HAL
archives-ouvertes.fr

Energy contracts management by stochastic programming techniques

Zhihao Cen, J. Frederic Bonnans, Thibault Christel

To cite this version:

Zhihao Cen, J. Frederic Bonnans, Thibault Christel. Energy contracts management by stochastic programming techniques. Annals of Operations Research, Springer Verlag, 2011, 200 (1), pp.199-222. 10.1007/s10479-011-0973-5 . inria-00486897v2

HAL Id: inria-00486897

https://hal.inria.fr/inria-00486897v2

Submitted on 1 Aug 2011

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.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Energy contracts management by stochastic programming techniques

Zhihao Cen — J. Frédéric Bonnans — Thibault Christel

N° 7289

May 2010

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7289--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SACLAY - ÎLE-DE-FRANCE

Energy contracts management by stochastic programming techniques

Zhihao Cen* , J. Frédéric Bonnans† , Thibault Christel‡

Thème NUM — Systèmes numériques
Équipes-Projets Commands

Rapport de recherche n° 7289 — May 2010 — 26 pages

Abstract: We consider the problem of optimal management of energy contracts, with bounds on the local (time step) amounts and global (whole period) amounts to be traded, integer constraint on the decision variables and uncertainty on prices only. After building a finite state Markov chain by using vectorial quantization tree method, we rely on the stochastic dual dynamic programming (SDDP) method to solve the continuous relaxation of this stochastic optimization problem. An heuristic for computing sub optimal solutions to the integer optimization problem, based on the Bellman values of the continuous relaxation, is provided. Combining the previous techniques, we are able to deal with high-dimension state variables problems. Numerical tests applied to realistic energy markets problems have been performed.

Key-words: stochastic programming, multi-stage, dual dynamic programming, quantization tree.

* Total, Paris La Defense, COMMANDS team, INRIA-Saclay and CMAP, Ecole Polytechnique, Palaiseau, France (zhihao.cen@polytechnique.edu).
† COMMANDS team, INRIA-Saclay, CMAP, Ecole Polytechnique, Palaiseau, and Laboratoire de Finance des Marchés de l'Energie, Paris, France (Frederic.Bonnans@inria.fr).
‡ Total, Paris La Defense, France (thibault.christel@total.com).

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Gestion de contrats d'énergie par des techniques de programmation stochastique

Résumé : Nous considérons le problème de la gestion optimale de contrats d'énergie, avec bornes sur les quantités locales et globales, des contraintes d'intégrité sur les variables de décision et une incertitude ne portant que sur les prix. Après avoir construit une chaîne de Markov en état finie par la méthode d'arbre de quantisation, nous nous appuyons sur la méthode de programmation stochastique dynamique duale (SDDP) pour résoudre la relaxation continue de ce problème d'optimisation stochastique. Une heuristique de calcul de solutions sous optimales du problème en nombres entiers, basée sur les valeurs de Bellman du problème relaxé, est proposée. Combinant les deux techniques précédentes, nous sommes en mesure de traiter des problèmes de grande dimension. Des tests numériques appliqués à des problèmes réalistes de marchés de l'énergie ont été réalisés.

Mots-clés : programmation stochastique, multi étapes, programmation dynamique duale, arbre de quantisation.

1 Introduction

The physical energy markets are articulated around long term supply and demand contracts. Their purpose is to accommodate both producers and consumers according to their needs. On one side, a producer will be keen on selling energy on a long term basis for cash flow certainties allowing him to schedule other investments. On the other side, it ensures the consumer of a certainty on volumes and acts as a hedge against short term uncertainties.

Long term energy contracts typically have the following features. First of all, they are associated with a granularity defining the periods at which the energy is exchanged. Secondly, the amounts of traded energy are contractually upper and lower bounded during each period as well as on the overall contract. The motivation behind these bounds is generally to allow the consumer to modulate his energy purchase. Finally, the price at which the energy is bought or sold at a given period is function of the index spot price which is a stochastic variable. Therefore, the optimal management of long term contracts implies for the option's holder to successively choose the best possible decision over the quantity to be bought or sold according to the price information available at that moment. Each decision taken will impact the remainder of the contract. In addition, we might consider the situation where the decisions to be made are discrete. Indeed, unlike pipeline natural gas where it is generally possible to assume that any quantity of gas can be exchanged as long as it fulfills the contractual boundaries, this is no longer true for sea shipped commodities such as Liquefied Natural Gas (LNG). In fact, the quantity of energy to be allocated is represented by the number of LNG vessels to allocate to a destination. The combination of stochastic and discrete decisions makes the problem very difficult, and it is natural to consider a continuous relaxation of the problem, at least as a first step.

For surveys on stochastic programming, we refer to [5, 18, 29]. Unless the underlying random space has finite support, our model is an optimization problem over infinite-dimensional function spaces, which is difficult to solve. Analytical solutions are not available except for extremely simple and unrealistic cases. Numerical approximation methods have been largely studied in the literature.

Most approximation schemes are based on discretization of the underlying random space by a scenario tree. A survey and evaluation of popular scenario generation techniques is provided in [19]. Among them, let us mention sampling-based methods [29, Chapter 3], moment matching methods [16], tree reduction methods [8, 13], recombining tree [20]. Once the underlying probability space becomes discrete, finite dimensional optimization algorithms can be applied without difficulty. However, obtaining a high precision for the optimal value and getting the optimal solution is still a big challenge.

Another main approximation method is to project the control function or cost-to-go function into a finite dimension functional space [4, 30, 31]. We have hence to solve a finite dimensional problem, whose dimension is the cardinality of the functional space. The advantage is to avoid discretizing the underlying probability space. Numerical simulation can be done based on Monte-Carlo methods, and thus the convergence results are given by classical Monte-Carlo methods properties as well as Hilbert space theory. This method has been successfully applied [27]. However, in practice, the result depends strongly on the basis functions chosen. Furthermore, when the state variable (and random

variable) dimension increases, the dimension of basis function space usually grows exponentially.

Apart from these two main approximation methods, a few techniques have been proposed for certain specific problems. For example, Pereira and Pinto [23] suggested a stochastic dual dynamic programming (SDDP) algorithm to solve the problem where randomness only appears on right-hand-side of constraint and is independent from one stage to another. Philpott and Guan [25] have studied the convergence of this method. Shapiro has recently analysed the statistical properties and rate of convergence of this method [28]. This method can also be applied to ARMA processes [17].

Finally, we note that Mo et al. [22] use a SDDP approach combined with a finite state Markov chain representation of the evolution of the price (which is scalar). A specific feature of our problem is that the random variables are functions of some Brownian motion, and therefore we have to design an approximation procedure in order to reduce the problem to one in which the random variables have a Markov chain representation.

The main contribution of this article is to use the vectorial quantization tree method introduced in Bally and Pagès [2, 1], which takes advantage of the fact that the random variable is Gaussian. We apply the stochastic dual decomposition method to the continuous relaxation of the quantized problem. The numerical test results present good performances when the two methods are combined. This article is organised as follows: in section 2, we present the model and its properties. In section 3, we introduce the quantization tree methods, and then in section 4 the algorithm. Finally, numerical tests are performed over realistic energy management problems in section 5. A heuristic method for computing integer suboptimal solutions based on the Bellman values obtained by continuous relaxation is also provided.

2 Problem formulations

2.1 Framework

We now formulate the class of problems of concern in the following setting. First of all, we have a discrete time Markov process $(\boldsymbol{\xi}_t), 0 \leq t \leq T$ in the probability space $L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathbb{R}^d)$, where $\mathcal{F}_t$ denotes the canonic filtration associated with $(\boldsymbol{\xi}_t) : \mathcal{F}_t := \sigma(\boldsymbol{\xi}_s, 0 \leq s \leq t)$. We may write its dynamics as

$$\boldsymbol{\xi}_{t+1} = f(W_t, \boldsymbol{\xi}_t, \alpha_t) \quad t = 0, \ldots, T-1, \tag{1}$$

where $\boldsymbol{\xi}_0$ is deterministic, $(\alpha_t)_{t\in[0,T-1]}$ is a sequence of deterministic parameters, and $(W_t)_{t\in[0,T-1]}$ is a sequence of independent random variables in $\mathbb{R}^d$, independent of $\mathcal{F}_t$; finally f is a measurable mapping.

The decision problem has the following expression:

$$\begin{aligned} \inf_{(u_t)} \quad & \mathbb{E}\left[\sum_{t=0}^{T-1} c_t(\boldsymbol{\xi}_t) \cdot u_t + g(\boldsymbol{\xi}_T, x_T)\right] \\ \text{subject to} \quad & u_t \in U_t, \\ & x_{t+1} = x_t + A_t u_t, \\ & x_0 = 0,\ x_T \in \mathfrak{X}_T \quad \text{almost surely;} \end{aligned} \tag{2}$$

where u_t is the control variable, assumed to be squared integrable, taking value in a compact nonempty convex polyhedral set $\mathfrak{U}_t \subset \mathbb{R}^n$, $U_t := L^2(\Omega, \mathcal{F}_t, \mathbf{P}; \mathfrak{U}_t)$ represents the feasible space, $x_t \in \mathbb{R}^m$ is the state variable representing the inventory levels, which is obviously $\mathcal{F}_{t-1}$ measurable by forward induction, $\mathfrak{X}_T$ is the set of admissible final state, assumed to be a nonempty polyhedral subset of $\mathbb{R}^m$, taking place in the final stage constraint over state variable x_T, $A_t \in \mathbb{R}^{m\times n}$ describes how the decisions impact the inventories, $c_t(\cdot) : \mathbb{R}^d \to \mathbb{R}^n$ is running cost unit, assumed to be Lipschitz, and finally $g(\boldsymbol{\xi}, x) : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ is a convex lower semi-continuous (l.s.c.) function of x, uniformly lower bounded and can often be interpreted as a penalty function for final state x_T.

Lemma 2.1. *If $\sum_{t=0}^{T-1} A_t \mathfrak{U}_t \cap \mathfrak{X}_{T+1} \neq \emptyset$, problem* (2) *has a nonempty set of solutions.*

Proof. The set of feasible policies is closed and convex, and hence, weakly closed. The final cost $\mathbb{E}[g(\boldsymbol{\xi}_T, x_T)]$ is a l.s.c. convex function over $L^2(\Omega, \mathcal{F}_T, \mathbf{P}; \mathbb{R}^m)$, and therefore is weakly l.s.c. in this space; so is the all cost function $(c_t(\boldsymbol{\xi}_t) \cdot u_t)$. Since the minimizing sequence u^k is bounded in $L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathbb{R}^n)$, extracting if necessary a subsequence, we may assume that it weakly converges in this space. Passing to the limit in both the state equation and in the constraint on the decision and using the weak l.s.c. of the cost function, we obtain that the weak limit of u^k is a solution of (2). □

The condition $\sum_{t=0}^{T-1} A_t u_t \cap \mathfrak{X}_T \neq \emptyset$ is assumed to be always satisfied in this article in order to make sure of the existence of a solution.

Remark 2.2. The function to be minimized is not strictly convex. Therefore the minimum is not necessarily unique.

2.2 Dynamic programming principle

The problem under study has the desired structure for the dynamic programming principle to hold (see [29, Chapter 3]). Let us define the realization up to time t of the random process as $\boldsymbol{\xi}_{[t]} = (\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_t)$, and of the state variable value as $x_{[t]} = (x_0, \ldots, x_t)$. The Bellman value at stage t, denoted by $Q(t, x_{[t]}, \boldsymbol{\xi}_{[t]})$, is the value of the variant of the original problem obtained when starting at time t with state variable (storage) x_t:

$$\begin{aligned} Q(t, x_{[t]}, \boldsymbol{\xi}_{[t]}) := \inf \quad & \mathbb{E}\left[\sum_{s=t}^{T-1} c_s(\boldsymbol{\xi}_s) \cdot u_s + g(\boldsymbol{\xi}_T, x_T)\right] \\ \text{subject to} \quad & u_s \in U_s, \mathcal{F}_s - \text{measurable}, \\ & x_{s+1} = x_s + A_s u_s, \\ & x_T \in \mathfrak{X}_T \quad \text{almost surely.} \end{aligned} \tag{3}$$

In view of the Markov property of $(\boldsymbol{\xi}_t)$, $(\boldsymbol{\xi}_s)_{t\leq s\leq T}$ only depends on $\boldsymbol{\xi}_t$ instead of $\boldsymbol{\xi}_{[t]}$. And x_t is $\mathcal{F}_{t-1}$ measurable because u_t is $\mathcal{F}_t$ measurable, then $Q(t, x_{[t]}, \boldsymbol{\xi}_{[t]})$ is actually a function of $(t, x_t, \boldsymbol{\xi}_t)$. The optimal value of the original problem is $Q(0, 0, \boldsymbol{\xi}_0)$.

The following *dynamic programming principle* is easily established:

$$\begin{aligned} Q(t, x_t, \boldsymbol{\xi}_t) = \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t, \\ & x_{t+1} = x_t + A_t u_t; \end{aligned} \tag{4}$$

where $\mathcal{Q}(\cdot,\cdot,\cdot)$ is the conditional expectation of the value at time $t+1$ obtained in state x_{t+1}:

$$\begin{aligned} \mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t) &:= \mathbb{E}\left[Q(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \tilde{v} \mathcal{F}_t\right] \\ &= \mathbb{E}\left[Q(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \tilde{v} x_{t+1}, \boldsymbol{\xi}_t\right] \end{aligned} \tag{5}$$

where the second equality comes from the fact that x_{t+1} is $\mathcal{F}_t$ measurable.

Finally the Bellman value function at final stage is:

$$Q(T, x_T, \boldsymbol{\xi}_T) = \begin{cases} g(\boldsymbol{\xi}_T, x_T) & \text{if } x_T \in \mathfrak{X}_T, \\ +\infty & \text{otherwise.} \end{cases} \tag{6}$$

2.3 Feasibility

The property of *relatively complete recourse* does not hold generally in this setting. Indeed, it may happen that for a given time t and state variable x_t, where the associated Bellman value is finite (implying the existence of a sequence of feasible decisions with the corresponding final time belongs to $\mathfrak{X}_T$), some $u_t \in \mathfrak{U}_t$ may result in a non feasible state variable $x_{t+1} := x_t + A_t u_t$, whose associated Bellman value will therefore be infinite. In other words, the final state constraint induces implicit constraints on the decision u_t at time t.

The SDDP approach includes a mechanism for dealing with this situation, by generating feasibility cuts (similar to Bender's feasibility cuts for a two stage problem but combined with backward recursion). However, generating feasibility cuts may be a long process and we prefer to deal with them by a direct approach, by taking advantage of the specific property that no uncertainty occurs in the state dynamics. In order to guarantee that the final constraints are fulfilled, let us define the sets of feasible control:

$$\mathfrak{U}^{ad} := \left\{ (u_t) \,\middle|\, \sum_{t=0}^{T-1} A_t u_t \in \mathfrak{X}_T,\ u_t \in \mathfrak{U}_t, t = 0, \ldots, T-1 \right\}. \tag{7}$$

and $U^{ad} = L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathfrak{U}^{ad})$.

At stage $t \in [0, T-1]$ and for a given state x_t is associated the (possibly empty) local set of feasible control

$$\mathfrak{U}_t^{ad}(x_t) := \left\{ u_t \,\middle|\, x_t + \sum_{s=t}^{T-1} A_s u_s \in \mathfrak{X}_T,\ u_s \in \mathfrak{U}_s, s = t, \ldots, T-1 \right\}. \tag{8}$$

and $U_t^{ad} = L^2(\Omega, \mathcal{F}_t, \mathbf{P}; \mathfrak{U}_t^{ad})$.

Hence, we can also define $\mathfrak{X}_t = \left\{ x \tilde{v} \mathfrak{U}_t^{ad}(x) \neq \emptyset \right\}$, such that only for $x_t \in \mathfrak{X}_t$, there exists a feasible control leading to the final state feasible set $\mathfrak{X}_{T+1}$:

$$\mathfrak{X}_t := \left\{ x_t \,\middle|\, \exists u_s \in \mathfrak{U}_s\ s \geq t,\ s.t.\ x_t + \sum_{s=t}^{T-1} A_s u_s \in \mathfrak{X}_T \right\} = \mathfrak{X}_T - \sum_{s=t}^{T-1} A_s \mathfrak{U}_s. \tag{9}$$

Since $\mathfrak{X}_T$ and $\mathfrak{U}_t$ are polyhedral sets, so is $\mathfrak{X}_t$. If we replace the decision u_t by a sequence $(u_t, \ldots, u_{T-1})$ such that $x_t \in \mathfrak{X}_t$, we get the property of relatively complete recourse. The counterpart is an increase of dimension of the decision variable, which in our LNG application (see second example in section 5) is acceptable.

2.4 Stochastic dual dynamic programming (SDDP) formulation

We next take advantage of a partial convexity property of the Bellman functions.

Proposition 2.3. *For any $t = 0$ to T, the Bellman value functions $Q(t, x_t, \boldsymbol{\xi}_t)$ and $\mathcal{Q}(t, x_t, \boldsymbol{\xi}_{t-1})$ are convex and lower semi-continuous functions of x_t.*

Proof. We prove the convexity and lower semi-continuous property by backward induction. In view of (6), both properties hold for function $Q(T, x_T, \boldsymbol{\xi}_T)$. The conditional expectation (5) keeps clearly the convexity on x_t since the conditional expectation operation is linear. As to lower semi-continuous property of $\mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t)$, we apply the Fatou lemma on $Q(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \leq \liminf_{x^k_{t+1} \to x_{t+1}} Q(t+1, x^k_{t+1}, \boldsymbol{\xi}_{t+1})$:

$$\begin{aligned}\mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t) &= \mathbb{E}[Q(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \tilde{v} \mathcal{F}_t] \\ &\leq \liminf_{x^k_{t+1} \to x_{t+1}} \mathbb{E}[Q(t+1, x^k_{t+1}, \boldsymbol{\xi}_{t+1}) \tilde{v} \mathcal{F}_t] = \liminf_{x^k_{t+1} \to x_{t+1}} \mathcal{Q}(t+1, x^k_{t+1}, \boldsymbol{\xi}_{t+1}).\end{aligned} \tag{10}$$

The convexity follows by induction over t, from $T-1$ to 0, using (4). The lower semi-continuous property, in view of theorem [26, Theorem 1.17] on parametric minimization, holds as the expression on right hand side of (4) is proper (which means that it is always greater than $-\infty$, and not always equal to $+\infty$), lower semi-continuous and level-bounded in u_t locally uniformly in x_t [26, Definition 1.16]. By induction, both properties follow for all $t = 0, \ldots, T$. □

Remark 2.4. $Q(t, x_t, \boldsymbol{\xi}_t)$ and $\mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t)$ are not in general convex functions of $\boldsymbol{\xi}_t$.

We denote by f^* the Fenchel conjugate of an extended real valued function f on an Euclidean space X, identified with its dual, defined for each $x^* \in X$ by

$$f^*(x^*) := \sup_{x \in X} (x^* \cdot x - f(x)). \tag{11}$$

Let f^{**} denote the biconjugate of f, defined by $f^{**}(x) := \sup_{x^* \in X^*} (x \cdot x^* - f^*(x^*))$. By the Moreau Fenchel theorem [26, Thm 12.1], $f = f^{**}$ whenever f is convex, lower semi-continuous and proper . Applying this theorem to $Q(t, x_t, \boldsymbol{\xi}_t)$ allows to write this function as the supremum of a family of affine functions of x, depending on $(t, \boldsymbol{\xi}_t)$:

$$Q(t, x_t, \boldsymbol{\xi}_t) = Q^{**}(t, x_t, \boldsymbol{\xi}_t) = \sup_{x^*_t \in \mathbb{R}^m} x^*_t \cdot x_t - Q^*(t, x^*_t, \boldsymbol{\xi}_t), \tag{12}$$

where Q^* denotes the conjugate of Q with respect to x only.

Thus, we can use the stochastic dual decomposition approach on the state variable x_t [5, 29], but not on the random variable $\boldsymbol{\xi}_t$. The corresponding formulation of the dynamic programming principle is

$$\begin{aligned}
Q(t, x_t, \boldsymbol{\xi}_t) = \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t) && (13)\\
= \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \mathcal{Q}^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_t) && (14)\\
\text{subject to} \quad & x_{t+1} = x_t + A_t u_t,\\
\text{(feasibility)} \quad & u_t \in \mathfrak{U}_t^{ad}(x_t),\\
\text{(optimality cut)} \quad & \begin{cases} \mathcal{Q}^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_t) \geq \mathbb{E}\left[Q^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \tilde{\upsilon} \mathcal{F}_t\right], \\ Q^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \geq x^* \cdot x_{t+1} - Q^*(t+1, x^*, \boldsymbol{\xi}_{t+1}), \forall x^*. \end{cases}
\end{aligned}$$

This is the formulation we will apply in our algorithm.

However, in this formulation, we still need to calculate conditional expectations. Many numerical methods have been proposed, such as PDE or Monte-Carlo methods. Here, we choose to discretize the random space. Considering the size of the portfolio that we are interested in (see second example in section 5), standard scenario trees will cause an increase of the dimension of the problem such that it will blow up numerically. To avoid this point, we rely on the vectorial quantization tree which builds a Markov chain with finitely many states.

2.5 Heuristic method for integer solution

In this paper, our main motivation consists of solving the LNG vessels routing problem. The decision variables describing how many cargos should be affected to a route are integers. The continuous problem presented in (2) can be considered as a continuous relaxation of the following integer problem describing the LNG vessel routing problem:

$$\begin{aligned}
\inf \quad & \mathbb{E}\left[\sum_{t=0}^{T-1} c_t(\boldsymbol{\xi}_t) \cdot u_t + g(\boldsymbol{\xi}_T, x_T)\right]\\
\text{subject to} \quad & u_t \in L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathbb{Z}^n \cap \mathfrak{U}_t),\\
& x_{t+1} = x_t + A_t u_t,\\
& x_0 = 0,\ x_T \in \mathfrak{X}_T \text{ almost surely;}
\end{aligned} \tag{15}$$

In order to assure the feasibility of the integer problem, $\mathfrak{U}^{ad} \cap \mathbb{Z}^{n \times T} \neq \emptyset$ is also assumed to be satisfied. As in section 2.3, we will define

$$\mathfrak{U}^{int,ad} := \left\{ (u_t) \,\middle|\, \sum_{t=0}^{T-1} A_t u_t \in \mathfrak{X}_T,\ u_t \in \mathfrak{U}_t \cap \mathbb{Z}^n, t = 0, \ldots, T-1 \right\}. \tag{16}$$

and

$$\mathfrak{U}_t^{int,ad}(x_t) := \left\{ u_t \,\middle|\, x_t + \sum_{s=t}^{T-1} A_s u_s \in \mathfrak{X}_T,\ u_s \in \mathfrak{U}_s \cap \mathbb{Z}^n, s = t, \ldots, T-1 \right\}. \tag{17}$$

Our heuristic method consists in finding the optimal integer control using the Bellman value obtained from the continuous (relaxation) problem:

$$\begin{aligned} Q(t, x_t, \boldsymbol{\xi}_t) = \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \mathcal{Q}^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_t) \\ \text{subject to} \quad & x_{t+1} = x_t + A_t u_t, \\ \text{(feasibility)} \quad & u_t \in \mathfrak{U}_t^{int,ad}(x_t), \\ \text{(optimality cut)} \quad & \begin{cases} \mathcal{Q}^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_t) \geq \mathbb{E}\left[Q^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1})\tilde{v}\mathcal{F}_t\right], \\ Q^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1}) \geq x^* \cdot x_{t+1} - Q^*(t+1, x^*, \boldsymbol{\xi}_{t+1}), \forall x^*. \end{cases} \end{aligned} \tag{18}$$

where $\mathcal{Q}^{**}(t+1, x_{t+1}, \boldsymbol{\xi}_t)$ is identical to its equivalent in (14) corresponding to the continuous relaxation formulation.

3 Vectorial Quantization Tree

The quantization methods were introduced in the early 1950's and have been applied in information science and signal processing. Recently, Bally and Pagès used this technique in order to discretize the probability space and later applied it to financial engineering [1, 2]. In this section, the main ideas and some important results of vectorial quantization are presented.

3.1 Optimal quantization

Optimal quantization of random vectors consists in finding the best possible approximation (in L^p norm sense) of a $\mathbb{R}^d-$valued random vector $\boldsymbol{\xi} \in L^p(\Omega, \mathcal{F}, \mathbf{P}; \mathbb{R}^d)$ by a measurable function $\phi(\boldsymbol{\xi})$ taking values in a finite set Γ of cardinality at most N values. Hence, we wish to solve the minimization problem:

$$\min \left\{ \|\boldsymbol{\xi} - \phi(\boldsymbol{\xi})\|_p \,\middle|\, \phi : \mathbb{R}^d \to \mathbb{R}^d, \text{measurable}, \#(\phi(\mathbb{R}^d)) \leq N \right\} \tag{19}$$

and let us denote by $\Gamma := \phi(\mathbb{R}^d)$ the set of quantized points. Let us denote the set of quantized points by $\hat{\boldsymbol{\xi}} := \phi(\boldsymbol{\xi})$. The above distance (19) is defined as the minimal *distortion* $D_N^{\mu,p} = \|\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}\|_p^p$ of quantization, where μ is the probability measure of $\boldsymbol{\xi}$. It is obvious to get that $\phi = \text{proj}_\Gamma$, where proj_Γ denotes the projection over the quantized points set $\Gamma = \left\{\xi^1, \ldots, \xi^n\right\}$ following the nearest neighbor rule of L^p distance.

We say that $C(\xi^i)$ is a *Voronoi tessellation* of N-tuple Γ if for all $\xi^i \in \Gamma$, $C(\xi^i)$ is a Borel set satisfying:

$$C(\xi^i) = \text{proj}_\Gamma^{-1}(\xi^i) \subset \left\{ \xi \,\middle|\, |\xi - \xi^i|_p = \min_{\xi^j \in \Gamma} |\xi - \xi^j|_p \right\}. \tag{20}$$

and we define the probability associated to ξ^i:

$$p^i = \mathbf{P}(\boldsymbol{\xi} \in C(\xi^i)) \tag{21}$$

Following the definition, a N-tuple has infinitely many Voronoi tessellations. However, each tessellation always has the same closure and the same boundary.

If the distribution of $\boldsymbol{\xi}$ is strongly continuous, i.e., the measure μ of any hyperplane is 0, we will not bother with boundaries and may think as if there was only one tessellation. This hypothesis is assumed to be satisfied in our application.

Therefore, the problem of finding the optimal quantization becomes a global optimization problem, which is in general difficult to solve. However, we can easily derive the following properties (e.g. [2, property 1-3]):

- The minimal distortion is decreasing with respect to N.
- $\lim_{N\to\infty} \min_{\hat{\boldsymbol{\xi}}\in\Gamma,\#(\Gamma)\leq N} \|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}\|_p = 0.$

One important property is the rate of the convergence to zero, which is referred as Zador's theorem.

Theorem 3.1. *[12, Theorem 6.2] If $\mathbb{E}[|\boldsymbol{\xi}|^{p+\eta}] < \infty$ for some $\eta > 0$, then,*

$$\lim_N \left(N^{p/d} \min_{\#(\Gamma)\leq N} \|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}\|_p^p \right) = J_{p,d} \left(\int |g|^{d/(d+p)}(u)du \right)^{1+p/d}$$

where $\mathbf{P}(du) = g(u)\lambda_d(du) + \nu, \nu \perp \lambda_d$ (λ_d Lebesgue measure on $\mathbb{R}^d$). The constant $J_{p,d}$ corresponds to the case of the uniform distribution on $[0,1]^d$.

We can obtain without difficulty the following corollary:

Corollary 3.2. $\min_{\hat{\boldsymbol{\xi}}\in\Gamma,\#(\Gamma)\leq N} \|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}\|_p = O(N^{1/d}).$

3.2 Quantization (recombining) tree

The quantization tree $(\hat{\boldsymbol{\xi}}_t)$ is a finite state Markov chain, such that at each stage t, $\hat{\boldsymbol{\xi}}_t$ is an optimal quantization of the original random variable $\boldsymbol{\xi}_t$ in the discrete time process.

Let us denote by N_t the number of quantization points at stage t, and set $N := \sum_{t=0}^T N_t$ such that $N_0 := 1$. We first choose the number of total quantization points N, which depends on the calculation capacity of the machine and on the required precision of the result.

The next step is to determine the number N_t of quantization points taken at each stage t. If $(\boldsymbol{\xi}_t) \in L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathbb{R}^d)$, following [2, proposition 8] in order to minimize the L^2 distortion of the whole process, we get that

$$N_t = \left\lceil \frac{a_t}{\sum_{\tau=0}^T a_\tau} N \right\rceil \quad \text{where} \quad a_t := \|\boldsymbol{\xi}_t\|_{d/(d+2)}^{d/2(1+d)} \qquad t = 0, \ldots, T. \tag{22}$$

We still need to evaluate the probability transitions for the Markov chain. Let p_t^{ij} be the transition probability from ξ_t^i to ξ_{t+1}^j, defined as

$$p_t^{ij} := \mathbf{P}\left[\mathrm{proj}_{\Gamma_{t+1}}(\boldsymbol{\xi}_{t+1}) = \xi_{t+1}^j \tilde{v} \mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t) = \xi_t^i\right] \tag{23}$$

In practice, an analytical computation of the probability transitions is out of reach for general distributions. One efficient way is to compute the transition by a Monte-Carlo method, i.e. p_t^{ij} is estimated by $\hat{p}_t^{ij}$

$$\hat{p}_t^{ij} := \frac{\sum_{m=1}^{M_p} \mathbf{1}_{\{\mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t^m)=\xi_t^i\}} \mathbf{1}_{\{\mathrm{proj}_{\Gamma_{t+1}}(\boldsymbol{\xi}_{t+1}^m)=\xi_{t+1}^j\}}}{\sum_{m=1}^{M} \mathbf{1}_{\{\mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t^m)=\xi_t^i\}}} \tag{24}$$

where M_p is the number of trajectories in Monte-Carlo simulation.

For the detail of quantization tree building, particularly the extended competitive learning vector quantization (CLVQ) algorithm, we refer to [2].

Remark 3.3. Pflug [24] considered the problem of generating a small size scenario tree (in order to be able to solve in an effective way stochastic programming problems) and set this problem in the framework of minimizing the distances between probability laws. In this paper we use a quantization approximation that also minimizes distances between probability laws, but in a different setting since we look for Markov-chain approximations.

In [24], Pflug studied a similar algorithm to discretize continuous-state processes into a non-recombining tree. But the quantization theory and CLVQ algorithm cover what he has proposed.

3.3 Dual dynamic programming formulation on quantization tree

The algorithm presented in this section for quantized problem is similar to the one in Mo et al. [22]. In the sequel we denote the quantized process by $(\hat{\boldsymbol{\xi}}_t)$, the associated filtration by $(\hat{\mathcal{F}}_t)$. And we represent the associated Bellman value by $\hat{Q}(t+1, x_{t+1}, \hat{\boldsymbol{\xi}}_t)$ to the problem similar to (2) in which random variables $(\boldsymbol{\xi}_t)$ are replaced by $(\hat{\boldsymbol{\xi}}_t)$. The corresponding dynamic programming principle reads

$$\begin{aligned} \hat{Q}(t, x_t, \hat{\boldsymbol{\xi}}_t) := \inf \quad & c_t(\hat{\boldsymbol{\xi}}_t) \cdot u_t + \hat{\mathcal{Q}}(t+1, x_{t+1}, \hat{\boldsymbol{\xi}}_t) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t^{ad}(x_t), \\ & x_{t+1} = x_t + A_t u_t; \end{aligned} \tag{25}$$

where $\hat{\mathcal{Q}}(\cdot,\cdot,\cdot)$ is the conditional expectation of the value at time $t+1$ obtained in state x_{t+1}:

$$\hat{\mathcal{Q}}(t+1, x_{t+1}, \hat{\boldsymbol{\xi}}_t) := \mathbb{E}\left[\hat{Q}(t+1, x_{t+1}, \hat{\boldsymbol{\xi}}_{t+1}) \tilde{v} \hat{\mathcal{F}}_t\right] \tag{26}$$

and

$$\hat{Q}(T, x_T, \hat{\boldsymbol{\xi}}_T) = \begin{cases} g(\hat{\boldsymbol{\xi}}_T, x_T) & \text{if } x_T \in \mathfrak{X}_T, \\ +\infty & \text{otherwise.} \end{cases} \tag{27}$$

In the algorithm to be studied below, only a finite number of affine lower bounds of the Bellman values $\hat{Q}(t, \cdot, \hat{\boldsymbol{\xi}}_t)$ are available. Denote by M_t^i the corresponding set for the realization of ξ_t^i. We see that the maximum of these affine lower bounds is itself a convex lower bound of the Bellman values $\hat{Q}(t, x_t, \boldsymbol{\xi}_t)$, denoted by :

$$\theta(t, x, \xi_t^i, M_t^i) := \max\left\{x^* \cdot x - e; \ \ (x^*, e) \in M_t^i\right\}. \tag{28}$$

And the associated conditional expectation appearing in the dynamic programming equation is then a lower bound of $\hat{\mathcal{Q}}(t+1, x_{t+1}, \boldsymbol{\xi}_t)$, when the Bellman values $\hat{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_{t+1})$ are replaced by the lower bounds $\theta(t+1, x, \xi_{t+1}^i, M_{t+1}^i)$:

$$\vartheta(t+1, \cdot, \xi_t^i, M_{t+1}) := \mathbb{E}[\theta(t+1, \cdot, \xi_{t+1}, M_{t+1}^j) \tilde{v} \hat{\mathcal{F}}_t] = \sum_{\xi_{t+1}^j \in \Gamma_{t+1}} \hat{p}_t^{ij} \theta(t+1, \cdot, \xi_{t+1}^j, M_{t+1}^j) \tag{29}$$

where $M_{t+1} := \left\{ M_{t+1}^j : \xi_{t+1}^j \in \Gamma_{t+1} \right\}$.

We may then write the resulting *approximate dynamic programming equation*, in which the Bellman value is replaced by its lower estimate, in the following form:

$$\begin{aligned} \inf \quad & c_t(\xi_t^i) \cdot u_t + \vartheta(t+1, x_{t+1}, \xi_t^i, M_{t+1}) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t^{ad}(x_t), \\ & x_{t+1} = x_t + A_t u_t, \\ & \begin{cases} \vartheta(t+1, x_{t+1}, \xi_t^i, M_{t+1}) = \sum_{\xi_{t+1}^j \in \Gamma_{t+1}} \hat{p}_t^{ij} \theta(t+1, x_{t+1}, \xi_{t+1}^j, M_{t+1}^j), \\ \theta(t+1, x_{t+1}, \xi_{t+1}^j, M_{t+1}^j) \geq x^* \cdot x_{t+1} - e, \ \forall (x^*, e) \in M_{t+1}^j. \end{cases} \end{aligned} \tag{30}$$

Starting from the initial state $x_0 = 0$, we begin by generating trajectories for $(\boldsymbol{\xi}_t)$, and for each of these trajectories, we solve the above linear program at each time step. Finally, we obtain a suboptimal policy (in the spirit of approximate dynamic programming). We will call this calculation a forward operation. Simultaneously, each time we solve (30), from the duality theory for linear programs, we obtain an affine lower bound of $\theta(t, \cdot, \hat{\boldsymbol{\xi}}_t)$ (and hence also of $\mathcal{Q}(t, \cdot, \hat{\boldsymbol{\xi}}_t)$). This is what we will call a backward elementary step. We perform a backward induction by propagating this information over a particular trajectory, from the final time to the initial one.

In the above discussion we do not use anymore the original (non quantized) random variable. However, it is sound, in order to obtain an upper bound of the optimal value, to use the original probability law for computing trajectories. At the same time, for computing the decision variable at each step, we solve the approximate dynamic programming problem where $\hat{\boldsymbol{\xi}}_t$ is taken as the projection of $\boldsymbol{\xi}_t$ over the quantized grid. The problem to be solved then is

$$\begin{aligned} \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \vartheta(t+1, x_{t+1}, \mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t) = \xi_t^i, M_{t+1}) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t^{ad}(x_t), \\ & x_{t+1} = x_t + A_t u_t, \\ & \begin{cases} \vartheta(t+1, x_{t+1}, \xi_t^i, M_{t+1}) = \sum_{\xi_{t+1}^j \in \Gamma_{t+1}} \hat{p}_t^{ij} \theta(t+1, x_{t+1}, \xi_{t+1}^j, M_{t+1}^j) \\ \theta(t+1, x_{t+1}, \xi_{t+1}^j, M_{t+1}^j) \geq x^* \cdot x_{t+1} - e, \ \forall (x^*, e) \in M_{t+1}^j. \end{cases} \end{aligned} \tag{31}$$

Remark 3.4. This formulation (31) corresponds to the multicut version in linear recourse problem, see [5, section 5.3].

3.4 Aggregated formulation

However, during the numerical tests, we came to realise that this formulation (31) was still too computationally intensive. We therefore provide an aggregated version of the above problem:

$$\begin{aligned} \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \vartheta(t+1, x_{t+1}, \mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t) = \xi_t^i, \mathcal{M}_{t+1}^i) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t^{ad}(x_t), \\ & x_{t+1} = x_t + A_t u_t, \\ & \vartheta(t+1, x_{t+1}, \xi_t^i, \mathcal{M}_{t+1}^i) \geq x^* \cdot x_{t+1} - e; \ \ \forall (x^*, e) \in \mathcal{M}_{t+1}^i, \end{aligned} \tag{32}$$

where the set $\mathcal{M}_{t+1}^i$ of affine lower bounds is obtained by aggregating the new lower bounds for each $\xi_{t+1}^j \in \Gamma_{t+1}$ (each time they are generated) into a single one taking into account the probability transitions:

$$\mathcal{M}_{t+1}^i := \left\{ (x^*, e) = \sum_{\xi_{t+1}^j \in \Gamma_{t+1}} \hat{p}_t^{ij}((x^*)^j, e^j),\ ((x^*)^j,\ e^j) \in M_{t+1}^j \right\}. \tag{33}$$

Therefore, this expression still provides lower bounds. It undoubtedly loses some information provided by optimality cuts in (31). Hence, more iterations will be required to converge (as proved in our numerical tests). But this formulation simplifies subproblem (31), and numerical tests indicate that it greatly reduces the CPU time.

Remark 3.5. This formulation (32) corresponds to the L-Shape method in linear recourse problem, see [5, section 5.1].

The version of heuristic method (18) for the integer solutions follows the same idea:

$$\begin{aligned} \inf \quad & c_t(\boldsymbol{\xi}_t) \cdot u_t + \vartheta(t+1, x_{t+1}, \mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t) = \xi_t^i, \mathcal{M}_{t+1}^i) \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t^{int,ad}(x_t), \\ & x_{t+1} = x_t + A_t u_t, \\ & \vartheta(t+1, x_{t+1}, \xi_t^i, \mathcal{M}_{t+1}^i) \geq x^* \cdot x_{t+1} - e;\ \forall (x^*, e) \in \mathcal{M}_{t+1}^i. \end{aligned} \tag{34}$$

However, it is only valid for the forward pass since we cannot obtain the dual information from a mixed integer programming problem.

4 Algorithm

After having discretized the probability space, it is natural to use the stochastic dual dynamic programming method on the resulting finite state Markov chain. Since the Bellman value (and its approximation by a collection of optimality cuts) is non-convex with respect to the underlying random variable, we need to calculate the cuts on every vertex of the quantization tree. The algorithm iterates over a forward pass and a backward pass. The forward pass generates a set of random trajectories following the dynamic of Markov chain (1) and runs the optimization on the fly on these trajectories. It provides an upper bound of the optimal value. The backward pass updates the optimality cuts on each vertex and computes a lower bound at the first stage.

4.1 Forward pass

The forward pass generates trajectories $\boldsymbol{\xi}^m, m = 1, \ldots, M_f$ following the dynamic of Markov chain (1), i.e. out of sample for quantization tree. For each trajectory, starting from the first stage ($x_0 = 0$), the current decision u_t and the state variable x_{t+1} for the next stage are computed by solving (32).

After having computed all the trajectories, an objective value v^m for each trajectory $m = 1, \ldots, M_f$ is produced. Thus, the forward statistic can be ob-

tained as follows:

$$\overline{v} = \frac{1}{M_f}\sum_{m=1}^{M_f} v^m \quad \text{and} \quad s = \frac{1}{M_f}\sqrt{\sum_{m=1}^{M_f}(v^m - \overline{v})^2}.$$

Remark 4.1. Also notice that the control solution (u_t) computed in this algorithm is sub-optimal since we approximate $Q(t, x_t, \boldsymbol{\xi}_t)$ by $\theta(t, x_t, \hat{\boldsymbol{\xi}}_t, M_t)$ (resp. $\mathcal{Q}(t+1, x_{t+1}, \boldsymbol{\xi}_t)$ by $\vartheta(t+1, x_{t+1}, \hat{\boldsymbol{\xi}}_t, \mathcal{M}_{t+1})$). However, (u_t) belongs to the feasible space (U_t^{ad}) (it necessarily belongs to L^2 since $\mathfrak{U}_t$ is compact), hence is feasible for the original problem. As it is a minimization problem, each v^m is sub-optimal, which implies that $\overline{v}$ is an upper bound in the statistical sense for the original problem, the 95% asymptotic confidence interval is given by $[\overline{v} - 1.96s, \overline{v} + 1.96s]$.

Remark 4.2. Like any Monte-Carlo type methods, the convergence rate is $1/\sqrt{M_f}$, which is slow. In addition, the number of subproblems to solve in the forward pass is proportional to the number of forward pass trajectories M_f. Therefore, some variance reduction techniques are very useful here. In practice, we use a quite efficient control variate technique, see section 5.

Remark 4.3. Since the calculation of each trajectory is independent, it is possible to use parallel computing in order to speed up the computation.

4.2 Backward pass

The goal of the backward pass is to update the optimality cuts used in (32). It follows the quantization tree, i.e. in sample calculation. We still use the formulation of sub-problem (32), where $\hat{\boldsymbol{\xi}}_t \in \Gamma_t$ and x_t have been stored during the previous forward pass. Once the sub-problem (32) has been solved, the optimality cut is computed from the dual value associated to the dynamic equation $x_{t+1} = x_t + A_t u_t$. We then add this cut into the optimality cuts collection of vertex M_t^i.

Remark 4.4. Optimality cuts have to be updated for each vertex of the quantization tree due to the non-convexity of $Q(t, x_t, \boldsymbol{\xi}_t)$ (resp. $\hat{Q}(t, x_t, \hat{\boldsymbol{\xi}}_t)$) with respect to $\boldsymbol{\xi}_t$ (resp. $\hat{\boldsymbol{\xi}}_t$). The number of sub-problems to solve in the backward pass is proportional to the quantization tree size N times the number of trajectories denoted M_b. Since the state variable values are stored during the forward pass, M_b is always smaller than M_f. But if we are using all M_f trajectories for the backward pass (M_f is relatively large owing to remark 4.2), the number of sub-problems would be $N \times M_f$ which is computationally unmanageable. So, one idea is to select only a small subsample of M_b samples to be computed in the backward pass in order to allow the program to finish in a reasonable time. Another reason is that in practice, some state values x_t are very close to each other, causing the sub-problems to return almost the same dual information which will not help us much approximate well $\hat{Q}(t, x_t, \hat{\boldsymbol{\xi}}_t)$.

However, the process of selecting M_b trajectories among the M_f ones depends on the nature of the problem on which the algorithm is applied. In the numerical tests in section 5, the selection follows the rejection method which consists in checking whether the distance between the new x_t and the existing ones is below a predefined threshold or not. This is a random way of satisfying the requirement that they are not too close to each other.

Remark 4.5. Similarly to the forward pass, the backward pass algorithm may also use parallel computation, because at stage t, the sub-problems (31) using different (x_t, ξ_t^i) are independent from each other.

After solving the sub-problem at the first stage, an optimal value $\underline{v}$ is obtained, which is a lower bound for the quantized problem, since this value is calculated by Benders decomposition. However, $\underline{v}$ is calculated according to the quantization tree, and would differ if calculated on another tree. The error caused by this approximation can be bounded, as discussed in the next subsection.

Remark 4.6. The stopping condition is to check whether $\underline{v} \in [\overline{v}-1.96s, \overline{v}+1.96s]$. However, the confidence interval is sometimes too large making this condition too straightforward to be satisfied. To avoid this problem, we add another stopping condition stating that the backward pass value $\underline{v}$ should be stable: $|\underline{v}^{it} - \underline{v}^{it-1}| \leq \epsilon |\underline{v}^{it}|$, where $it \in \mathbb{Z}_{++}$ is the number of iteration of the algorithm.

Recently, Shapiro has exposed in [28] that the above stopping condition is relatively weak if the confidence interval is large. He proposed that the upper bound is $\overline{v} + 1.96s$ and the following stopping condition:

$$\underline{v} \in [\overline{v} + 1.96s - \varepsilon, \overline{v} + 1.96s + \varepsilon] \tag{35}$$

where ε is a parameter.

In this article, we take the stopping conditions

$$\underline{v} \in [\overline{v} - \varrho s, \overline{v} + \varrho s] \tag{36}$$

where ϱ is a parameter which is stricter than the one proposed by Pereira and Pinto [23].

4.3 Convergence result and error analysis on quantization

In this section, we establish the convergence result of the above algorithm and provide an error analysis of discretization by quantization.

We first show the convergence of the algorithm when the forward pass follows the discretized distribution – quantization tree. As we have noticed in remark 3.4, the algorithm is equivalent to L-Shape method. The convergence result on quantized forward pass follows 2 arguments.

- The convergence result of L-Shape method on finite probability distribution is given in Birge and Louveaux [5, chapter7, theorem 1], where the main argument is finiteness of optimality cuts and feasibility cuts. Thus, after generating all feasible cuts and optimality cuts, the backward pass value does not increase anymore.

- Since now the forward pass follows discretized distribution, the forward value of Monte Carlo method is itself a random variable whose mean is the upper bound of L-Shape method which converges to lower bound by finite convergence of L-Shape method.

Remark 4.7. In practice, we can first simulate the forward pass following the discretized distribution and let the forward value and backward value converge. Then we can change the forward simulation to following original distribution to compute the final result.

Now, we study the convergence of the upper bound and the lower bound once we let forward pass follow continuous distribution. In order to estimate the error caused by quantization, we refer to the result in Heitsch and Römisch [14, 15].

Theorem 4.8. *[14, Theorem 2.1] Assume that the solution set $\mathfrak{U}^{ad}$ is nonempty, and bounded. Then, there exists positive constants L and δ such that the estimate*

$$|v(\boldsymbol{\xi}) - v(\hat{\boldsymbol{\xi}})| \leq L(\|\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}\|_2 + D_f(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})) \tag{37}$$

holds for all random element $\hat{\boldsymbol{\xi}} \in L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; \mathbb{R}^d)$ with $\|\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}\|_2 \leq \delta$, where

$$D_f(\boldsymbol{\xi}_t, \hat{\boldsymbol{\xi}}_t) = \inf_{u \in S(\boldsymbol{\xi}), \hat{u} \in S(\hat{\boldsymbol{\xi}})} \sum_{t=2}^{T-1} \max\left\{\|u_t - \mathbb{E}[u_t \tilde{v} \mathcal{F}_t(\hat{\boldsymbol{\xi}})]\|_2, \|\hat{u}_t - \mathbb{E}[\hat{u}_t \tilde{v} \mathcal{F}_t(\boldsymbol{\xi})]\|_2\right\}, \tag{38}$$

where $S(\boldsymbol{\xi})$ (resp. $S(\hat{\boldsymbol{\xi}})$) is the optimal solution of with random variable $\boldsymbol{\xi}$ (resp. $\hat{\boldsymbol{\xi}}$), and $v(\boldsymbol{\xi})$ (resp. $v(\hat{\boldsymbol{\xi}})$) are the related optimal values.

Since $(\hat{\boldsymbol{\xi}}_t)$ is the quantization of $(\boldsymbol{\xi}_t)$, then $\mathcal{F}_t(\hat{\boldsymbol{\xi}}) \subset \mathcal{F}_t(\boldsymbol{\xi})$. The optimal control $\hat{u}_t$ is $\mathcal{F}_t(\hat{\boldsymbol{\xi}})$ measurable, hence it is $\mathcal{F}_t(\boldsymbol{\xi})$ measurable. Then, we have for (38):

$$D_f(\boldsymbol{\xi}_t, \hat{\boldsymbol{\xi}}_t) = \inf_{u \in S(\boldsymbol{\xi}), \hat{u} \in S(\hat{\boldsymbol{\xi}})} \sum_{t=2}^{T-1} \|u_t - \mathbb{E}[u_t \tilde{v} \mathcal{F}_t(\hat{\boldsymbol{\xi}})]\|_2 \tag{39}$$

Our framework is a little different from the problem studied in [14, 15], since our problem possesses a final cost function $g(\boldsymbol{\xi}_T, x_{T+1})$. However, their result is still suitable in our framework if the Lipschitz property of this function holds. We will omit the proof, which is similar to that in [14].

This error bound consists in 2 terms, the first one of the right hand side (37) is the l^2-distance between the original random process and the approximated one, the second one is the distance between the 2 filtrations over the optimal control. The first term can be measured by theorem 3.1, which provides the convergence rate to 0. However, it is very difficult to provide a convergence rate to (39).

Remark 4.9. There are other references which establish stability results of optimal value of control Markov process. Let us mention Gordienko et al. [11] on average cost Markov control process based on Kantorovich distance and the references therein. However, their problem is to minimize an average or discounted cost for an infinite horizon problem. Therefore, applying their analyst in our algorithm is not obvious.

4.4 Algorithm

The whole algorithm is summarized in figure 1.

5 Numerical tests

In this section, two numerical tests are performed. The first one is a swing option, which is an example where both state variable and random variable are scalar. The second one concerns a LNG portfolio optimization, which is far more complex. The algorithm has been successfully applied on both examples.

Step 0 **Construction quantization tree (possible off-line)**
build one quantization tree and compute transition probability.
Step 1 **Initialisation**
let $u_0 \in \mathfrak{U}_0^{ad}$ the current optimal solution and compute state variable value x_1;
let $\underline{v}^0 = -\infty$.
Step 2 **Forward pass**
create M_f diffusion processes following dynamic (1);
for $t = 1, \ldots, T-1$ do:
  for $m = 1, \ldots, M_f$ do:
    calculate $\hat{\boldsymbol{\xi}}_t^m = \mathrm{proj}_{\Gamma_t}(\boldsymbol{\xi}_t^m)$,
    solve the subproblem (32);
compute the forward statistic $\overline{v}$ and s.
Step 3 **Backward pass**
for $t = T-1, \ldots, 1$ do:
  for $m = 1, \ldots, M_b$ do:
    solve the subproblem (32) for all vertices in Γ_t;
    calculate new optimality cut and add it into M_t;
$t = 0$(first stage)
  solve the first stage problem and calculate the backward value $\underline{v}^{it}$.
Step 4 **Check stopping condition**
if $\underline{v} \in [\overline{v} - 1.96s, \overline{v} + 1.96s]$ and $|\underline{v}^{it} - \underline{v}^{it-1}| \leq \epsilon |\underline{v}^{it-1}|$
  STOP;
else
  go to Step 2.

Figure 1: Algorithm

5.1 Price model

In this paper, we assume that the random process, taking part in the modeling of futures commodity price evolution, follows the celebrated Black model [6]. Although simplistic, this model is equivalent to the one factor Gaussian HJM model (see [7]) which is well adapted for forward curve modeling. This model is discussed in the practice-based literature, Eydeland and Wolyniec [9, Chapter 5], and Lai et al. [10]. Since we are using the vectorial quantization method to discretize the underlying random process, we are not limited to that model. Other financial price models can also be applied.

The price model takes as parameters the market volatilities σ as well as the forward curves F_0. The Black model states that future prices of commodities are martingales whose expectations correspond to the original forward curve level F_0 due to lack of arbitrage opportunities. The forward contract price F_s^t at time s, with maturity (also called tenor) t, follows under the equivalent martingale measure (EMM) $\mathbf{P}$ the following dynamic:

$$\ln \frac{(F_{t+1}^s)^i}{(F_t^s)^i} = \sigma^i W_t^i - \frac{1}{2}(\sigma^i)^2 \quad i = 1, \ldots, d; \tag{40}$$

where W_t^i is a standard normal distribution $\mathcal{N}(0,1)$, with correlation $\mathrm{corr}(W_t^i, W_t^j) = \rho_{ij}$.

Then, the spot prices at time t are simply obtained by taking the t-expiring forward contracts at time t: $\boldsymbol{\xi}_t = F_t^t$,

$$\boldsymbol{\xi}_t^i := (F_t^t)^i = (F_0^t)^i \exp\left(\sum_{s=0}^{t-1} \sigma^i W_s^i - \frac{1}{2}(\sigma^i)^2 t\right) \quad i = 1, \ldots, d; \tag{41}$$

where (F_0^t) is the original forward curve observed. Setting

$$S_t^i := \exp\left(\sum_{s=0}^{t-1} \sigma^i W_s^i - \frac{1}{2}(\sigma^i)^2 t\right),$$

we then have $\boldsymbol{\xi}_t^i = (F_0^t)^i S_t^i$.

5.2 Swing option

Swing options are fairly common in the energy market. It allows its holder to purchase a total amount of commodity during a fixed time period. A basic swing option contract can be formulated as:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\sum_{t=0}^{T-1} (\boldsymbol{\xi}_t - K_t) \cdot u_t\right] \\ \text{subject to} \quad & u_t \in L^2(\Omega, (\mathcal{F}_t), \mathbf{P}; [0,1]) \\ & \sum_{t=0}^{T-1} u_t \in [L, U] \text{ almost surely} \end{aligned} \tag{42}$$

where K_t is the strike price, assumed to be deterministic, L and U are lower and upper bounds of the final condition.

Let us first flip the sign of the objective function to $(K_t - \boldsymbol{\xi}_t)$ in order to switch to a minimization problem. In this test, we consider that the strike price equals to the original forward price observed nowadays $K_t = F_0^t$ without loss of generality.

In the case where $(L, U) \in \mathbb{Z}^2$, Bardou et al. [3] show that the optimal control has bang-bang property, i.e. $u_t \in \{0, 1\}$. Thus, dynamic programming can be directly applied in this framework by discretizing the state space x_t. Bardou et al. have applied this idea to discretize the state variable and use a quantization tree to approximate the underlying random space. However, in the presence of more complex conditions on the control variable such as $u_t \in [l(x_t), u(x_t)]$ (like in most of storage capacities technical features), the bang-bang property will not hold anymore, and furthermore the discretization of the state space x_t may become intractable.

5.2.1 Control variate

Since the forward pass produces a large confidence interval with a small M_f, the following control variate technique has been introduced to reduce it. Let us consider the following variable:

$$v_{cv}(\boldsymbol{\xi}) = \sum_{t=0}^{T-1} (K_t - \boldsymbol{\xi}_t). \tag{43}$$

Observe that $E[v_{cv}(\boldsymbol{\xi})] = 0$ since $\mathbb{E}[\boldsymbol{\xi}_t] = F_0^t = K_t$.

5.2.2 Numerical result

In the following test, without losing any generality, we take $F(0,t) = 1$, $\forall t = 0, \ldots, T-1$ in the price model (40). The results are compared to another method using bang-bang strategy and the Longstaff-Schwartz method [21]. During the simulation, the quantization tree has 3000 points in total. We set parameters $M_f = 1000$ and $M_b = 10$ in Monte-Carlo simulation. We use 1000 scenarios in the Longstaff-Schwartz method.

$T = 50$		$[L,U] = [0,25]$			$[L,U] = [20,30]$		
	σ	$0.5/\sqrt{T}$	$0.75/\sqrt{T}$	$1.0/\sqrt{T}$	$0.5/\sqrt{T}$	$0.75/\sqrt{T}$	$1.0/\sqrt{T}$
SDDP.	$\underline{v}$	4.31	6.43	8.49	1.96	2.93	3.88
	c.i.	$[4.04, 4.40]$	$[5.91, 6.51]$	$[7.86, 8.62]$	$[1.77, 1.97]$	$[2.66, 3.00]$	$[3.68, 4.16]$
BB+LSM.	c.i.	$[4.19, 4.47]$	$[6.23, 6.71]$	$[8.22, 8.94]$	$[1.81, 2.17]$	$[2.69, 3.31]$	$[3.57, 4.42]$

Table 1: Quantization tree + dual dynamic programming versus bang-bang strategy + Longstaff Schwartz method for pricing swing option.

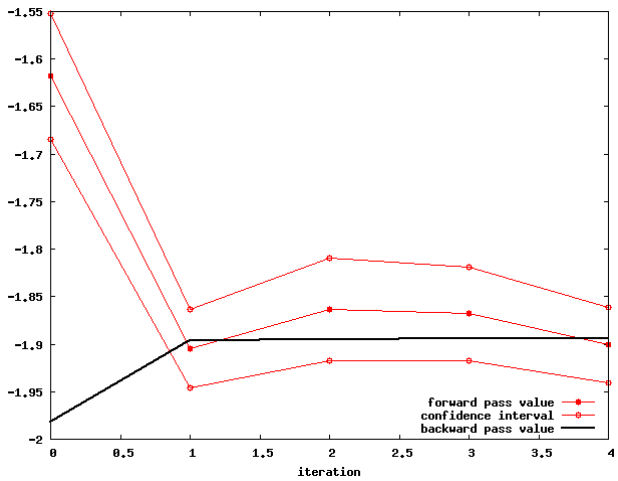

Figure 2: SDDP iteration for pricing a swing option using quantization method. $[L,U] = [20,30]$, $\sigma = 0.5/\sqrt{50}$

Figure 2 shows the convergence between the upper and lower bounds for the swing option pricing problem. It only takes 5 iterations for the algorithm to converge which represents less than 1 minute of CPU time. Notice also that the standard deviation associated to the upper bound is less than 1% of the optimal value. Compared to other numerical methods, we see that there is no real advantage in using this method in terms of computation time. However, the example in the next section will evidence its superiority over the other algorithms which fail to provide accurate results in a reasonable time.

5.3 Dynamic portfolio optimization

We now consider a gas trading portfolio. A trading company purchases liquefied natural gas (LNG.) from a set of producing countries indexed at a price formula and sells it to consuming countries at another other price formula (see figure 3 for the main market and table 2 for the price formulae). Annual quantity and price formulae have been agreed contractually, the latter are functions of the future prices of major energy markets $\boldsymbol{\xi}_t$, such as crude oil price (OIL), north American natural gas price (NA NG), and Europe natural gas price (EU NG). The goal is to find a strategy that optimizes the portfolio value. The decision is made over the quantity of LNG transported on each route. *At that point, we relax the discrete nature of the decision variables.* The first difficulty lies in the number of indices we have to consider when we model the portfolio: $\boldsymbol{\xi}_t \in \mathbb{R}^3$. The second difficulty is linked to the dimension of the state variable $x_t \in \mathbb{R}^m$, corresponding to the total number of production and delivery countries, which is also in high dimension, $m = 6$ in the simulation.

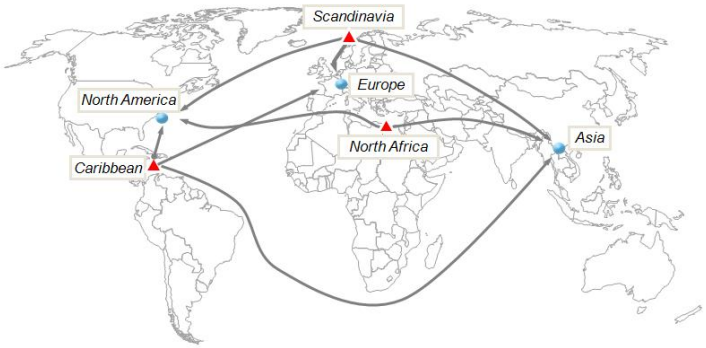

Figure 3: A fictive supply and demand portfolio, as well as the possible routes. ▲: producing country; •: consuming country.

Port	Cargo Size	Annual QC.*	Monthly QC.*	Price formula**
Caribbean	3.0	[48.0, 54.0]	[0.0, 6.0]	NA NG − 0.1
Scandinavia	3.0	[24.0, 30.0]	[0.0, 3.0]	$\begin{cases} 0.05\text{OIL} + 2.5 & \text{if OIL} \leq 75 \\ 0.07\text{OIL} + 1.0 & \text{otherwise} \end{cases}$
North Africa	4.0	[100.0, 100.0]	[0.0, 12.0]	$\begin{cases} 0.9\text{NA NG} + 0.4 & \text{if NA NG} \leq 5 \\ 0.8\text{NA NG} + 0.9 & \text{otherwise} \end{cases}$
North American	3.0, 4.0	[84.0, 88.0]	[0.0, 8.0]	NA NG
Europe	3.0, 4.0	[68.0, 76.0]	[0.0, 8.0]	EU NG
Asia	3.0, 4.0	[20.0, 20.0]	[0.0, 4.0]	0.08OIL − 0.8

Table 2: Constraints and price formulae

* The quantity unit is in TBtu. MMBtu stands for a million British thermal unit, a TBtu is a trillion British thermal unit thus equivalent to a Million MMBtu.

** The price unit is \$/MMBtu.

So, the problem is

$$\begin{aligned} \sup \quad & \mathbb{E}\left[\sum_{t=0}^{T-1} \tilde{c}_t(\boldsymbol{\xi}_t) \cdot u_t\right] \\ \text{subject to} \quad & u_t \in U_t \quad \text{(monthly constraint)} \\ & \sum_{t=0}^{T-1} A_t u_t \in \mathfrak{X}_T \quad \text{(annual constraint)} \end{aligned} \tag{44}$$

Let us flip the sign of the objective function: $c_t = -\tilde{c}_t$ to switch to a minimization problem.

The forward price F_0^t in (40) is read from the energy market. Here is the value used in numerical test (see figure 4 - 5).

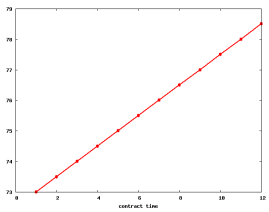

Figure 4: OIL price (US\$ /barrel)

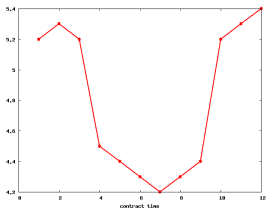

Figure 5: EU NG price (US\$ / MMBtu)

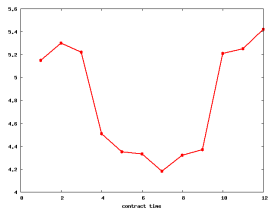

Figure 6: NA NG price (US\$ / MMBtu)

The quantization tree was processed with $N = 36000$ $\mathbb{R}^3$-valued elementary quantizers dispatched on $T = 12$ stages. The estimation of transition probability was carried out by Monte-Carlo method using 10^9 samples.

5.3.1 Control variate

Similarly to the previous example, we keep using a control variate to reduce the large confidence interval. Let us consider the following variable:

$$v_{cv}(\boldsymbol{\xi}) = \sum_{t=0}^{T-1} c_t(\boldsymbol{\xi}_t) u_t^* \tag{45}$$

where u_t^* is one of the optimal solutions for problem

$$\begin{aligned} \inf \quad & \sum_{t=0}^{T-1} \mathbb{E}\left[c_t(\boldsymbol{\xi}_t)\right] u_t \\ \text{subject to} \quad & u_t \in \mathfrak{U}_t \quad \sum_{t=0}^{T-1} A_t u_t \in \mathfrak{X}_T \end{aligned} \tag{46}$$

Let v_{cv}^* be the optimal value.

This is a deterministic linear program, which is straightforward to solve. The only technical point is to evaluate $\mathbb{E}[c_t(\boldsymbol{\xi}_t)]$. One possible method consists in using quantization:

$$\mathbb{E}[c_t(\boldsymbol{\xi}_t)] \approx \sum_{\xi_t^i \in \Gamma_t} p_t^i c_t(\xi_t^i)$$

Clearly we have $\mathbb{E}[v_{cv}(\boldsymbol{\xi})] = v_{cv}^*$.

Remark 5.1. The (46) is nothing else but the deterministic counterpart formulation, whose value is also known in financial engineering as the intrinsic value of the option. Thus the optimal value v_{cv}^* is considered as a good reference to compare the optimal value obtained by stochastic programming.

5.3.2 Results

The parameters driving the stochastic processes in the test are set as follows: volatility $\sigma_1 = \sigma_2 = \sigma_3 = 40\%$, and correlation $\rho_{12} = 0.7$ $\rho_{13} = 0.2$ $\rho_{23} = 0.4$. And the parameters driving the SDDP part of the algorithm are forward pass samples $M_f = 3000$ to 6000, backward pass sample $M_b = 8$ to 12. The program is written in C++ with Cplex 10.1, and the tests were performed on a PC with a 2.2 GHz Dual Core AMD CPU and 16GByte main memory.

The numerical results are shown in Figure 7.

Notice that the value of the backward pass is almost stable from the 6th iteration, and enters the confidence interval of the upper bound several times. The stopping condition taken here is (36) proposed by Shapiro in [28] where we take $\varrho = 0.5$. Furthermore, the confidence interval of the upper bound generated with control variate represents circa 4% of the forward pass value. This shows that the control variate proves to be adequate in this example. Moreover, the algorithm converges in 17 iterations, which is efficient. However, if we test this algorithm with larger volatilities, or with more producing/consuming countries, more iterations will be required to converge.

Below are the graphs representing the CPU time for each iteration (figure 8 and figure 9):

This program does not consume a lot of time during each iteration. However the calculation time increases sharply when the iteration goes on. We know that

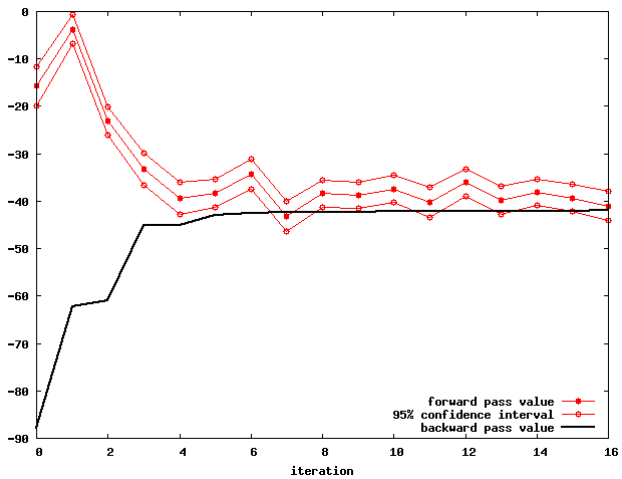

Figure 7: SDDP iteration result for portfolio optimization.

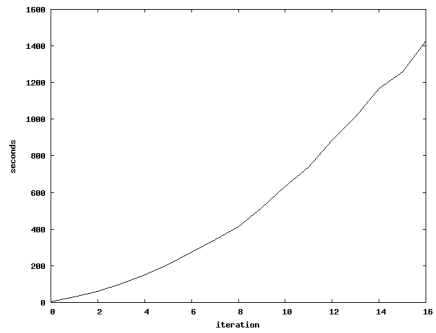

Figure 8: forward CPU time

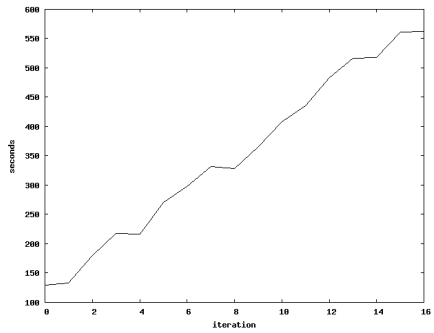

Figure 9: backward CPU time

the forward computation time is proportional to the number of forward samples M_f, and that the backward computing time is proportional to the number of vertices in the quantization tree N and to the number of backward samples M_b. Hence, we can modify these parameters to reduce the calculation time when it becomes too long. But if we do so, we will lose the precision by enlarging the confidence interval.

Integer solution As explained in section 2.5, we can compute one sub optimal integer solution by combining the Bellman values of the continuous relaxation with the integer constraints on the decision variables during the forward step.

The results are presented in table 3. The control variate value of -8.3946 is obtained when solving (46) taking account of the integer constraints on the control variable. The heuristic gives an expected value of -13.4142, with standard deviation of 0.93317, for 5000 forward samples.

As mentioned in remark 5.1, the deterministic counterpart value gives a good insight on the gain obtained by using stochastic programming. In the contin-

	upper bound	standard deviation of u.b.	lower bound	intrinsic value
Continuous	-40.9976	1.56231	-42.0263	-8.9247
Integer	-13.4142	0.93317		-8.39455

Table 3: Comparison of optimal value of continuous problem and integer problem

uous relaxation case, the optimal value got through stochastic programming is about 5 times better than the deterministic counterpart value. Concerning the integer variable problem, although only an upper bound could be generated, it is nevertheless about 50% better than the one of the deterministic counterpart. Therefore we conclude that the optimal value of the stochastic integer program lies somewhere between -41.00 (continuous relaxation optimal value) and -13.41 (integer sub-optimal value). Reducing this interval is the object of our future work.

Acknowledgments The authors thank to two anonymous referees for their useful remarks, in particular by bringing our attention to papers Mo et al. [22] and Gordienko et al. [11].

References

[1] V. Bally and G. Pagès. Error analysis of the quantization algorithm for obstacle problems. *Stochastic Processes & Their Applications*, 106(1):1–40, 2003.

[2] V. Bally and G. Pagès. A quantization algorithm for solving discrete time multi-dimensional optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003.

[3] O. Bardou, S. Bouthemy, and G. Pagès. Optimal quantization for the pricing of swing options. *Applied Mathematical Finance*, 16(2):183–217, 2009.

[4] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

[5] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer, New York, 2000.

[6] F. Black. The pricing of commodity contracts. *Journal of Financial Economy*, 3:167–179, 1976.

[7] R.A. Jarrow D.C Heath and A. Morton. Bond pricing and the term structure of interest rates : a new methodology for contingent claim valuation. *Econometrica*, 60:77–105, 1992.

[8] J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming–an approach using probability metrics. *Math. Program. Ser. A*, 95:493–511, 2003.

[9] A. Eydeland and K. Wolyniec. *Energy and Power Risk Management: New Developments in Modeling, Pricing and Hedging.* John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[10] G.Lai, F. Margot, and N. Secomandi. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Oper. Res.*, 58:564–582, May 2010.

[11] Evgueni Gordienko, Enrique Lemus-RodrÃguez, and RaÃºl Montes-de Oca. Average cost markov control processes: stability with respect to the kantorovich metric. *Mathematical Methods of Operations Research*, 70:13–33, 2009. 10.1007/s00186-008-0229-6.

[12] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions.* Springer-Verlag, Berlin, 2000.

[13] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Opti. and Appli.*, 24:187–206, 2003.

[14] H. Heitsch, W. Römisch, and C. Strugarek. Stability of multistage stochastic programs. *SIAM J. on Optimization*, 17:511–525, 2006.

[15] Holger Heitsch and Werner Römisch. Stability and scenario trees for multistage stochastic programs. In Gerd Infanger, editor, *Stochastic Programming*, volume 150 of *International Series in Operations Research & Management Science*, pages 139–164. Springer New York, 2011.

[16] K. Hoyland, M. Kaut, and S.W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24:169–185, 2003.

[17] G. Infanger and D.P. Morton. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.

[18] P. Kall and S.W. Wallace. *Stochastic Programming.* John Wiley & Sons, Chichester, 1994.

[19] M. Kaut and S.W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257 – 271, 2007.

[20] C. Küchler and S. Vigerske. Decomposition of multistage stochastic programs with recombining scenario trees.

[21] F. Longstaff and E. Schwartz. Valuing american options by simulation : a simple least squares approach. *Review of Financial Studies*, 14:113–148, 2001.

[22] B. Mo, A. Gjelsvik, and A. Grundt. Integrated risk management of hydro power scheduling and contract management. *IEEE Transactions on Power Systems*, 16(2):216–221, 2001.

[23] M.V.F. Pereira and L.M.V.G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.

[24] G.Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Math. Program. Ser. B*, 89:251–271, 2001.

[25] A.B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operation Research Letters*, 36:450–455, 2008.

[26] R.T. Rockafellar and R. J-B. Wets. *Variational analysis*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag, first edition, 1997.

[27] B. Van Roy and J.N. Tsitsiklis. Regression methods for pricing complex american-style options. *IEEE transactions on Neural Networks*, 12(4):694–703, 2001.

[28] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.

[29] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming : Modelling and Theory*. SIAM, Philadelphia, 2009.

[30] J.N. Tsitsiklis. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997.

[31] J.N. Tsitsiklis and B. Van Roy. Optimal stopping of Markov processes : Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE transactions on automatic control*, 44(10):1840–1851, 1999.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399