HAL
archives-ouvertes.fr

A comparison of sample-based Stochastic Optimal Control methods

Pierre Girardeau

To cite this version:

Pierre Girardeau. A comparison of sample-based Stochastic Optimal Control methods. 2010. <hal-00454394>

HAL Id: hal-00454394

https://hal.archives-ouvertes.fr/hal-00454394

Submitted on 8 Feb 2010

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.

A COMPARISON OF SAMPLE-BASED STOCHASTIC OPTIMAL CONTROL METHODS

PIERRE GIRARDEAU

Abstract. In this paper, we compare the performance of two scenario-based numerical methods to solve stochastic optimal control problems: scenario trees and particles. The problem consists in finding strategies to control a dynamical system perturbed by exogenous noises so as to minimize some expected cost along a discrete and finite time horizon. We introduce the Mean Squared Error (MSE) which is the expected L^2-distance between the strategy given by the algorithm and the optimal strategy, as a performance indicator for the two models. We study the behaviour of the MSE with respect to the number of scenarios used for discretization. The first model, widely studied in the Stochastic Programming community, consists in approximating the noise diffusion using a scenario tree representation. On a numerical example, we observe that the number of scenarios needed to obtain a given precision grows exponentially with the time horizon. In that sense, our conclusion on scenario trees is equivalent to the one in the work by Shapiro (2006) and has been widely noticed by practitioners. However, in the second part, we show using the same example that, by mixing Stochastic Programming and Dynamic Programming ideas, the particle method described by Carpentier et al (2009) copes with this numerical difficulty: the number of scenarios needed to obtain a given precision now does not depend on the time horizon. Unfortunately, we also observe that serious obstacles still arise from the system state space dimension.

Introduction

Consider a controlled dynamical system, affected by some exogenous noises, say uncertain parameters for instance. Stochastic optimal control consists in driving this system, having at each time step partial or total observations of those noises, so as to minimize some expected cost integrated through time and while satisfying a number of constraints. Many applications can be handled through such a model. For instance, think of a power producer that has to plan the use of a set of power plants (the control would be the production of the plants at each time step) in order to supply an uncertain power demand (the noises) while minimizing a production cost over a certain time period. We here consider discrete and finite time horizon. We are hence looking for feedback functions which, at each time step and for each possible observation of the system, provide a decision to be taken. We consider the laws of the random variables involved to be continuous; consequently this is an infinite-dimensional optimization problem. In most situations, no analytic solution can be found and one has to consider approximations of the original problem to be able to solve it numerically.

Date: February 8, 2010.

2010 *Mathematics Subject Classification.* Primary 93E20; Secondary 49M05, 49L20.

Key words and phrases. Stochastic optimal control, Stochastic programming, Particle methods.

The author would like to thank his PhD supervisors Pierre Carpentier and Guy Cohen for their helpful remarks on previous versions of this paper.

In the manner of the Monte Carlo approach for computing the expectation of a random variable, we seek a tractable approximation of the probabilistic structure of the original problem based on samples of some noise scenarios. A scenario consists in a sample of the noises along the whole time horizon. Note that such a resolution method is stochastic in itself. Indeed, the solution given by such a method is a random variable in the sense that it depends on the scenarios sampled and used in the resolution. Therefore, when studying the performance of an algorithm based on such an approach, a variance term naturally appears, representing the sensitivity of the solution regarding these scenarios. This variance term is added to the squared bias term which arises from the necessity to approximate the solution in feedback by a function in a predefined space. Note that it is also present in the performance evaluation of deterministic techniques, like Dynamic Programming (Bertsekas, 2000) for instance.

For stochastic optimal control problems, it is common to represent the diffusion of "likely futures" using a scenario tree structure, leading to so-called multistage stochastic programs. This kind of representation goes back to Dantzig (1955) and has been widely studied within the Stochastic Programming community (see Prékopa, 1995; Shapiro and Ruszczynski, 2003, for a broad overview of this approach). This methodology consists in discretizing the probabilistic structure of the problem, then rewriting the constraints and objective function of the original problem on this discrete structure and finally solving it using a well-suited mathematical programming technique. In other words, the problem is solved for a particular sample of the random variables (a scenario tree) and the solution is hence a random variable in itself. We have to keep this fact in mind while evaluating the error. The main interest of such a methodology is that it leads back to a deterministic problem, on which we can apply classical tools of numerical optimization (Bonnans et al, 2006). For instance, one may then try to solve large-scale problems using decomposition techniques (see Carpentier et al, 1995; Higle and Sen, 1996; Shapiro and Ruszczynski, 2003, Ch. 3).

Despite the benefits of this methodology, it has been already observed that, in order to obtain a given precision, the number of scenarios needed to build the scenario tree has to grow exponentially with the time horizon of the problem. One of the aims of this paper is to highlight and to quantify this practical observation on an example. Thus, we start in Section 1 by introducing the Mean Squared Error, that is the distance between the strategy obtained using scenario trees and the (supposedly unique) optimal strategy. This is the indicator we use to estimate the performance of the method. Then, in Section 2, we study on a numerical example the relation linking the Mean Squared Error to the number of scenarios used for the discretization. Shapiro (2006) obtains similar conclusions by working on the performance regarding the objective function and using large deviations tools.

We show that this negative observation is to be attributed to the scenario tree approach rather than to be considered as a feature of the original problem. Indeed, when using the particle method (Dallagi, 2007; Carpentier et al, 2009), in Section 3 on the same example, we show that the number of scenarios needed to obtain a given precision does not grow when the time horizon gets longer. This methodology, which mixes ideas from Stochastic Programming and Dynamic Programming, is a variational technique which consists in writing the optimality conditions of the optimization problem first, and then solving them using sampling techniques. This is a gradient-like method that builds an adaptive mesh over the state space as gradient iterations progress towards the solution. This mesh aims at discretizing the space in the regions mostly visited by the state vector at the optimum. According to the results obtained on the example we present here, the algorithm seems to be

well-suited for multi-stage problems. However, with this approach, it seems that the difficulties arising from the dimension of the state space remain; this point is discussed in Section 4.

1. Mathematical formulation

Consider a controlled dynamic system, affected by random variables called noises. We aim to find strategies that minimize some expected cost over a finite time horizon, while satisfying a number of constraints. We suppose that the problem has a unique solution and propose to compare the approximate strategies using the Mean Squared Error (MSE).

1.1. The problem.

We denote by T the finite time horizon of the problem and consider discrete time $\{0, \dots, T\}$. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. Three kinds of random variables on this space are involved in the problem[1]:

- the state[2] $\boldsymbol{X} = (\boldsymbol{X}_t)_{t=0,\dots,T}$, the values of which lie in a finite-dimensional vector space $\mathbb{X}$;
- the control $\boldsymbol{U} = (\boldsymbol{U}_t)_{t=0,\dots,T-1}$, the values of which lie in a finite-dimensional vector space $\mathbb{U}$;
- the noise $\boldsymbol{W} = (\boldsymbol{W}_t)_{t=0,\dots,T}$, the values of which lie in a finite-dimensional vector space $\mathbb{W}$, equipped with the σ-field $\mathcal{W}$ and the probability $\mathbb{P}_{\boldsymbol{W}}$, which is the transport of the probability $\mathbb{P}$ by $\boldsymbol{W}$.

Since some of the numerical methods we use in the following are gradient-based, it is natural to assume that the state and control lie in a Hilbert space. Hence we assume all three random variables are in $L^2(\Omega, \mathcal{A}, \mathbb{P})$. The noise variable $\boldsymbol{W}$ is given, i.e. its probability law is known, while the state and control variables are optimization variables. We here suppose that the information structure has perfect memory: at each time step t, the noise $\boldsymbol{W}_t$ is observed and kept in memory, in such a way that the decision at time t is based on the knowledge of $(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t)$. Hence we introduce the σ-field $\mathcal{F}_t = \sigma\{\boldsymbol{W}_0, \dots, \boldsymbol{W}_t\}$ that represents the information available at time t, the associated filtration $(\mathcal{F}_t)_{t=0,\dots,T}$, and require the control at time t to be measurable with respect to $\mathcal{F}_t$. This constraint will be written: $\boldsymbol{U}_t \preceq \mathcal{F}_t$.

At time 0, we assume that the value of the state is the realization of some random variable $\boldsymbol{W}_0$. Then, at each time step t, based on the available information, a decision $\boldsymbol{U}_t$ is taken, a cost $C_t(\boldsymbol{X}_t, \boldsymbol{U}_t)$ is incurred and the state evolves according to $\boldsymbol{X}_{t+1} = f_t(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1})$. At the end of the time period, a final cost $V(\boldsymbol{X}_T)$ is incurred.

The aim is to find a pair $(\boldsymbol{X}, \boldsymbol{U})$ that minimizes the expected sum of the costs over the time horizon while satisfying the dynamics and measurability constraints.

[1] Throughout this paper, random variables will be denoted by bold letters (e.g. $\boldsymbol{W} \in L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^p)$).

[2] In the Stochastic Optimal Control framework, the terminology "state" has a rather precise sense: this is the minimal information that completely sums up the past of the system in order to compute its future behaviour and cost function knowing all future inputs (see Powell, 2007, Def. 5.4.1). In that sense, the variable $\boldsymbol{X}$ can only be considered as the state variable when assumption 1 is made later on in §1.2. Until then, our use of this terminology is somewhat abusive.

Mathematically speaking, the problem we consider may be formulated as follows:

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} C_t\left(\boldsymbol{X}_t,\boldsymbol{U}_t\right)+V\left(\boldsymbol{X}_T\right)\right), \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}=f_t\left(\boldsymbol{X}_t,\boldsymbol{U}_t,\boldsymbol{W}_{t+1}\right), \qquad \forall t=0,\ldots,T-1, \tag{1b}$$

$$\boldsymbol{X}_0=\boldsymbol{W}_0, \tag{1c}$$

$$\boldsymbol{U}_t \preceq \mathcal{F}_t, \qquad \forall t=0,\ldots,T. \tag{1d}$$

Equations (1b) and (1c) describe the dynamics of the state variable and are $\mathbb{P}$-almost sure relations. Equation (1d) states the information structure of the problem: it is the measurability constraint that forces the decision to be a function of all the past noises, and to be independent of future noise values. Hence we refer to this relation as the non-anticipativity constraint.

1.2. **Mean Squared Error (MSE) for strategies.** Now suppose that Problem (1) has a unique solution denoted by $(\boldsymbol{X}^*,\boldsymbol{U}^*)$. Suppose that we also have a numerical method, based on a scenario discretization, that gives us an approximate solution of (1) that we denote by $(\boldsymbol{X}^\sharp,\boldsymbol{U}^\sharp)$. Since this approximate solution given by the numerical method depends on, say, N samples, the pair $(\boldsymbol{X}^\sharp,\boldsymbol{U}^\sharp)$ lies in the probability space associated with these samples, that is $(\mathbb{W},\mathcal{W},\mathbb{P}_{\boldsymbol{W}})^{\otimes N}$. For instance, scenario tree-based methods compute values of the optimal decision for a finite number of state values that depend on the samples used for building the tree structure.

We make the following assumption:

Assumption 1. Random variables $\boldsymbol{W}_0,\ldots,\boldsymbol{W}_T$ are independent.

Under this assumption, the problem lies in the framework of Dynamic Programming (DP). In particular, we know that optimal controls can be expressed as feedback functions depending only on the state variable x. In other words, there exists some sequence of functions γ^* such that $\boldsymbol{U}_t^*=\gamma_t^*(\boldsymbol{X}_t^*)$.

In order to be able to compare the optimal and approximate solutions, we need to produce a solution of the same nature as the optimal solution, that is a feedback function such as γ^* out of $(\boldsymbol{X}^\sharp,\boldsymbol{U}^\sharp)$. We hence suppose that we have such an interpolation/regression operator that gives us a feedback function $\boldsymbol{\Gamma}^\sharp=(\boldsymbol{\Gamma}_t^\sharp)_{t=0,\ldots,T}$ out of the approximate solution $(\boldsymbol{X}^\sharp,\boldsymbol{U}^\sharp)$. More details on the interpolation/regression operators we use are given in §2.2. Note that the approximate strategy $\boldsymbol{\Gamma}^\sharp=(\boldsymbol{\Gamma}_t^\sharp)_{t=0,\ldots,T-1}$ is a random variable that lies in the samples probability space introduced earlier, that is $(\mathbb{W},\mathcal{W},\mathbb{P}_{\boldsymbol{W}})^{\otimes N}$.

We are interested in evaluating the efficiency of our numerical method for solving Problem (1) by computing the distance between the approximate strategy $\boldsymbol{\Gamma}^\sharp$ and the optimal one γ^*. Since those functions take values over the state space, we need to define some measure on this space. In our context, the measure that seems the most natural is the one corresponding to the density of the optimal state $\boldsymbol{X}^*$. Indeed, using this measure, we allocate more weight to regions which are often "visited" by the optimal state and we do not take into account decisions in regions that are never "visited" by the optimal state, since those decisions are unlikely to be used in practice. We denote by μ_t^* the density of the optimal state at time t.

We can now introduce the MSE associated with strategy $\boldsymbol{\Gamma}^\sharp$, which is our performance indicator in this study:

$$\text{MSE} \triangleq \mathbb{E}\left(\sum_{t=0}^{T-1}\int_{\mathbb{X}}\left\|\gamma_t^*\left(x\right)-\boldsymbol{\Gamma}_t^\sharp\left(x\right)\right\|^2\mu_t^*\left(x\right)\mathrm{d}x\right).$$

Note that the expectation in the MSE is taken over all possible samples; in other words, it lies on the samples probability space $(\mathbb{W}, \mathcal{W}, \mathbb{P}_{\boldsymbol{W}})^{\otimes N}$. The MSE is thus the expected distance between the optimal strategy and the approximate strategy. Recall that this expectation is taken with respect to the drawings used by the algorithm to produce strategy $\boldsymbol{\Gamma}^\sharp$. It is common to decompose the MSE using the so-called variance and squared bias. For this purpose, we introduce the expectation $\gamma^\sharp$ of the approximate feedback function $\boldsymbol{\Gamma}^\sharp$. Then

$$\begin{aligned}\mathrm{MSE} = &\underbrace{\sum_{t=0}^{T-1} \int_{\mathbb{X}} \left\| \gamma_t^*(x) - \gamma_t^\sharp(x) \right\|^2 \mu_t^*(x)\,\mathrm{d}x}_{\text{Squared bias}} \\ &+ \underbrace{\mathbb{E}\left(\sum_{t=0}^{T-1} \int_{\mathbb{X}} \left\| \gamma_t^\sharp(x) - \boldsymbol{\Gamma}_t^\sharp(x) \right\|^2 \mu_t^*(x)\,\mathrm{d}x \right)}_{\text{Variance}}. \end{aligned} \tag{2}$$

The squared bias is the square of the distance between the optimal strategy and the expected approximate strategy, while the variance is the expected square of the distance between the approximate strategy and the expected approximate strategy. Both are real values.

Remark 1. Another relevant choice as a performance indicator would have been the following:

(1) Having the approximate strategy $\boldsymbol{\Gamma}^\sharp$, build the"true" state and control obtained while simulating this strategy, that is the pair $(\boldsymbol{X}^\dagger, \boldsymbol{U}^\dagger)$ that satisfies the dynamics (1b) and (1c):

$$\boldsymbol{X}_0^\dagger \triangleq \boldsymbol{W}_0,$$
$$\boldsymbol{X}_{t+1}^\dagger \triangleq f_t\left(\boldsymbol{X}_t^\dagger, \boldsymbol{U}_t^\dagger, \boldsymbol{W}_{t+1}\right), \text{ with } \boldsymbol{U}_t^\dagger \triangleq \boldsymbol{\Gamma}_t^\sharp(\boldsymbol{X}_t^\dagger), \qquad \forall t = 0, \ldots, T-1.$$

(2) Since the optimal decision $\boldsymbol{U}^*$ is a random variable on $(\Omega, \mathcal{A}, \mathbb{P})$ and the decision $\boldsymbol{U}^\dagger$ is a random variable that lies on the tensor product of $(\Omega, \mathcal{A}, \mathbb{P})$ and the samples probability space $(\mathbb{W}, \mathcal{W}, \mathbb{P}_{\boldsymbol{W}})^{\otimes N}$, compute the expectation:

$$\mathbb{E}\left(\left\| \boldsymbol{U}^\dagger - \boldsymbol{U}^* \right\|_{\mathbb{U}}^2 \right),$$

where the expectation lies on the tensor product of the probability spaces $(\Omega, \mathcal{A}, \mathbb{P})$ and $(\mathbb{W}, \mathcal{W}, \mathbb{P}_{\boldsymbol{W}})^{\otimes N}$. Note that this quantity may also be written $\mathbb{E}(\|\boldsymbol{U}^\dagger - \boldsymbol{U}^*\|^2_{L^2(\Omega,\mathcal{A},\mathbb{P})})$, where the expectation now lies on the samples probability space only.

This quantity is another relevant performance indicator since it is positive and it equals zero if and only if $\boldsymbol{U}^\dagger$ equals $\boldsymbol{U}^*$ almost surely. Moreover, we note that the choice of $\boldsymbol{U}^\dagger$ for evaluating the performance of the numerical method is the one made by Shapiro (2006), even though the comparison is performed using the objective function rather than the strategies.

1.3. **Computing the MSE.** For computing the squared bias and the variance as defined in Equation (2), we must face two main issues.

(1) At each time step t, we must compute integrals over the whole state space with respect to the density μ_t^*. In the following example, we are able to explicitly compute the optimal control strategy. We can then numerically integrate the Fokker-Planck equation to derive the density of the optimal state on a sufficiently dense grid and perform the integration by quadrature

techniques. However, this technique suffers for the same *curse of dimensionality* as DP. For this reason we choose to perform the integration using a Quasi-Monte Carlo technique. This sampling method aims at distributing the samples associated with density μ_t^* equally over the state space using so-called low-discrepancy quasi-random sequences. This provides a very efficient convergence speed for the numerical computation of expectations. There exist numerous methods of Quasi-Monte Carlo sampling (see Niederreiter, 1992, for a survey of these methods).

(2) We must be able to compute the expectation in the variance term. This expectation is taken over the samples used during the algorithm to produce the approximate strategy, e.g. the scenarios in the case of a scenario tree technique. We approximate this expectation by Monte Carlo, by performing the experiment a great number of times independently one from another. In our case, 10^4 experiments were enough to precisely evaluate the variance.

We are now able to estimate the efficiency of numerical methods using the MSE. We point out the relation between the error and the parameters of the methods, namely the number of scenarios used to discretize randomness.

2. Scenario tree-based methods

In the context of discrete time and finite horizon stochastic optimal control problems, a popular and widely used resolution technique consists in approximating the information structure of Problem (1) using scenario trees. This methodology has been widely studied by the Stochastic Programming community. We do not detail this method and refer to Shapiro and Ruszczynski (2003) for a more in depth presentation of Stochastic Programming.

2.1. **Brief presentation.** Because of the measurability constraints in Problem (1), and since there is no reason for the noise probability densities to have finite support, decision variables $\boldsymbol{X}$ and $\boldsymbol{U}$ are in general infinite-dimensional. Hence, except for some very particular cases where an explicit solution can be found, solving the underlying optimization problem directly is intractable. Scenario trees provide a way to approximate filtrations using noise samples to produce finite support approximations.

Let $\mathcal{N}$ be the set of all nodes in the tree, $\mathcal{R}$ be the set of the root nodes and $\mathcal{L}$ the set of the leaves. Building a scenario tree consists in defining the following functions:

- the time function $\theta : \mathcal{N} \to \{0, \dots, T\}$ which, with every node, associates the corresponding time step and the corresponding multi-application θ^{-1} which, to every time step $t \in \{0, \dots, T\}$, associates the set of all nodes at time t;
- the function $\nu : \mathcal{N} \backslash \mathcal{R} \to \mathcal{N} \backslash \mathcal{L}$ which, with each non-root node, associates the preceding node in the tree;
- the weight function $\pi : \mathcal{N} \to [0, 1]$ which, with every node in the tree, associates the probability to go through that node (the sum of all $\pi(i)$, with $i \in \theta^{-1}(t)$ is equal to 1, for each t);
- the multi-application $F = \nu^{-1}$ which, with every node i in the tree, associates the set of its successors (we impose the convention that $F(i) = \emptyset, \forall i \in \mathcal{L}$);
- the function F^+ which, with each node i, associates the whole sub-tree starting from i: $F^+(i) = F(i) \cup F^2(i) \cup \dots \cup F^{T-\theta(i)}(i)$.

We consider a regularly branching tree, i.e. with a constant branching factor denoted by n_b, so that $\text{card}(F(i)) = n_b$, for every node $i \in \mathcal{N} \backslash \mathcal{L}$. At time $t = 0$,

we draw a n_b-sample of the random variable $\boldsymbol{W}_0$, that we denote by $(\boldsymbol{W}'^i)_{i\in\theta^{-1}(0)}$. For each root node i, we draw at time $t = 1$ a n_b-sample of the random variable $\boldsymbol{W}_1$[3], that we denote by $(\boldsymbol{W}'^j)_{j\in F(i)}$. Samples of the random variable $\boldsymbol{W}_1$ are drawn independently root node per root node. This procedure continues until final time $t = T$, so that we have n_b^{T+1} leaves to the tree. Hence one can note that this can reveal a heavy procedure when T becomes large, since we need to draw n_b^{T+1} samples of $\boldsymbol{W}_T$.

With this material we can now define a discretized version of Problem (1):

$$\min_{\boldsymbol{X}',\boldsymbol{U}'} \quad \sum_{i\in\mathcal{N}\backslash\mathcal{L}} \pi(i)\cdot C_{\theta(i)}\left(\boldsymbol{X}'^i,\boldsymbol{U}'^i\right) + \sum_{i\in\mathcal{L}} V\left(\boldsymbol{X}'^i\right), \tag{3a}$$

$$\text{s.c.} \quad \boldsymbol{X}'^i = f_{\theta(\nu(i))}\left(\boldsymbol{X}'^{\nu(i)},\boldsymbol{U}'^{\nu(i)},\boldsymbol{W}'^i\right), \qquad \forall i\in\mathcal{N}\backslash\mathcal{R}, \tag{3b}$$

$$\boldsymbol{X}'^i = \boldsymbol{W}'^i, \qquad \forall i\in\mathcal{R}. \tag{3c}$$

The expectation in the objective function (1a) has been replaced in (3a) by a discrete weighted sum of costs on the tree nodes. Equations (3b) and (3c) correspond to Equations (1b) and (1c). The only constraint that seems to be missing is Equation (1d). Actually, it is now reflected in the structure of the tree. Indeed, starting from any node in the tree, there is only one possibility for going back to the root (we know exactly the values of all past noises), but there is a whole sub-tree that goes to the set of leaves, corresponding to the final time step (we only know the laws of future noises). In other words, the non-anticipativity constraint is coded in the structure of the tree.

In most cases, the scenario tree is not drawn directly. One draws a number of scenarios independently from one another, and then one builds a tree structure. Building such a tree in an efficient way (regarding the implied discretization error) is a challenging task. There exists a large literature on the way one can build scenario trees in a reasonable way (Pflug, 2001; Heitsch and Römisch, 2003), as well as on stability of the underlying optimization problem regarding this discretization (Heitsch et al, 2006). We will not focus on this point here. As it has already been mentioned, we suppose that the tree has been built using a branching factor n_b, which does not depend on time, leading to n_b^{T+1} leaves. Recall that this procedure requires the drawing of $N \triangleq n_b^{T+1}$ different scenarios.

2.2. **Case study.** We apply this methodology to a simple instance where Problems (1) and (3) can be solved analytically. Then, we apply the methodology described in §1.3 in order to compute the MSE and to point out its relation with the number of scenarios N used to discretize the original problem. Let ε be some positive real number, the problem we consider is the following:

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\varepsilon\sum_{t=0}^{T-1}\boldsymbol{U}_t^2 + \boldsymbol{X}_T^2\right), \tag{4a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{U}_t + \boldsymbol{W}_{t+1}, \quad \forall t = 0,\dots,T-1, \tag{4b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{4c}$$

$$\boldsymbol{U}_t \preceq \mathcal{F}_t. \tag{4d}$$

The state and control variables lie in $L^2(\Omega,\mathcal{A},\mathbb{P};\mathbb{R})$. Noise variables $\boldsymbol{W}_0,\dots,\boldsymbol{W}_T$ are i.i.d. random variables with uniform law on $[-1,1]$. One easily shows that the

[3]This tree building procedure, often called conditional sampling, usually requires to draw samples of $\boldsymbol{W}_1$ knowing the value of the preceding noises, that is $\boldsymbol{W}_0$. Because of Assumption 1, this is not needed here.

optimal control for Problem (4) reads $\boldsymbol{U}_t^* = \gamma_t^*(\boldsymbol{X}_t^*)$ with:

$$\gamma_t^*(x) = -\frac{x}{T-t+\varepsilon}, \qquad \forall x \in \mathbb{R}. \tag{5}$$

We now solve the problem on the tree. On each node i, one has a noise sample $\boldsymbol{W}'^i$ and solving the problem on the tree leads to values $\boldsymbol{X}'^i$ and $\boldsymbol{U}'^i$ for the state and control, respectively. One easily shows that:

$$\boldsymbol{U}'^i = -\frac{\boldsymbol{X}'^i + \frac{1}{n_b^{T-\theta(i)}} \sum_{j \in F^+(i)} \boldsymbol{W}'^j}{T-\theta(i)+\varepsilon}, \qquad \forall i \in \mathcal{N} \backslash \mathcal{L}.$$

We now need to build a feedback function from these state and control values. This can be performed using a interpolation/regression operator. We here use the simplest one, which is the nearest neighbour interpolation operator (see Gersho and Gray, 1992, §10.4, for a rigorous study of this operator and its possible implementations). At each time step t, we build the Voronoi diagram of the set of state values $\{\boldsymbol{X}'^i\}_{i \in \theta^{-1}(t)}$. We denote by $\boldsymbol{C}'^i$ the cell for which $\boldsymbol{X}'^i$ is the center. This cell is a random variable in itself, because it depends on the random variables drawn to build the scenario tree. On this cell, the feedback function will be constant, equal to $\boldsymbol{U}'^i$. Thus we obtain the following strategy at time t:

$$\boldsymbol{\Gamma}_t^\sharp(x) = \sum_{i \in \theta^{-1}(t)} \boldsymbol{U}'^i \mathbf{1}_{\boldsymbol{C}'^i}(x). \tag{6}$$

Remark 2. The fact that the Voronoi cells $\boldsymbol{C}'^i$ are random variables, depending on the scenarios, makes the theoretical study of the error associated with strategy $\boldsymbol{\Gamma}^\sharp$ intricate. Indeed, in order to compute the expected approximate strategy $\gamma^\sharp$, as well as to evaluate the variance term in Equation (2), we have to compute expectations over all possible scenario drawings, leading through Equation (6) to expectations over all possible Voronoi cells. This is generally a challenging task. This is the reason why we here appeal to numerical experiments to study the behaviour of the error with respect to the number of scenarios.

The results for this approach are presented in Figure 1 for $T = 4$ and a branching factor of $n_b = 3$. The approximate strategy $\boldsymbol{\Gamma}_0^\sharp$ for the first time step is hence a piecewise constant function with 3 pieces. On the right-hand side of Figure 1, we draw 4 samples of this strategy (dashed curves) to emphasize that the scenario-tree method is stochastic: each sample $\boldsymbol{\Gamma}_0^\sharp$ is derived from a different scenario tree. We also draw the averaged strategy $\gamma_0^\sharp$ over the 10^4 scenarios (dotted curve). One can remark that even if sample strategies are not continuous functions, the average strategy seems to be continuous, even smooth. On the left-hand side of figure 1, we draw the exact strategy (solid curve) obtained using equation (5) and the same average approximate strategy as on the right-hand side (dotted curve).

We can observe both the variance and the bias terms of the MSE on Figure 1: the variance term is the average L^2-distance between the dashed curves (of course if there were 10^4 of them) and the dotted curve on the right hand-side, whereas the bias term is the L^2-distance between the solid curve and the dotted curve on the left hand-side. From Equation (4c), the density of the optimal state, under which we compute the L^2-distances cited above, equals the density of $\boldsymbol{W}_0$, which follows a uniform law on $[-1, 1]$.

We are now able to compute the MSE using the protocol described in §1.3, for several values of the branching factor. We draw in Figure 2 the evolution of both the squared bias and the variance with respect to the branching factor, for the strategy at each time step (here $T = 4$). For computational time reasons, we were not able to perform this experiment for strategies at the third and fourth time steps

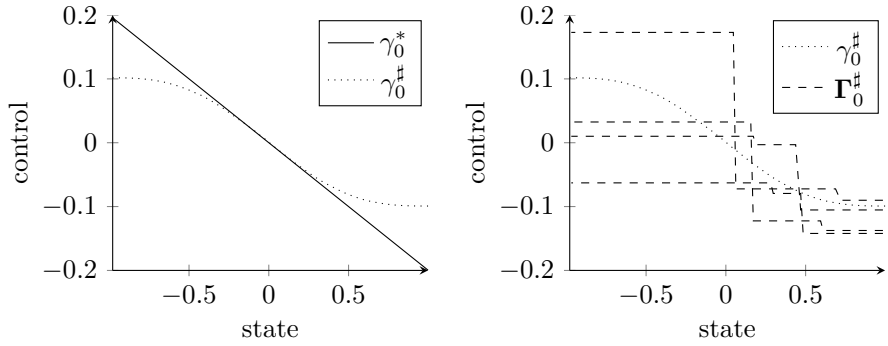

FIGURE 1. Strategies for scenario trees

when the branching factor is more than 10 (the number of nodes involved at these time steps becomes too large).

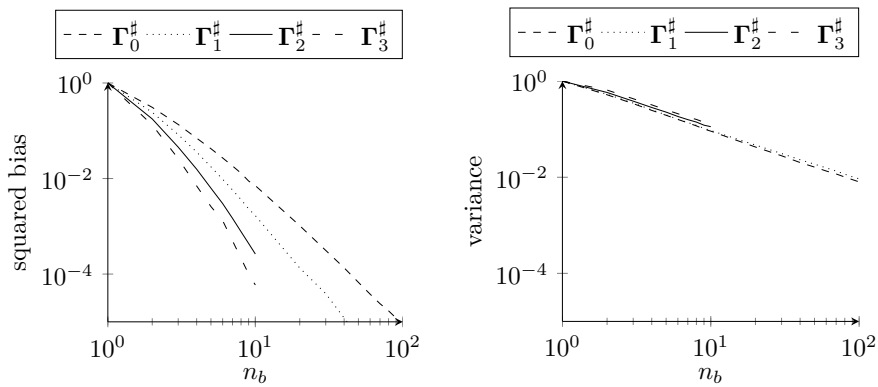

FIGURE 2. Squared bias and variance for scenario trees

On this figure which uses a logarithmic scaling on both axes, one can observe that the variance term on the right-hand side obviously dominates the squared bias term on the left-hand side. Moreover, while the convergence rate for the variance seems to be constant with respect to time, the bias decreases with a highest rate as time grows. This is natural since the tree has more nodes in the last time steps than in the first ones to represent the approximate strategy.

A careful inspection of the variance shows that the MSE appears experimentally to have a convergence rate of n_b^{-1} for every time step. In other words, the number of scenarios needed to obtain a given precision varies exponentially with respect to the time horizon: it is proportional to n_b^T. Still, recall that in order to have a branching factor n_b equal to 10^2, since we have a 4-period problem, we need to draw 10^8 scenarios!

3. Particle methods

We here propose a similar study using another numerical method based on scenarios: the particle method. This variational technique consists in first writing the optimality conditions for the problem. Then, scenarios are used to compute the expectations that lie in the optimality conditions. We briefly present the idea of this method and then study its error with respect to the number of scenarios used

for sampling. A detailed description can be found in the work by Carpentier et al (2009).

3.1. **Brief presentation.** The particle method consists in solving first-order optimality conditions that have been approximated through sampling. We first present the optimality conditions and then describe the sampling procedure. The resulting approximate optimality conditions are solved numerically using a gradient technique.

We suppose that the cost function and the system dynamics are differentiable with respect to both the state and the control, and that their derivatives are Lipschitz-continuous.

We suppose that Assumption 1 is still in force. In this context, Carpentier et al (2009, Theorem 2.6) state that, if a solution $(\boldsymbol{X}^*, \boldsymbol{U}^*)$ to Problem 1 exists, then, for every time step $t = 0, \dots, T$, there exists a random variable $\boldsymbol{\Lambda}_t^* \in L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{X})$, called the adjoint state, such that the following conditions hold:

$$\boldsymbol{X}_0^* = \boldsymbol{W}_0, \tag{7a}$$

$$\boldsymbol{X}_{t+1}^* = f_t\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*, \boldsymbol{W}_{t+1}\right), \tag{7b}$$

$$\boldsymbol{\Lambda}_T^* = \frac{\partial V}{\partial x}\left(\boldsymbol{X}_T^*\right)^\top, \tag{7c}$$

$$\boldsymbol{\Lambda}_t^* = \frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*, \boldsymbol{W}_{t+1}\right)^\top \boldsymbol{\Lambda}_{t+1}^* \middle| \boldsymbol{X}_t^*\right), \tag{7d}$$

$$0 = \frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*, \boldsymbol{W}_{t+1}\right)^\top \boldsymbol{\Lambda}_{t+1}^* \middle| \boldsymbol{X}_t^*\right). \tag{7e}$$

Note that these optimality conditions are specific to the Markovian case and are not obtained in a straightforward manner. The authors first write first-order optimality conditions conditionally to the filtration $\mathcal{F}_t$. Then Assumption 1 is used at each time step to recursively replace the conditional expectations with respect to $\mathcal{F}_t$ by conditional expectations with respect to the optimal state $\boldsymbol{X}_t^*$.

Equations (7a) and (7b) simply recall the state dynamics of the system. Equations (7c) and (7d) introduce a backwards relation on the adjoint states. Actually, the adjoint state at time t may be seen as the sensitivity of the optimal cost from time t to the end of the time horizon with respect to the state variable at time t. Hence there exists a relation between Equations (7c) and (7d) and the classical DP equation. Finally, Equation (7e) states that the derivative of the optimal future cost at a given time step with respect to the control at the same time step equals zero at the optimum. This is the classical first-order optimality condition when no additional constraints (apart for the non-anticipativity constraints) are imposed to the control.

Note that the adjoint state $\boldsymbol{\Lambda}_{t+1}^*$ in Equations (7d) and (7e) is, by construction, measurable with respect to the state variable $\boldsymbol{X}_{t+1}^* = f_t\left(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*, \boldsymbol{W}_{t+1}\right)$. Because of Assumption 1, the conditional expectations in Equations (7d) and (7e) are actually expectations that are only supported by the random variable $\boldsymbol{W}_{t+1}$. One has to keep this in mind when discretizing these conditions.

We now want solve these conditions numerically using sampling techniques and gradient methods. Suppose we have N samples $\boldsymbol{W}'^1, \dots, \boldsymbol{W}'^N$, called scenarios or noise particles, for the random variable $\boldsymbol{W}$. After the k-th iteration, suppose we have N samples for the current state $\boldsymbol{X}^{(k)}$, control $\boldsymbol{U}^{(k)}$, and adjoint state $\boldsymbol{\Lambda}^{(k)}$. In order to compute conditions (7d) and (7e), we need to evaluate expectations of functions involving $\boldsymbol{\Lambda}_{t+1}^{(k)}$, knowing the value $\boldsymbol{X}_t^{(k)}$ of the state at time t, for every particle $k = 1, \dots, N$. We have two ingredients to perform this operation:

(1) at the optimum, we know that for every time step $t = 0, \dots, T-1$, the adjoint state $\boldsymbol{\Lambda}^*_{t+1}$ is a function of the state $\boldsymbol{X}^*_{t+1} = f_t(\boldsymbol{X}^*_t, \boldsymbol{U}^*_t, \boldsymbol{W}_{t+1})$, where $\boldsymbol{U}^*_t$ is measurable with respect to $\boldsymbol{X}^*_t$, and $\boldsymbol{W}_{t+1}$ is independent of $\boldsymbol{X}^*_t$;
(2) we have N samples of all the random variables involved, namely $\boldsymbol{X}^{(k)}$, $\boldsymbol{U}^{(k)}$, $\boldsymbol{\Lambda}^{(k)}$ and $\boldsymbol{W}$.

All we need is to define a regression operator, based on the current state and adjoint state samples, that associates with every value of the current state an estimate for the corresponding value of the adjoint state. At time t, we denote this regression operator by $\tilde{\boldsymbol{\Lambda}}_t$. Using this material, we are now able to write the discretized optimality conditions, namely the initial condition on the state for every particle $i = 1, \dots, N$:

$$\boldsymbol{X}'^i_0 = \boldsymbol{W}'^i_0, \tag{8a}$$

the state dynamics for every time step $t = 0, \dots, T-1$, and for every particle $i = 1, \dots, N$:

$$\boldsymbol{X}'^i_{t+1} = f_t\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t, \boldsymbol{W}'^i_{t+1}\right), \tag{8b}$$

the final condition on the adjoint state, for every particle $i = 1, \dots, N$:

$$\boldsymbol{\Lambda}'^i_T = \frac{\partial V}{\partial x}\left(\boldsymbol{X}'^i_T\right)^\top, \tag{8c}$$

the backwards adjoint state dynamics, for every time step $t = T-1, \dots, 0$ and every particle $i = 1, \dots, N$:

$$\begin{aligned}\boldsymbol{\Lambda}'^i_t = \frac{1}{N}\sum_{j=1}^{N}\Bigg(&\frac{\partial C_t}{\partial x}\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t\right)^\top \\ &+ \frac{\partial f_t}{\partial x}\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t, \boldsymbol{W}'^j_{t+1}\right)^\top \tilde{\boldsymbol{\Lambda}}_{t+1}\left(f_t\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t, \boldsymbol{W}'^j_{t+1}\right)\right)\Bigg),\end{aligned} \tag{8d}$$

and the stationarity condition, for every time step $t = T-1, \dots, 0$ and every particle $i = 1, \dots, N$:

$$\begin{aligned}0 = \frac{1}{N}\sum_{j=1}^{N}\Bigg(&\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t\right)^\top \\ &+ \frac{\partial f_t}{\partial u}\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t, \boldsymbol{W}'^j_{t+1}\right)^\top \tilde{\boldsymbol{\Lambda}}_{t+1}\left(f_t\left(\boldsymbol{X}'^i_t, \boldsymbol{U}'^i_t, \boldsymbol{W}'^j_{t+1}\right)\right)\Bigg),\end{aligned} \tag{8e}$$

Note that in both equations (8d) and (8e), the sums only involve the index j, and hence the noise variable only. This corresponds to the fact that the conditional expectations in (7d) and (7e) with respect to $\boldsymbol{X}^*_t$ are in fact expectations.

The algorithm then consists in solving these conditions using a gradient method. Thus, each iteration k of the particle method consists in three steps:

(1) integrate the N state dynamics (8a) and (8b) using the current control particles $\boldsymbol{U}'^{1,(k)}, \dots, \boldsymbol{U}'^{N,(k)}$;
(2) integrate the N backwards adjoint state dynamics (8c) and (8d) using the current state and noise particles;
(3) compute the N "gradient particles" $\boldsymbol{G}'^{1,(k)}, \dots, \boldsymbol{G}'^{N,(k)}$ (the right-hand side of equation (8e)) and update every control particle i using a gradient step:

$$\boldsymbol{U}'^{i,(k+1)} = \boldsymbol{U}'^{i,(k)} - \rho \boldsymbol{G}'^{i,(k)}.$$

The algorithm stops when all gradient particles are sufficiently small, ensuring that condition (8e) is almost fulfilled, for every $i = 1, \dots, N$.

Remark 3. Note that, at each iteration, we have N samples (or particles) for the state and control at every time step. Using a regression operator, we are able to compute a feedback function such as the one introduced in Section 1.2, denoted by $\mathbf{\Gamma}^\sharp$. This feedback function associates a decision with every possible state of the system and this is the quantity on which we base our error analysis.

3.2. **Case study.** using the same example as in §2.2, we represent in the right-hand side of Figure 3 some samples of the approximate strategy (dashed curves) obtained by the particle method using 3^4 scenarios for discretization. Observe that these are piecewise constant functions with 3^4 pieces, instead of only 3 pieces when using scenario trees. Indeed, using the particle method one has the same number of nodes at every time step. In other words, all the scenarios are used at each time step. In the left-hand side of the same figure we draw the average approximate

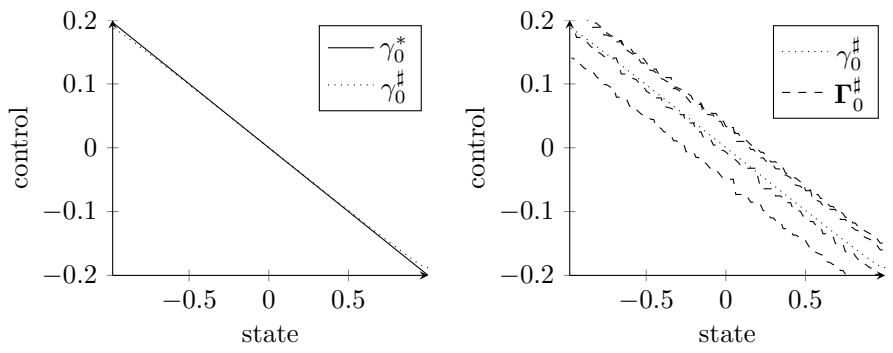

FIGURE 3. Strategies for particle methods

strategy (dotted curve) and the optimal strategy (solid curve). The bias term in the MSE is still the L^2-distance between the dotted curve and the solid curve, while the variance term is the average of the L^2-distances between the dashed curves and the dotted curve. Using the same number of scenarios as we did for for scenario trees in §2.2, we observe that both the bias and variance look much smaller to those observed in Figure 1.

We can now compute the MSE for several values of the number N of scenarios used for discretization. We observe in figure 4 how the squared bias and the variance associated with each strategy, i.e. for each time step, behave when the number of scenarios grows. Note that the squared bias, on the left-hand side is dominated by the variance, on the right-hand side. A careful inspection of the latter shows that the term leads experimentally to a convergence rate that is slightly less than N^{-1}. When comparing with the results obtained using scenario trees, one should notice that the x-axes do not refer to the same quantities in Figures 2 and 4. Hence, we here observe a convergence rate in N^{-1}, N being the number of scenarios, while we observed a convergence rate of $n_b^{-1} = N^{-\frac{1}{T}}$ in the case of scenario trees, which is much smaller.

Unlike scenario trees, the error associated with particle methods does not depend on the time horizon on this example: it is the same at every time step.

4. The state space dimension issue

Most numerical methods that aim at solving stochastic optimal control problems encounter difficulties when the dimension of the state space becomes large. Since we

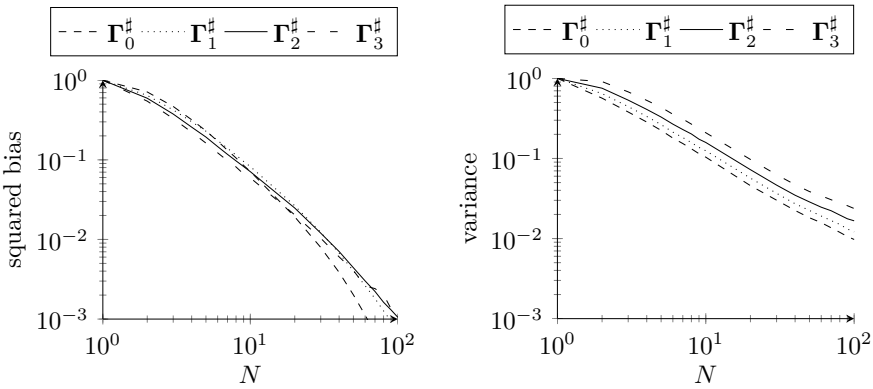

FIGURE 4. Squared bias and variance for particles.

are looking for strategies, that is functions that map a decision from every possible state of the system, even just storing such functions becomes challenging when the dimension of the state space is large. In the stochastic optimal control community framework, this difficulty is known as the *curse of dimensionality*.

Note that particle methods are not prone to the same curse of dimensionality as DP, for instance. While DP encounters difficulties because it requires to build a somewhat regular grid to cover the state space, particle methods are designed to construct an adaptive mesh over the state space: the state particles are built so as to concentrate in regions that are often visited by the optimal state. As such, regardless of the dimension of the state space, whenever the dispersion of the optimal state is small, the particle method shall produce satisfying results.

In the example we here study, the density of the optimal state does not seem to have this "tightness property". On a simple extension of Problem (4) to a two-dimensional state space case, we observe in Figure 5 how the squared bias and the variance associated with each strategy, i.e. for each time step, behave when the number of scenarios grows. Note that the squared bias, on the left-hand side, now

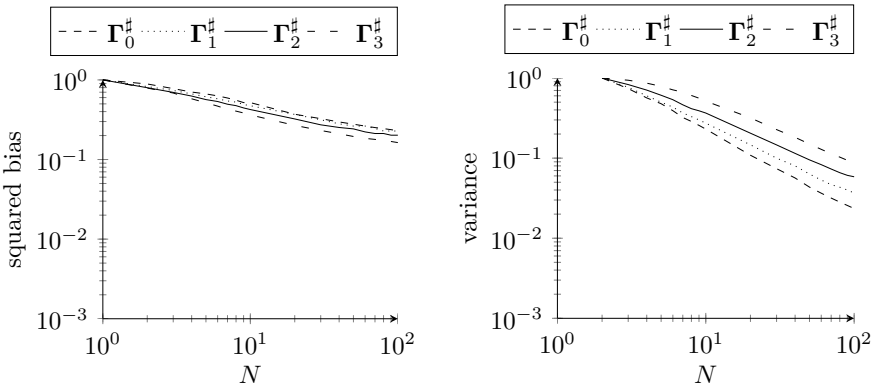

FIGURE 5. Squared bias and variance for particles in dimension 2.

prevails over the variance term, on the right-hand side. A careful inspection of the latter shows that the term leads experimentally to a convergence that is slightly less than $N^{-0.5}$. The variance itself also seems to decrease at a lower rate than in the one-dimensional case. Both of these observations indicate that the performance

of the method could benefit from a better regression operator than the nearest neighbour that we used here.

Note that, for computational time reasons, we cannot produce such experimental results for higher dimensions. Hence we are not able to make a thorough study of the relation between the dimension of the state space and the convergence rate of both the squared bias and the variance.

On the other hand, when dealing with large state space dimensions, a clever idea is to make use of decomposition schemes in order to replace the solving of a high dimensional problem by the iterative solving of several smaller problems. This is indeed a common and powerful technique when using scenario trees (see Carpentier et al, 1995; Higle and Sen, 1996; Shapiro and Ruszczynski, 2003, Ch. 3). Both the improvement of the regression operator and the use of a decomposition scheme are the subject of ongoing research.

Conclusion

We presented a numerical study concerning the comparison of two scenario-based approaches for solving stochastic optimal control problems with a discrete finite horizon. In the first section, we introduced the performance indicator that was used to compare both methods, namely the Mean Squared Error (MSE).

The first approach we considered was scenario tree modeling. We observed that the number of scenarios needed to obtain a given accuracy grew exponentially with the time horizon, making the implementation for multi-stage problems hardly tractable.

The second approach we considered was particle methods. In this case, we observed that the associated MSE does not depend on time. Thus, this recent approach seems to be well-suited for multi-stage problems. Unfortunately, it seems that its performance deteriorates rapidly when the dimension of the state space increases. However, this method is quite new and as such has received little attention. Its implementation and hence performance would certainly benefit from computational studies. Future works will be concerned with this topic.

References

Bertsekas D (2000) Dynamic Programming and Optimal Control, vol 1 & 2, 2nd edn. Athena Scientific

Bonnans JF, Gilbert JC, Lemaréchal C, Sagastizàbal CA (2006) Numerical Optimization, Theoretical and Practical Aspects, Universitext, vol XIV, 2nd edn. Springer

Carpentier P, Cohen G, Culioli JC (1995) Stochastic optimal control and decomposition-coordination methods. In: Durier R, Michelot C (eds) Recent Developments in Optimization. Lecture Notes in Economics and Mathematical Systems, 429, Springer Verlag, Berlin, pp 72–103

Carpentier P, Cohen G, Dallagi A (2009) Particle Methods for Stochastic Optimal Control Problems, arXiv:0907.4663v1

Dallagi A (2007) Méthodes particulaires en commande optimale stochastique. Thèse de doctorat, Université Paris 1, Panthéon-Sorbonne

Dantzig G (1955) Linear programming under uncertainty. Management Science 1:197–206

Gersho A, Gray R (1992) Vector Quantization and Signal Compression. Kluwer Academic Press/Springer

Heitsch H, Römisch W (2003) Scenario Reduction Algorithms in Stochastic Programming. Computational Optimization and Applications 24:187–206

Heitsch H, Römisch W, Strugarek C (2006) Stability of multistage stochastic programs. SIAM Journal on Optimization 17:511–525

Higle J, Sen S (1996) Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming. Kluwer Academic Publishers, Dordrecht

Niederreiter H (1992) Random Number Generation and Quasi-Monte Carlo Methods. CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia

Pflug G (2001) Scenario tree generation for multiperiod financial optimization by optimal discretization. Mathematical Programming 89:251–271

Powell WB (2007) Approximate Dynamic Programming. Wiley Series in Probability and Statistics, John Wiley & Sons

Prékopa A (1995) Stochastic Programming. Kluwer, Dordrecht

Shapiro A (2006) On complexity of multistage stochastic programs. Operations Research Letters 34:1–8

Shapiro A, Ruszczynski A (eds) (2003) Stochastic Programming. Elsevier, Amsterdam

P. Girardeau, EDF R&D, 1, avenue du Général de Gaulle, F-92141 Clamart Cedex, France, also with Université Paris-Est, CERMICS and ENSTA.

E-mail address: pierre.girardeau@cermics.enpc.fr