QUANTITATIVE FINANCE
RESEARCH CENTRE

UNIVERSITY OF
TECHNOLOGY SYDNEY

QUANTITATIVE FINANCE RESEARCH CENTRE

Research Paper 352 January 2015

Algorithms for Optimal Control of Stochastic Switching Systems

Juri Hinz and Nicholas Yap

ISSN 1441-8010 www.qfrc.uts.edu.au

Algorithms for optimal control of stochastic switching systems

Juri Hinz and Nicholas Yap
University of Technology Sydney
juri.hinz@uts.edu.au

December 29, 2014

Abstract

Optimal control problems of switching type with linear state dynamics are ubiquitous in applications of stochastic optimization. For high-dimensional problems of this type, solutions which utilize some convexity related properties are useful. For such problems, we present novel algorithmic solutions which require minimal assumptions while demonstrating remarkable computational efficiency. Furthermore, we devise procedures of the primal-dual kind to assess the distance to optimality of these approximate solutions.

1 Introduction

When making decisions under uncertainty, the major difficulty is to determine how to update estimates and decisions in order to achieve optimality over a given time period. These kind of questions are often framed within the realm of *Markov decision theory* which can be viewed as discrete-time *optimal stochastic control*.

The theoretical underpinnings of Markov decision theory are now well-understood. Rigorous mathematical treatments are available in textbook form (see [2], [4], [12] and [23]).

However, practical applications remain persistently challenging despite the rich arsenal of theoretical tools that are currently available. In this context, *approximate dynamic programming* (see [22]) grew from attempts at providing simultaneously practically implementable heuristics and theoretical insights as to why they perform well in practice.

In order to control a large system, a practical approach to dealing with the high dimensionality of the state space is to first achieve a finite discretization of it. Alternatively, one can rely on an efficient approximation of functions on this space. In this spirit, function-based methods suggest to approximate value functions on the state space. One such method is the least squares Monte Carlo approach which suggests an approximation by a suitably parameterized set of basis functions. As these parameters are computed by performing successive regressions, this method is placed within the regression-based method family.

Following [7, 28, 29], the contribution [18] became the source of subsequent research focused on its theoretical justification. Convergence issues are addressed in [8] and later generalized in [27, 9] and [10], extensions to multiple exercise rights were considered in [6], and studied in [3] where the connections to statistical learning theory and the theory of empirical processes is emphasized. For an overview of the applications of Monte Carlo methods in financial engineering we refer the interested reader to Glasserman's book [13] and to the literature cited therein. Beyond financial applications, function approximation methods have also been used to capture local behavior of value functions, and advanced regression methods, e.g. kernel methods [20, 21], local polynomial regression [11], and neural networks [5], have been established.

One of the main advantages of regression-based methods is that they reduce computations to simple linear algebraic operations in low dimension. However, due to the successive iterations required by the implementation of the dynamic programming principle, computed solutions often exhibit instability. The thrust of this paper is to present certain numerical aspects of a novel function-based method, which utilizes some convexity assumptions to establish stable and fast solutions in terms of algebraic matrix operations. Although our approach requires us to focus on a rather specific problem structure, it covers a large number of important practical applications.

The goal of the present work is to extend and apply a concepts suggested in [15] to overcome its rather restrictive assumptions. With this extension, our methodology applies to a broad problem class. Furthermore, we develop a technique to asses the quality of our numerical solutions in terms of estimation of their distance to optimality. This is realized by a computation of the so-called confidence intervals (or *bounds*), when evaluating value functions at state space variables. We present a novel and numerically efficient algorithm for estimation of the value functions from above and below.

The remainder of the paper shall be structured as follows. After introducing a general framework in Section 3, Section 4 presents the notion of a *convex switching system* and discusses solutions to this stochastic problem

class. In Section 5, we review and analyze the numerical scheme of [15] that provides fast and stable solutions to convex switching problems. Section 7 represents a first step to relaxing the requirement of convexity. A remarkable generalization is achieved Section 8 where a method has been devised that allows us to by-pass any convexity requirement while yielding significant improvements in computation time. Another major contribution of the paper is presented in Section 9, where we suggest an adaptation of the approach [25] to obtain recursive schemes for upper bound estimates of an approximate solution. Section 10 provides two numerical examples.

2 Markov decision theory

We begin by reviewing the classical framework of finite-horizon Markov decision theory, where we closely follow Chapter 2 of [2] and tailor it to suit our purposes. Consider a system on the finite time horizon $0, \dots, T$ whose state varies in a measurable space $(E, \mathcal{E})$ and is affected by elements from a set A of possible actions. For each $a \in A$, we assume that $K_t^a(x, dx')$ is a stochastic transition kernel on $(E, \mathcal{E})$. Consider a fixed sequence $(X_t)_{t=0}^T$ of random variables which can be thought of as coordinate projections acting on the product $E^{\{0,\dots,T\}}$ of copies of $(E, \mathcal{E})$. A mapping $\pi_t : E \mapsto A$ which describes the action that the controller of the system takes at time t is called a *decision rule*. A sequence of decision rules $\pi = (\pi_t)_{t=0}^{T-1}$ is called a *policy*. For each initial point $x_0 \in E$ and each policy $\pi = (\pi_t)_{t=0}^{T-1}$, there exists a probability measure $\mathbb{P}^{x_0,\pi}$ for which $\mathbb{P}^{x_0,\pi}(X_0 = x_0) = 1$ and where

$$\mathbb{P}^{x_0,\pi}(X_{t+1} \in B \,|\, X_0, \dots, X_t) = K_t^{\pi_t(X_t)}(X_t, B) \tag{1}$$

holds for each measurable $B \in \mathcal{E}$ and $t = 0, \dots, T-1$. That is, given that system is in state X_t at time t, the action $a = \pi_t(X_t)$ is used to pick the transition probability $K_t^{a=\pi_t(X_t)}(X_t, \cdot)$ which assigns the random evolution of the state from X_t to X_{t+1} with the distribution $K_t^{\pi_t(X_t)}(X_t, \cdot)$. For the sake of notational convenience, we use $\mathcal{K}_t^a$ to denote the one-step transition operator associated with the transition kernel K_t^a when the action $a \in A$ is chosen. In other words, for each action $a \in A$ the operator $\mathcal{K}_t^a$ acts on functions φ by

$$(\mathcal{K}_t^a \varphi)(x) = \int_E \varphi(x') K_t^a(x, dx') \qquad x \in E, \tag{2}$$

whenever the above integrals are well-defined.

At each time t, we are given the *t-step reward function* $r_t : E \times A \mapsto \mathbb{R}$, where $r_t(x, a)$ represents the reward for applying an action $a \in A$ when the

state of the system is $x \in E$ at time t. At the end of the time horizon, at time T, it is assumed that no action can be taken. Here, if the system is in a state x, a *scrap value* $r_T(x)$, which is described by a pre-specified *scrap function* $r_T : E \to \mathbb{R}$, is collected. Given an initial point x_0, our goal is to maximize the expected finite-horizon total reward, in other words to find the argument $\pi^* = (\pi_t^*)_{t=0}^{T-1}$ such that

$$\pi^* = \operatorname{argmax}_{\pi \in \mathcal{A}} \mathbb{E}^{x_0,\pi} \left(\sum_{t=0}^{T-1} r_t(X_t, \pi_t(X_t)) + r_T(X_T) \right), \tag{3}$$

where $\mathcal{A}$ is the set of all policies, and $\mathbb{E}^{x,\pi}$ denotes the expectation over the controlled Markov chain defined by (1). The maximization (3) is well-defined under diverse additional assumptions (see [2], p. 199).

The calculation of the optimal policy is addressed in the following setting. We introduce for $t = 0, \ldots, T-1$ the *Bellman operator*

$$\mathcal{T}_t v(x) = \sup_{a \in A} \left(r_t(x, a) + \mathcal{K}_t^a v(x) \right), \quad x \in E \tag{4}$$

which acts on each measurable function $v : E \to \mathbb{R}$ where the integrals $\mathcal{K}_t^a v$ for all $a \in A$ exist. Further, consider the *Bellman recursion*

$$v_T = r_T, \quad v_t = \mathcal{T}_t v_{t+1} \qquad \text{for } t = T-1, \ldots, 0. \tag{5}$$

Under appropriate assumptions, there exists a recursive solution $(v_t^*)_{t=0}^T$ to the Bellman recursion, which gives the so-called *value functions* and determines an optimal policy π^* via

$$\pi_t^*(x) = \operatorname{argmax}_{a \in A} \left(r_t(x, a) + \mathcal{K}_t^a v_{t+1}^*(x) \right), \quad x \in E$$

for all $t = 0, \ldots, T-1$.

3 Convex switching systems

For the remainder of this work, we concentrate on Markov decision problems which satisfy specific additional assumptions under which the solutions to the Bellman recursion exist. This enables us to focus on finding numerical approximations.

Consider a Markov decision model whose state evolution consists of one discrete and one continuous component. To be more specific, we assume that the state space $E = P \times \mathbb{R}^d$ is the product of a finite space P and

the Euclidean space $\mathbb{R}^d$. We suppose that the first component $p \in P$ is deterministically driven by a finite set A of actions in terms of a function

$$\alpha : P \times A \to A, \quad (p, a) \to \alpha(p, a),$$

where $\alpha(p, a) \in P$ is the new value of the discrete component of the state if its previous discrete component value was p and the action $a \in A$ was taken by the controller. Furthermore, we assume that the continuous state component evolves as an uncontrolled Markov process $(Z_t)_{t=0}^T$ on $\mathbb{R}^d$ whose evolution is driven by random linear transformations

$$Z_{t+1} = W_{t+1} Z_t$$

with pre-specified independent and integrable disturbance matrices $(W_t)_{t=1}^T$. Finally, let us assume that the reward functions

$$r_t(p, z, a), \quad t = 0, \ldots, T-1, \quad p \in P, \quad a \in A$$

and scrap functions

$$r_T(p, z), \quad p \in P$$

are convex and globally Lipschitz continuous in the continuous component of the state space $z \in \mathbb{R}^d$. In this setting, the transition operators are given by

$$\mathcal{K}_t^a v(p, z) = \mathbb{E}(v(\alpha(p, a), W_{t+1} z)), \qquad t = 0, \ldots, T-1, \;\; a \in A \tag{6}$$

and the Bellman operators are

$$\mathcal{T}_t v(p, z) = \sup_{a \in A} \big(r_t(p, z, a) + \mathbb{E}(v(\alpha(p, a), W_{t+1} z)) \big) \tag{7}$$

for all $p \in P$, $z \in \mathbb{R}^d$ and $t = 0, \ldots, T-1$. Markov decision problems satisfying these assumptions are referred to as *convex switching systems* in what follows.

4 Algorithmic solutions

For such systems, the backward induction described by (5) solves our control problem. However, by inspecting the Bellman operator

$$\mathcal{T}_t v(p, z) = \max_{a \in A} \big(r_t(p, z, a) + \mathbb{E}(v(\alpha(p, a), W_{t+1} z)) \big), \tag{8}$$

we see that solving the Bellman recursion results in a number of problems, the most pressing of which is that one requires a point-wise solution for

each $z \in \mathbb{R}^d$. In [15], a method was presented that targeted a solution in a "functional" form. We now provide a detailed account and justification of their approach.

First, by approximating the expectation in the Bellman operator in (8) by finite summation, we obtain the *modified* Bellman operator $\mathcal{T}_t^n$ that acts on a given value function according to

$$\mathcal{T}_t^n v(p,z) = \max_{a\in A}\left(r_t(p,z,a) + \sum_{k=1}^{n} \nu_{t+1}(k) v(\alpha(p,a), W_{t+1}(k)z)\right) \tag{9}$$

where $(W_{t+1}(k))_{k=1}^n$ represents appropriate realizations of disturbances with the corresponding probability weights $(\nu_{t+1}(k))_{k=1}^n$. By replacing the true Bellman operator (8) in the backward induction of (5) by its modified counterpart that is given by (9), we obtain the modified induction

$$v_T^n = r_T, \quad v_t = \mathcal{T}_t^n v_{t+1}^n \qquad \text{for } t = T-1,\ldots,0. \tag{10}$$

Although the integration is now replaced by a finite sum, determining $(v_t^n)_{t=0}^T$ is still algorithmically intractable as the calculation must be performed at each point $z \in \mathbb{R}^d$. At this point, we turn to the important observation that since the scrap and reward functions, $r_t(p,z,a)$, $t = 0,\ldots,T-1$ and $r_T(p,z)$, used in (9) and (10) are convex in the continuous component, then the resulting value functions $(v_t^n)_{t=0}^T$ must also be convex in the same component.

We now suggest an approximation of these functions $(v_t^n)_{t=0}^T$ in terms of maxima over a finite selection of their sub-gradients. Before we can begin to explain the advantage of such a piecewise linear approximation, we need to first establish a few concepts.

First, let us refer to a countable subset $G \subset \mathbb{R}^d$ as a *grid*. For a grid G, the *sub-gradient envelope* $\mathcal{S}_G f$ of a convex function f is defined to be the maximum of sub-gradients $\nabla_g f$ of f at each grid point $g \in G$ and so

$$\mathcal{S}_G f = \vee_{g\in G} \nabla_g f.$$

Given a family $\{(W_t(k))_{t=1}^T : k = 1,\ldots,n\}$ of trajectories of disturbances that increases with $n \in \mathbb{N}$ and a family of grids $(G^m)_{m\in\mathbb{N}}$ whose tightness increases with $m \in \mathbb{N}$, we define for each $n, m \in \mathbb{N}$ the *double modified* Bellman operators $\mathcal{T}_t^{m,n}$ for $t = 0,\ldots,T-1$

$$(\mathcal{T}_t^{m,n} v)(p,\cdot) = \mathcal{S}_{G^m} \max_{a\in A}\left(r_t(p,\cdot,a) + \sum_{k=1}^{n} \nu_{t+1}(k) v(\alpha(p,a), W_{t+1}(k)\cdot)\right)$$

Using these operators, the double modified value functions $(v_t^{m,n})_{t=0}^T$ are obtained from the backward induction which starts with

$$v_T^{m,n}(p,\cdot) = \mathcal{S}_{G^m} r_T(p,\cdot), \quad p \in P \tag{11}$$

and recursively determines functions

$$v_t^{m,n} = \mathcal{T}_t^{m,n} v_{t+1}^{m,n}, \;\; t = T-1, \dots 0. \tag{12}$$

Obviously, this approach involves two approximation parameters $n \in \mathbb{N}$ and $m \in \mathbb{N}$, which correspond to the sampled disturbances $(W_{t+1}(k))_{k=1}^n$ with their weights $(\nu_{t+1}(n))_{k=1}^n \subset \mathbb{R}_+^d$ and the grid tightening $(G^m)_{m\in\mathbb{N}}$. Under appropriate assumptions, this scheme enjoys excellent convergence properties (see [15]). However, we shall now focus solely on its algorithmic aspect.

Since the double-modified backward induction (11) and (12) returns value functions $(v_t^{m,n})_{t=0}^T$ which are piecewise linear and convex (in the continuous component), we now address an appropriate representation of such functions in terms of matrices in order to re-write the backward induction algorithm (11) and (12) in a compact matrix form.

A matrix with d columns is called a *matrix representative* of a piecewise linear convex function $l : \mathbb{R}^d \to \mathbb{R}$ if it holds that $l(z) = \max(Lz)$ for all $z \in \mathbb{R}^d$. We shall use the expression $l \sim L$ if a piecewise linear convex function l possesses a matrix representative L. It turns out that the formation of a sub-gradient envelope can be directly described in terms of matrix representatives. Namely, if l possesses a matrix representative L then its sub-gradient envelope $\mathcal{S}_G$ on the grid $G = \{g^1, \dots, g^m\}$ possess a matrix representative $\Upsilon_G[L]$ where the row-re-arrangement operator Υ_G is defined by

$$\Upsilon_G[L]_{i,\cdot} = L_{\mathrm{argmax}(Lg^i),\cdot} \qquad \text{for all } i = 1, \dots, m.$$

In other words, when Υ_G is applied to a matrix L with d columns, the result $\Upsilon_G[L]$ of the row-rearrangement yields an $m \times d$ matrix whose i-th row is the row of L at which the maximum in Lg^i at the i-th grid point is attained. As mentioned above, the relation between the sub-gradient envelope of a function and its matrix representative is thus given in terms of the row-rearrangement operator Υ_G:

$$l \sim L \quad \Longrightarrow \quad \mathcal{S}_G l \sim \Upsilon_G[L].$$

A similar relation holds for the summation of piecewise linear and convex functions, followed by sub-gradient envelope. Namely, it corresponds to a straight summation of their matrix representatives, after row-rearrangement:

$$l \sim L, \;\; f \sim F \quad \Longrightarrow \quad \mathcal{S}_G(f+l) \sim \Upsilon_G[L] + \Upsilon_G[F].$$

Similarly, maximization of piecewise linear and convex functions, followed by sub-gradient envelope is realized on matrix level by binding by rows of matrix representatives, followed by the row-rearrangement:

$$l \sim L, \;\; f \sim F \quad \Longrightarrow \quad \mathcal{S}_G(l \vee f) \sim \Upsilon_G[L \sqcup F].$$

Here, the binding-by-row operation $L \sqcup F$ performs a row concatenation of the two matrices L and F. Let us also introduce an equivalent but algorithmically more convenient procedure of maximization on the level of matrix representatives for later use. Given a grid $G = \{g^1, \ldots, g^m\} \subset \mathbb{R}^d$ and $m \times d$ matrices $F(a)$, $a \in A$, we introduce

$$F := \bigsqcup_{a \in A} F(a)$$

to denote a $m \times d$ matrix F whose i-th row

$$F_{i,} = F_{i,}(a(i)), \quad i = 1, \ldots, m$$

equals to the i-th row of the matrix $F(a(i))$ where the maximum at the i-th grid point g^i is reached, i.e.

$$a(i) = \operatorname{argmax}\{F_{i,}(a) \cdot g^i : i = 1, \ldots, m\}.$$

This maximization is used to obtain a sub-gradient envelope of the maximum over a family f^a, $a \in A$ of piecewise linear and convex functions in terms of the matrix representatives of their sub-gradient envelopes:

$$\mathcal{S}_G f^a \sim F(a), \;\; a \in A \quad \Rightarrow \quad \mathcal{S}_G(\bigvee_{a \in A} f^a) \sim \bigsqcup_{a \in A} F(a).$$

Finally, we emphasize that determining the sub-gradient envelope of the composition of a function with a linear mapping corresponds to a simple matrix product followed by a row-rearrangement. In other words, for each $d \times d$-matrix W, it holds that

$$l \sim L \quad \Longrightarrow \quad \mathcal{S}_G(l(W\cdot)) \sim \Upsilon_G[LW].$$

Observe that the rows of the matrix F representing a sub-gradient envelope $\mathcal{S}_G f$ of a convex piecewise linear function f can always be arranged such that $F = \Upsilon_G(F)$ holds. We say that a the sub-gradient representative F is in the *normal form* if it holds that $F = \Upsilon_G(F)$.

Since the double-modified backward induction involves maximization, summations and compositions with linear mappings applied to piecewise linear convex functions, it can be rewritten in terms of matrix operations. Let

us present the resulting algorithm:

Pre-calculations: Given a grid $G^m = \{g^1, \dots, g^m\}$, implement the row-rearrangement operator $\Upsilon = \Upsilon_{G^m}$ and the row maximization operator $\sqcup_{a\in A}$. Determine a distribution sampling $(W_t(k))_{k=1}^n$ of each disturbance W_t with the corresponding weights $(\nu_t(k))_{k=1}^n$ for $t = 1, \dots, T$. Given reward functions $(r_t)_{t=0}^{T-1}$ and scrap value r_T, determine the normal form of the matrix representatives of their sub-gradient envelopes

$$\mathcal{S}_{G^m} r_t(p, \cdot, a) \sim R_t(p, a), \qquad \mathcal{S}_{G^m} r_T(p, \cdot) \sim R_T(p)$$

for $t = 0, \dots, T-1$, $p \in P$ and $a \in A$. Introduce matrix representatives $V_t(p)$ for $t = 0, \dots, T$, $p \in P$ of each value function by

$$v_t^{n,m}(p, \cdot) \sim V_t(p) \quad \text{for } t = 0, \dots, T,\ p \in P$$

which are obtained via the following matrix of the backward induction:

Initialization: Start with the matrices

$$V_T(p) = R_T(p), \qquad \text{for all } p \in P.$$

Recursion: For $t = T-1, \dots, 0$ calculate for $p \in P$

$$V_t(p) = \sqcup_{a\in A}\big(R_t(p, a) + \sum_{k=1}^{n} \nu_{t+1}(k)\Upsilon[V_{t+1}(\alpha(p, a)) \cdot W_{t+1}(k)]\big) \tag{13}$$

5 Non-convex extension

In this section, we demonstrate that for non-convex of reward and scrap functions the above algorithm can be adapted, if the functions are representable as a difference of two convex functions. More precisely, assume that for all $t = 0, \dots, T-1$, and $p \in P$, $a \in A$ it holds that

$$r_t(p, \cdot, a) = \breve{r}_t(p, \cdot, a) - \hat{r}_t(p, \cdot, a), \tag{14}$$

and

$$r_T(p, \cdot) = \breve{r}_T(p, \cdot) - \hat{r}_T(p, \cdot) \tag{15}$$

with convex functions $\breve{r}_t(p, \cdot, a)$, $\hat{r}_t(p, \cdot, a)$, $\breve{r}_T(p, \cdot)$ and $\hat{r}_T(p, \cdot)$ for $p \in P$. Given such represention, the idea is to decompose the backward induction into parallel procedures that operate on convex functions. Suppose that at the step t, the value function v_{t+1} can be represented as a difference

$v_{t+1} = \breve{v}_{t+1} - \hat{v}_{t+1}$ of convex functions $\breve{v}_{t+1}(p,\cdot)$ and $\hat{v}_{t+1}(p,\cdot)$ for $p \in P$. With this, we have

$$\begin{aligned} \mathcal{T}_t v(p,z) &= \sup_{a\in A} \left(r_t(p,z,a) + \mathcal{K}_t^a v_{t+1}(p,z)\right) \\ &= \sup_{a\in A} \left([\breve{r}_t(p,z,a) + \mathcal{K}_t^a \breve{v}_{t+1}(p,z)] - [\hat{r}_t(p,z,a) + \mathcal{K}_t^a \hat{v}_{t+1}(p,z)] \right) \end{aligned}$$

showing that before maximization in $a \in A$, the result is obtained as difference of two convex functions. However, a direct application of convex function maximization (i.e. the use of the row maximization operator $\sqcup$) is not compatible with this decomposition. Therefore, we require a way to express the maximum over differences of convex functions as difference of two convex functions. The following simple observation helps here.

Consider for each $a \in A$ the difference $\breve{f}_a - \hat{f}_a$ of two convex functions $\breve{f}_a$ and $\hat{f}_a$ and let $\hat{f} := \sum_{a\in A} \hat{f}_a$. Then for each $a \in A$ the functions $\breve{f}_a - \hat{f}_a + \hat{f}$ and $\hat{f}$ are convex and yield the desired decomposition

$$\max_{a\in A}(\breve{f}_a - \hat{f}_a) = \max_{a\in A}(\breve{f}_a - \hat{f}_a + \hat{f}) - \hat{f}. \tag{16}$$

Having this approach in mind, we propose the following algorithm:

Pre-calculation: Decompose the reward $(r_t)_{t=0}^{T-1}$ and scrap r_T functions into a difference of convex functions as in (14) and (15) with their (normal form) matrix representatives

$$\begin{aligned} \mathcal{S}_{G^m}\breve{r}_t(p,\cdot,a) \sim \breve{R}_t(p,a), \qquad \mathcal{S}_{G^m}\hat{r}_t(p,\cdot,a) \sim \hat{R}_t(p,a), \\ \mathcal{S}_{G^m}\breve{r}_T(p,\cdot) \sim \breve{R}_T(p), \qquad \mathcal{S}_{G^m}\hat{r}_T(p,\cdot) \sim \hat{R}_T(p) \end{aligned} \tag{17}$$

for all $t = 0,\ldots,T-1$, $p \in P$ and $a \in A$. Introduce the approximate value functions $(v_t^{n,m})_{t=0}^T$ which possess the decomposition

$$v_t^{n,m} = \breve{v}_t^{n,m} - \hat{v}_t^{n,m} \tag{18}$$

where $\breve{v}_t^{n,m}(p,\cdot)$ and $\hat{v}_t^{n,m}(p,\cdot)$ are piecewise linear convex functions with matrix representatives

$$\breve{v}_t^{n,m}(p,\cdot) \sim \breve{V}_t(p) \quad \text{and} \quad \hat{v}_t^{n,m}(p,\cdot) \sim \hat{V}_t(p) \tag{19}$$

for $t = 0,\ldots,T$, $p \in P$.

Initialization: Start with the matrices

$$\breve{V}_T(p) = \breve{R}_T(p) \quad \text{and} \quad \hat{V}_T(p) = \hat{R}_T(p), \qquad \text{for all } p \in P.$$

Recursion: For $t = T-1, \ldots, 1$, calculate

$$\begin{array}{l} \breve{\Psi}_t(p,a) = \breve{R}_t(p,a) + \sum_{k=1}^n \nu_{t+1}(k)\Upsilon[\breve{V}_{t+1}(\alpha(p,a)) \cdot W_{t+1}(k)] \\ \hat{\Psi}_t(p,a) = \hat{R}_t(p,a) + \sum_{k=1}^n \nu_{t+1}(k)\Upsilon[\hat{V}_{t+1}(\alpha(p,a)) \cdot W_{t+1}(k)] \end{array} \tag{20}$$

and determine

$$\begin{array}{l} \hat{V}_t(p) = \sum_{a\in A} \hat{\Psi}_t(p,a) \\ \breve{V}_t(p) = \bigsqcup_{a\in A}(\breve{\Psi}_t(p,a) - \hat{\Psi}_t(p,a) + \hat{V}_t(p)). \end{array} \tag{21}$$

for all $p \in P$.

6 An efficient approximation

Although numerical experiments indicate stable and reliable results, it seems that the computational performance suffers from the fact that most of the calculation time is being spent on matrix rearrangements required by the operator Υ. We see from (13) that in order to calculate

$$\sum_{k=1}^{n} \nu_{t+1}(k)\Upsilon[V_{t+1}(\alpha(p,a)) \cdot W_{t+1}(k)] \tag{22}$$

at each step of the recursion, row-rearrangement must be performed n times, once for each disturbance matrix multiplication. This task becomes increasingly demanding for larger values of the disturbance sampling sizes n, particularly in high dimensions. Before we proceed, let us omit the time index $t+1$ in (22) to ease notation. We then focus on the two major sources of computational effort in evaluation of this expression, namely

$$\begin{array}{c} \text{the rearrangement } \Upsilon[VW(k)] \text{ of} \\ \text{large matrices } V \cdot W(k) \end{array} \tag{23}$$

and

$$\begin{array}{c} \text{the summation of matrices } \Upsilon[V \cdot W(k)] \text{ over} \\ \text{a large index range } k = 1, \ldots, n. \end{array} \tag{24}$$

The remainder of this section will be divided into two parts. In Section 7.1, we present a method that approximates (22), and addresses both problems simultaneously. The improvement in computational effort makes it feasible to obtain *approximate* solutions for large grids and distribution samples sizes. Furthermore, we will see that unlike (22), this approximation does not require $V = V_{t+1}$ to be convex. In Section 7.2, we derive a suitable first order approximation that provides a efficient way of evaluating

functions without having to decompose them into convex components. By combining this approximation with the method in Section 7.1, we obtain an efficient algorithm where we are no longer encumbered by the requirement of convexity.

6.1 Estimating the conditional expectation

The crucial point is that one can approximate the procedure in (23) by replacing the row-rearrangement operation with an appropriate matrix multiplication. More precisely, for $k = 1, \ldots, n$ we

$$\begin{array}{c} \text{construct a matrix } Y(k) \text{ such that} \\ Y(k)VW(k) \text{ approximates } \Upsilon[VW(k)]. \end{array} \tag{25}$$

Before we justify the approximation (25), let first us see how it can be used to address the computational problem associated with (24). Given (25), we now have the following approximation to (22)

$$\sum_{k=1}^{n} \nu(k)\Upsilon[VW(k)] \approx \sum_{k=1}^{n} \nu(k)Y(k)VW(k) \tag{26}$$

and this in turn requires an efficient calculation of sums of matrices. In practical examples, the distribution sample size n and the grid size m (row number of V) will typically be orders of magnitude of the dimension d of the disturbance matrices $W(k)$. For instance, to achieve an acceptable level of numerical convergence in typical applications, the sample size n and the grid size m must be chosen in the range of several thousands, whereas the state size dimension d is typically of several dozens. This insight shows that a significant reduction in computational effort can be achieved by an additive decomposition of the disturbance realizations. Assume that disturbance matrix W is represented as the linear combination

$$W = \bar{W} + \sum_{j=1}^{J} \epsilon_j E(j) \tag{27}$$

with non-random matrices $\bar{W}$ and $(E(j))_{j=1}^{J}$, and random coefficients $(\varepsilon_j)_{j=1}^{J}$. With this decomposition, each realization $W(k)$ of the disturbance matrix W is obtained as

$$W(k) = \bar{W} + \sum_{j=1}^{J} \epsilon_j(k) E(j), \qquad k = 1, \ldots, n. \tag{28}$$

Utilizing this, we obtain the following interchange of summations on the right-hand side of (26):

$$\sum_{k=1}^{n} \nu(k) Y(k) V W(k) = \left(\sum_{k=1}^{n} \nu(k) Y(k) \right) V \bar{W} + \sum_{j=1}^{J} \left(\sum_{k=1}^{n} \nu(k) \epsilon_j(k) Y(k) \right) V E(j).$$

If one pre-computes the following matrices

$$D_0 = \sum_{k=1}^{n} \nu(k) Y(k), \qquad D_j = \sum_{k=1}^{n} \nu(k) \epsilon_j(k) Y(k), \quad j = 1, \ldots, J, \tag{29}$$

we then obtain a significant simplification to (26)

$$\sum_{k=1}^{n} \nu(k) Y(k) V W(k) = D_0 V \bar{W} + \sum_{j=1}^{J} D_j V E(j) \tag{30}$$

which only involves a low number of matrix summations and multiplications. We shall denote this efficient calculation of the conditional expectation by $\mathcal{E}$ where

$$\mathcal{E}(V) := D_0 V \bar{W} + \sum_{j=1}^{J} D_j V E(j). \tag{31}$$

We now address the justification of the approximation in (25). Suppose that the grid $\{g^1, \ldots, g^m\}$ is represented by the matrix G where each row i contains row vector, representing the grid point g^i. Thus G will consist of m rows with $G_{i,\cdot} = g^i$ for $i = 1, \ldots, m$. Now let $\tilde{L} = \Upsilon[L]$ be the result the application of Υ to a matrix L. The matrix $\tilde{L}$ is then characterized by the following requirements:

$$\begin{aligned} &\tilde{L} = \Upsilon[L] \text{ consists of } m \text{ rows which are obtained} \\ &\qquad \text{from the rows of } L \text{ by a arrangement,} \end{aligned} \tag{32}$$

such that

$$\tilde{L}_{i,\cdot} \cdot G_{i,\cdot}^{\top} \geq L_{j,\cdot} \cdot G_{i,\cdot}^{\top} \quad \text{for all } i, j = 1, \ldots, m. \tag{33}$$

According to requirement (32), we therefore assume that

$$\begin{aligned} &\Upsilon[VW(k)] \text{ consists of } m \text{ rows which are obtained} \\ &\text{from the rows of } VW(k) \text{ by row-rearrangement.} \end{aligned} \tag{34}$$

Since any row rearrangement can be achieved by a left-multiplication with appropriate matrix, there will always exist a permutation matrix $Y_V(k)$ such that

$$Y_V(k) V W(k) = \Upsilon[VW(k)]. \tag{35}$$

Computing each $Y_V(k)$ requires great effort since it is not only dependent on $W(k)$, but also on each V. We suggest determining a reasonable surrogate $Y(k)$ for $Y_V(k)$ which depends only on $W(k)$ and not on V. Since $Y_V(k)$ must satisfy (35), we observe with (33) in mind that

$$(Y_V(k)V)_{i,\cdot} \cdot (W(k)G)_{i,\cdot}^\top \geq V_{j,\cdot} \cdot (W(k)G)_{i,\cdot}^\top \quad \text{for } i,j = 1,\ldots,m. \tag{36}$$

Now, for each $i = 1,\ldots,m$ consider the row $(W(k)G)_{i,\cdot}$ and determine the closest row $G_{h_k(i),\cdot}$ in the original grid matrix by

$$h_k(i) = \operatorname{argmin}\{j = 1,\ldots,m \,:\, \|(W(k)G)_{i,\cdot} - G_{j,\cdot}\|\}, \quad i = 1,\ldots,m. \tag{37}$$

With this proximity function $h_k : \{1,\ldots,m\} \to \{1,\ldots,m\}$, we may consider, in place of the relation (36), the condition

$$(Y(k)V)_{i,\cdot} \cdot G_{h_k(i),\cdot}^\top \geq V_{j,\cdot} \cdot G_{h_k(i),\cdot}^\top \quad \text{for all } i,j = 1,\ldots,m \tag{38}$$

with an appropriate permutation matrix $Y(k)$. While (38) is clearly not equivalent to (36), it does provide a reasonable approximation when the grid is sufficiently dense. Now define $Y(k)$ to be such that

$$Y(k)_{i,j} = \begin{cases} 1, & \text{if } j = h_k(i) \\ 0, & \text{otherwise} \end{cases} \tag{39}$$

and observe that with this permutation matrix $Y(k)$, the following assertion

$$V_{h_k(i),\cdot} \cdot G_{h_k(i),\cdot}^\top \geq V_{j,\cdot} \cdot G_{h_k(i),\cdot}^\top \quad \text{for all } i,j = 1,\ldots,m$$

holds if V is in the normal form $\Upsilon[V] = V$. That is, the required approximation (25) is determined by (39).

The pre-calculations involved in the approximation of (22) (i.e. computing $D_0,\ldots,D_J$) are computationally demanding. Thus, a gain in computation performance can only be realized if disturbances $(W_t)_{t=1}^T$ are identically distributed whereby the pre-calculations need only be done once. In this case, the ideas presented in this section will be encapsulated in the following algorithm.

Pre-calculations: Determine a sampling $(W(k))_{k=1}^n$ from the target distribution. For each disturbance $W(k)$, find the corresponding permutation matrix $Y(k)$ as in (39) using the proximity function (37). Use these matrices and the components of the decomposition described in (28) of each $W(k)$ to compute the matrices (29).

Continuation: Execute the algorithm (17) – (21) but replace (20) with

$$\begin{aligned} \breve{\Psi}_t(p,a) &= \breve{R}_t(p,a) + \mathcal{E}[\breve{V}_{t+1}(\alpha(p,a))] \quad \text{and} \\ \hat{\Psi}_t(p,a) &= \hat{R}_t(p,a) + \mathcal{E}[\hat{V}_{t+1}(\alpha(p,a))]. \end{aligned}$$

by substituting the conditional expectations with its efficient counterpart (31).

6.2 A direct approach

So far, we have worked with parallel procedures on convex functions. However, an important point to note is that in no part of the efficient conditional expectation procedure was the convexity of the target function required. With this in mind, we shall now present a further simplification to this algorithm based on a first-order approximation. Previously, we considered the convex decomposition $f = \hat{f} - \breve{f}$ of a non-convex function function f where the two convex piecewise linear functions $\hat{f}$ and $\breve{f}$ with respective matrix representatives $\hat{F}$ and $\breve{F}$. The value $f(z)$ at point z is then calculated as

$$f(z) = \max(\hat{F}z) - \max(\breve{F}z).$$

However, if only the matrix difference $\hat{F} - \breve{F}$ is known then it is possible to use a first-order approximation

$$f(z) \approx (\hat{F} - \breve{F})(h(z)).$$

where h is the so-called host function of the underlying grid G

$$h(z) = \operatorname{argmin}\{\|z - g\| : g \in G\}$$

which returns to each argument $z \in \mathbb{R}^d$ the so-called host - the point on the grid with the smallest distance to z. The first-order approximation uses the difference $\hat{F} - \breve{F}$ directly and unlike convex decomposition, does not require a separate calculation of convex and concave parts. If one decides to use this first-order approximation to access the functions, then there is no need to trace convex and concave part separately. This gives a significant simplification and results in the following direct algorithm:

Pre-calculations: Determine the operator $\mathcal{E}$ as in (31), under the assumptions required therefore. Determine for $p \in P$, $a \in A$ and $t = 0, \ldots, T-1$ the matrices

$$R_t(p,a) = \hat{R}_t(p,a) - \breve{R}_t(p,a), \quad R_T(p) = \hat{R}_T(p) - \breve{R}_T(p), \tag{40}$$

which are obtained as in (17). Introduce the approximate value functions, their convex decomposition and representatives as in (18) and (19). The matrices

$$V_t(p) = \breve{V}_t(p) - \hat{V}_t(p) \quad \text{for } t = 0, \ldots, T,\ p \in P$$

are obtained via the following scheme:

Initialization: Start with the matrices

$$V_T(p) = \breve{R}_T(p) - \hat{R}_T(p), \qquad \text{for all } p \in P$$

Recursion: For $t = T-1, \ldots, 1$ calculate for $p \in P$

$$V_t(p) = \bigsqcup_{a \in A} \left(R_t(p,a) + \mathcal{E}(V_{t+1}(p,a)) \right). \tag{41}$$

Remark: Unlike in the convex decomposition case (17) – (21), the *direct* algorithm (40) – (41) merely returns the difference $V_t(p) = \check{V}_t(p) - \hat{V}_t(p)$. That is, the access to the approximate value functions is provided via

$$v_t^{m,n}(p,z) \approx V_t(p) \cdot h(z),$$

using the host function $z \mapsto h(z) = \operatorname{argmin}\{\|z-g\| \ : \ g \in G^m\}$ of the grid G^m. In particular, we suggest an approximation of the optimal policy $\pi^{m,n} = (\pi_t^{m,n})_{t=0}^{T-1}$ as

$$\pi_t^{m,n}(p,z) = \operatorname{argmax}_{a \in A} \left((R_t(p,a) + \mathcal{E}(V_{t+1}(p,a))) \cdot h(z) \right), \tag{42}$$

for $t = 0, \ldots, T-1$, $z \in \mathbb{R}^d$, $p \in P$. To obtain an efficient implementation of a host function h, a tree-like structure on the grid can be used which may be established using hierarchical clustering methods.

7 Solution diagnostics

In Section 7.1, we derived a heuristic method to obtain an efficient approximation to the conditional expectation in the Bellman recursion. In Section 7.2, we saw that by combining this with a first order approximation, we were then able to obtain the approximation (42) for a given grid to the optimal policy in (3). In order to address the distance to optimality of this approximate solution, we first need to outline an appropriate measure for this distance.

Suppose we are given an arbitrary policy $\pi = (\pi_t)_{t=0}^{T-1}$. For such a policy one can define an associated set of *policy values* $(v_t^\pi(p,z))_{t=0}^T$ that follow the recursion

$$\begin{aligned} v_T^\pi(p,z) &= r_T(p,z) && (43) \\ v_t^\pi(p,z) &= r_t(p,z,\pi_t(p,z)) + \mathbb{E}(v_{t+1}^\pi(\alpha(p,\pi_t(p,z)), W_{t+1}z)), && (44) \end{aligned}$$

for $t = T-1, \ldots, 0$. Let us consider a switching system which starts in a given initial position $p_0^\pi = p_0 \in P$ and state $Z_0 = z_0 \in \mathbb{R}^d$. At any time t, the actions and new positions are determined recursively, following policy $\pi = (\pi_t)_{t=0}^{T-1}$ as

$$a_t^\pi := \pi_t(p_t^\pi, Z_t), \quad p_{t+1}^\pi := \alpha(p_t^\pi, a_t^\pi), \quad t = 0, \ldots, T-1.$$

These values define a *policy run* $\mathcal{V}_0^\pi(p_0^\pi, z_0)$ where

$$\mathcal{V}_0^\pi(p_0^\pi, Z_0) = \sum_{s=0}^{T-1} r_t(p_s^\pi, Z_s, a_s^\pi) + r_T(p_T^\pi, Z_T)$$

According to the definition, $v_0^\pi(p_0, z_0)$ is the expected value of the policy run

$$v_0^\pi(p_0, z_0) = \mathbb{E}(\mathcal{V}_0^\pi(p_0, z_0)) \quad p \in P, \quad z \in \mathbb{R}^d.$$

In practice, one can use Monte Carlo to estimate this value since given a sequence $(\omega_k)_{k\in\mathbb{N}}$ independent random draws,

$$v_0^\pi(p_0, z_0) = \mathbb{E}(\mathcal{V}_0^\pi(p_0, z_0)) = \lim_{k\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathcal{V}_0^\pi(p_0, z_0)(\omega_k) \tag{45}$$

holds true due to the strong law of large numbers.

Such a Monte Carlo procedure may be useful estimating the performance of a given strategy π. However, it does not clarify how far the value $v_0^\pi(p_0, z_0)$ is from what is theoretically possible $v_0^{\pi^*}(p_0, z_0)$ is the optimal policy π^* was known.

In the reminder of the section, we suggest a sound solution to this question in terms of a *diagnostic method*. Given a starting point (p_0, z_0), we explain how the gap

$$[v_0^\pi(p_0,\ z_0), v_0^{\pi^*}(p_0, z_0)] \tag{46}$$

between a given strategy π and the optimal strategy π^* can be assessed. Our methodology is based on a finite sample $\{\omega_1, \ldots, \omega_K\}$ of trajectory realizations and utilizes to a build-in variance reduction technique to derive *tight confidence bounds* for upper and lower estimates of the interval (46).

Let us focus on the upper bound first. Consider a sequence $\varphi = (\varphi_t)_{t=1}^T$ of random mappings

$$\varphi_t : P \times \mathbb{R}^d \times A \times \Omega \to \mathbb{R}, \qquad (p, z, a, \omega) \mapsto \varphi_t(p, z, a)(\omega), \tag{47}$$

which for $t = 1, \ldots, T$ satisfy

$$\mathbb{E}(\varphi_t(p, z, a)) = 0, \qquad p \in P, z \in \mathbb{R}^d, a \in A \tag{48}$$

such that the σ-algebras

$$\sigma(\varphi_t(p, z, a), W_t; a \in A, z \in \mathbb{R}^d), \qquad t = 1, \ldots, T, \tag{49}$$

are independent. Given these mappings $\varphi = (\varphi_t)_{t=1}^T$, we now introduce the random functions $(\bar{v}_t^\varphi)_{t=0}^T$

$$\bar{v}_t^\varphi : P \times \mathbb{R}^d \times \Omega \to \mathbb{R}, \qquad t = 0, \ldots, T$$

which are recursively defined for $t = T, \dots, 1$ via

$$\begin{aligned} \bar{v}_T^{\varphi}(p,z) &= r_T(p,z) && (50)\\ \bar{v}_t^{\varphi}(p,z) &= \max_{a\in A}\big(r_t(p,z,a) + \varphi_{t+1}(p,z,a) + \bar{v}_{t+1}^{\varphi}(\alpha(p,a), W_{t+1}z)\big). && (51) \end{aligned}$$

Using $(\bar{v}_t^{\varphi})_{t=0}^T$, the following theorem holds:

Theorem 1. *(i)For each policy $\pi = (\pi_t)_{t=0}^{T-1}$, it holds that the policy values $(v_t^{\pi})_{t=0}^T$ are dominated from above*

$$v_t^{\pi}(p,z) \le \mathbb{E}(\bar{v}_t^{\varphi}(p,z)), \qquad \text{for all } t = 0,\dots,T,\ p \in P,\ z \in \mathbb{R}^d. \tag{52}$$

(ii) Given the policy values $(v_t^{\pi^*})_{t=0}^T$ associated with the optimal policy $\pi^* = (\pi_t^*)_{t=0}^{T-1}$, let $(\varphi_t^*)_{t=1}^T$ be defined by

$$\varphi_{t+1}^*(p,z,a) = \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)) - v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) \tag{53}$$

for all $p \in P$, $z \in \mathbb{R}^d$, $a \in A$ and $t = 0,\dots,T-1$. It then holds that the mappings $(\varphi_t^*)_{t=1}^T$ satisfy (47) – (49) and that (52) holds with equality

$$v_t^{\pi^*}(p,z) = \bar{v}_t^{\varphi^*}(p,z), \qquad \text{for all } t = 0,\dots,T,\ p \in P,\ z \in \mathbb{R}^d. \tag{54}$$

Proof. (i) The value $(v_t^{\pi})_{t=0}^T$ of the policy $\pi = (\pi_t)_{t=0}^{T-1}$ satisfies the recursion (44). Using this recursion and (48) we obtain

$$\begin{aligned} v_t^{\pi}(p,z) &= \mathbb{E}(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z)))\\ &\quad + \mathbb{E}(v_{t+1}^{\pi}(\alpha(p,\pi_t(p,z)), W_{t+1}z)). \end{aligned} \tag{55}$$

Now, let us prove the assertion (52) by induction. For $t = T$, the inequality (52) holds with equality because of the initialization

$$v_T^{\pi} = r_T = \bar{v}_T^{\varphi} \tag{56}$$

in (43) and (50). Given the induction assumption

$$v_{t+1}^{\pi}(p,z) \le \mathbb{E}(\bar{v}_{t+1}^{\varphi}(p,z)), \qquad \text{for all } p \in P,\ z \in \mathbb{R}^d,$$

we use (49) to conclude that

$$v_{t+1}^{\pi}(\alpha(p,\pi_t(p,z)), W_{t+1}z) \le \mathbb{E}(\bar{v}_{t+1}^{\varphi}(\alpha(p,\pi_t(p,z)), W_{t+1}z) \,|\, W_{t+1})$$

holds for all $p \in P$, $z \in \mathbb{R}^d$. Using this, we obtain in (55) an estimate

$$\begin{aligned} v_t^{\pi}(p,z) &\le \mathbb{E}\left(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z))\right)\\ &\quad + \mathbb{E}\left(\mathbb{E}(\bar{v}_{t+1}^{\varphi}(\alpha(p,\pi_t(p,z)), W_{t+1}z) \,|\, W_{t+1})\right) \end{aligned}$$

from which the assertion follows

$$\begin{aligned} v_t^{\pi}(p,z) &\leq \mathbb{E}\Big(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z)) + \bar{v}_{t+1}^{\varphi}(\alpha(p,\pi_t(p,z)), W_{t+1}z)\Big) \\ &\leq \mathbb{E}\left(\max_{a\in A}\left[r_t(p,z,a) + \varphi_{t+1}(p,z,a) + \bar{v}_{t+1}^{\varphi}(\alpha(p,a), W_{t+1}z)\right]\right) \\ &\leq \mathbb{E}\left(\bar{v}_t^{\varphi}(p,z)\right), \end{aligned}$$

where the last step results from the recursion (51).

(ii) Now suppose that π^* is an optimal policy and define $\varphi^* = (\varphi_t^*)_{t=1}^T$ as in (53), which satisfies the assumption (48). Furthermore, the independence (49) holds since for $t = 0, \ldots, T-1$ the random component in $\varphi_{t+1}^*(p,z,a)$ is

$$v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)$$

which is in turn a function of W_{t+1}. Let us now verify the assertion (54). By induction which is started as in (50) we can assume that $v_{t+1}^{\pi^*} = \bar{v}_{t+1}^{\varphi^*}$ holds. Using this, we conclude for all $p \in P$, $z \in \mathbb{R}^d$ and $a \in A$ the assertion

$$\begin{aligned} &r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + \bar{v}_{t+1}^{\varphi^*}(\alpha(p,a), W_{t+1}z) \\ &\quad = \; r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z). \end{aligned} \tag{57}$$

On the other hand, using (53), we infer that for all $p \in P$, $z \in \mathbb{R}^d$ and $a \in A$ it holds that

$$\begin{aligned} &r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) \\ &\quad = \; r_t(p,z,a) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)) \qquad (58) \\ &\qquad -v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) + v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) \\ &\quad = \; r_t(p,z,a) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)) \qquad (59) \end{aligned}$$

Now, in the recursion (51) we replace (57) by (59) to obtain the desired result (54)

$$\begin{aligned} \bar{v}_t^{\varphi^*}(p,z) &= \max_{a\in A}\Big(r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + \bar{v}_{t+1}^{\varphi^*}(\alpha(p,a), W_{t+1}z)\Big) \\ &= \max_{a\in A}\left(r_t(p,z,a) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z))\right) \\ &= v_t^{\pi^*}(p,z), \qquad p \in P,\;\; z \in \mathbb{R}^d,\;\; a \in A. \end{aligned}$$

□

Let us elaborate on a practical application of this technique. Suppose that we attempt to assess the distance to optimality of an approximate policy $\tilde{\pi}$,

obtained by a numerical procedures described previously. According to (i) of the Theorem 1, arbitrary $(\varphi_t)_{t=1}^T$ satisfying (48) and (49) yields an upper bound

$$v_0^{\tilde{\pi}}(p,z) \leq v_0^{\pi^*}(p,z) \leq \mathbb{E}(\bar{v}_0^{\varphi}(p,z)) \qquad p \in P,\ \ z \in \mathbb{R}^d, \tag{60}$$

Note that the expectation $\mathbb{E}(\bar{v}_0^{\varphi}(p,z))$ will be estimated via Monte Carlo. Thus, we obtain the following estimation procedure:

Upper bound estimation:

1) Given a switching system, implement $(\varphi_t)_{t=1}^T$ which fulfills (47), (48) and (49).

2) Chose a number $K \in \mathbb{N}$ of Monte-Carlo trials and obtain for $k = 1, \ldots, K$ independent realizations $(W_t(\omega_k))_{t=1}^T$ of disturbances.

3) Starting at $z_0^k := z_0 \in \mathbb{R}^d$, define for $k = 1, \ldots, K$ the trajectories $(z_t^k)_{t=0}^T$ recursively

$$z_{t+1}^k = W_{t+1}(\omega_k) z_t^k, \qquad t = 0, \ldots, T-1$$

and determine realizations

$$\varphi_{t+1}(p, z_t^k, a)(\omega_k), \qquad t = 0, \ldots, T-1, \quad k = 1, \ldots, K.$$

4) For each $k = 1, \ldots, K$ initialize the recursion at $t = T$ as

$$\bar{v}_T^{\varphi}(p, z_T^k) = r_T(p, z_T^k) \qquad \text{for all } p \in P$$

and continue for $t = T-1, \ldots, 0$ by

$$\bar{v}_t^{\varphi}(p, z_t^k) \ = \ \max_{a \in A} \big(r_t(p, z_t^k, a) + \varphi_{t+1}(p, z_t^k, a)(\omega_k) + \bar{v}_{t+1}^{\varphi}(\alpha(p,a), z_{t+1}^k) \big).$$

Store the value as $\bar{v}_0^{\varphi}(p, z_0^k)$ for $k = 1, \ldots, K$.

5) Determine the sample mean $\frac{1}{K} \sum_{k=1}^K \bar{v}_0^{\varphi}(p, z_0^k)$ and its upper confidence bound to estimate $v_0^{\pi^*}(p, z_0)$ from above.

To obtain a tight upper bound, $(\varphi_t)_{t=1}^T$ must be chosen accordingly. Thereby, the assertion (ii) of Theorem 1 suggests an appropriate choice. Namely, in the hypothetical case that the optimal policy value functions $(v_t^{\pi *})_{t=0}^T$ are known, the $(\varphi_t^*)_{t=1}^T$ is obtained via (53) will give an exact and non-random upper bound. In practice, this situation is not feasible, since an optimal strategy π^* is not known. Instead, we suggest using an approximate value function $(\tilde{\varphi}_t)_{t=0}^T$, returned by one of the algorithms described in this work.

That is, following (53), a reasonable candidate for $t = 0, \ldots, T-1$ could be given as

$$\varphi_{t+1}(p, z, a) \quad = \quad \mathbb{E}(\tilde{v}_{t+1}(\alpha(p, a), W_{t+1}z)) - \tilde{v}_{t+1}(\alpha(p, a), W_{t+1}z). \quad (61)$$

However, note that this choice involves an exact calculation of expectation $\mathbb{E}(\tilde{v}_{t+1}(\alpha(p, a), W_{t+1}z))$, which is not possible in practice. For this reason, we suggest a modification. We introduce φ_{t+1} as in (61), with the expectation replaced by an arithmetic mean over a number I of independent copies $(W_{t+1}^{(i)})_{i=1}^{I}$ of W_{t+1}. That is, given independent random variables W_{t+1} and $W_{t+1}^{(i)}$ for $i = 1, \ldots, I$ and $t = 0, \ldots, T-1$ such that the distribution of $W_{t+1}^{(i)}$ equals to that of W_{t+1}, we define for all $t = 0, \ldots, T-1$, $a \in A$, $p \in P$, and $z \in \mathbb{R}^d$

$$\varphi_{t+1}(p, z, a) = \frac{1}{I} \sum_{i=1}^{I} \tilde{v}_{t+1}(\alpha(p, a), W_{t+1}^{(i)}z) - \tilde{v}_{t+1}(\alpha(p, a), W_{t+1}z). \quad (62)$$

With this definition, $(\varphi_t)_{t=1}^{T}$ satisfies conditions (48) and (49), and we thus obtain a valid and computable upper bound.

Let us turn now to the estimation of the lower boundary of the interval (46). Given a strategy $\pi = (\pi_t)_{t=0}^{T-1}$, the value $v_0^{\pi}(p_0, z_0)$ can in principle be approached as in (45) from test runs of the strategy in a series of independent back-testing experiments. However, it turns out that a slight adaptation of the upper bound technique provides far better results, due to a built-in variance reduction technique. Similar to part (ii) of the previous theorem, which indicates that the variance of the Monte Carlo trials reduces if approximate solution is close to the optimal one, we establish a recursive procedure with a control variate built-in. The idea is simple: Given a nearly-optimal policy $\pi = (\pi_t)_{t=0}^{T-1}$ we alter the recursion (50), (51) replacing the maximization by an exact choice of the action according to the policy $\pi = (\pi_t)_{t=0}^{T-1}$.

Given a sequence $\varphi = (\varphi_t)_{t=1}^{T}$ satisfying (48) and (49) we introduce the random functions $(\underline{v}_t^{\pi,\varphi})_{t=0}^{T}$

$$\underline{v}_t^{\pi,\varphi} : P \times \mathbb{R}^d \times \Omega \to \mathbb{R}, \qquad t = 0, \ldots, T$$

which are recursively defined for $t = T, \ldots, 1$ via

$$\underline{v}_T^{\pi,\varphi}(p, z) \quad = \quad r_T(p, z) \quad (63)$$

$$\underline{v}_t^{\pi,\varphi}(p, z) \quad = \quad r_t(p, z, \pi_t(p, z)) + \varphi_{t+1}(p, z, \pi_t(p, z)) + \underline{v}_{t+1}^{\pi,\varphi}(\alpha(p, \pi_t(p, z)), W_{t+1}z). \quad (64)$$

The following theorem holds for $(\underline{v}_t^{\pi,\varphi})_{t=0}^{T}$.

Theorem 2. *(i) Given $\varphi = (\varphi_t)_{t=1}^T$ as in (47) satisfying (49) and a policy $\pi = (\pi_t)_{t=0}^{T-1}$, introduce $(\underline{v}_t^{\pi,\varphi})_{t=0}^T$ by (63), (64). It holds that*

$$v_t^\pi(p,z) = \mathbb{E}(\underline{v}_t^{\pi,\varphi}(p,z)), \qquad \text{for all } t = 0,\dots,T,\ p \in P,\ z \in \mathbb{R}^d. \tag{65}$$

(ii) Given the value $(v_t^{\pi^*})_{t=0}^T$ of the optimal policy $\pi^* = (\pi_t^*)_{t=0}^{T-1}$, define $(\varphi_t^*)_{t=1}^T$

$$\varphi_{t+1}^*(p,z,a) = \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a),W_{t+1}z)) - v_{t+1}^{\pi^*}(\alpha(p,a),W_{t+1}z) \tag{66}$$

for all $p \in P$, $z \in \mathbb{R}^d$, $a \in A$ and $t = 0,\dots,T-1$. Then the mappings $(\varphi_t^*)_{t=1}^T$ satisfy (47) – (49) such that (65) holds with equality:

$$v_t^{\pi^*}(p,z) = \underline{v}_t^{\pi^*,\varphi^*}(p,z), \qquad \text{for all } t = 0,\dots,T,\ p \in P,\ z \in \mathbb{R}^d. \tag{67}$$

Proof. (i) The value $(v_t^\pi)_{t=0}^T$ of the policy $\pi = (\pi_t)_{t=0}^{T-1}$ satisfies the recursion (44). Using this recursion and (48) we obtain

$$\begin{aligned} v_t^\pi(p,z) &= \mathbb{E}(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z))) \\ &\quad +\mathbb{E}(v_{t+1}^\pi(\alpha(p,\pi_t(p,z)),W_{t+1}z). \end{aligned} \tag{68}$$

Now, let us prove the assertion (65) by induction. For $t = T$ the inequality (65) holds with equality because of the initialization

$$v_T^\pi = r_T = \underline{v}_T^{\pi,\varphi} \tag{69}$$

in (43) and (63). Given the induction assumption

$$v_{t+1}^\pi(p,z) = \mathbb{E}(\underline{v}_{t+1}^{\pi,\varphi}(p,z)), \qquad \text{for all } p \in P,\ z \in \mathbb{R}^d,$$

we use (49) to conclude that

$$v_{t+1}^\pi(\alpha(p,\pi_t(p,z)),W_{t+1}z) = \mathbb{E}(\underline{v}_{t+1}^{\pi,\varphi}(\alpha(p,\pi_t(p,z)))\,|\,W_{t+1})$$

holds for all $p \in P$, $z \in \mathbb{R}^d$. Using this, we obtain in (68) the equality

$$\begin{aligned} v_t^\pi(p,z) &= \mathbb{E}\left(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z))\right) \\ &\quad +\mathbb{E}\left(\mathbb{E}(\underline{v}_{t+1}^{\pi,\varphi}(\alpha(p,\pi_t(p,z)))\,|\,W_{t+1})\right) \\ &= \mathbb{E}\big(r_t(p,z,\pi_t(p,z)) + \varphi_{t+1}(p,z,\pi_t(p,z)) + \underline{v}_{t+1}^{\pi,\varphi}(\alpha(p,\pi_t(p,z)))\big). \end{aligned}$$

By using the recursion (64), the assertion (65) follows.

(ii) Let us now verify the assertion (67). By induction which is started as in (63) we can assume that $v_{t+1}^{\pi^*} = \underline{v}_{t+1}^{\pi^*,\varphi^*}$ holds. Using this, we conclude for all $p \in P$, $z \in \mathbb{R}^d$ and $a \in A$ the assertion

$$\begin{aligned} &r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + \underline{v}_{t+1}^{\pi^*,\varphi^*}(\alpha(p,a),W_{t+1}z) \\ &\quad = r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + v_{t+1}^{\pi^*}(\alpha(p,a),W_{t+1}z). \end{aligned} \tag{70}$$

On the other hand, using (53), we infer that for all $p \in P$, $z \in \mathbb{R}^d$ and $a \in A$ it holds that

$$\begin{aligned} & r_t(p,z,a) + \varphi_{t+1}^*(p,z,a) + v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) \\ &= r_t(p,z,a) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)) \\ &\quad - v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) + v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z) \\ &= r_t(p,z,a) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,a), W_{t+1}z)) \end{aligned} \tag{71}$$

Now, in the recursion (64) we replace (70) by (71) to obtain the desired claim (54):

$$\begin{aligned} \underline{v}_t^{\pi^*,\varphi^*}(p,z) &= r_t(p,z,\pi_t^*(p,z)) + \varphi_{t+1}^*(p,z,\pi_t^*(p,z)) \\ &\quad + \underline{v}_{t+1}^{\pi^*,\varphi^*}(\alpha(p,\pi_t^*(p,z)), W_{t+1}z) \\ &= r_t(p,z,\pi_t^*(p,z)) + \mathbb{E}(v_{t+1}^{\pi^*}(\alpha(p,\pi_t^*(p,z)), W_{t+1}z) \\ &= v_t^{\pi^*}(p,z), \qquad p \in P,\ \ z \in \mathbb{R}^d,\ \ a \in A. \end{aligned}$$

□

The practical implementation of the lower bound estimation is based on the same realization of $(\varphi_t)_{t=1}^T$ as in (62), using independent copies of disturbances. Let us summarize this procedure as follows:

Lower bound estimation:

1) Given approximate value functions $(\tilde{v}_t)_{t=0}^T$ and a corresponding strategy $\tilde{\pi} = (\tilde{\pi}_t)_{t=0}^{T-1}$, chose $\varphi = (\varphi_t)_{t=0}^{T-1}$ as in (62).

2) Given $K \in \mathbb{N}$ Monte-Carlo trials, obtain for $k = 1, \ldots, K$ independent realizations $(W_t(\omega_k))_{t=1}^T$ of disturbances.

3) Starting at $z_0^k := z_0 \in \mathbb{R}^d$, define for $k = 1, \ldots, K$ trajectories $(z_t^k)_{t=0}^T$ recursively
$$z_{t+1}^k = W_{t+1}(\omega_k) z_t^k, \qquad t = 0, \ldots, T-1$$
and determine realizations
$$\varphi_{t+1}(p, z_t^k, a)(\omega_k), \qquad t = 0, \ldots, T-1, \quad k = 1, \ldots, K.$$

4) For each $k = 1, \ldots, K$ initialize the recursion at $t = T$ as
$$\underline{v}_T^{\tilde{\pi},\varphi}(p, z_T^k) = r_T(p, z_T^k) \qquad \text{for all } p \in P$$

and continue for $t = T-1, \ldots, 0$ and for all $p \in P$ by

$$\begin{aligned} \underline{v}_t^{\tilde{\pi},\varphi}(p, z_t^k) &= r_t(p, z_t^k, \tilde{\pi}_t(p, z_t^k)) + \varphi_{t+1}(p, z_t^k, \tilde{\pi}_t(p, z_t^k))(\omega_k) \\ &\quad + \underline{v}_{t+1}^{\tilde{\pi},\varphi}(\alpha(p, \tilde{\pi}_t(p, z_t^k)), z_{t+1}^k). \end{aligned} \tag{72}$$

Store the value as $\underline{v}_0^{\tilde{\pi},\varphi}(p, z_0^k)$ for $k = 1, \ldots, K$, $p \in P$.

5) Calculate the sample mean $\frac{1}{K}\sum_{k=1}^{K} \underline{v}_0^{\tilde{\pi},\varphi}(p, z_0^k)$ and use its lower confidence bounds to estimate $v_0^{\pi^*}(p, z_0)$ for $p \in P$ from below.

8 Examples

In the literature, the estimation of a complementary upper bound for the optimal stopping problem relies heavily on the concept of martingale duality and has been addressed in [14], [24] and [16]. From a computational perspective, achieving a sufficiently tight upper bound is equivalent to constructing a "good" martingale and tractable algorithm to do so was given by [1]. Upper bound methods have since been extended to the more general class of optimal multiple stopping problems by [19], [26] and [17]. Finally, the work [25] generalizes this technique to a wider class of discrete-time stochastic control problems. The combination of upper and lower bound methods is known as *primal-dual simulation*. In this section, we compare our technique to results achieved using standard least-squares regression method.

We will now perform value function approximations using the method outlined in Section 7 and the associated diagnostics established in Section 8 on two examples of Markov decision problems, an optimal stopping and an optimal multiple exercise problem. Optimal stopping problems are an important subclass of Markov decision problems (see Chapters 10 and 11 of [2]), whose upper bound estimation using duality is well-studied. As an illustration of our approach, we obtain in Section 9.1 bounds on the price of the *Bermudan* put option, a practically important discrete time optimal stopping problem. In Section 9.2, we use these methods to obtain numerical solutions to an optimal multiple stopping problem - the *swing* option (see [6]). A swing option allows the holder to buy a fixed quantity of the underlying at a predetermined price more than once before the maturity of the option. However there is a limit to the maximum number of times this can be done. Swing options are predominant in commodity markets, particularly in the energy sector.

For both applications, we will consider the evolution of the discounted asset price $(S_t)_{t=0}^T$ in discrete time, with respect to a risk-neutral measure. The dynamics $(S_t)_{t=0}^T$ of the discounted price depends on the asset type.

For the Bermudan put, the discounted price process $(S_t)_{t=0}^T$ is modelled as a martingale in the risk-neutral measure. For the swing option, we suppose that the price process $(S_t)_{t=0}^T$ is modeled by the exponential of an Ornstein-Uhlenbeck process to explain the mean-reverting price property naturally expected for commodity prices.

The logarithm $(\tilde{Z}_t)_{t=0}^T$ of the price forms the continuous component of our state dynamics. In practice, a further transformation of the state space is usually required before linear state dynamics can be achieved. In most cases, an augmentation with 1 via

$$Z_t = \begin{bmatrix} \tilde{Z}_t \\ 1 \end{bmatrix}, \qquad t = 0, \ldots, T.$$

is needed to represent the evolution of the continuous state component. In this representation, the system state follows a multiplicative dynamic

$$Z_{t+1} = W_{t+1} Z_t, \qquad t = 0, \ldots, T-1$$

with independent and identically distributed matrix-valued random variables $(W_t)_{t=1}^T$. The entries of these disturbance matrices reflect the underlying process model.

The grid choice is a key ingredient in the algorithm. For multi-variate state processes, a convenient way of grid construction is by simulation of appropriate trajectories. Thus, we create a grid of a desired size by simulating and storing a sufficient number of paths of $(Z_t)_{t=0}^{k_p T}$ of an appropriate length $k_p T \in \mathbb{N}$. In our examples, we have used a number of steps that is twice of the time horizon ($k_p = 2$). The distribution of disturbances is approximated by a discrete distribution. For this, a sample of $(W(k))_{k=1}^n$ of independent realizations was generated and stored. All required steps from Section 7 and the Monte-Carlo simulation for diagnostics refer to this discrete distribution approximation. For bound computations, we use confidence intervals based on K simulated trajectories. More precisely, we quote the intervals as

$$\left[\underline{\mu} - \Phi^{-1}(1 - \frac{x}{2}) \frac{\underline{\sigma}}{\sqrt{K}}, \quad \overline{\mu} + \Phi^{-1}(1 - \frac{x}{2}) \frac{\overline{\sigma}}{\sqrt{K}} \right] \tag{73}$$

where $1-x$ denotes the confidence level and $(\underline{\mu}, \underline{\sigma})$ and $(\overline{\mu}, \overline{\sigma})$ denote the sample mean and sample standard deviation of $(\underline{v}_0^{\tilde{\pi},\varphi}(p, z_0^k))_{k=1}^K$ and $(\bar{v}_0^{\varphi}(p, z_0^k))_{k=1}^K$ respectively.

8.1 The Bermudan put option

This option allows the holder to sell the underlying asset at a pre-specified strike price on a discrete set of exercise dates up to and including the ex-

piry date of the option. The fair price of a Bermudian put is given by the supremum

$$\sup_{\tau} \mathbb{E}[(Ke^{-\rho\tau} - S_\tau)^+]$$

where τ runs through all $\{0, \ldots, T\}$-valued stopping times. First let us express this control problem as a switching system. We use the position set $P = \{1, 2\}$ to indicate whether the option has been exercised ($p = 1$) or not ($p = 2$). The action set $A = \{1, 2\}$ represents the choice between exercising ($a = 1$) or not exercising ($a = 2$). The control α of the discrete component of the state space

$$(\alpha(p,a))_{p,a=1}^2 \sim \begin{bmatrix} \alpha(1,1) & \alpha(1,2) \\ \alpha(2,1) & \alpha(2,2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

ensures that $p = 1$ is absorbing. The continuous state space component follows

$$\tilde{Z}_{t+1} = \tilde{Z}_t + \gamma + \beta\epsilon_{t+1}, \quad \tilde{Z}_0 = \ln(S_0) \tag{74}$$

where $(\varepsilon)_{t=1}^T$ are independent standard normally distributed random variables. We set the parameters as $\gamma = -\frac{1}{2}\sigma^2\Delta$ and $\beta = \sigma\sqrt{\Delta}$ where $\Delta > 0$ is the time duration (in years) from time point t to $t+1$ and $\sigma > 0$ represents the volatility of the process that is measured on yearly scale. Given this price process, the disturbances are given as

$$W_{t+1} = \begin{bmatrix} 1 & \gamma + \beta\epsilon_{t+1} \\ 0 & 1 \end{bmatrix}, \quad t = 0, ..., T-1.$$

For the two-dimensional space $\mathbb{R}^2$ with the evolution of the continuous component as above, let us now determine all reward and the scrap functions. Consider a realization of the continuous component $(z^{(1)}, z^{(2)}) \in \mathbb{R}^2$ at the current time $t = 0, \ldots T-1$, then, given a Bermudian put option, the action $a \in \{1, 2\}$ leads to the reward

$$r_t(p, (z^{(1)}, z^{(2)}), a) = (Ke^{-\rho t} - e^{z^{(1)}})^+ \big(p - \alpha(p,a)\big) \tag{75}$$

for all $p \in P$ and $a \in A$. At final time T, we suppose that the put option is exercised automatically, which gives the scrap value

$$r_T(p, (z^{(1)}, z^{(2)})) = (Ke^{-\rho T} - e^{z^{(1)}})^+ \big(p - \alpha(p,1)\big) \tag{76}$$

for all $p \in P$. Note that the reward and scrap functions are not convex in the continuous component $z = (z^{(1)}, z^{(2)})$. Hence we decompose them into the difference of two convex functions

$$r_t(p, \cdot, a) = \breve{r}_t(p, \cdot, a) - \hat{r}_t(p, \cdot, a) \tag{77}$$

$$r_T(p, \cdot) = \breve{r}_T(p, \cdot) - \hat{r}_T(p, \cdot) \tag{78}$$

Table 1: Bermudan put option numerical results

S_0	σ	maturity	confidence interval	LSM mean	LSM se
36	0.2	1	[4.4763, 4.4768]	4.472	.0100
36	0.2	2	[4.8296, 4.8312]	4.821	.0120
36	0.4	1	[7.0989, 7.0992]	7.091	.0200
36	0.4	2	[8.4965, 8.4968]	8.488	.0240
38	0.2	1	[3.2481, 3.2489]	3.244	.0090
38	0.2	2	[3.7355, 3.7370]	3.735	.0110
38	0.4	1	[6.1451, 6.1452]	6.139	.0190
38	0.4	2	[7.6580, 7.6583]	7.669	.0220
40	0.2	1	[2.3119, 2.3129]	2.313	.0090
40	0.2	2	[2.8765, 2.8776]	2.879	.0100
40	0.4	1	[5.3093, 5.3094]	5.308	.0180
40	0.4	2	[6.9075, 6.9077]	6.921	.0220
42	0.2	1	[1.6150, 1.6158]	1.617	.0070
42	0.2	2	[2.2053, 2.2060]	2.206	.0100
42	0.4	1	[4.5797, 4.5798]	4.588	.0170
42	0.4	2	[6.2351, 6.2354]	6.243	.0210
44	0.2	1	[1.1081, 1.1087]	1.118	.0070
44	0.2	2	[1.6836, 1.6843]	1.675	.0090
44	0.4	1	[3.9449, 3.9450]	3.957	.0170
44	0.4	2	[5.6324, 5.6326]	5.622	.0210

These results were produced using a grid size of $m = 1024$ and disturbances of size $n = 4096$. Diagnostics is based on $K = 1024$ sample paths and 99% confidence bounds are calculated by setting $x = 0.01$ in (73). For comparison, the means and standard errors obtained by least squares Monte Carlo are given in the last two columns *LSM mean* and *LSM se* respectively, they are cited from [18], where numbers were quoted with three decimal points.

given by

$$\begin{aligned}
\breve{r}_t(p,(z^{(1)},z^{(2)}),a) &= (e^{z^{(1)}} - Ke^{-\rho t})^+\big(p - \alpha(p,a)\big)\\
\hat{r}_t(p,(z^{(1)},z^{(2)}),a) &= (e^{z^{(1)}} - Ke^{-\rho t})\big(p - \alpha(p,a)\big)\\
\breve{r}_T(p,(z^{(1)},z^{(2)})) &= (e^{z^{(1)}} - Ke^{-\rho T})^+\big(p - \alpha(p,1)\big)\\
\hat{r}_T(p,(z^{(1)},z^{(2)})) &= (e^{z^{(1)}} - Ke^{-\rho T})\big(p - \alpha(p,1)\big)
\end{aligned}$$

for all $p \in P, a \in A$ and $(z^{(1)}, z^{(2)}) \in \mathbb{R}^2$.

We compare our results with the low-biased estimates given in the literature for the Bermudan put where the risk-free rate is 0.06 and the strike is set at 40. The results are given in Table 9.1 for different combinations of initial prices, volatilities and maturities.

8.2 The swing option

We consider a specific case of the swing option, referred to as a *unit-time refraction period* condition. This condition limits the holder to exercise one right at a any time. Given the discounted asset price $(S_t)_{t=0}^T$, the price of a swing option with N rights is given by the supremum

$$\sup_{0 \le \tau_1 < \cdots < \tau_N \le T} \mathbb{E}\Big[\sum_{k=1}^{N} (S_{\tau_k} - Ke^{-\rho\tau_k})^+\Big]$$

over all stopping times $\tau_1, \ldots, \tau_N$ with values in $\{0, \ldots, T\}$. In order to represent this control problem as a switching system, we use the position set $P = \{1, \ldots, N+1\}$ to represent the number of rights remaining. That is $p \in P$ stands for the situation when there are $p-1$ rights remaining to be exercised. The action set $A = \{1, 2\}$ is the same as in the case of the Bermudan Put with control matrix α given by

$$(\alpha(p,a))_{p,a} \sim \begin{bmatrix} \alpha(1,1) & \alpha(1,2) \\ \alpha(2,1) & \alpha(2,2) \\ \cdots & \cdots \\ \alpha(N+1,1) & \alpha(N+1,2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ \cdots & \cdots \\ N & N+1 \end{bmatrix}.$$

Having modeled the discounted stock price process as an exponential mean-reverting process with reversion parameter $\kappa \in [0, 1[$, long run mean $\mu > 0$ and volatility $\sigma > 0$, we obtain the logarithm of the discounted price process as

$$\tilde{Z}_{t+1} = (1-\kappa)(\tilde{Z}_t - \mu) + \mu + \sigma\epsilon_{t+1}, \quad \tilde{Z}_0 = \ln(S_0). \tag{79}$$

with the disturbance matrix

$$W_{t+1} = \begin{bmatrix} (1-\kappa) & \kappa\mu + \sigma\epsilon_{t+1} \\ 0 & 1 \end{bmatrix}, \quad t = 0, ..., T-1.$$

The reward and scrap values are given by

$$r_t(p, (z^{(1)}, z^{(2)}), a) = (e^{z^{(1)}} - Ke^{-\rho t})^+ \big(p - \alpha(p,a)\big) \quad t = 0, \ldots, T-1 \tag{80}$$

and

$$r_T(p,(z^{(1)},z^{(2)})) = (e^{z^{(1)}} - Ke^{-\rho T})^+\big(p - \alpha(p,1)\big) \tag{81}$$

respectively for all $p \in P$ and $a \in A$.

In Table 9.2, we compare our results to those given in [19] with bounds on the swing option price where the underlying process is assumed to follow the dynamics (79) with parameters

$$\rho = 0, \quad \kappa = 0.9, \quad \mu = 0, \quad \sigma = 0.5, \quad S_0 = 1, \quad K = 0 \quad \text{and} \quad T = 1000.$$

Table 2: Swing option numerical results

	CSS	MH
Position (Rights + 1)	confidence interval	confidence interval
2	[4.737, 4.761]	[4.773, 4.794]
3	[9.005, 9.031]	[9.016, 9.091]
4	[13.001, 13.026]	[12.959, 13.100]
5	[16.805, 16.830]	[16.773, 16.906]
6	[20.465, 20.491]	[20.439, 20.580]
11	[37.339, 37.363]	[37.305, 37.540]
16	[52.694, 52.718]	[52.670, 53.009]
21	[67.070, 67.095]	[67.050, 67.525]
31	[93.811, 93.835]	[93.662, 94.519]
41	[118.639, 118.663]	[118.353, 119.625]
51	[142.059, 142.084]	[141.703, 143.360]
61	[164.368, 164.392]	[163.960, 166.037]
71	[185.757, 185.781]	[185.335, 187.729]
81	[206.362, 206.386]	[205.844, 208.702]
91	[226.284, 226.308]	[225.676, 228.985]
101	[245.601, 245.625]	[244.910, 248.651]

Results were produced using a grid size of $m = 1024$ and disturbances of size $n = 1024$. Diagnostics is based on $K = 1024$ sample paths and 99% confidence bounds are calculated by setting $x = 0.01$. The columns under *MH* denote the results from [19].

9 Conclusion

In this work we present a novel class of algorithms to solve stochastic switching problems whose processes follow linear state space dynamics. Our methodology is directly applicable to high-dimensional problems and shows remarkable numerical efficiency and excellent precision. More importantly, we adapt the primal-dual approach to estimate the distance to optimality of approximate solutions using Monte-Carlo techniques. With this, we establish a sound and reliable diagnostics and quality assessment tool for a posterior justification of the numerical approximation. The authors believe that such combination of efficient numerical schemes with a subsequent diagnostic check can be very useful in practical applications. This approach may help in development and justification of further approximate methods. In this context, natural extensions of the present scheme (say, from linear to piecewise linear dynamics) can be examined in detail. We address this promising direction in further research.

References

[1] L. Andersen and M. Broadie. A primal-dual simulation algorithm for pricing multidimensional American options. *Management Science*, 50:1222–1234, 2004.

[2] N. Bäuerle and U. Rieder. *Markov Decision Processes with Applications to Finance*. Springer, Heidelberg, 2011.

[3] A. Belomestny, N. Kolodko and J. Schoenmakers. Regression methods for stochastic control problems and their convergence analysis. *SIAM J. Control Optim.*, 48(5):3562–3588, 2010.

[4] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

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

[6] R. Carmona and N. Touzi. Optimal multiple stopping and valuation of swing options. *Mathematical Finance*, 18:239 – 268, 2008.

[7] J. F. Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19:19–30, 1996.

[8] E. Clement, D. Lamberton, and P. Protter. An analysis of the Longstaff-Schwartz algorithm for American option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

[9] D. Egloff. Monte Carlo algorithms for optimal stopping and statistical learning. *Appl. Probab.*, 15:1396–1432, 2005.

[10] D. Egloff, M. Kohler, and N. Todorovic. A dynamic look-ahead Monte Carlo algorithm. *Appl. Appl. Probab.*, 17:1138–1171, 2007.

[11] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall, 1996.

[12] E. A. Feinberg and A. Shwartz. *Handbook of Markov Decision Processes*. Kluwer Academic, 2002.

[13] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.

[14] M. Haugh and L. Kogan. Pricing American options: A duality approach. *Oper. Res.*, 52:258–270, 2004.

[15] J. Hinz. Optimal stochastic switching under convexity assumptions. *SIAM Journal on Control and Optimization*, 52(1):164–188, 2014.

[16] F. Jamshidian. Numeraire-invariant option pricing & American, Bermudan, and trigger stream rollover. July 2004. Version 1.6.

[17] M. S. Joshi and N. Yap. The multiplicative dual for multiple-exercise options. *http://papers.ssrn.com/sol3/papers.cfm?abstract_ id=2430558*, 2014.

[18] F. Longstaff and E. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, (14):113–147, 2001.

[19] N. Meinshausen and B. M. Hambly. Monte Carlo methods for the valuation of multiple-exercise options. *Mathematical Finance*, 14(4):557–583, 2004.

[20] D. Ormoneit and P. Glynn. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002.

[21] D. Ormoneit and P. Glynn. Kernel-based reinforcement learning in average-cost problems. *IEEE Transactions in Automatic Control*, 47:1624–1636, 2002.

[22] W. B. Powell. *Approximate dynamic programming: Solving the curses of dimensionality.* Wiley, 2007.

[23] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, 1994.

[24] L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12:271–286, 2002.

[25] L. C. G Rogers. Pathwise stochastic optimal control. *SIAM J. Control Optimisation*, 46:1116–1132, 2007.

[26] J. Schoenmakers. A pure martingale dual for multiple stopping. *Finance and Stochastics*, 16:319–334, 2012.

[27] L. Stentoft. Convergence of the least squares Monte Carlo approach to American option valuation. *Management Science*, 50(9):576–611, 2004.

[28] J. N. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 2001.

[29] 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.