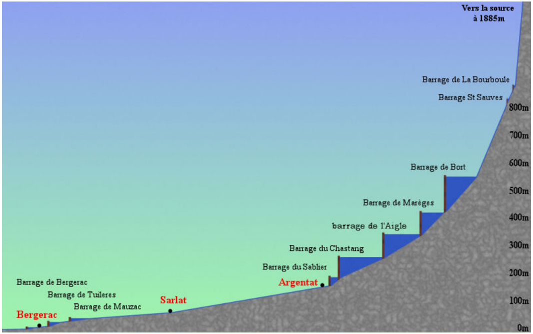
# Dual Approximate Dynamic Programming for Large Scale Hydro Valleys

P. Carpentier, J-P. Chancelier, V. Leclère, F. Pacaud
*supported by the FMJH Program Gaspard Monge for Optimization.*

ENSTA ParisTech and ENPC ParisTech, France

# 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.

# How to push the curse of dimensionality limits?

### Aggregation methods

- fast to run method
- require some homogeneity between units

### Stochastic Dual Dynamic Programming (SDDP)

- efficient method for this kind of problems
- strong assumptions (convexity, linearity)

### Dual Approximate Dynamic Programming (DADP)

- spatial decomposition method
- complexity almost linear in the number of dams
- approximation methods in the stochastic framework

**This talk**: present numerical results for large-scale hydro valleys using DADP, and comparison with DP and SDDP.

# Lecture outline

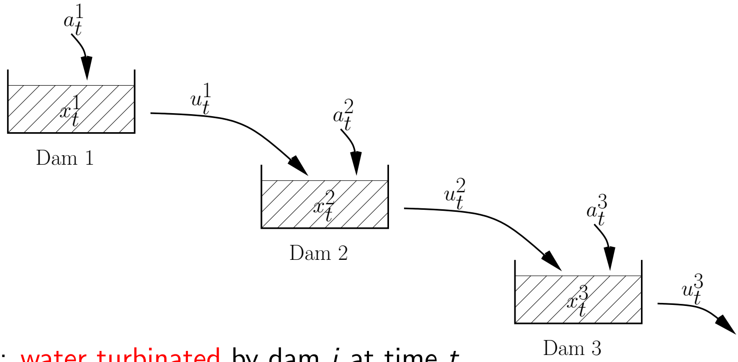
1. Dams management problem
   - Hydro valley modeling
   - Optimization problem
2. DADP in a nutshell
   - Spatial decomposition
   - Constraint weakening
3. Numerical experiments
   - Academic examples
   - More realistic examples

1. Dams management problem
   - Hydro valley modeling
   - Optimization problem

2. DADP in a nutshell
   - Spatial decomposition
   - Constraint weakening

3. Numerical experiments
   - Academic examples
   - More realistic examples

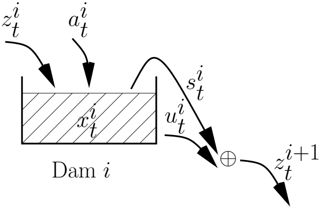
# Operating scheme




$u_t^i$ : 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 $t$,
$p_t^i$ : water price at dam $i$ at time $t$,

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

# Dynamics and cost functions




Dam dynamics:

$$x_{t+1}^i = f_t^i(x_t^i, u_t^i, w_t^i, z_t^i) \ , \\ = x_t^i - u_t^i + a_t^i + z_t^i - s_t^i \ ,$$

$z_t^{i+1}$ being the outflow of dam $i$:

$$z_t^{i+1} = g_t^i(x_t^i, u_t^i, w_t^i, z_t^i) \ , \\ = u_t^i + \underbrace{\max\left\{0, x_t^i - u_t^i + a_t^i + z_t^i - \overline{x}^i\right\}}_{s_t^i} \ .$$

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

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 $T$: $K^i(x_T^i) = -a^i \min\{0, x_T^i - \widehat{x}^i\}^2$.

1. Dams management problem
   - Hydro valley modeling
   - Optimization problem
2. DADP in a nutshell
   - Spatial decomposition
   - Constraint weakening
3. Numerical experiments
   - Academic examples
   - More realistic examples

# 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:

$$\mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i,\mathbf{U}_t^i,\mathbf{W}_t^i,\mathbf{Z}_t^i)\ , \ \forall i\ , \ \forall t\ ,$$

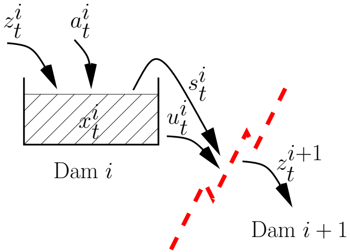
$$\sigma\left(\mathbf{U}_t^i\right) \subset \sigma\left(\mathbf{W}_0,\ldots,\mathbf{W}_t\right)\ , \quad \forall i\ , \ \forall t\ ,$$

$$\mathbf{Z}_t^{i+1} = g_t^i(\mathbf{X}_t^i,\mathbf{U}_t^i,\mathbf{W}_t^i,\mathbf{Z}_t^i)\ , \ \forall i\ , \ \forall t\ .$$

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

# 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 $\mathbf{\Lambda}_t^{i+1}$ is a random variable.
- Minimize the dual problem (using a gradient-like algorithm).




- At iteration $k$, the duality term:

  $$\mathbf{\Lambda}_t^{i+1,(k)} \cdot \big(\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)\big) ,$$

  is the difference of two terms:

  - $\mathbf{\Lambda}_t^{i+1,(k)} \cdot \mathbf{Z}_t^{i+1}$ $\rightsquigarrow$ dam $i+1$,
  - $\mathbf{\Lambda}_t^{i+1,(k)} \cdot g_t^i(\cdots)$ $\rightsquigarrow$ dam $i$.

- Dam by dam decomposition for the maximization in $(\mathbf{X}, \mathbf{U}, \mathbf{Z})$ at $\mathbf{\Lambda}_t^{i+1,(k)}$ fixed.

1. Dams management problem
   - Hydro valley modeling
   - Optimization problem
2. DADP in a nutshell
   - Spatial decomposition
   - Constraint weakening
3. Numerical experiments
   - Academic examples
   - More realistic examples

# DADP core idea

The $i$-th subproblem writes:

$$\max_{\mathbf{U}^i,\mathbf{Z}^i,\mathbf{X}^i} \mathbb{E}\Bigg( \sum_{t=0}^{T-1} \Big( L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) + \boldsymbol{\Lambda}_t^{i,(k)} \cdot \mathbf{Z}_t^i - \boldsymbol{\Lambda}_t^{i+1,(k)} \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \Big) + K^i(\mathbf{X}_T^i) \Bigg) ,$$

but $\boldsymbol{\Lambda}_t^{i,(k)}$ depends on the whole past of noises $(\mathbf{W}_0, \ldots, \mathbf{W}_t)$...

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 $\mathbf{Y}_t^i$:

$$\mathbb{E}\big(\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \mid \mathbf{Y}_t^i\big) = 0 ,$$

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

# DADP core idea

The $i$-th subproblem writes:

$$\max_{\mathbf{U}^i,\mathbf{Z}^i,\mathbf{X}^i} \mathbb{E}\Bigg( \sum_{t=0}^{T-1} \Big( L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) + \mathbf{\Lambda}_t^{i,(k)} \cdot \mathbf{Z}_t^i - \mathbf{\Lambda}_t^{i+1,(k)} \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \Big) + K^i(\mathbf{X}_T^i) \Bigg) ,$$

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

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 $\mathbf{Y}_t^i$:

$$\mathbb{E}\big(\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \bigm| \mathbf{Y}_t^i\big) = 0 ,$$

- 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.

This constraint relaxation is equivalent to replace the original multiplier $\boldsymbol{\Lambda}_t^{i,(k)}$ by its conditional expectation $\mathbb{E}\big(\boldsymbol{\Lambda}_t^{i,(k)} \mid \mathbf{Y}_t^{i-1}\big)$.

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

$$\max_{\mathbf{U}^i,\mathbf{Z}^i,\mathbf{X}^i} \mathbb{E}\bigg( \sum_{t=0}^{T-1} \Big( L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) + \mathbb{E}\big(\boldsymbol{\Lambda}_t^{i,(k)} \mid \mathbf{Y}_t^{i-1}\big) \cdot \mathbf{Z}_t^i - \mathbb{E}\big(\boldsymbol{\Lambda}_t^{i+1,(k)} \mid \mathbf{Y}_t^{i}\big) \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)\Big) + K^i(\mathbf{X}_T^i)\bigg) .$$

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

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

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

$$\mathbb{E}\big(\boldsymbol{\Lambda}_t^{i,(k+1)}\big) = \mathbb{E}\big(\boldsymbol{\Lambda}_t^{i,(k)}\big) + \rho_t \mathbb{E}\Big(\mathbf{Z}_t^{i+1,(k)} - g_t^i\big(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t^i, \mathbf{Z}_t^{i,(k)}\big)\Big) .$$

3. The constraints taken into account by DADP are in fact:

$$\mathbb{E}\Big(\mathbf{Z}_t^{i+1} - g_t^i\big(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i\big)\Big) = 0 .$$

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

# How to regain admissible policies?

We have computed $N$ local Bellman functions $V_t^i$ 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:

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

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

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

with $x_{t+1}^i = f_t^i(x^i, u^i, w_t^i, z^i)$ and $z^{i+1} = g_t^i(x^i, u^i, w_t^i, z^i)$

# How to regain admissible policies?

We have computed $N$ local Bellman functions $V_t^i$ 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:

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

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

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

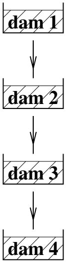
$$\textit{with} \quad x_{t+1}^i = f_t^i(x^i, u^i, w_t^i, z^i) \quad \text{and} \quad z^{i+1} = g_t^i(x^i, u^i, w_t^i, z^i) \ .$$

# Full optimization and simulation process

- **Optimization**
  - Apply DADP and compute the cost-to-go functions $V_t^i$.
  - Form the approximated global Bellman functions $\widehat{V}_t$.

- **Simulation**
  - Draw a large number of noise scenarios.
  - Compute the control values along each scenario by solving the one-step DP problems involving the $\widehat{V}_t$'s, thus satisfying all the constraints of the initial problem:
    $\rightsquigarrow$ payoff value for each scenario,
    $\rightsquigarrow$ state and control trajectories.
  - Evaluate the quality of the solution: mean payoff,...

# Academic case studies of increasing complexity



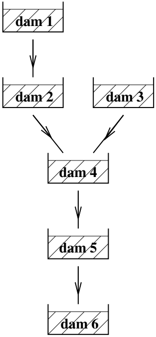

*Discretization*

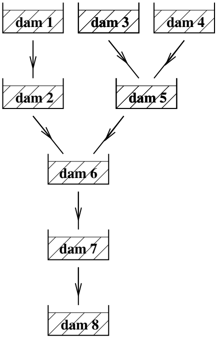
$T \rightsquigarrow 12$

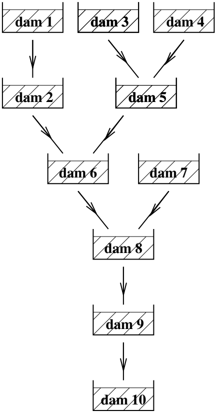
$\mathbf{X} \rightsquigarrow 41$

$\mathbf{U} \rightsquigarrow 6$

$\mathbf{W} \rightsquigarrow 10$




4-Dams

6-Dams

8-Dams

10-Dams

# Optimal values and computational times

| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DP CPU time | $1.6\ 10^{3}$' | $\sim 10^{8}$' | $\sim \infty$ | $\sim \infty$ |
| DP value | 3743 | N.A. | N.A. | N.A. |
| SDDP value | 3742 | 7026 | 11834 | 17069 |
| SDDP CPU time | 5' | 7' | 9' | 50' |
| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |

Table: Results obtained by DP and SDDP[1]

| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DADP CPU time | 7' | 12' | 17' | 24' |
| DADP value | 3667 | 6816 | 11573 | 16760 |
| Gap with SDDP | $-2.0\%$ | $-3.0\%$ | $-2.2\%$ | $-1.8\%$ |

Table: Results obtained by DADP "Expectation"

[1] SDDP method is an alternative to DP, pushing away the dimension barrier.

# Optimal values and computational times

| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DP CPU time | *1.6 $10^3$'* | $\sim$ *$10^8$'* | $\sim\infty$ | $\sim\infty$ |
| DP value | 3743 | N.A. | N.A. | N.A. |
| SDDP value | 3742 | 7026 | 11834 | 17069 |
| SDDP CPU time | *5'* | *7'* | *9'* | *50'* |
| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |

Table: Results obtained by DP and SDDP[1]

| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DADP CPU time | *7'* | *12'* | *17'* | *24'* |
| DADP value | 3667 | 6816 | 11573 | 16760 |
| Gap with SDDP | $-2.0\%$ | $-3.0\%$ | $-2.2\%$ | $-1.8\%$ |

Table: Results obtained by DADP "Expectation"

[1] SDDP method is an alternative to DP, pushing away the dimension barrier.

# Optimal values and computational times

| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DP CPU time | *1.6 $10^3$'* | $\sim$ *$10^8$'* | $\sim \infty$ | $\sim \infty$ |
| DP value | 3743 | N.A. | N.A. | N.A. |
| SDDP value | 3742 | 7026 | 11834 | 17069 |
| SDDP CPU time | *5'* | *7'* | *9'* | *50'* |
| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |

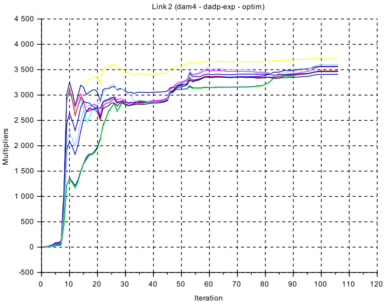
Table: Results obtained by DP and SDDP[1]

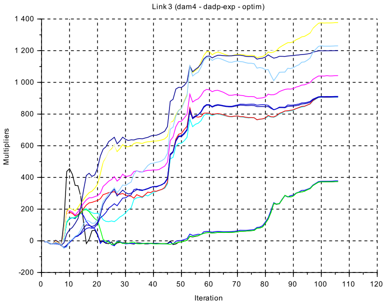
| Valley | **4-Dams** | **6-Dams** | **8-Dams** | **10-Dams** |
|---|---|---|---|---|
| DADP CPU time | *7'* | *12'* | *17'* | *24'* |
| DADP value | 3667 | 6816 | 11573 | 16760 |
| Gap with SDDP | **−2.0%** | **−3.0%** | **−2.2%** | **−1.8%** |

Table: Results obtained by DADP "Expectation"

[1]SDDP method is an alternative to DP, pushing away the dimension barrier.

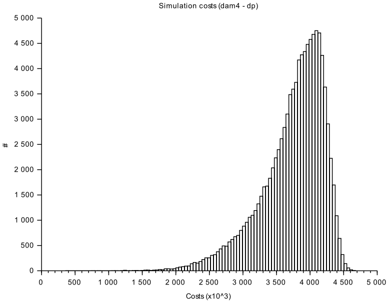
# 4-Dams valley in detail: DADP convergence



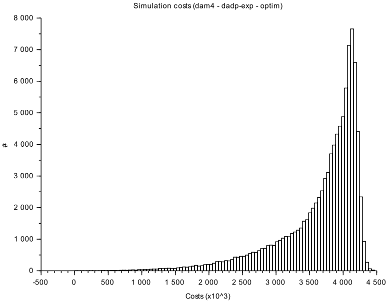




Multipliers convergence (dam1↔dam2 and dam2↔dam3)

# 4-Dams valley in detail: payoff distributions

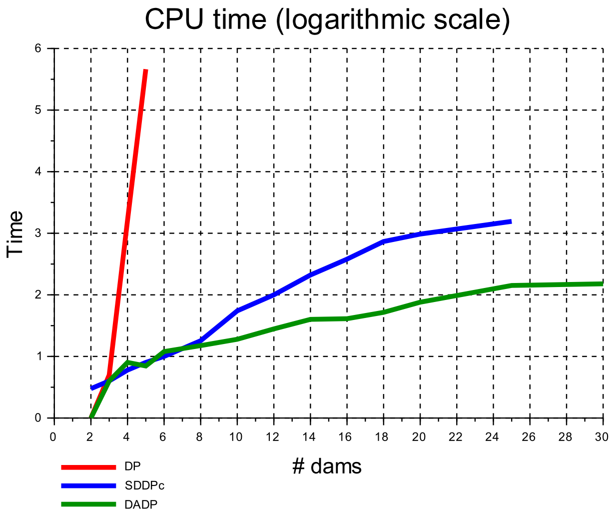



DP payoff distribution

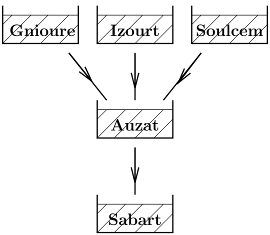



DADP payoff distribution

# CPU time summary



# Two “true” valleys

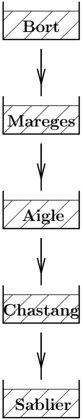



*Discretization*

$T \rightsquigarrow 12$, $\mathbf{W} \rightsquigarrow 10$

realistic grids for $\mathbf{U}$ and $\mathbf{X}$

Vicdessos




Dordogne

# Results

| Valley | **Vicdessos** | **Dordogne** |
|---|---|---|
| SDDP CPU time | *10*' | *17*' |
| SDDP value | 2244 | 22136 |

Table: Results obtained by SDDP

| Valley | **Vicdessos** | **Dordogne** |
|---|---|---|
| DADP CPU time | *10*' | *155*' |
| DADP value | 2237 | 21499 |
| Gap with $\mathrm{SDDP_d}$ | $-0.3\%$ | $-2.8\%$ |

Table: Results obtained by DADP "Expectation"

# Results

| Valley | **Vicdessos** | **Dordogne** |
|---|---|---|
| SDDP CPU time | *10*' | *17*' |
| SDDP value | 2244 | 22136 |

Table: Results obtained by SDDP

| Valley | **Vicdessos** | **Dordogne** |
|---|---|---|
| DADP CPU time | *10*' | *155*' |
| DADP value | 2237 | 21499 |
| Gap with $\mathrm{SDDP_d}$ | $-$**0.3%** | $-$**2.8%** |

Table: Results obtained by DADP "Expectation"

# Conclusions and perspectives

**Conclusions for DADP**

- Fast numerical convergence of the 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 (smart grids).
- Connection with other decomposition methods.
- Theoretical study.

P. Carpentier et G. Cohen.
*Décomposition-coordination en optimisation déterministe et stochastique.*
En préparation, Springer, 2016.

P. Girardeau.
*Résolution de grands problèmes en optimisation stochastique dynamique.*
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.

K. Barty, P. Carpentier, G. Cohen and P. Girardeau,
*Price decomposition in large-scale stochastic optimal control.*
*arXiv*, math.OC 1012.2092, 2010.