



Stochastic decomposition applied to large-scale hydro valleys management

Pierre Carpentier, Jean-Philippe Chancelier, Vincent Leclère, François Pacaud

► **To cite this version:**

Pierre Carpentier, Jean-Philippe Chancelier, Vincent Leclère, François Pacaud. Stochastic decomposition applied to large-scale hydro valleys management. *European Journal of Operational Research*, Elsevier, 2018, 10.1016/j.ejor.2018.05.025 . hal-01526775v2

HAL Id: hal-01526775

<https://hal-enpc.archives-ouvertes.fr/hal-01526775v2>

Submitted on 25 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Stochastic decomposition applied to large-scale hydro valleys management

P. Carpentier^a, J.-Ph. Chancelier^b, V. Leclère^{b,*}, F. Pacaud^{c,b}

^aUMA, ENSTA ParisTech, Université Paris-Saclay, 828 bd des Maréchaux, 91762 Palaiseau cedex France

^bUniversité Paris-Est, CERMICS (ENPC), F-77455 Marne-la-Vallée, FRANCE
^cEfficacity, 14-20 boulevard Newton, F-77455 Marne-la-Vallée, FRANCE

Abstract

We are interested in optimally controlling a discrete time dynamical system that can be influenced by exogenous uncertainties. This is generally called a Stochastic Optimal Control (SOC) problem and the Dynamic Programming (DP) principle is one of the standard ways of solving it. Unfortunately, DP faces the so-called curse of dimensionality: the complexity of solving DP equations grows exponentially with the dimension of the variable that is sufficient to take optimal decisions (the so-called state variable). For a large class of SOC problems, which includes important practical applications in energy management, we propose an original way of obtaining near optimal controls. The algorithm we introduce is based on Lagrangian relaxation, of which the application to decomposition is well-known in the deterministic framework. However, its application to such closed-loop problems is not straightforward and an additional statistical approximation concerning the dual process is needed. The resulting methodology is called Dual Approximate Dynamic Programming (DADP). We briefly present DADP, give interpretations and enlighten the error induced by the approximation. The paper is mainly devoted to applying DADP to the management of large hydro valleys. The modeling of such systems is presented, as well as the practical implementation of the methodology. Numerical results are provided on several valleys, and we compare our approach with the state of the art SDDP method.

Keywords: Discrete time stochastic optimal control, Decomposition methods, Dynamic programming, Energy management

2010 MSC: 90-08, 90C06, 90C15, 90C39, 49M27

*Corresponding author

Email addresses: pierre.carpentier@ensta-paristech.fr (P. Carpentier), jpc@cermics.enpc.fr (J.-Ph. Chancelier), vincent.leclere@enpc.fr (V. Leclère), f.pacaud@efficacity.com (F. Pacaud)

1. Introduction

1.1. Large-scale systems and energy applications

Consider a controlled dynamical system over a discrete and finite time horizon. This system may be influenced by exogenous noises that affect its behavior. Assume that, at every instant t , the decision maker designs a control based on all the observations of noises available up to time t . We are thus looking for strategies (or policies), that is, feedback functions that map every instant and every possible history of the system to a decision to be made.

We can find typical applications in the field of energy management. Consider a power producer that owns a certain number of power units. Each unit has its own local characteristics such as physical constraints that restrain the set of feasible decisions, and induces a production cost or a revenue. The power producer control the power units so that an overall goal is met. A classical example is the so-called unit commitment problem (see Takriti et al. (1996)) where the producer has to satisfy a global power demand at every instant. The power demand, as well as other parameters such as unit breakdowns, are random. The producer is looking for strategies that minimize the overall expected production cost, over a given time horizon. Another application, which is considered in this paper, is the management of a large-scale hydro valley: here the power producer manages a cascade of dams, and maximizes the revenue obtained by selling the energy produced by turbinating the water inside the dams. Both natural inflows in water reservoirs and energy prices are random. In all these problems, the number of power units and the number of time steps are usually large (see de Matos et al. (2015)).

1.2. Standard resolution methods

One classical approach when dealing with stochastic dynamic optimization problems is to discretize the random inputs of the problem using a scenario tree. Such an approach has been widely studied within the stochastic programming community (see Heitsch & Römisch (2009), Shapiro et al. (2009)), and used to model and solve energy problems, e.g. by Pflug & Pichler (2014). One of the advantages of such a technique is that, as soon as the scenario tree is drawn, the derived problem can be treated by classical mathematical programming techniques. Thus, a number of decomposition methodologies have been proposed (see for instance Rockafellar & Wets (1991), Carpentier et al. (1996), Ruszczyński (1997), (Ruszczyński & Shapiro, 2003, Chap. 3) and applied to energy planning problems (see Baccard et al. (2001)). Ways to combine the discretization of the expected value together with the discretization of information in a general setting have been presented in Heitsch et al. (2006), Pflug & Pichler (2014) and Carpentier et al. (2015)). However, in a multi-stage setting, this methodology suffers from the drawback that arises with scenario trees: as it was pointed out by Shapiro (2006), the number of scenarios needed to achieve a given accuracy grows exponentially with the number of time steps of the problem.

The other natural approach to solve SOC problems is to rely on the Dynamic Programming (DP) principle (see Bellman (1957), Puterman (1994)). The core

of the DP approach is the definition of a state variable that is, roughly speaking, the variable that, in conjunction with the time variable, is sufficient to take an optimal decision at every instant. It does not have the drawback of the scenario trees concerning the number of time steps since strategies are, in this context, depending on a state variable whose space dimension does not grow with time (usually linked to the number of power units in the case of power management). However, DP suffers from another drawback which is the so-called *curse of dimensionality*: the complexity of solving the DP equation grows exponentially with the state space dimension. Hence, solving the DP equation by brute force is generally intractable when the state space dimension goes beyond several units. In Vezolle et al. (2009), the authors were able to solve DP on a 10 state variables energy management problem, using parallel computation coupled with adequate data distribution, but the DP limits are around 5 state variables in a straightforward use of the method.

Another popular idea is to represent the value functions (solutions of the DP equation) as a linear combination of a priori chosen basis functions (see Bertsekas & Tsitsiklis (1996)). This approach, called Approximate Dynamic Programming (ADP) has become very popular and the reader is referred to Powell (2011) and Bertsekas (2012) for a precise description of ADP. This approximation drastically reduces the complexity of solving the DP equation. However, in order to be practically efficient, such an approach requires some a priori information about the problem, in order to define a well suited functional subspace. Indeed, there is no systematic means to choose the basis functions and several choices have been proposed in the literature (see Tsitsiklis & Van Roy (1996)).

Last but not least is the popular DP-based method called Stochastic Dual Dynamic Programming (SDDP). Starting with the seminal work of Van Slyke & Wets (1969), the SDDP method has been designed in Pereira & Pinto (1991). It has been widely used in the energy management context and lately regained interest in the Stochastic Programming community (see Shapiro (2011) and references therein). The idea is to extend Kelley's cutting plane method to the case of multi-stage stochastic problems. Alternatively it can be seen as a multistage Benders (or L-shaped) decomposition method with sampling. It consists of a succession of forward (trajectory computation) and backward (Bellman function refining) passes that ultimately aims at approaching the Bellman function as the supremum of affine hyperplanes (cuts) generated during the backward passes.

1.3. Decomposition approach

When dealing with large-scale optimization problems, the decomposition-coordination approach aims at finding a solution to the original problem by iteratively solving subproblems of smaller dimension. In the deterministic case, several types of decomposition have been proposed (e.g. by prices, by quantities or by interaction prediction) and unified in Cohen (1980) using a general framework called Auxiliary Problem Principle. In the open-loop stochastic case, i.e. when controls do not rely on any observation, it is proposed in Cohen & Culioli

(1990) to take advantage of both decomposition techniques and stochastic gradient algorithms. The natural extension of these techniques to the closed-loop stochastic case (see Barty et al. (2009)), i.e. when the control is a function of the available observations, fails to provide decomposed state dependent strategies. Indeed, the optimal strategy of a subproblem depends on the state of the whole system, and not only on the local state.

We recently proposed a way to use price decomposition within the closed-loop stochastic case. The coupling constraints, namely the constraints preventing the problem from being naturally decomposed, are dualized using a Lagrange multiplier (price). At each iteration, the price decomposition algorithm solves each subproblem using the current price, and then uses the solutions to update it. In the stochastic context, the price is a random process whose dynamics is not available, so the subproblems do not in general fall into the Markovian setting. However, in a specific instance of this problem (see Strugarek (2006)), the author exhibited a dynamics for the optimal multiplier and showed that these dynamics were independent from the decision variables. Hence it was possible to come down to the Markovian framework and use DP to solve the subproblems. Following this idea, it is proposed in Barty et al. (2010) to choose a parameterized dynamics for these multipliers in such a way that solving subproblems using DP becomes possible. While the approach, called Dual Approximate Dynamic Programming (DADP), showed promising results on numerical examples, it suffered from the fact that the induced restrained dual space is non-convex, leading to some numerical instabilities. Moreover, it was not possible to give convergence results for the algorithm. The method has then been improved both from the theoretical and from the practical point of view. The core idea is to replace the current Lagrange multiplier by its conditional expectation with respect to some *information process*, at every iteration. This information process has to be a priori chosen and adapted to the natural filtration. Moreover, if the information process is driven by a dynamic, the state in each subproblem then consists of the original state augmented by the information process, making the resolution of the subproblem tractable by DP. Interestingly, approximating the multipliers by their conditional expectations is equivalent to solving a relaxed primal problem where the almost-sure coupling constraint has been replaced by its conditional expectation with respect to the information variable, yielding a lower bound of the true optimal cost. Further, the solutions obtained by the DADP algorithm do not necessarily satisfy the initial almost-sure coupling constraint, so we must rely on a heuristic procedure to produce a feasible solution to the original problem.

1.4. Contents of the paper

The main contribution of the paper is to give a practical algorithm aiming at solving large scale stochastic optimal control problems and providing closed-loop strategies. The numerous approximations used in the algorithm, and especially the one allowing for feasible strategies, make difficult to theoretically assess the quality of the solution finally adopted. Nevertheless, numerical implementation

shows that the method is promising to solve large scale stochastic optimization problems such as those encountered in the field of energy management.

The paper is organized as follows. In §2, we present the hydro valley management problem, the corresponding general SOC formulation and the DP principle. We then focus on spatial decomposition of such a problem and the difficulties of using DP at the subproblem level. In §3, we present the DADP method and give different interpretations. We then propose a way to recover an admissible solution from the DADP results and we briefly discuss the theoretical and practical questions associated to the convergence and implementation of the method. Finally, in §4, we apply the DADP method to the management of hydro valleys. Different examples, corresponding to either academic or realistic valleys, are described. A comparison of the method with SDDP is outlined.

1.5. Notations

We will use the following notations, considering a probability space $(\Omega, \mathcal{A}, \mathbb{P})$:

- $\llbracket i, j \rrbracket$ is the set of integers between i and j ;
- bold letters are used for random variables, normal font for their realizations;
- $\mathbf{X} \preceq \mathcal{F}_t$ (resp. $\mathbf{X} \preceq \mathbf{Y}$) means that the random variable \mathbf{X} is measurable with respect to the σ -algebra \mathcal{F}_t (resp. with respect to the σ -algebra generated by \mathbf{Y} , denoted by $\sigma(\mathbf{Y})$);
- x generally stands for the state, u for the control, w for an exogeneous noise;
- f_t stands for a dynamics, that is, a transition function modeling the system evolution along time, L_t stands for a cost function at time t , K stands for a final cost function;
- V_t represents a Bellman's value function at time t ;
- the notation \mathbf{X}^i (resp. \mathbf{U}^i and \mathbf{Z}^i) stands for the discrete time state process $(\mathbf{X}_0^i, \dots, \mathbf{X}_T^i)$ (resp. the two control processes $(\mathbf{U}_0^i, \dots, \mathbf{U}_{T-1}^i)$ and $(\mathbf{Z}_0^i, \dots, \mathbf{Z}_{T-1}^i)$).

2. Mathematical formulation

In this section, we present the modeling of a hydro valley and the associated optimization framework.

2.1. A generic formulation

We are interested in solving a multistage stochastic optimal control problem over a discrete-time horizon $\llbracket 0, T \rrbracket$. In this problem we consider multiple stochastic systems indexed by $i \in \llbracket 1, N \rrbracket$, that follow independent dynamics but that must satisfy a coupling constraint.

More precisely, we want to address the following problem

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

$$\text{s.t. } \mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t), \quad \mathbf{X}_0^i \text{ given}, \quad (1b)$$

$$\mathbf{U}_t^i \preceq \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t), \quad (1c)$$

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

Constraints (1b) represent the dynamics and constraints (1c) are the non-anticipativity constraints, that is, the fact that each control \mathbf{U}_t^i at time t , considered as a random variable, has to be *measurable* with respect to the sigma-field $\sigma(\mathbf{W}_0, \dots, \mathbf{W}_t)$ generated by noises up to time t . The last constraints (1d) express the interactions between the production units i . They represent an *additive* coupling with respect to the different production units, which is termed the “spatial coupling of the problem”. Such a general modeling covers other cases than the cascade problem, such that the unit commitment problem, or the problem of exchanging energy on a smart grid.

2.2. Dams management problem

We consider a hydro valley constituted of N cascaded dams as represented in Figure 1. The water turbinated at a dam produces energy which is sold on electricity markets, and then enters the nearest downstream dam. The overall goal of the decision maker is to maximize the profit obtained by selling the produced energy on a market. We consider that the hydro valley manager acts as a price follower, in the sense that energy prices are independent of energy produced by the hydro valley. Note that the valley geometry may be more complicated than a pure cascade: see for example the valleys represented at Figure 4.

The representative variables of dam i at time t are u_t^i for the turbinated water, x_t^i for the current water volume, a_t^i for the natural water inflow entering dam i , p_t^i for the market value of the water at dam i . The randomness is given by $w_t^i = (a_t^i, p_t^i)$. The modeling of a dam takes into account a possible overflow: the spilled water does not produce electricity, but enters the next downstream dam.

We now cast the problem in the generic framework presented at §2.1, with a slight abuse of notation (\mathbf{U}_t^i stands for $(\mathbf{U}_t^i, \mathbf{Z}_t^i)$ here).

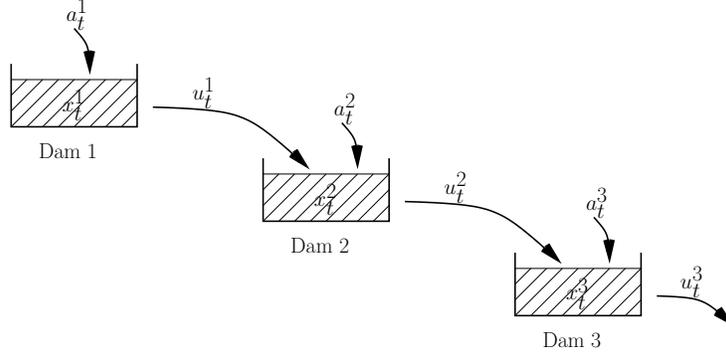


Figure 1: Operating scheme of a hydro valley with 3 dams.

- The dam dynamics (corresponding to Equation (1b)) reads

$$x_{t+1}^i = x_t^i - u_t^i + a_t^i + z_t^i - s_t^i = f_t^i(x_t^i, (u_t^i, z_t^i), w_t^i), \quad (2a)$$

where s_t^i is the volume of water spilled by overflowing the dam:

$$s_t^i = \max \{0, x_t^i - u_t^i + a_t^i + z_t^i - \bar{x}^i\}. \quad (2b)$$

The constant value \bar{x}^i stands for the maximal capacity of dam i . The outflow of dam i , that is, the sum of the turbinated water and of the spilled water, is denoted by z_t^{i+1} :

$$z_t^{i+1} = u_t^i + s_t^i = g_t^i(x_t^i, (u_t^i, z_t^i), w_t^i). \quad (2c)$$

This last equation corresponds to Equation (1d) in the general framework. Note that the dynamic equations (2a) are nonlinear because of the max operator in the definition (2b) of the spilled water volume. We assume the *Hazard-Decision* information structure: the control u_t^i applied at time t is chosen once the noise w_t^i at time t has been observed. It is thus possible to ensure that the dam always remains above its minimal admissible volume \underline{x}^i by limiting the control range: $\underline{u}^i \leq u_t^i \leq \min \{\bar{u}^i, x_t^i + a_t^i + z_t^i - \underline{x}^i\}$.

Remark 1. As will be seen in §4, the typical time step length we use is the month (with a time horizon of one year). It is thus reasonable to assume the Hazard-Decision framework, the control applied for a given month being in fact implemented each day taking into account the observed information on a daily basis.

- The objective function of dam i is the sum of different terms.
 - The cost at each time $t \in \llbracket 0, T - 1 \rrbracket$ is: $L_t^i(x_t^i, (u_t^i, z_t^i), w_t^i) = -p_t^i u_t^i + \varepsilon(u_t^i)^2$. The first linear term corresponds to the opposite of the profit when selling the energy produced by the turbinated water on

the energy market. The second term $\varepsilon(u_t^i)^2$ models the operating cost of the turbine as a quadratic term, and is usually small. This last term ensures the strong convexity of the cost function.

- The final cost at time T is: $K^i(x_T^i) = \alpha^i \min\{0, \widehat{x}^i - x_T^i\}^2$. It corresponds to a quadratic penalization around a target value \widehat{x}^i representing the desired water volume in the dam at the end of the time horizon.

Both functions appear in the cost (1a) in the generic problem formulation.

2.3. Dynamic Programming like approaches

In the remainder of the paper, we assume that we are in the so-called *white noise* setting.

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

This assumption can be alleviated, in the case where it is possible to identify a dynamics in the noise process (such as an ARMA model), and by incorporating this new dynamics in the state variables (see Maceira & Damazio (2006) on this topic).

Under Assumption 1, Dynamic Programming (DP) applies to Problem (1): there is no optimality loss to seek each control \mathbf{U}_t^i at time t as a function of both the state and the noise at time t . Then, the Bellman functions V_t are obtained by solving the Dynamic Programming equation backwards in time:

$$V_T(x_T) = \sum_{i=1}^N K^i(x_T^i), \quad (3a)$$

$$V_t(x_t) = \mathbb{E} \left(\min_{u_t^1, \dots, u_t^N} \sum_{i=1}^N L_t^i(x_t^i, u_t^i, \mathbf{W}_t) + V_{t+1}(f_t(x_t, u_t, \mathbf{W}_t)) \right). \quad (3b)$$

where $x_t = (x_t^1, \dots, x_t^N)$, $u_t = (u_t^1, \dots, u_t^N)$ and $f_t(x_t, u_t, \mathbf{W}_t)$ is the collection of new states $f_t^i(x_t^i, u_t^i, \mathbf{W}_t)$.

The DP equation is agnostic to whether the state and control variables are continuous or discrete, whether the constraints and the cost functions are convex or not, etc. However, in order to exhaustively solve the DP equation, we need to have discrete state, and to be able to solve each equation to optimality. In practice, the method is subject to the curse of dimensionality and cannot be used for large-scale optimization problems. For example, applying DP to dams management problems is practically untractable for more than five dams (see the results given at §4.3).

Another way to compute the Bellman functions associated to Problem (1) is to use the Stochastic Dual Dynamic Programming (SDDP) method. The method has been first described in Pereira & Pinto (1991), and its convergence has been analyzed in Philpott & Guan (2008) for the linear case and in Girardeau et al. (2015) for the general convex case. SDDP recursively constructs

an approximation of each Bellman function as the supremum of a number of affine functions, thus exploiting the convexity of the Bellman functions (arising from the convexity of the cost and constraint functions). SDDP has been used for a long time for solving large-scale hydrothermal problems (see de Matos et al. (2015) and the references therein) and allows to push the limits of DP in terms of state dimension (see the results given at §4.4).

2.4. Spatial coupling and approach by duality

A standard way to tackle large-scale optimization problems is to use Lagrange relaxation in order to split the original problem into a collection of smaller subproblems by dualizing coupling constraints. As far as Problem (1) is concerned, we have in mind to use DP for solving the subproblems and thus want to dualize the spatial coupling constraints (1d) in order to formulate subproblems, each incorporating a single dam. The associated Lagrangian \mathcal{L} is accordingly

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

where the multiplier $\boldsymbol{\lambda}_t$ associated to Constraint (1d) is a random variable. From the measurability of the variables \mathbf{X}_t^i , \mathbf{U}_t^i and \mathbf{W}_t , we can assume without loss of optimality that the multipliers $\boldsymbol{\lambda}_t$ are $\sigma(\mathbf{W}_0, \dots, \mathbf{W}_t)$ -measurable random variables.

In order to be able to apply duality theory to the problem (which is mandatory for algorithmic resolution), we make the two following assumptions.

Assumption 2. A saddle point of the Lagrangian \mathcal{L} exists.

Assumption 3. The Uzawa algorithm applies to compute a saddle-point of \mathcal{L} (see (Ekeland & Temam, 1999, Chap. VII) for a complete presentation).

Assumption 2 corresponds to a Constraint Qualification condition and ensures the existence of an optimal multiplier. Assumption 3 allows to use a (dual) gradient ascent algorithm to compute the optimal multiplier. An important question in order to be able to satisfy these two assumptions is the choice of the spaces where the various random variables of the problem are living in. Duality theory and associated algorithms have been extensively studied in the framework of Hilbert spaces (see Ekeland & Temam (1999)), but the transition to the framework of stochastic optimal control poses difficult challenges (Rockafellar (1968, 1971)), which will be briefly presented at §3.4. One way to get rid of these difficulties is to assume that the space Ω is finite, assumption also

required for convergence of SDDP.¹

When using the Uzawa algorithm to compute a saddle-point of the Lagrangian, the minimization step with respect to $(\mathbf{X}^i, \mathbf{U}^i)_{i \in \llbracket 1, N \rrbracket}$ splits in N independent subproblems each depending on a single pair $(\mathbf{X}^i, \mathbf{U}^i)$, and therefore allows for a dam by dam decomposition. More precisely, the k -th iteration of the Uzawa algorithm consists of the two following steps.

1. Solve Subproblem i , $i \in \llbracket 1, N \rrbracket$, with fixed $\boldsymbol{\lambda}^{(k)}$:

$$\min_{\mathbf{X}^i, \mathbf{U}^i} \mathbb{E} \left[\sum_{t=0}^{T-1} L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) + \boldsymbol{\lambda}_t^{(k)} \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) + K^i(\mathbf{X}_T^i) \right] \quad (4a)$$

$$s.t. \quad \mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t), \mathbf{X}_0^i \text{ given} \quad (4b)$$

$$\mathbf{U}_t^i \preceq \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t), \quad (4c)$$

whose solution is denoted $(\mathbf{U}^{i,(k)}, \mathbf{X}^{i,(k)})$.

2. Use a gradient step to update the multipliers $\boldsymbol{\lambda}_t$:

$$\boldsymbol{\lambda}_t^{(k+1)} = \boldsymbol{\lambda}_t^{(k)} + \rho_t \left(\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t) \right). \quad (5)$$

Note that even if Subproblem (4) only involves the “physical” state variable \mathbf{X}_t^i and the control variable \mathbf{U}_t^i , a situation which seems favorable to DP, it also involves two exogenous random processes, namely \mathbf{W} and $\boldsymbol{\lambda}^{(k)}$. The white noise Assumption 1 applies for the first process \mathbf{W} , but not for the second one $\boldsymbol{\lambda}^{(k)}$, so that the state of the system cannot be summarized by the physical state \mathbf{X}_t^i ! Moreover if we just use the fact that $\boldsymbol{\lambda}_t^{(k)}$ is measurable with respect to the past noises, the state of the system must incorporate all noises prior to time t , that is, $(\mathbf{W}_0, \dots, \mathbf{W}_t)$. The state size of the subproblem increases with time. Without some additional knowledge on the process $\boldsymbol{\lambda}^{(k)}$, DP cannot be applied in a straightforward manner: something has to be compressed in order to use Dynamic Programming.

3. Dual Approximate Dynamic Programming

In Strugarek (2006), for a very specific instance of Problem (1), the author exhibited the dynamics of the optimal multiplier of the coupling constraint (1d). Hence it was possible to come down to the Markovian framework and to use DP to solve the subproblems (4) with an augmented space, namely the “physical” state \mathbf{X}_t^i and the state associated to the multiplier’s dynamics. Following this idea for a general Problem (1), Barty et al. (2010) proposed to choose a parameterized dynamics for the multiplier: then solving the subproblems using DP

¹Recall that the aim of the present paper is mainly to present numerical results. The reader is referred to Leclère (2014) for these difficult theoretical questions.

became possible, the parameters defining the multiplier dynamics being updated at each iteration of the Uzawa algorithm. This new approach, called Dual Approximate Dynamic Programming (DADP), has then largely improved through a series of PhD theses (Girardeau (2010), Alais (2013) and Leclère (2014)) both from the theoretical and from the practical point of view. We give here a brief overview of the current DADP method.

3.1. DADP core idea and associated algorithm

In order to overcome the obstacle explained at §2.4 concerning the measurability of random variables $\lambda_t^{(k)}$, we choose a random variable \mathbf{Y}_t at each time t , each \mathbf{Y}_t being measurable with respect to the noises $(\mathbf{W}_0, \dots, \mathbf{W}_t)$ up to time t . We call $\mathbf{Y} = (\mathbf{Y}_0, \dots, \mathbf{Y}_{T-1})$ the *information process* associated to Problem (1).

3.1.1. Method foundation

The core idea of DADP is to replace the multiplier $\lambda_t^{(k)}$ by its conditional expectation $\mathbb{E}[\lambda_t^{(k)} | \mathbf{Y}_t]$ with respect to \mathbf{Y}_t . From an intuitive point of view, the resulting optimization problem will be a good approximation of the original one if \mathbf{Y}_t is close to the random variable $\lambda_t^{(k)}$. Note that we require that the information process is not influenced by controls because introducing a dependency of the conditioning term with respect to the control would lead to very serious difficulties for optimization.

Using this core idea, we replace Subproblem (4) by:

$$\min_{\mathbf{X}^i, \mathbf{U}^i} \quad \mathbb{E} \left[\sum_{t=0}^{T-1} \left(L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) + K^i(\mathbf{X}_T^i) \right) + \mathbb{E}[\lambda_t^{(k)} | \mathbf{Y}_t] \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) \right], \quad (6a)$$

$$\text{s.t.} \quad \mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t), \quad \mathbf{X}_0^i \text{ given}, \quad (6b)$$

$$\mathbf{U}_t^i \preceq \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t). \quad (6c)$$

According to the Doob property (Dellacherie & Meyer, 1975, Chapter 1, p. 18), the \mathbf{Y}_t -measurable random variable $\mathbb{E}[\lambda_t^{(k)} | \mathbf{Y}_t]$ can be represented by a measurable mapping $\mu_t^{(k)}$, that is,

$$\mu_t^{(k)}(y) = \mathbb{E}[\lambda_t^{(k)} | \mathbf{Y}_t = y], \quad (7)$$

so that Subproblem (6) in fact involves the two fixed random processes \mathbf{W} and \mathbf{Y} . If the process \mathbf{Y} follows a non-controlled Markovian dynamics driven by the noise process \mathbf{W} , i.e. if there exist functions h_t such that $\mathbf{Y}_{t+1} = h_t(\mathbf{Y}_t, \mathbf{W}_t)$ then $(\mathbf{X}_t^i, \mathbf{Y}_t)$ is a valid state for the subproblem and DP applies.

3.1.2. DADP algorithm

Assume that the information process \mathbf{Y} follows the dynamics $\mathbf{Y}_{t+1} = h_t(\mathbf{Y}_t, \mathbf{W}_t)$.

- The first step of the DADP algorithm at iteration k consists of solving all the subproblems (6) with $\boldsymbol{\lambda}_t^{(k)}$ fixed, that is, with $\mu_t^{(k)}(\cdot)$ given. It is done by solving the Bellman functions associated to each subproblem i , that is,

$$\begin{aligned} V_T^{i,(k)}(x^i, y) &= K^i(x), \\ V_t^{i,(k)}(x^i, y) &= \mathbb{E}[Q_t^{i,(k)}(x^i, y, \mathbf{W}_t)], \end{aligned}$$

where $Q_t^{i,(k)}(x^i, y, w_t)$ is the value of

$$\begin{aligned} \min_{u^i} \quad & L_t^i(x^i, u^i, w_t) + \mu_t^{(k)}(y) \cdot \Theta_t^i(x^i, u^i, w_t) + V_{t+1}^{i,(k)}(x_{t+1}^i, y_{t+1}^i) \\ \text{s.t.} \quad & x_{t+1}^i = f_t^i(x^i, u^i, w_t), \\ & y_{t+1} = h_t(y, w_t). \end{aligned}$$

Storing the argmin obtained during the Bellman resolution, we obtain the optimal feedback laws $\gamma_t^{i,(k)}$ as functions of both the state (x^i, y) and the noise w_t at time t . These functions allow to compute the optimal state and control processes $(\mathbf{U}^{i,(k)}, \mathbf{X}^{i,(k)})$ of subproblem i at iteration k . Starting from $\mathbf{X}_0^{i,(k)} = \mathbf{X}_0^i$ the optimal control and state variables are obtained by applying the optimal feedback laws from $t = 0$ up to $T - 1$:

$$\begin{aligned} \mathbf{U}_t^{i,(k)} &= \gamma_t^{i,(k)}(\mathbf{X}_t^{i,(k)}, \mathbf{Y}_t, \mathbf{W}_t), \\ \mathbf{X}_{t+1}^{i,(k)} &= f_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t). \end{aligned}$$

- The second step of the DADP algorithm consists of updating the multiplier process $\boldsymbol{\lambda}^{(k)}$. Instead of updating the multipliers themselves by the standard gradient formula

$$\boldsymbol{\lambda}_t^{(k+1)} = \boldsymbol{\lambda}_t^{(k)} + \rho_t \left(\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t) \right), \quad (8)$$

it is sufficient to deal with their conditional expectations w.r.t. \mathbf{Y}_t . Using the optimal processes $\mathbf{X}^{i,(k)}$ and $\mathbf{U}^{i,(k)}$ obtained at the previous step of the algorithm for all subproblems, the *conditional* deviation from the coupling constraint is represented by a measurable mapping $\Delta_t^{(k)}$:

$$\Delta_t^{(k)}(y_t) = \mathbb{E} \left[\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t) \mid \mathbf{Y}_t = y_t \right]. \quad (9)$$

Gathering the functional representations (7) and (9) of the conditional multiplier and of the conditional deviation, the gradient update reduces to the following functional expression:

$$\mu_t^{(k+1)}(\cdot) = \mu_t^{(k)}(\cdot) + \rho_t \Delta_t^{(k)}(\cdot). \quad (10)$$

This last equation is equivalent to the multipliers conditional expectation update:

$$\mathbb{E}[\boldsymbol{\lambda}_t^{(k+1)} \mid \mathbf{Y}_t] = \mathbb{E}[\boldsymbol{\lambda}_t^{(k)} \mid \mathbf{Y}_t] + \rho_t \mathbb{E}\left[\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t) \mid \mathbf{Y}_t\right]. \quad (11)$$

From a practical point of view, computing the gradients using Formula (9), instead of (8) opens the way to important numerical improvements in the DADP algorithm. Indeed, instead of a gradient formula in a large space, we can use more sophisticated direction descent algorithms: as a matter of fact, if the support of the random variable \mathbf{Y}_t is finite, it becomes possible to efficiently implement a quasi-Newton method, thus obtaining a much faster convergence than the one of the standard gradient ascent method (see §4.3.2 for details).

DADP algorithm is depicted in Figure 2.

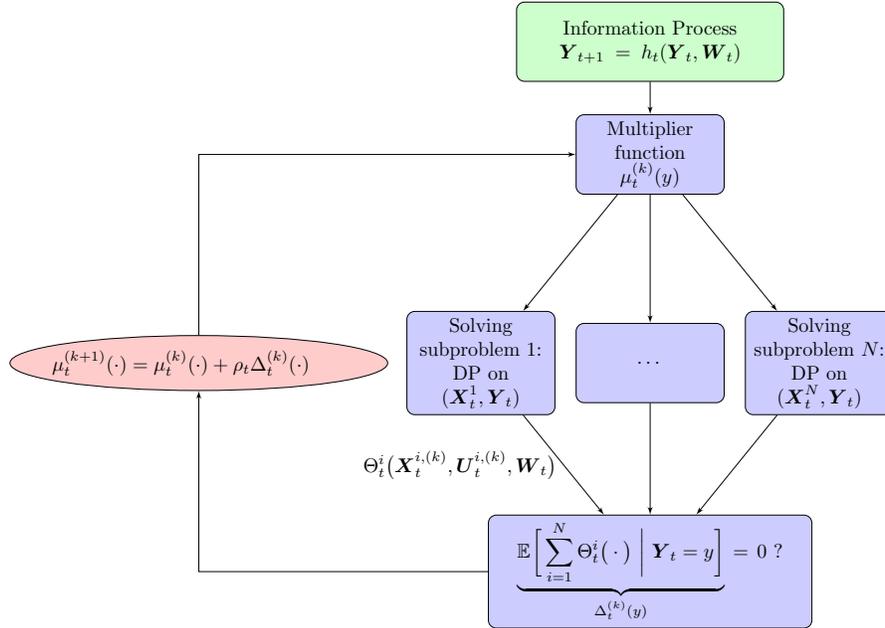


Figure 2: DADP flowchart.

3.2. DADP interpretations

The DADP method, as it has been presented up to now, makes use of an *approximation* of the optimal multiplier, that is, the multiplier $\boldsymbol{\lambda}_t$ is replaced by its conditional expectation $\mathbb{E}[\boldsymbol{\lambda}_t \mid \mathbf{Y}_t]$. Such an approximation is equivalent

to a *decision-rule* approach for the dual problem (see also Kuhn et al. (2011)), obtained by imposing that the dual variables λ_t is measurable with respect to \mathbf{Y}_t .

DADP may also be viewed as a *relaxation* of the constraints in the primal problem. More precisely, we replace the almost sure coupling constraint (1d) by the following conditional expectation constraint,

$$\mathbb{E} \left[\sum_{i=1}^N \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) \mid \mathbf{Y}_t \right] = 0. \quad (12)$$

Proposition 1. *Assume that the Lagrangian associated with this relaxed problem has a saddle point. Then the DADP algorithm on Problem (1) can be interpreted as the Uzawa algorithm applied to the relaxed Problem.*

Proof. Consider the duality term $\mathbb{E} [\mathbb{E}[\lambda_t^{(k)} \mid \mathbf{Y}_t] \cdot \Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t)]$ which appears in the cost function of subproblem i in DADP. This term can be written equivalently $\mathbb{E}[\lambda_t^{(k)} \cdot \mathbb{E}[\Theta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t) \mid \mathbf{Y}_t]]$, which corresponds to the dualization of the coupling constraint handled in the relaxed problem. \square

DADP thus consists of replacing an *almost-sure* constraint by its *conditional expectation* w.r.t. the information variable \mathbf{Y}_t . From this interpretation, we deduce that the optimal value provided by DADP is a guaranteed *lower bound* of the optimal value of Problem (1).

3.3. Admissibility recovery

Solving the relaxed problem, that is Problem (1), where constraints (1d) is replaced by the less binding constraints (12), does not necessarily yield a solution admissible for Problem (1). Nevertheless it produces at each time t a set of N local Bellman functions $V_t^{i,\infty}$, each depending on the extended state (x_t^i, y_t) . We use these functions to produce an *approximation* V_t^∞ of the “true” Bellman function V_t of the global state (x_t^1, \dots, x_t^N) by simply summing the local Bellman functions:

$$V_t^\infty(x_t^1, \dots, x_t^N, y_t) = \sum_{i=1}^N V_t^{i,\infty}(x_t^i, y_t).$$

We then obtain an admissible feedback policy for Problem (1): for any value of the state (x_t^1, \dots, x_t^N) , any value of the information y_t and any value of the noise w_t at time t , the control value is obtained by solving the following one-step

DP problem

$$\begin{aligned}
& \min_{(u_t^1, \dots, u_t^N)} \sum_{i=1}^N L_t^i(x_t^i, u_t^i, w_t^i) + V_{t+1}^\infty(x_{t+1}^1, \dots, x_{t+1}^N, y_{t+1}), \\
& \text{s.t. } x_{t+1}^i = f_t^i(x_t^i, u_t^i, w_t^i), \quad i \in \llbracket 1, N \rrbracket, \\
& \quad y_{t+1} = h_t(y_t^i, w_t^i), \\
& \quad \sum_{i=1}^N \Theta_t^i(x_t^i, u_t^i, w_t^i) = 0.
\end{aligned}$$

In this framework, DADP can be viewed as a tool allowing to compute approximated Bellman functions for Problem (1) which in turns yields an online admissible feedback policy for Problem (1).

Applying this online feedback policy along a bunch of noises scenarios allows to compute a Monte Carlo approximation of the cost, which is accordingly a stochastic *upper bound* of the optimal value of Problem (1).

3.4. Theoretical and practical questions

The theoretical questions linked to DADP are addressed in Leclère (2014), and the practical ones in Girardeau (2010) and Alais (2013).

3.4.1. Theoretical questions

In the DADP approach, we treat the coupling constraints of a stochastic optimization problem by duality methods and solve it using the Uzawa algorithm. The Uzawa algorithm is a dual ascent method which is usually described in an Hilbert space such as $L^2(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{R}^n)$, but we cannot guarantee the existence of an optimal multiplier in such a space. To overcome the difficulty, the approach consists of extending the setting to the non-reflexive Banach space $L^\infty(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{R}^n)$, to give conditions for the existence of an optimal multiplier in $L^1(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^n)$ (rather than in the dual space of L^∞) and to study the Uzawa algorithm convergence in this space.

3.4.2. Practical questions

An important practical question is the choice of the information variables \mathbf{Y}_t . We present here some possibilities.

1. *Perfect memory*: $\mathbf{Y}_t = (\mathbf{W}_0, \dots, \mathbf{W}_t)$.

From the measurability properties of $\lambda_t^{(k)}$, we have $\mathbb{E}[\lambda_t^{(k)} \mid \mathbf{Y}_t] = \lambda_t^{(k)}$, that is, there is no approximation! Indeed a valid state for each subproblem is $(\mathbf{X}_t, \mathbf{W}_0, \dots, \mathbf{W}_t)$: the state is growing with time.

2. *Minimal information*: $\mathbf{Y}_t = 0$.

Here $\lambda_t^{(k)}$ is approximated by its expectation $\mathbb{E}[\lambda_t^{(k)}]$. The information variable does not deliver any online information, and a valid state for subproblem i is \mathbf{X}_t^i .

3. *Dynamic information:* $\mathbf{Y}_{t+1} = h_t(\mathbf{Y}_t, \mathbf{W}_{t+1})$.

This choice corresponds to a number of possibilities, as mimicking the state of another unit, or adding a hidden dynamics. A valid state for subproblem i is $(\mathbf{X}_t^i, \mathbf{Y}_t)$.

The question of accelerating the DADP algorithm by using a more sophisticated method than the simple gradient ascent method in the multiplier update step has been discussed at the end of §3.1.2. Numerical experiments have shown that it has a great impact on the convergence speed of the method (see §4.3.2). Another improvement would be to replace the standard Lagrangian by an augmented Lagrangian.

4. Numerical experiments

In this section, we present numerical results obtained on a large selection of hydro valleys. Some of these valleys (see Figure 4) correspond to academic examples, in the sense that their characteristics (size of dams, range of controls, inflows values) do not rely on existing valleys. These examples allow us to quantify the performance of different optimization methods (DP, DADP and SDDP) on problems of increasing size, from a valley incorporating 4 dams, and thus solvable by DP, up to a valley with 30 dams, and thus facing the curse of dimensionality (§4.3 and §4.4). We also present two instances corresponding to more realistic hydro valleys, where the models respect the orders of magnitude of the dam sizes of existing valleys (§4.5).

All the results presented here have been obtained using a 3.4GHz, 4 cores – 8 threads Intel® Xeon® E3 based computer.

4.1. Application of DADP to a hydro valley

We go back to the problem formulation presented at §2.2. In order to implement the DADP algorithm, we dualize the coupling constraints

$$\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) = 0, \quad (13)$$

and we denote by λ_t^{i+1} the associated multiplier (random variable).

When minimizing the dual problem at iteration k of the algorithm, the product of (13) with a given multiplier by $\lambda_t^{i+1,(k)}$ is additive with respect to the dams, that is, the term $\lambda_t^{i+1,(k)} \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)$ pertains to dam i subproblem, whereas the term $\lambda_t^{i+1,(k)} \cdot \mathbf{Z}_t^{i+1}$ pertains to dam $i+1$ subproblem, hence leading to a dam by dam decomposition for the dual problem maximization in $(\mathbf{X}, \mathbf{U}, \mathbf{Z})$ at $\lambda_t^{i+1,(k)}$ fixed.

4.1.1. DADP implementation

The DADP method consists of choosing a multiplier process \mathbf{Y} and then replacing the coupling constraints by their conditional expectations with respect

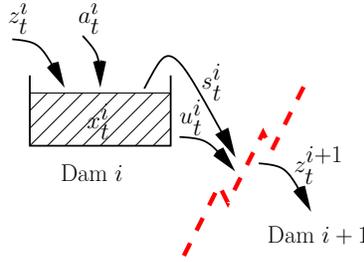


Figure 3: Decomposition by dam.

to \mathbf{Y}_t . Here we adopt the choice $\mathbf{Y}_t = 0$ (minimal information), so that Constraints (13) are replaced in the approximated problem by their expectations:

$$\mathbb{E}[\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)] = 0. \quad (14)$$

The expression of Subproblem (6) attached to dam i reads

$$\min_{\mathbf{U}^i, \mathbf{Z}^i, \mathbf{X}^i} \mathbb{E} \left[\sum_{t=0}^{T-1} \left(L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) + \mathbb{E}[\boldsymbol{\lambda}_t^{i,(k)}] \cdot \mathbf{Z}_t^i \right. \right. \quad (15a)$$

$$\left. \left. - \mathbb{E}[\boldsymbol{\lambda}_t^{i+1,(k)}] \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \right) + K^i(\mathbf{X}_T^i) \right], \quad (15b)$$

$$s.t. \quad \mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i), \quad \mathbf{X}_0^i \text{ given} \quad (15c)$$

$$\mathbf{U}_t^i \leq \sigma(\mathbf{W}_0, \dots, \mathbf{W}_t). \quad (15d)$$

Because of the crude relaxation due to a constant \mathbf{Y}_t^i , 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 all subproblems involve a 1-dimensional state variable, that is, the dam stock \mathbf{X}_t^i , and hence are easily solvable by Dynamic Programming. We denote by $(\mathbf{U}^{i,(k)}, \mathbf{Z}^{i,(k)}, \mathbf{X}^{i,(k)})$ the optimal solution of each subproblem i , and by $V_t^{i,(k)}(x^i)$ the Bellman function obtained for each dam i at time t .

With the choice of constant information variables \mathbf{Y}_t^i , the coordination update step (11) reduces to

$$\mathbb{E}[\boldsymbol{\lambda}_t^{i,(k+1)}] = \mathbb{E}[\boldsymbol{\lambda}_t^{i,(k)}] + \rho_t \mathbb{E}[\mathbf{Z}_t^{i+1,(k)} - g_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t^i, \mathbf{Z}_t^{i,(k)})], \quad (16)$$

that is, a collection of deterministic equations involving the expectation of (13) which is easily estimated by a Monte Carlo approach.

Assume that DADP converges, leading to optimal Bellman functions $V_t^{i,\infty}$. We know that the initial almost-sure coupling constraints are not satisfied. To recover admissibility, we use the heuristic rule proposed at §3.3, solving the

following deterministic one-step DP problem:

$$\min_{(u^1, \dots, u^N)} \sum_{i=1}^N L_t^i(x^i, u^i, w_t^i, z^i) + V_{t+1}^\infty(x_{t+1}^1, \dots, x_{t+1}^N), \quad (17a)$$

$$s.t. \quad x_{t+1}^i = f_t^i(x^i, u^i, w_t^i, z^i) \quad \forall i, \quad (17b)$$

$$z^{i+1} = g_t^i(x^i, u^i, w_t^i, z^i) \quad \forall i. \quad (17c)$$

4.1.2. Complete process

We can summarize the whole process as follows. In the *optimization stage* we first compute the local Bellman functions $V_t^{i,\infty}$, and form the approximate global Bellman functions V_t^∞ by summing the local ones. In the *simulation stage*, we evaluate by Monte-Carlo the strategy induced by V_t^∞ . We draw a large number of noise scenario, and compute the admissible control values along each scenario by solving Problem (17), from $t = 0$ to $t = T - 1$, and storing payoffs.

4.2. SDDP implementation

As explained in §4.3, the controls of the original problem are discrete, which is a difficulty for SDDP implementation though recent extension has been proposed in Zou et al. (2017). In the *optimization stage* we relax the integrity constraints to obtain relaxed Bellman value functions V_t^∞ . Then, in the simulation stage, we use these relaxed Bellman value functions to design policies taking into account the discrete controls by solving problems akin to Problem (17). Furthermore, we consider that the spillage is a control variable, so as to render the dynamics linear, which is the convex-costs SDDP framework.

The whole process of SDDP is as follows. In the optimization stage, lower approximations of Bellman functions V_t are built iteratively. At iteration k , the procedure consists of two passes.

- During the *forward pass*, we sample a scenario of noise. We then simulate a stock trajectory by using the current approximation of the Bellman functions. This is done by successively solving one-step DP problem, akin to Problem (17), where V_{t+1}^∞ is replaced by its current piecewise linear outer-approximation, to determine the next stock value. Note that each of these one-step DP problem is a continuous quadratic programming (QP) problem.
- In the *backward pass*, duality theory allows to find subgradient of lower approximations of the Bellman functions. This subgradients are computed along the trajectory obtained during the forward pass, and used to construct valid cuts, that is hyperplanes that are lower than the Bellman functions. Those cuts are then added to the current outer-approximations of the Bellman functions.

In order to assess the convergence of the SDDP algorithm, we compute (say every 20 iterations of SDDP) a Monte Carlo approximation of the expected

cost value with its associated 95% confidence interval, and compare the upper value of the confidence interval with the lower bound provided by SDDP up to a given threshold in order to stop the algorithm (see Shapiro (2011)). In our experiments, the Monte Carlo simulation has been made using 10,000 scenarios, and the relative convergence threshold was around 0.5%. The *simulation stage* is identical to the one described at §4.1.2 using the global Bellman’s value function obtained by SDDP .

We have used a version of SDDP implemented in Julia (StochDynamicProgramming package²) built on top of the JuMP package used as a modeler (see Dunning et al. (2017)). The QP problems are solved using CPLEX 12.5. Every 10 iterations, redundant cuts are removed thanks to the limited memory level-1 heuristic described in Guigues (2017). Indeed, without cuts removal, the resolution of each QP becomes too slow as the number of cuts increases along iterations.

4.3. Results obtained for academic valleys

We model a first collection of hydro valleys including from 4 to 12 dams, with arborescent geometries (see Figure 4).

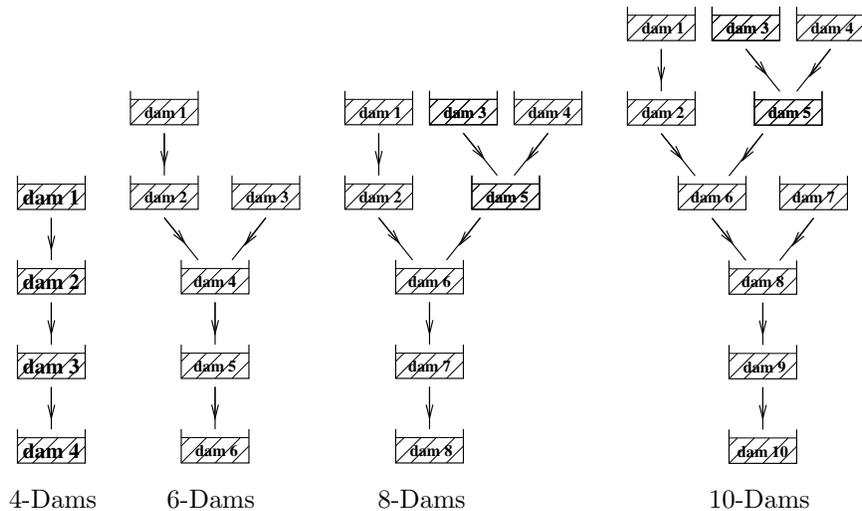


Figure 4: Some academic examples of hydro valleys.

The optimization problem is stated on a time horizon of one year, with a monthly time step ($T = 12$). All the dams have more or less the same maximal volume. The maximal amount of turbinated water for each dam varies with the location of the dam in the valley (more capacity for a downstream dam than for an upstream dam), as well as the random inflows in a dam (more inflow for

²See the github link <https://github.com/JuliaOpt/StochDynamicProgramming.jl>.

an upstream dam than for a downstream dam). We assume discrete probability laws with finite supports for the inflows, and deterministic market prices. We also assume that the available turbine controls are discrete, so that each dam is in fact modeled using a discrete Markov chain. These valleys do not correspond to realistic valleys, in the sense that a true valley incorporates dams with very heterogeneous sizes.

4.3.1. SDDP convergence

We first illustrate the convergence of the SDDP algorithm for the 8-Dams valley on Figure 5 (note that most of the valleys display a similar convergence pattern). As explained at §4.2, the exact lower bound given by SDDP (black curve) increases along the iterations, and the gap between this lower bound and the upper value of the confidence interval (red curve) is less than 0.5% at iteration 140.

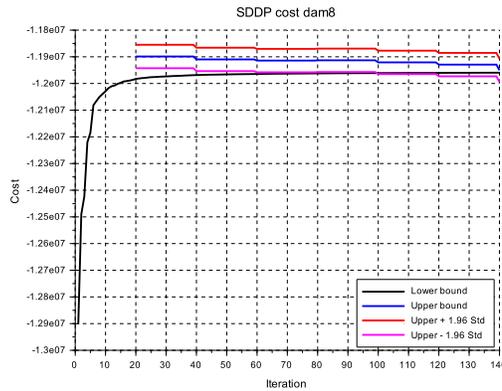


Figure 5: Convergence of SDDP for the 8-Dams valley

Note that, in our experiments, this stopping criterion approximately matches the classical SDDP convergence stopping criterion proposed in Pereira & Pinto (1991) corresponding to the fact that the lower bound provided by SDDP becomes greater than the lower value of the confidence interval.

4.3.2. DADP convergence

Let us first detail the method used for the update of the multipliers involved by DADP. Thanks to the choice of constant information variables, the gradient expression involved in the update formula (16) is an expectation, that can be approximated by a Monte Carlo approach. We draw a collection of statistically independent scenarios of $\{\mathbf{W}_t\}$ and then compute at iteration k of DADP the optimal solutions $\{\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{Z}_t^{i,(k)}\}$ of Subproblem (15) along each scenario. One has to note that this collection of scenarios is independent of the one

used during the simulation stage of the complete process described at §4.1.2. We thus obtain realizations of $(\mathbf{Z}_t^{i+1,(k)} - g_t^i(\mathbf{X}_t^{i,(k)}, \mathbf{U}_t^{i,(k)}, \mathbf{W}_t^i, \mathbf{Z}_t^{i,(k)}))$, whose arithmetic mean gives the (approximated) gradient component at time t for the coupling between dam i and dam $i + 1$. This gradient can be used either in the standard steepest ascent method such as in (16), or in a more sophisticated algorithm such as the conjugate gradient or the quasi-Newton method. We use in our numerical experiments a solver (limited memory BFGS) of the MODULOPT library from INRIA by Gilbert & Jonsson (2007). For all the valleys we studied, the convergence was fast (around 100 iterations regardless of the problem size). Figure 6 represents the evolution of the multipliers λ_t^i for the 8-Dams valley along the iterations of the algorithm.

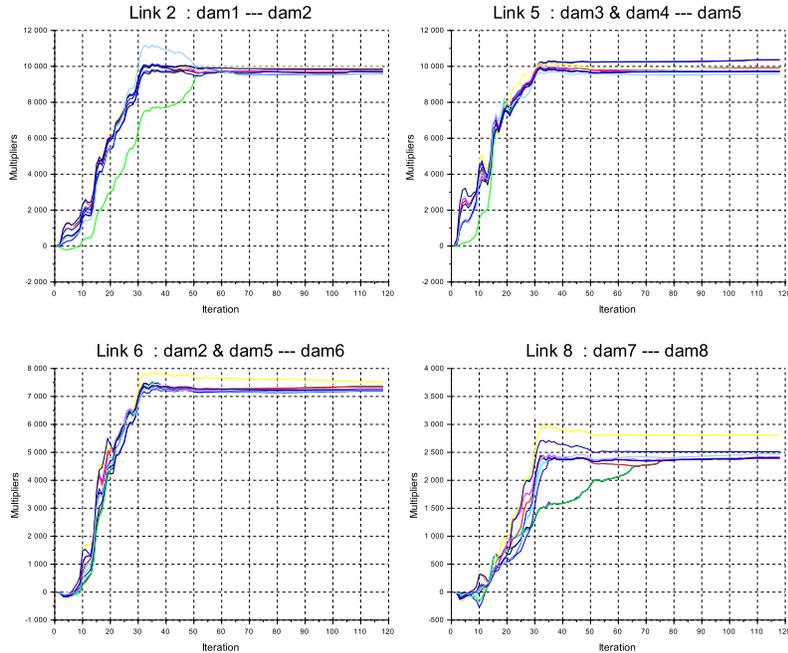


Figure 6: 8-Dams multipliers: dam1→dam2 , dam3-4→dam5 , dam2-5→dam6 , dam7→dam8

The order of magnitude of the optimal multipliers decreases with the geographical position of the link in the hydro valley. Nevertheless, the convergence rate is very similar for all links: this practical consideration remains true for almost all valleys, and it explains why the number of iterations required for the DADP convergence does not vary too much with the size of the valley.

4.3.3. Methods comparison

We solve Problem (1) for the first collection of academic valleys by 3 different methods:

1. the standard Dynamic Programming method (DP), when possible,
2. the SDDP presented at §4.2,
3. the DADP method.

All these methods produce Bellman functions (optimization stage described at §4.1.2), whose quality is evaluated by the simulation stage of §4.1.2. The obtained results are given in Table 1. The lines “CPU time” correspond to the time (in minute) needed to compute the Bellman functions (optimization stage), whereas the lines “value” indicate the cost obtained by Monte Carlo on the initial model (simulation stage, performed using a 100,000 scenarios sample, except for the 12-Dams valley for which a smaller sample set was used to reduce the computational load). The comparisons between the different cost values for the same valley are thus relevant. For both SDDP and DADP, we also give the lower bound corresponding to the Bellman value obtained at the end of the optimization stage.

Valley	4-Dams	6-Dams	8-Dams	10-Dams	12-Dams
DP CPU time	1600'	$\sim 10^8$ '	$\sim \infty$	$\sim \infty$	$\sim \infty$
DP value	-3743	N.A.	N.A.	N.A.	N.A.
SDDP CPU time	6'	10'	13'	50'	97'
SDDP value	-3742	-7027	-11830	-17070	~ -17000
SDDP lower bound	-3754	-7050	-11960	-17260	-19490
DADP CPU time	7'	12'	18'	24'	22'
DADP value	-3667	-6816	-11570	-16760	~ -17000
DADP lower bound	-3996	-7522	-12450	-17930	-20480
Gap DADP/SDDP	2.0%	3.0%	2.2%	1.8%	?

Table 1: Results obtained by DP, SDDP and DADP

We first note that a direct use of DP is only possible for the 4-Dams valley: it corresponds to the well-known curse of dimensionality inherent to DP. The value given by DP is the true optimal cost value for the 4-Dams valley and can be used as the reference value. The SDDP method, although relying on the integrity constraints relaxation in the optimization stage (hence a not so tight lower bound), gives excellent results for the 4-Dams valley: we thus elect SDDP as the reference method in order to evaluate the DADP method. Note that the CPU time remains reasonable, the optimization problems inside SDDP corresponding to a continuous linear-quadratic formulation (here solved using the CPLEX commercial solver).

Remark 2. Note however that all the methods we are comparing face the curse of dimensionality associated to the combinatorics of the control during the simulation stage, as the controls associated to the whole valley have to be enumerated at each time t along each scenario. This is the reason why the value obtained for the 12-Dams valley have been computed using 1,000 scenarios (100,000 for the others valleys) and hence are not so accurate.

We now turn to the DADP method. We first notice that the lower bound given by the method is rather bad (as a consequence of solving a problem with

relaxed coupling constraints in the optimization stage), but the values obtained in the simulation stage are reasonable compared to the ones given by SDDP (as indicated by the last line of Table 1). The most noticeable point is that the CPU time needed for the optimization stage seems to grow more slowly for DADP than for SDDP. This aspect will be highlighted in §4.4.

Let us finally materialize more finely the difference in the results between SDDP and DADP. Beyond average values given in Table 1, Figure 7 represents the payoff empirical probability laws (optimal cost over the time horizon), obtained by the simulation stage using 100,000 scenarios, for both SDDP and DADP. We observe that, although the expectations are fairly close, the shapes of the two distributions differ significantly.

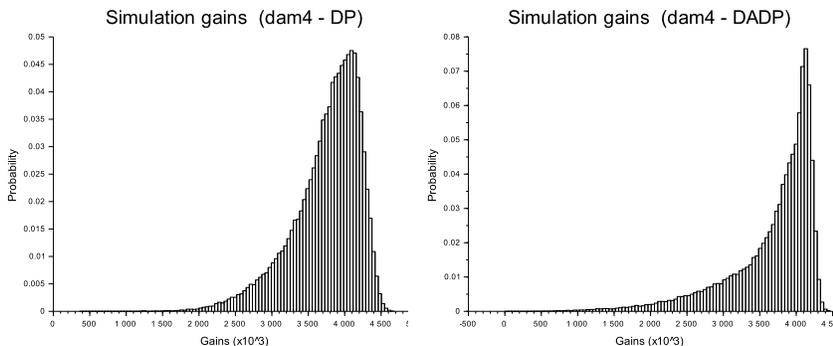


Figure 7: 4-Dams payoff distributions: SDDP (left) — DADP (right)

4.4. Challenging the curse of dimensionality

The experiments made in §4.3 seem to indicate that DADP is less sensitive to the size of the valley than the SDDP method. In order to validate this observation, we design a new collection of academic hydro valleys incorporating from 14 up to 30 dams. It is of course no longer possible to perform the simulation stage for these instances: the combinatorics induced by the set of possible values of the controls is too large to allow simulation of the valley behavior along a large set of scenarios. We thus limit ourselves to the computation of the Bellman functions (optimization stage). The corresponding results are reported in Table 2.

It appears that the CPU time required for the DADP method grows linearly with the number of dams, while the growth rate of SDDP is more or less exponential. Figure 8 shows how the CPU time varies for the three methods. As expected, DP is only implementable for small instances, say up to 5 dams. Eventually, the limits of SDDP and DADP have not really been reached, but DADP displays a near linear rate of CPU time allowing to tackle instances of even greater size.

Valley	14-Dams	18-Dams	20-Dams	25-Dams	30-Dams
SDDP CPU time	210'	585'	970'	1560'	2750'
SDDP lower bound	-32024	-46917	-61454	-79440	-100430
DADP CPU time	40'	50'	75'	140'	150'
DADP lower bound	-32981	-48095	-62802	-80993	-101990

Table 2: SDDP and DADP comparison for large academic valleys

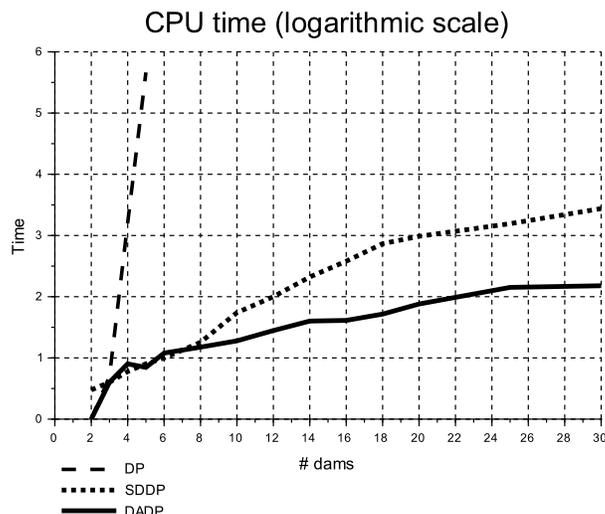


Figure 8: CPU time comparison

4.5. Results for two realistic valleys

We finally model two hydro valleys corresponding to existing systems in France, namely the Vicdessos valley and the Dordogne river (see Figure 9).

The optimization problem is stated again on a one year horizon, with a monthly time step. What mainly differ here from the academic examples used at §4.3 are the characteristics of the dams. For example, the Dordogne river valley encompasses large dams (as “Bort” whose capacity is say 400) and small dams (as “Mareges” the capacity of which is equal to 35, that is, ten times smaller). This heterogeneity induces numerical difficulties, for example the requirement to have a wide range of possible controls for the small downstream reservoirs, or the need to use a fine discretization for the state grid in DP-like methods. We again assume discrete probability laws with finite support for the inflows, and we also assume that the available turbine controls are discrete.

The comparison results of SDDP and DADP are given in Table 3. As for the academic examples, SDDP displays the best results and is therefore used as the reference. The large number of possible discrete controls penalizes the

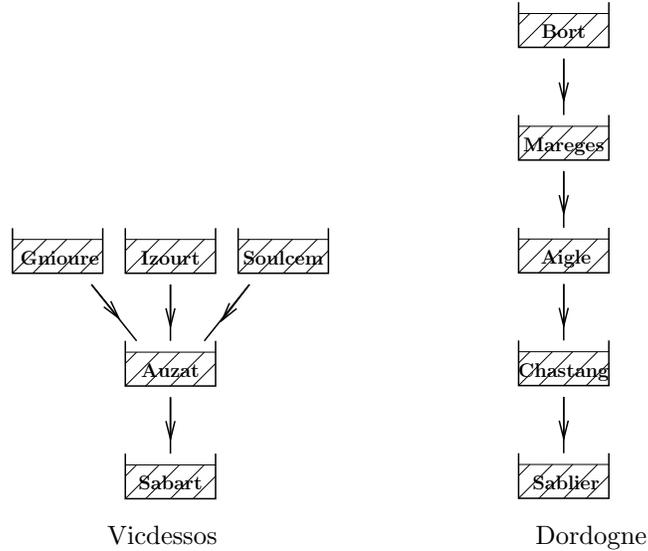


Figure 9: Two realistic hydro valleys.

Valley	Vicdessos	Dordogne
SDDP CPU time	<i>9'</i>	<i>17'</i>
SDDP value	-2244	-22150
SDDP lower bound	-2258	-22310
DADP CPU time	<i>9'</i>	<i>210'</i>
DADP value	-2238	-21650
DADP lower bound	-2286	-22990
Gap DADP/SDDP	-0.3%	-2.2%

Table 3: Results obtained by SDDP and DADP

DADP method, although the gap between SDDP and DADP remains limited.

5. Conclusion

In this article, we have depicted a method called DADP which allows to tackle large-scale stochastic optimal control problems in discrete time, such as the ones found in the field of energy management. We have presented the practical aspects of the method, without deepening in the theoretical issues arising in the foundations of the method. Lots of numerical experiments have been presented on hydro valley problems (“chained models”), which complements the ones already made on unit commitment problems (“flower models”) Barty et al. (2010). The main conclusions are that DADP converges fast and gives near-optimal results even when using a “crude” relaxation (here a constant

information process \mathbf{Y}). More precisely, DADP allows to deal with optimization problems that are out of the scope of standard Dynamic Programming, and beats SDDP for very large hydro valleys in terms of CPU time. We thus hope to be able to implement DADP for very large stochastic optimal control problems such as the ones encountered in smart management of urban districts, involving hundreds of houses and thus hundred of states variables. Such problems are formulated on a short-term time scale (typically a one day horizon with 15 minutes time steps), and incorporate on/off devices. In that new context, controls will have to be modeled using discrete variables (whereas this assumption was not mandatory for the study presented in this paper). Moreover, on a short-term time scale, the randomness of the markets prices plays an important role, and it will thus be necessary to take them into account as a noise in the problem.

We plan to extend this study in two directions. First to implement the DADP method for general spatial structures (not only “flower models” or “chain model”, but “smart-grid models” involving a generic graph). The second direction is to implement more sophisticated decomposition methods than price decomposition. On the one hand we want to use decomposition schemes such that resource allocation or interaction prediction principle Cohen (1978). On the other hand we want to use augmented Lagrangian based methods such as alternating direction method of multiplier (ADMM) and proximal decomposition algorithm (PDA) for decomposition in order to obtain the nice convergence properties of this kind of methods (see Lenoir & Mahey (2017) for a survey).

Finally, let us mention that a theoretical work has begun in order to provide foundations of the method (Leclère (2014)). It includes conditions for existence of a multiplier in the L^1 space when the optimization problem is posed in L^∞ and conditions for convergence of the Uzawa algorithm in L^∞ . A lot of work remains to be done on these questions, mainly to relax the continuity assumption in order to be able to deal with extended functions, and to obtain more general assumptions ensuring the convergence of the Uzawa algorithm.

Acknowledgments. This research benefited from the support of the FMJH “Program Gaspard Monge for optimization and operations research”, and from the support from EDF.

References

- Alais, J.-C. (2013). *Risque et optimisation pour le management d’énergies*. Thèse de doctorat Université Paris-Est.
- Bacaud, L., Lemaréchal, C., Renaud, A., & Sagastizábal, C. A. (2001). Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioner. *Computational Optimization and Applications*, 20, 227–244.
- Barty, K., Carpentier, P., & Girardeau, P. (2010). Decomposition of large-scale stochastic optimal control problems. *RAIRO Operations Research*, 44, 167–183.

- Barty, K., Roy, J.-S., & Strugarek, C. (2009). A stochastic gradient type algorithm for closed-loop problems. *Mathematical Programming, Series A*, *119*, 51–78.
- Bellman, R. (1957). *Dynamic Programming*. New Jersey: Princeton University Press.
- Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*. (4th ed.). Athena Scientific.
- Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.
- Carpentier, P., Chancelier, J.-P., Cohen, G., & De Lara, M. (2015). *Stochastic Multi-Stage Optimization* volume 75 of *Probability Theory and Stochastic Modelling*. Springer.
- Carpentier, P., Cohen, C., Culioli, J.-C., & Renaud, A. (1996). Stochastic optimization of unit commitment: a new decomposition framework. *IEEE Transactions on Power Systems*, *11*, 1067–1073.
- Cohen, G. (1978). Optimization by Decomposition and Coordination: A Unified Approach. *IEEE Transactions on Automatic Control*, *23*, 222–232.
- Cohen, G. (1980). Auxiliary Problem Principle and decomposition of optimization problems. *Journal of Optimization Theory and Applications*, *32*, 277–305.
- Cohen, G., & Culioli, J.-C. (1990). Decomposition Coordination Algorithms for Stochastic Optimization. *SIAM Journal on Control and Optimization*, *28*, 1372–1403.
- Dellacherie, C., & Meyer, P. A. (1975). *Probabilités et potentiel*. Hermann.
- Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, *59*, 295–320.
- Ekeland, I., & Temam, R. (1999). *Convex Analysis and Variational Problems* volume 28 of *Classics in Applied Mathematics*. SIAM.
- Gilbert, J. C., & Jonsson, X. (2007). Libopt—an environment for testing solvers on heterogeneous collections of problems.
- Girardeau, P. (2010). *Résolution de grands problèmes en optimisation stochastique dynamique*. Thèse de doctorat Université Paris-Est.
- Girardeau, P., Leclere, V., & Philpott, A. B. (2015). On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, *40*, 130–145.

- Guigues, V. (2017). Dual dynamic programming with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, 258, 47–57.
- Heitsch, H., & Römisch, W. (2009). Scenario tree modelling for multistage stochastic programs. *Mathematical Programming*, 118, 371–406.
- Heitsch, H., Römisch, W., & Strugarek, C. (2006). Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17, 511–525.
- Kuhn, D., Wiesemann, W., & Georghiou, A. (2011). Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130, 177–209.
- Leclère, V. (2014). *Contributions aux méthodes de décomposition en optimisation stochastique*. Thèse de doctorat Université Paris-Est.
- Lenoir, A., & Mahey, P. (2017). A survey on operator splitting and decomposition of convex programs. *RAIRO Operations Research*, 51, 17–41.
- Maceira, M. E. P., & Damazio, J. M. (2006). Use of par(p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the brazilian hydropower system. *Probability in the Engineering and Informational Sciences*, 20, 143–156.
- de Matos, V. L., Philpott, A. B., & Finardi, E. C. (2015). Improving the performance of stochastic dual dynamic programming. *Journal of Computational and Applied Mathematics*, 290, 196–208.
- Pereira, M. V. F., & Pinto, L. M. V. G. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52, 359–375.
- Pflug, G. C., & Pichler, A. (2014). *Multistage Stochastic Optimization*. Springer.
- Philpott, A. B., & Guan, Z. (2008). On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36, 450–455.
- Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality* volume 842. John Wiley & Sons.
- Puterman, M. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons.
- Rockafellar, R. T. (1968). Integrals which are convex functionals. *Pacific Journal of Mathematics*, 24, 525–539.
- Rockafellar, R. T. (1971). Integrals which are convex functionals, ii. *Pacific Journal of Mathematics*, 39, 439–469.

- Rockafellar, R. T., & Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.*, *16*, 119–147.
- Ruszczynski, A. (1997). Decomposition methods in stochastic programming. *Mathematical programming*, *79*, 333–353.
- Ruszczynski, A., & Shapiro, A. (Eds.) (2003). *Stochastic Programming* volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier.
- Shapiro, A. (2006). On complexity of multistage stochastic programs. *Operations Research Letters*, *34*, 1–8.
- Shapiro, A. (2011). Analysis of Stochastic Dual Dynamic Programming Method. *European Journal of Operational Research*, *209*, 63–72.
- Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, Philadelphia.
- Strugarek, C. (2006). *Approches variationnelles et autres contributions en optimisation stochastique*. Thèse de doctorat École Nationale des Ponts et Chaussées.
- Takriti, S., Birge, J. R., & Long, E. (1996). A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, *11*, 1497–1508.
- Tsitsiklis, J. N., & Van Roy, B. (1996). Feature-based methods for large-scale dynamic programming. *Machine Learning*, *22*, 59–94.
- Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, *17*, 638–663.
- Vezolle, P., Vialle, S., & Warin, X. (2009). Large Scale Experiment and Optimization of a Distributed Stochastic Control Algorithm. Application to Energy Management Problems. In *International workshop on Large-Scale Parallel Processing (LSPP 2009)*. Rome, Italy.
- Zou, J., Ahmed, S., & Sun, X. A. (2017). Stochastic dual dynamic integer programming. *Mathematical Programming*, (pp. 1–42).