Dual Approximate Dynamic Programming for Large Scale Hydro Valleys

Pierre Carpentier and Jean-Philippe Chancelier [1]

ENSTA ParisTech and ENPC ParisTech

CMM Workshop, January 2016

ENSTA ParisTech université PARIS-SACLAY

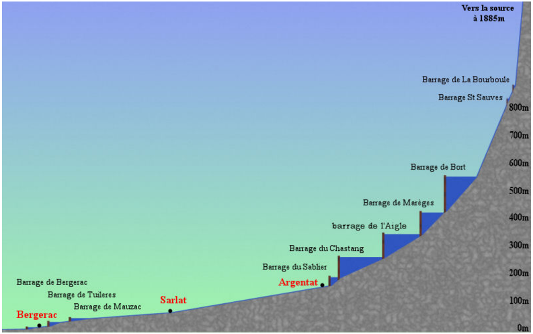
CMM Center for Mathematical Modeling

École des Ponts ParisTech

[1] Joint work with J.-C. Alais,
supported by the FMJH Program Gaspard Monge for Optimization.

Motivation

Electricity production management for hydro valleys



- *1 year time horizon*: compute each month the Bellman functions ("water values")
- *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

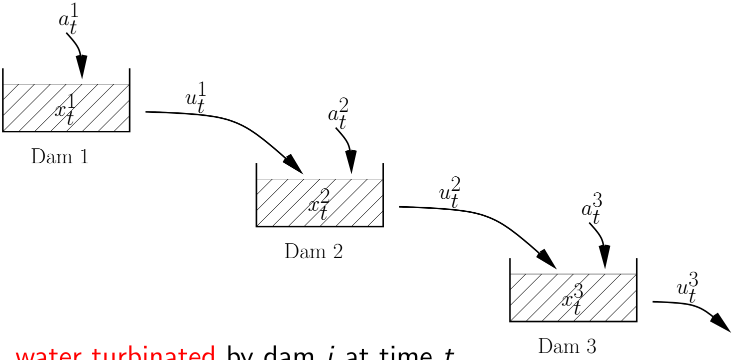
This talk: present numerical results for large-scale hydro valleys using DADP, and comparison with DP and SDDP.

Lecture outline

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

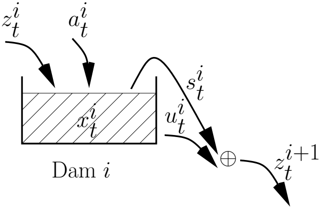
Operating scheme



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

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

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

Stochastic optimization problem

The global optimization problem reads:

$$\max_{(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Z})} \mathbb{E}\left(\sum_{i=1}^{N}\left(\sum_{t=0}^{T-1} L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_t^i, \boldsymbol{Z}_t^i) + K^i(\boldsymbol{X}_T^i)\right)\right),$$

subject to:

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

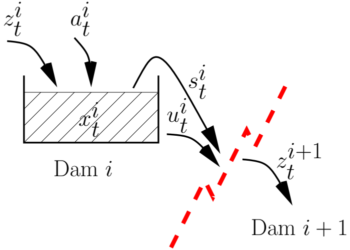
$$\sigma\big(\boldsymbol{U}_t^i\big) \subset \sigma\big(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t\big) \ , \qquad \forall i \ , \ \ \forall t \ ,$$

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

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

Price decomposition

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



- At iteration k, the duality term:

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

  is the difference of two terms:

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

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

DADP core idea

The i-th subproblem writes:

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

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

The core idea of DADP is

- to replace the constraint $\boldsymbol{Z}_t^{i+1} - g_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_t^i, \boldsymbol{Z}_t^i) = 0$ by its conditional expectation with respect to $\boldsymbol{Y}_t^i$:

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

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

DADP core idea

The i-th subproblem writes:

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

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

The core idea of DADP is

- to replace the constraint $\boldsymbol{Z}_t^{i+1} - g_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_t^i, \boldsymbol{Z}_t^i) = 0$ by its conditional expectation with respect to $\boldsymbol{Y}_t^i$:

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

- where $(\boldsymbol{Y}_0^i, \ldots, \boldsymbol{Y}_{T-1}^i)$ 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 $\mathbb{E}\big(\boldsymbol{\Lambda}_t^{i,(k)} \mid \boldsymbol{Y}_t^{i-1}\big)$.

The expression of the i-th subproblem becomes:

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

If the process $\boldsymbol{Y}^{i-1}$ follows a dynamical equation, DP applies.

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

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:

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

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

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

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 $(x^1, \ldots, x^N)$ in order to design the decisions by using one-step Dynamic Programming.

Heuristic proposal:

$$V_t(x^1, \ldots, x^N) = \sum_{i=1}^{N} V_t^i(x^i) ,$$

the one-step DP problem to solve at time t being:

$$\max_{(u^1, \ldots, u^N)} \quad \sum_{i=1}^{N} L_t^i(x^i, u^i, w_t^i, z^i) + V_{t+1}^i(x_{t+1}^i) ,$$

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

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 by using one-step Dynamic Programming.

Heuristic proposal:

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

the one-step DP problem to solve at time t being:

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

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

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 by using one-step Dynamic Programming.

Heuristic proposal:

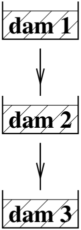
$$V_t(x^1, \dots, x^N) = \sum_{i=1}^{N} V_t^i(x^i) \ ,$$

the one-step DP problem to solve at time t being:

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

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

Three case studies



Discretization

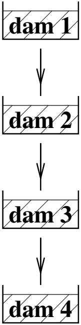
$T \rightsquigarrow 12$

$\boldsymbol{X} \rightsquigarrow 41$

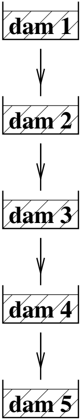
$\boldsymbol{U} \rightsquigarrow 6$

$\boldsymbol{W} \rightsquigarrow 10$

"3Dams" Valley



"4Dams" Valley



"5Dams" Valley

Results

Valley	**3Dams**	**4Dams**	**5Dams**
DP CPU time	*5*'	*1630*'	*677000*'
DP value	2482.0	3742.7	4685.1

Table: Results obtained by DP

Valley	**3Dams**	**4Dams**	**5Dams**
DADP CPU time	*3*'	*6*'	*5*'
DADP value	2401.3	3667.0	4633.7
Gap with DP	−**3.2%**	−**2.0%**	−**1.1%**
Dual value	*2687.5*	*3995.8*	*4885.9*

Table: Results obtained by DADP "Expectation"

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

Results

Valley	**3Dams**	**4Dams**	**5Dams**
DP CPU time	*5'*	*1630'*	*677000'*
DP value	2482.0	3742.7	4685.1
$\mathrm{SDDP_d}$ value	2474.2	3736.4	4672.2
$\mathrm{SDDP_d}$ CPU time	*0.3'*	*2'*	*16'*

Table: Results obtained by DP and $\mathrm{SDDP_d}$

Valley	**3Dams**	**4Dams**	**5Dams**
DADP CPU time	*3'*	*6'*	*5'*
DADP value	2401.3	3667.0	4633.7
Gap with DP	-3.2%	-2.0%	-1.1%
Dual value	*2687.5*	*3995.8*	*4885.9*

Table: Results obtained by DADP "Expectation"

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

Results

Valley	**3Dams**	**4Dams**	**5Dams**
DP CPU time	*5'*	*1630'*	*677000'*
DP value	2482.0	3742.7	4685.1
$\mathrm{SDDP_d}$ value	2474.2	3736.4	4672.2
$\mathrm{SDDP_d}$ CPU time	*0.3'*	*2'*	*16'*

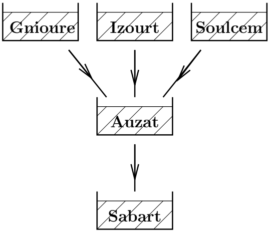
Table: Results obtained by DP and $\mathrm{SDDP_d}$

Valley	**3Dams**	**4Dams**	**5Dams**
DADP CPU time	*3'*	*6'*	*5'*
DADP value	2401.3	3667.0	4633.7
Gap with DP	$-\mathbf{3.2\%}$	$-\mathbf{2.0\%}$	$-\mathbf{1.1\%}$
Dual value	*2687.5*	*3995.8*	*4885.9*

Table: Results obtained by DADP "Expectation"

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

Three valleys

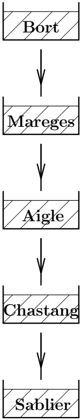


Discretization

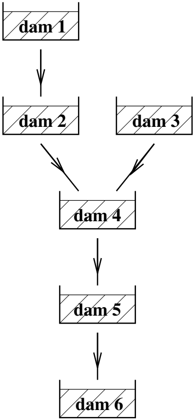
$\boldsymbol{T} \rightsquigarrow 12$, $\boldsymbol{W} \rightsquigarrow 10$

fine grids for $\boldsymbol{X}$ and $\boldsymbol{U}$

Vicdessos Valley



Dordogne Valley



Stoopt Valley

Results

Valley	**Vicdessos**	**Dordogne**	**Stoopt**
$\mathrm{SDDP_d}$ CPU time	*90*'	*40000*'	*320*'
$\mathrm{SDDP_d}$ value	2232.1	22028.9	7014.8

Table: Results obtained by $\mathrm{SDDP_d}$

Valley	**Vicdessos**	**Dordogne**	**Stoopt**
DADP CPU time	*10*'	*155*'	*13*'
DADP value	2237.4	21499.8	6816.5
Gap with $\mathrm{SDDP_d}$	$+0.2\%$	-2.4%	-2.8%
Dual value	*2285.6*	*22991.1*	*7521.9*

Table: Results obtained by DADP "Expectation"

Results

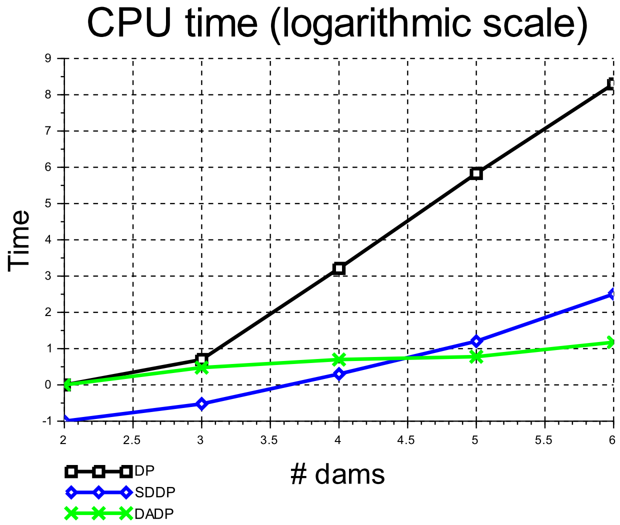
Valley	**Vicdessos**	**Dordogne**	**Stoopt**
$\mathrm{SDDP_d}$ CPU time	*90'*	*40000'*	*320'*
$\mathrm{SDDP_d}$ value	2232.1	22028.9	7014.8

Table: Results obtained by $\mathrm{SDDP_d}$

Valley	**Vicdessos**	**Dordogne**	**Stoopt**
DADP CPU time	*10'*	*155'*	*13'*
DADP value	2237.4	21499.8	6816.5
Gap with $\mathrm{SDDP_d}$	$+$**0.2%**	$-$**2.4%**	$-$**2.8%**
Dual value	*2285.6*	*22991.1*	*7521.9*

Table: Results obtained by DADP “Expectation”

CPU time comparison



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.