathbb{R}$ ,  $i \in \llbracket 1, N \rrbracket$  be a local objective
- $\mathcal{U}_{\text{ad}}^i$  be a subset of  $\mathcal{U}^i$
- $\Theta^i : \mathcal{U}^i \rightarrow \mathcal{C}^i$  be a local constraint mapping
- $S$  be a subset of  $\mathcal{C} = \mathcal{C}^1 \times \dots \times \mathcal{C}^N$

We denote by  $S^*$  the **dual cone** of  $S$

$$S^* = \{\lambda \in \mathcal{C}^* \mid \langle \lambda, r \rangle \geq 0 \quad \forall r \in S\}$$



# Price and resource value functions

For each  $i \in \llbracket 1, N \rrbracket$ ,

- for any **price**  $\lambda^i \in (\mathcal{C}^i)^*$ , we define the **local price value**

$$\underline{V}_0^i[\lambda^i] = \min_{u^i \in \mathcal{U}_{\text{ad}}^i} J^i(u^i) + \langle \lambda^i, \Theta^i(u^i) \rangle$$

- for any **resource**  $r^i \in \mathcal{C}^i$ , we define the **local resource value**

$$\overline{V}_0^i[r^i] = \min_{u^i \in \mathcal{U}_{\text{ad}}^i} J^i(u^i) \quad \text{s.t.} \quad \Theta^i(u^i) = r^i$$

## Theorem 1 (Upper and lower bounds for optimal value)

- For any **admissible price**  $\lambda = (\lambda^1, \dots, \lambda^N) \in S^*$
- For any **admissible resource**  $r = (r^1, \dots, r^N) \in -S$

$$\sum_{i=1}^N \underline{V}_0^i[\lambda^i] \leq V_0^\# \leq \sum_{i=1}^N \overline{V}_0^i[r^i]$$

# The case of multistage stochastic optimization

Assume that the **local price value**

$$\underline{V}_0^i[\lambda^i] = \min_{u^i \in \mathcal{U}_{\text{ad}}^i} J^i(u^i) + \langle \lambda^i, \Theta^i(u^i) \rangle,$$

corresponds to a **stochastic optimal control problem**

$$\begin{aligned} \underline{V}_0^i[\Lambda^i](x_0^i) &= \min_{x^i, u^i} \mathbb{E} \left( \sum_{t=0}^{T-1} L_t^i(x_t^i, u_t^i, w_{t+1}) + \langle \Lambda_t^i, \Theta_t^i(x_t^i, u_t^i) \rangle + K^i(x_T^i) \right) \\ \text{s.t. } x_{t+1}^i &= f_t^i(x_t^i, u_t^i, w_{t+1}), \quad x_0^i = f_{-1}^i(w_0) \\ \sigma(u_t^i) &\subset \sigma(w_0, \dots, w_t) \end{aligned}$$

This local control problem can be solved by **Dynamic Programming** (DP) under restrictive assumptions:

- the noise process  $W$  is a **white noise** process
- the price process  $\Lambda^i$  follows a **dynamics in small dimension**

DP leads to a collection  $\{\underline{V}_t^i[\Lambda^i]\}_{t \in [0, T]}$  of **local price value functions**

# The case of multistage stochastic optimization

II

Similar considerations hold true for the **local resource value**

$$\bar{V}_0^i[r^i] = \min_{u^i \in \mathcal{U}_{\text{ad}}^i} J^i(u^i) \quad \text{s.t.} \quad \Theta^i(u^i) = r^i$$

considered as a **stochastic optimal control problem**

$$\begin{aligned} \bar{V}_0^i[\mathbf{R}^i](x_0^i) &= \min_{\mathbf{x}^i, \mathbf{u}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}) + K^i(\mathbf{x}_T^i) \right) \\ \text{s.t. } \mathbf{x}_{t+1}^i &= f_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}), \quad \mathbf{x}_0^i = f_1^i(\mathbf{w}_0) \\ \sigma(\mathbf{u}_t^i) &\subset \sigma(\mathbf{w}_0, \dots, \mathbf{w}_t) \\ \Theta_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i) &= \mathbf{R}_t^i \end{aligned}$$

Provided that the **dynamics** of the resource process  $\mathbf{R}^i$  is **small**,  
DP leads to a collection  $\{\bar{V}_t^i[\mathbf{R}^i]\}_{t \in \llbracket 0, T \rrbracket}$  of **local resource value functions**

# Mix of spatial and temporal decompositions

For any **admissible price process**  $\Lambda \in S^*$  and any **admissible resource process**  $R \in -S$ , we have bounds of the optimal value  $V_0^\#$

$$\sum_{i=1}^N \underline{V}_0^i[\Lambda^i](x_0^i) \leq V_0^\# \leq \sum_{i=1}^N \overline{V}_0^i[R^i](x_0^i)$$

① To obtain the bounds, we perform **spatial decompositions** giving

- a collection  $\{\underline{V}_0^i[\Lambda^i](x_0^i)\}_{i \in \llbracket 1, N \rrbracket}$  of price local values
- a collection  $\{\overline{V}_0^i[R^i](x_0^i)\}_{i \in \llbracket 1, N \rrbracket}$  of resource local values

*The computation of these local values can be performed in **parallel***

② To compute each local value, we perform **temporal decomposition** based on **Dynamic Programming**. For each  $i$ , we obtain

- a sequence  $\{\underline{V}_t^i[\Lambda^i]\}_{t \in \llbracket 0, T \rrbracket}$  of price local value functions
- a sequence  $\{\overline{V}_t^i[R^i]\}_{t \in \llbracket 0, T \rrbracket}$  of resource local value functions

*The computation of these local values functions is done **sequentially***

# Mix of spatial and temporal decompositions

II

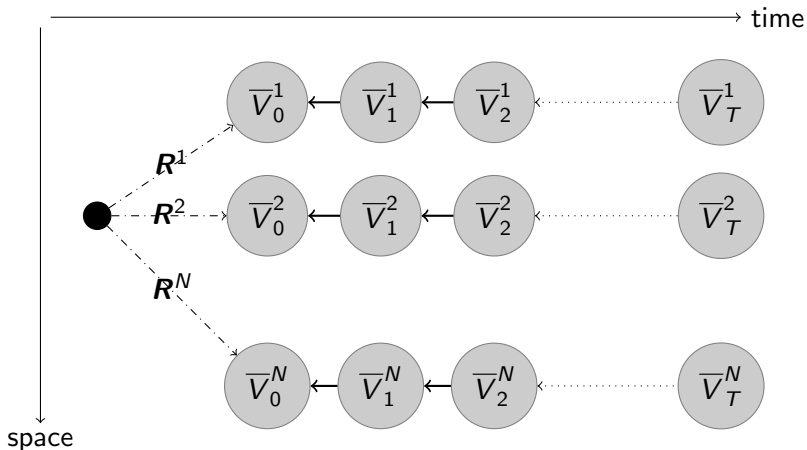


Figure: The case of resource decomposition

## The case of deterministic price and resource processes

We assume that  $W$  is a **white noise process**, and we restrict ourselves to **deterministic** admissible processes  $r \in -S$ ,  $\lambda \in S^*$

- The **local value functions**  $\underline{V}_t^i[\lambda^i]$  and  $\overline{V}_t^i[r^i]$  are easy to compute because they **only depend** on the local state variable  $x^i$
- It is easy to obtain **tighter bounds** by **maximizing** the lower bound w.r.t. prices and **minimizing** the upper bound w.r.t. resources

$$\sup_{\lambda \in S^*} \sum_{i=1}^N \underline{V}_0^i[\lambda^i](x_0^i) \leq V_0^\# \leq \inf_{r \in -S} \sum_{i=1}^N \overline{V}_0^i[r^i](x_0^i)$$

## Heuristic procedure to produce online admissible policies

The **local value functions**  $\underline{V}_t^i[\lambda^i]$  and  $\bar{V}_t^i[r^i]$  allow the computation of **global policies** by solving static optimization problems

- In the case of local **price** value functions, the policy is obtained as

$$\begin{aligned} \underline{\gamma}_t(x_t^1, \dots, x_t^N) \in \arg \min_{u_t^1, \dots, u_t^N} \mathbb{E} \left( \sum_{i=1}^N L_t^i(x_t^i, u_t^i, \mathbf{W}_{t+1}) + \sum_{i=1}^N \underline{V}_{t+1}^i[\lambda^i](x_{t+1}^i) \right) \\ \text{s.t. } \mathbf{X}_{t+1}^i = f_t^i(x_t^i, u_t^i, \mathbf{W}_{t+1}), \quad \forall i \in \llbracket 1, N \rrbracket \\ (\Theta_t(x_t^1, u_t^1), \dots, \Theta_t(x_t^N, u_t^N)) \in -S_t \end{aligned}$$

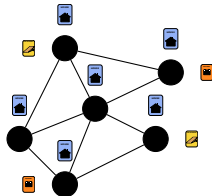
- A global policy based on **resource** value functions is also available

Estimating the expected cost of such policies by Monte Carlo simulation leads to a **statistical upper bound** of the optimal cost of the problem

# Progress status

- First, we have obtained **lower** and **upper** bounds for a global optimization problem with coupling constraints thanks to two **spatial decomposition** schemes
  - price decomposition
  - resource decomposition
- Second, we have computed the lower and upper bounds by dynamic programming (**temporal decomposition**)
- Using the price and resource Bellman value functions, we have devised two **online policies** for the **global** problem

We will apply these decomposition schemes to **large-scale network problems**





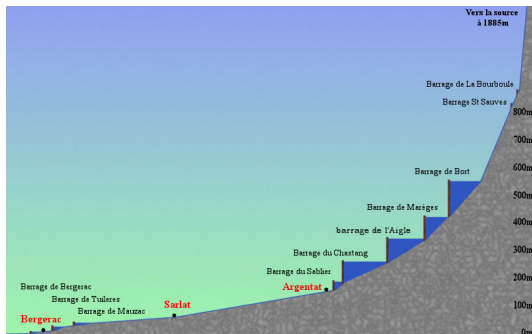
- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

# The Durance cascade



# Motivation

## Electricity production management for hydro valleys

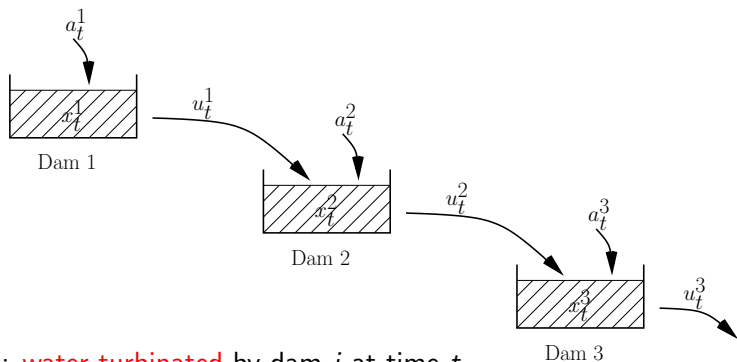


- *1 year time horizon:*  
compute each month the “values of water” (Bellman functions)
- *stochastic framework:*  
rain, market prices
- *large-scale valley:*  
5 dams and more

We wish to remain within the scope of **Dynamic Programming**.

- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
  
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
  
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

## Operating scheme



$u_t^i$  : water turbinated by dam  $i$  at time  $t$ ,

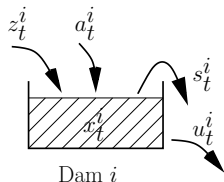
$x_t^i$  : water volume of dam  $i$  at time  $t$ ,

$a_t^i$  : water inflow at dam  $i$  at time  $t$ ,

$p_t^i$  : market price at dam  $i$  at time  $t$ ,

**Randomness:**  $w_t^i = (a_t^i, p_t^i)$  and  $w_t = (w_t^1, \dots, w_t^N)$ .

# Dynamics and costs functions



Dam dynamics:

$$\begin{aligned}
 x_{t+1}^i &= f_t^i(x_t^i, u_t^i, w_t^i, z_t^i), \\
 &= x_t^i - u_t^i + a_t^i + z_t^i - s_t^i, \\
 z_t^{i+1} &\text{ being the outflow of dam } i: \\
 z_t^{i+1} &= g_t^i(x_t^i, u_t^i, w_t^i, z_t^i), \\
 &= u_t^i + \underbrace{\max\{0, x_t^i - u_t^i + a_t^i + z_t^i - \bar{x}^i\}}_{s_t^i}.
 \end{aligned}$$

We assume the **Hazard-Decision** information structure ( $u_t^i$  is chosen once  $w_t^i$  is observed), so that  $\underline{u}^i \leq u_t^i \leq \min\{\bar{u}^i, x_t^i + a_t^i + z_t^i - \underline{x}^i\}$ .

Gain at time  $t < T$ :  $L_t^i(x_t^i, u_t^i, w_t^i, z_t^i) = p_t^i u_t^i - \epsilon (u_t^i)^2$ .

Final gain at time  $T$ :  $K^i(x_T^i) = -a^i \min\{0, x_T^i - \hat{x}^i\}^2$ .

# Stochastic optimization problem

The **global optimization** problem reads:

$$\max_{(\mathbf{x}, \mathbf{u}, \mathbf{z})} \mathbb{E} \left( \sum_{i=1}^N \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) + K^i(\mathbf{x}_T^i) \right) \right),$$

subject to:

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

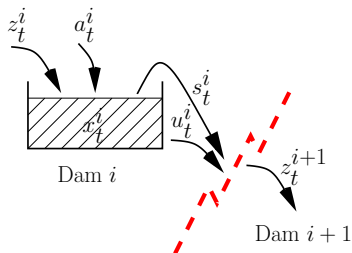
$$\sigma(\mathbf{u}_t^i) \subset \sigma(\mathbf{w}_0, \dots, \mathbf{w}_t), \quad \forall i, \quad \forall t,$$

$$\mathbf{z}_t^{i+1} = g_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i), \quad \forall i, \quad \forall t.$$

**Assumption.** Noises  $\mathbf{w}_0, \dots, \mathbf{w}_{T-1}$  are *independent over time*.

# Standard price decomposition

- Dualize the coupling constraints  $\mathbf{Z}_t^{i+1} = g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)$ .  
 Note that the associated multiplier  $\Lambda_t^{i+1}$  is a **random variable**.
- Minimize the **dual problem** (using a gradient-like algorithm).



- At iteration  $k$ , the duality term:  

$$\Lambda_t^{i+1,(k)} \cdot (\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i))$$
 is the difference of two terms:
  - $\Lambda_t^{i+1,(k)} \cdot \mathbf{Z}_t^{i+1} \rightsquigarrow$  dam  $i+1$ ,
  - $\Lambda_t^{i+1,(k)} \cdot g_t^i(\dots) \rightsquigarrow$  dam  $i$ .
- **Dam by dam decomposition** for the maximization in  $(\mathbf{X}, \mathbf{U}, \mathbf{Z})$  at  $\Lambda_t^{i+1,(k)}$  fixed.



# Application of DADP

The  $i$ -th subproblem writes:

$$\max_{\mathbf{u}^i, \mathbf{z}^i, \mathbf{x}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) + \boldsymbol{\Lambda}_t^{i,(k)} \cdot \mathbf{z}_t^i - \boldsymbol{\Lambda}_t^{i+1,(k)} \cdot g_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) \right) + K^i(\mathbf{x}_T^i) \right),$$

but  $\boldsymbol{\Lambda}_t^{i,(k)}$  depends on the **whole past** of noises  $(\mathbf{w}_0, \dots, \mathbf{w}_t) \dots$

We recall that the **core idea** of DADP is

- to **replace** the constraint  $\mathbf{z}_t^{i+1} - g_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) = 0$  by its **conditional expectation** with respect to  $\mathbf{Y}_t^i$ :

$$\mathbb{E}(\mathbf{z}_t^{i+1} - g_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) \mid \mathbf{Y}_t^i) = 0,$$

- where  $(\mathbf{Y}_0^i, \dots, \mathbf{Y}_{T-1}^i)$  is a “well-chosen” **information process**.

## Subproblems in DADP

DADP thus consists of a **constraint relaxation**, which is equivalent to replace the multiplier  $\Lambda_t^{i,(k)}$  by its **conditional expectation**  $\mathbb{E}(\Lambda_t^{i,(k)} \mid \mathbf{Y}_t^{i-1})$ .

The expression of the  $i$ -th subproblem becomes:

$$\max_{\mathbf{u}^i, \mathbf{z}^i, \mathbf{x}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} \left( L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) + \mathbb{E}(\Lambda_t^{i,(k)} \mid \mathbf{Y}_t^{i-1}) \cdot \mathbf{z}_t^i - \mathbb{E}(\Lambda_t^{i+1,(k)} \mid \mathbf{Y}_t^i) \cdot \mathbf{g}_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i) + K^i(\mathbf{x}_T^i) \right) \right).$$

If each process  $\mathbf{Y}^i$  follows a dynamical equation, **DP applies**.

## A crude relaxation: $\mathbf{Y}_t^i \equiv \text{cste}$

- 1 The multipliers  $\boldsymbol{\Lambda}_t^{i,(k)}$  appear only in the subproblems by means of their expectations  $\mathbb{E}(\boldsymbol{\Lambda}_t^{i,(k)})$ , so that each subproblem involves a **1-dimensional** state variable.
- 2 For the gradient algorithm, the coordination task reduces to:

$$\mathbb{E}(\boldsymbol{\Lambda}_t^{i,(k+1)}) = \mathbb{E}(\boldsymbol{\Lambda}_t^{i,(k)}) - \rho_t \mathbb{E}(\mathbf{z}_t^{i+1,(k)} - \mathbf{g}_t^i(\mathbf{x}_t^{i,(k)}, \mathbf{u}_t^{i,(k)}, \mathbf{w}_t^i, \mathbf{z}_t^{i,(k)})) .$$

- 3 The constraints taken into account by DADP are

$$\mathbb{E}(\mathbf{z}_t^{i+1} - \mathbf{g}_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_t^i, \mathbf{z}_t^i)) = 0 .$$

The DADP solutions do not satisfy the initial constraints: we need to use an **heuristic method** to regain admissibility.

## Admissible online policies for the global problem

We have computed  $N$  local Bellman functions  $\underline{V}_t^i$  at each time  $t$ , each depending on a single state variable  $x^i$ , whereas we need one global Bellman function  $V_t$  depending on the global state  $(x^1, \dots, x^N)$  in order to design the decisions at time  $t$ .

**Heuristic procedure:** form the following global Bellman function:

$$\widehat{V}_t(x^1, \dots, x^N) = \sum_{i=1}^N \underline{V}_t^i(x^i),$$

and solve at each time  $t$  the one-step DP problem:

$$\max_{u, z} \sum_{i=1}^N L_t^i(x^i, u^i, w_t^i, z^i) + \widehat{V}_{t+1}(x_{t+1}^1, \dots, x_{t+1}^N),$$

$$\text{s.t. } x_{t+1}^i = f_t^i(x^i, u^i, w_t^i, z^i), \quad z^{i+1} = g_t^i(x^i, u^i, w_t^i, z^i) \quad \forall i.$$

## Bounds for the problem optimal cost

Let  $V_0^\#$  be the **optimal** value of the global optimization problem.

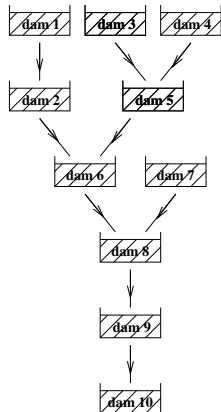
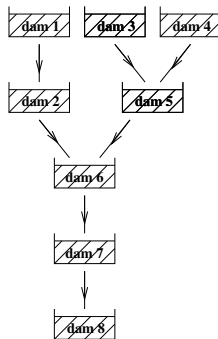
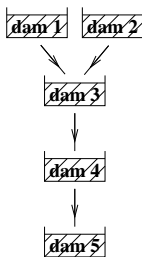
- 1 As already noticed, the optimal value computed by DADP, that is, the **sum of the optimal values of the subproblems** once the optimal multiplier  $\mathbb{E}(\Lambda_t)$  have been obtained, is an **exact upper bound**<sup>2</sup> of  $V_0^\#$ .
- 2 The **expected value** associated to the **admissible policy** induced by the sum of the local Bellman functions is a **lower bound** of global problem optimal value. This expected value being evaluated by Monte Carlo, we in fact have at disposal a **statistical lower bound** of  $V_0^\#$ .

---

<sup>2</sup>and not a lower bound because we are dealing with a **maximization problem**

- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

## Four case studies



*Discretization*

$$T \rightsquigarrow 12$$

$$X \rightsquigarrow 41$$

$$U \rightsquigarrow 6$$

$$W \rightsquigarrow 10$$

3-Dams

5-Dams

8-Dams

10-Dams

## Results

Valley	3-Dams	5-Dams	8-Dams	10-Dams
DP CPU time	5'	461200'	N.A.	N.A.
DP value	2482.3	4681.6	N.A.	N.A.
SDDP exact UB	2491.3	4694.1	11958.3	17256.0
SDDP value	2481.6	4680.9	11834.4	17069.3
SDDP CPU time	3'	7'	13'	50'

Table: Results obtained by DP and SDDP

Valley	3-Dams	5-Dams	8-Dams	10-Dams
DADP CPU time	3'	5'	12'	24'
DADP exact UB	2687.5	4885.9	12451.0	17933.5
DADP value	2401.6	4633.7	11573.0	16759.8
Gap with SDDP	-3.2%	-1.0%	-2.2%	-1.8%

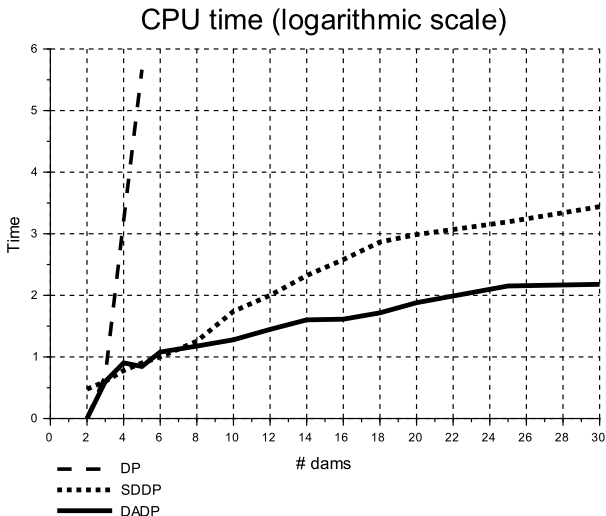
Table: Results obtained by DADP

---

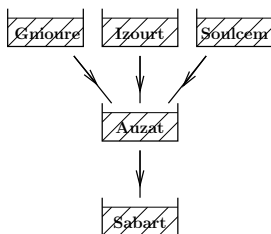
Results obtained using a 4 cores – 8 threads Intel®Core i7 based computer.



# Challenging the curse of dimensionality



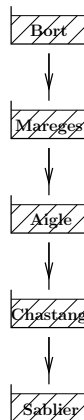
## Two realistic valleys



*Discretization*

$T \rightsquigarrow 12$ ,  $W \rightsquigarrow 10$

realistic grids for  $U$  and  $X$



Vicdessos

Dordogne

# Results

Valley	Vicdessos	Dordogne
SDDP CPU time	9'	17'
SDDP exact UB	2258.0	22310.0
SDDP value	2244.3	22136.1

Table: Results obtained by SDDP

Valley	Vicdessos	Dordogne
DADP CPU time	10'	210'
DADP exact UB	2285.6	22991.1
DADP value	2237.4	21650.8
Gap with SDDP	-0.3%	-2.2%

Table: Results obtained by DADP

# Conclusions and perspectives

## Conclusions for this study

- Fast numerical convergence of the DADP method.
- Near-optimal results even when using a “crude” relaxation.
- Method that can be used for very large valleys

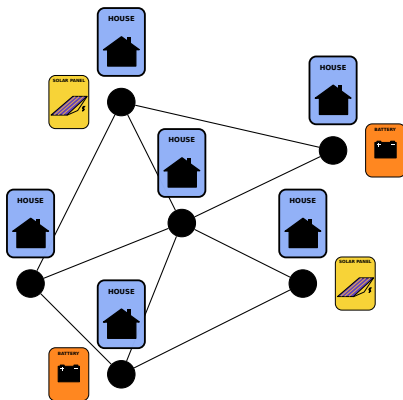
## General perspectives

- Apply to more complex topologies (microgrids).
- Use other decomposition methods (resource, prediction).
- Study the theoretical questions (convergence. . . ).

- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

# Motivation

We consider a *peer-to-peer* microgrid where houses exchange energy, and we formulate it as a **large-scale stochastic** optimization problem

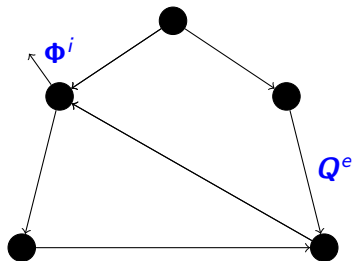


**How to manage it in an (sub)optimal manner?**

- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

## Network and flows

Directed graph  $G = (\mathcal{V}, \mathcal{E})$



- $Q_t^e$  flow through edge  $e$ ,
- $\Phi_t^i$  flow imported at node  $i$

Let  $A$  be the *node-edge* incidence matrix

Each node corresponds to a building with its own devices (battery, hot water tank, solar panel...)

At each time  $t \in \llbracket 0, T-1 \rrbracket$ , the **Kirchhoff current law** couples node and edge flows

$$A Q_t + \Phi_t = 0$$



## Optimization problem at a given node

At each **node**  $i \in \mathcal{V}$ , given a node flow process  $\Phi^i$ , we minimize the house cost

$$J_{\mathcal{V}}^i(\Phi^i) = \min_{\mathbf{x}^i, \mathbf{u}^i} \mathbb{E} \left( \sum_{t=0}^{T-1} L_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}^i) + K^i(\mathbf{x}_T^i) \right)$$

subject to, for all  $t \in \llbracket 0, T-1 \rrbracket$

i) **nodal dynamics** constraints (battery, hot water tank)

$$\mathbf{x}_{t+1}^i = f_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}^i)$$

ii) **non-anticipativity** constraints (future remains unknown)

$$\sigma(\mathbf{u}_t^i) \subset \sigma(\mathbf{w}_0, \dots, \mathbf{w}_{t+1})$$

iii) **nodal load balance** equations (demand - production = import)

$$\Delta_t^i(\mathbf{x}_t^i, \mathbf{u}_t^i, \mathbf{w}_{t+1}^i) = \Phi_t^i$$

# Transportation cost and global optimization problem

We define the **network cost** as the sum over time and **edges** of the costs of flows  $Q_t^e$  through the edges of the network

$$J_{\mathcal{E}}(\mathbf{Q}) = \mathbb{E} \left( \sum_{t=0}^{T-1} \sum_{e \in \mathcal{E}} l_t^e(Q_t^e) \right)$$

This transportation cost is **additive** in space, in time and in uncertainty!

The global optimization problem is obtained by gathering all elements

$$\begin{aligned} V_0^{\mathcal{E}} &= \min_{\Phi, \mathbf{Q}} \sum_{i \in \mathcal{V}} J_i(\Phi^i) + J_{\mathcal{E}}(\mathbf{Q}) \\ &\text{s.t. } A\mathbf{Q} + \Phi = 0 \end{aligned}$$

# Transportation cost and global optimization problem

We define the **network cost** as the sum over time and **edges** of the costs of flows  $Q_t^e$  through the edges of the network

$$J_{\mathcal{E}}(\mathbf{Q}) = \mathbb{E} \left( \sum_{t=0}^{T-1} \sum_{e \in \mathcal{E}} l_t^e(\mathbf{Q}_t^e) \right)$$

This transportation cost is **additive** in space, in time and in uncertainty!

The global **optimization problem** is obtained by gathering all elements

$$\begin{aligned} V_0^\# &= \min_{\Phi, \mathbf{Q}} \sum_{i \in \mathcal{V}} J_V^i(\Phi^i) + J_{\mathcal{E}}(\mathbf{Q}) \\ &\text{s.t. } A\mathbf{Q} + \Phi = 0 \end{aligned}$$

# Price and resource decompositions

- **Price** problem:

$$\begin{aligned} \underline{V}_0[\Lambda] &= \min_{\Phi, Q} \sum_{i \in \mathcal{V}} J_V^i(\Phi^i) + J_E(Q) + \langle \Lambda, AQ + \Phi \rangle \\ &= \sum_{i \in \mathcal{V}} \underbrace{\left( \min_{\Phi^i} J_V^i(\Phi^i) + \langle \Lambda^i, \Phi^i \rangle \right)}_{\text{Node } i\text{'s subproblem}} + \underbrace{\left( \min_Q J_E(Q) + \langle A^T \Lambda, Q \rangle \right)}_{\text{Network subproblem}} \end{aligned}$$

- **Resource** problem:

$$\begin{aligned} \overline{V}_0[R] &= \min_{\Phi, Q} \sum_{i \in \mathcal{V}} J_V^i(\Phi^i) + J_E(Q) \quad \text{s.t.} \quad AR + \Phi = 0, \quad Q = R \\ &= \sum_{i \in \mathcal{V}} \left( \min_{\Phi^i} J_V^i(\Phi^i) \quad \text{s.t.} \quad \Phi^i = -(AR)^i \right) + \left( \min_Q J_E(Q) \quad \text{s.t.} \quad Q = R \right) \end{aligned}$$

Find deterministic processes  $\hat{\lambda}$  and  $\hat{r}$  with a gap as small as possible

$$\sup_{\hat{\lambda}} \underline{V}_0[\hat{\lambda}] \leq V_0^* \leq \inf_{\hat{r}} \overline{V}_0[\hat{r}]$$

# Price and resource decompositions

- **Price** problem:

$$\begin{aligned} \underline{V}_0[\Lambda] &= \min_{\Phi, Q} \sum_{i \in \mathcal{V}} J_V^i(\Phi^i) + J_E(Q) + \langle \Lambda, AQ + \Phi \rangle \\ &= \sum_{i \in \mathcal{V}} \underbrace{\left( \min_{\Phi^i} J_V^i(\Phi^i) + \langle \Lambda^i, \Phi^i \rangle \right)}_{\text{Node } i\text{'s subproblem}} + \underbrace{\left( \min_Q J_E(Q) + \langle A^T \Lambda, Q \rangle \right)}_{\text{Network subproblem}} \end{aligned}$$

- **Resource** problem:

$$\begin{aligned} \overline{V}_0[R] &= \min_{\Phi, Q} \sum_{i \in \mathcal{V}} J_V^i(\Phi^i) + J_E(Q) \quad \text{s.t.} \quad AR + \Phi = 0, \quad Q = R \\ &= \sum_{i \in \mathcal{V}} \left( \min_{\Phi^i} J_V^i(\Phi^i) \quad \text{s.t.} \quad \Phi^i = -(AR)^i \right) + \left( \min_Q J_E(Q) \quad \text{s.t.} \quad Q = R \right) \end{aligned}$$

Find **deterministic** processes  $\hat{\lambda}$  and  $\hat{r}$  with a **gap** as small as possible

$$\sup_{\lambda} \underline{V}_0[\lambda] \leq V_0^\# \leq \inf_r \overline{V}_0[r]$$

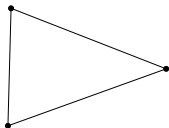
## Progress status

- We have formulated a **multistage stochastic optimization** problem on a graph
- We are able to handle the coupling Kirchhoff constraints by the two methods presented earlier
  - Price decomposition
  - Resource decomposition
- Now, we show the scalability of decomposition algorithms (we solve problems with up to **48 buildings**)

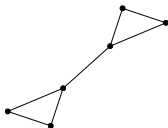
- 1 Mixing spatial and temporal decompositions
  - Problem formulation and price decomposition
  - Dual approximate dynamic programming (DADP)
  - Upper and lower bounds for large scale SOC problems
- 2 Application to dams management problems
  - Hydro valley modeling
  - Numerical experiments
- 3 Application to microgrids management problems
  - Urban microgrid modeling
  - Numerical experiments

# Different urban configurations

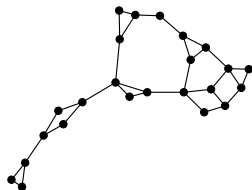
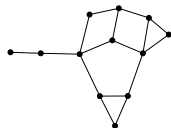
**3-Nodes**



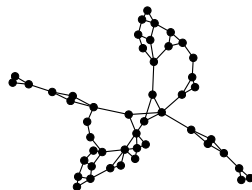
**6-Nodes**



**12-Nodes**



**24-Nodes**



**48-Nodes**



## Problem settings

Thanks to the periodicity of demands and electricity tariffs, the microgrid management problem can be solved day by day

- One day horizon with a 15mn time step:  $T = 96$
- Weather corresponds to a sunny day in Paris (*June 28, 2015*)
- We mix three kinds of buildings
  - 1 battery + electrical hot water tank
  - 2 solar panel + electrical hot water tank
  - 3 electrical hot water tank

and we suppose that all consumers are sharing their devices

# Algorithms implemented on the problem

## SDDP

We use the SDDP algorithm to solve the problem **globally**...

- but noises  $W_t^1, \dots, W_t^N$  are **independent node by node**, so that the support size of the noise may be **huge** ( $|\text{supp}(W_t^i)|^N$ ). We must **resample the noise** to be able to compute the cuts

## Price decomposition

Spatial decomposition and maximization w.r.t. a **deterministic price**  $\lambda$

- Each nodal subproblem solved by a DP-like method
- Maximisation w.r.t.  $\lambda$  by Quasi-Newton (BFGS) method

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho^{(k)} H^{(k)} \nabla \underline{V}_0[\lambda^{(k)}]$$

- Oracle  $\nabla \underline{V}_0[\lambda]$  estimated by Monte Carlo ( $N^{scen} = 1,000$ )

## Resource decomposition

Spatial decomposition and minimization w.r.t. a **deterministic resource** process  $r$

# Exact upper and lower bounds on the global problem

	Network	3-Nodes	6-Nodes	12-Nodes	24-Nodes	48-Nodes
State dim.	$ \mathbb{X} $	4	8	16	32	64
SDDP	time	1'	3'	10'	79'	453'
SDDP	LB	225.2	455.9	889.7	1752.8	3310.3
Price	time	6'	14'	29'	41'	128'
Price	LB	213.7	447.3	896.7	1787.0	3396.4
Resource	time	3'	7'	22'	49'	91'
Resource	UB	253.9	527.3	1053.7	2105.4	4016.6

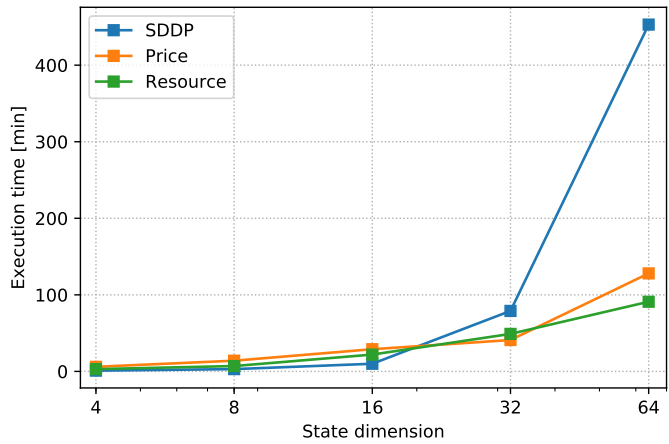
For the **48-Nodes** microgrid,

- price decomposition gives a (slightly) **better exact lower bound** than SDDP

$$\underbrace{3310.3}_{V_0[\text{sddp}]} \leq \underbrace{3396.4}_{V_0[\text{price}]} \leq V_0^\# \leq \underbrace{4016.6}_{\bar{V}_0[\text{resource}]}$$

- price decomposition is more than **3 times faster** than SDDP

# Time evolution



# Policy evaluation by Monte Carlo (1,000 scenarios)

	3-Nodes	6-Nodes	12-Nodes	24-Nodes	48-Nodes
SDDP policy	226 ± 0.6	471 ± 0.8	936 ± 1.1	1859 ± 1.6	3550 ± 2.3
Price policy	228 ± 0.6	464 ± 0.8	923 ± 1.2	1839 ± 1.6	3490 ± 2.3
Gap	+0.9 %	-1.5%	-1.4%	-1.1%	-1.7%
Resource policy	229 ± 0.6	471 ± 0.8	931 ± 1.1	1856 ± 1.6	3503 ± 2.2
Gap	+1.3 %	0.0%	-0.5%	-0.2%	-1.2%

All the cost values above are **statistical upper bounds** of  $V_0^\#$

For the **48-Nodes** microgrid,

- price policy **beats** SDDP policy and resource policy

$$V_0^\# \leq \underbrace{3490}_{C[\text{price}]} \leq \underbrace{3503}_{C[\text{resource}]} \leq \underbrace{3550}_{C[\text{sddp}]}$$

- the **exact upper bound** given by resource decomposition is **not so tight**

$$\underbrace{3396.4}_{V_0[\text{price}]} \leq V_0^\# \leq \underbrace{3490}_{C[\text{price}]} \leq \underbrace{3503}_{C[\text{resource}]} \leq \underbrace{4016.6}_{V_0[\text{resource}]}$$

gap
<3%
≈ 3%
>18%

## Conclusions

- We have two algorithms that **decompose spatially and temporally** a large-scale optimization problem under coupling constraints.
- In our case study, **price decomposition beats SDDP** for large instances ( $\geq 24$  nodes)
  - in computing time (more than twice faster)
  - in precision (more than 1% better)
- **Price decomposition** gives (in a surprising way) a **tight lower bound**, whereas the **upper bound** given by **resource decomposition** is **weak** (which is understandable on the case study)
- **Can we obtain tighter bounds?** *especially for resource decomposition...*  
If we select properly price  $\lambda$  and resource  $R$  processes among the class of **Markovian** processes (instead of **deterministic** ones) we can obtain “better” nodal value functions (with an extended local state)

## Further details in

F. Pacaud

*Decentralized Optimization Methods for Efficient Energy Management under Stochasticity*

**PhD Thesis, Université Paris Est, 2018**

P. Carpentier, J.-P. Chancelier, V. Leclère and F. Pacaud

*Stochastic decomposition applied to large-scale hydro valleys management*

**"European Journal of Operational Research, 270(3):1086–1098, 2018**

P. Carpentier, J.-P. Chancelier, M. De Lara and F. Pacaud

*Mixed Spatial and Temporal Decompositions for Large-Scale Multistage Stochastic Optimization Problem*

**Journal of Optimization Theory and Applications, 186:985–1005, 2020**

# THANK YOU FOR YOUR ATTENTION



C. Strugarek.

*Approches variationnelles et contributions en optimisation stochastique.*

Thèse de doctorat, Ecole Nationale des Ponts et Chaussées, 2006.



P. Girardeau.

*Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande.*

Thèse de doctorat, Université Paris-Est, 2010.



J.-C. Alais.

*Risque et optimisation pour le management d'énergies.*

Thèse de doctorat, Université Paris-Est, 2013.



V. Leclère.

*Contributions aux méthodes de décomposition en optimisation stochastique.*

Thèse de doctorat, Université Paris-Est, 2014.







F. Pacaud.

*Decentralized Optimization Methods for Efficient Energy Management under Stochasticity.*

Thèse de doctorat, Université Paris-Est, 2018.



-  K. Barty, P. Carpentier and P. Girardeau.  
Decomposition of large-scale stochastic optimal control problems.  
*RAIRO Operations Research*, 44, 167-183, 2010.
-  K. Barty, P. Carpentier, G. Cohen and P. Girardeau,  
Price decomposition in large-scale stochastic optimal control.  
*arXiv*, math.OC 1012.2092, 2010.
-  P. Carpentier, J.-Ph. Chancelier, V. Leclère and F. Pacaud.  
Stochastic decomposition applied to large-scale hydro valleys  
management.  
*European Journal of Operational Research*, 270, 1086–1098, 2018.
-  P. Carpentier, J.-Ph. Chancelier, M. De Lara and F. Pacaud.  
Mixed Spatial and Temporal Decompositions for Large-Scale Multistage  
Stochastic Optimization Problem.  
*Journal of Optimization Theory and Applications*, 186, 985–1005, 2020