Abstract

This paper considers the pricing of operational flexibility. By extending the recently develoed duality ideas for American option pricing we develop a dual representation for the problem and give algorithms for calculating upper bounds for the price of such optionality. We have also analysed the influence of the choice of basis functions to the results computed.

Contents

1 Introduction

A fundamental problem encountered in exotic energy derivatives is the pricing of operational flexibility. Power merchants and electricity trading firms increasingly have dedicated scheduling teams that control the operation of generating assets like power plants, storage facilities, etc. The controllers aim to maximize the total cashflow to the firm, subject to the engineering, regulatory and other market constraints.

Quantifying the value (hereafter called "price" for simplicity) of such optionality is therefore of great interest. Carmona and Ludkovski (2005) [2] use the idea of the regression Monte Carlo approach to compute the price via simulation techniques. But this approach only produces a lower bound for the value of a tolling agreement, since the price is computed by approximating the optimal scheduling rule.

Recently, a duality approach based on the work of Davis and Karatzas (1994) [3] was proposed independently by Haugh and Kogan (2001)[4] and Rogers (2002) [6] to compute an upper bound for American options. Rogers (2002) generates a good approximation to the optimal martingale process by constructing a portfolio to hedge the option. But as Rogers states in his paper:"There are few general rules so far; the selection of the martingales appears to be more art than science".

Meinshausen and Hambly (2004)[5] developed this idea to price the multiple-exercise case, specifically the swing option with a single exercise opportunity at every exercise time. Aleksandrov and Hambly (2008) [1] extended [5] to incorporate extra constraints, such as different volume restrictions each day and a total volume restriction of the contract.

Our work is to develop the duality ideas from [2], [1], [6] and [5] to find an upper bound on the tolling agreement's price. To reduce the complexity of this problem, we have simplified the model to one-dimensional (X_t), which can be explained as the spark spread. We give two numerical examples and write down all the difficulties we have run into.

The structure of the paper is as follows. We begin by introducing the financial engineering problem we study and our method to implement the duality approach in section 2. Section 3 gives a first numerical example where we try three different algorithms to compute the upper bound. Section 4 gives a second numerical example where operator's action will impact the price of market.

2 Problem Setup

2.1 General Situation

Financial players in the energy markets have increasing interest in owning energy assets, but the construction and maintenance of these assets is extremely capital intensive. In order to avoid this difficulty, the idea of a tolling agreement was invented. A tolling agreement temporarily transfers the scheduling flexibility of the asset in return for a fixed payment.

Let us consider a typical tolling agreement that gives control of a power plant for a period of T time epochs. We suppose that besides running the plant at full capacity, or turning it off completely, there also exist a total of $M_d - 2 (M_d \in \mathbb{N}, M_d \geq 2)$ intermediate operation modes. To each mode $m (0 \leq m \leq (M_d - 1), m \in \mathbb{N})$, we associate a payoff rate (in dollars or utility) at time $t (0 \leq t \leq T)$: $\psi_m(t)$. The payoff may be related to many factors such as the prices of fuel, gas and electricity, to establish a more abstract model, we denote by (X_t) all these factors supposed here to be a Markov process. We denote the corresponding production cashflow rate as $\psi_m(X_t)$.

Changing from one mode i to another mode j is costly, requiring extra fuel and various costs. We denote by $C_{i,j}(t, X_t)$ the switching costs from mode i to mode j with potential dependence on time t and current state X_t.

Assumption 2.1. *The switching costs are strictly positive with $C_{i,j} > \epsilon_C > 0$, for all $i \neq j$ and some $\epsilon_C > 0$. Also, $C_{i,i} = 0$, and $C_{i,j}$ satisfies the triangle inequality*

$$C_{i,j} \leq C_{i,k} + C_{k,j}, \quad \textit{for any } \{i, j, k\} \in \{0, 1, ..., M_d\}^3.$$

Let $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t^X), \mathbb{P})$ be a stochastic basis. For our model, we take a d-dimensional $(X_t) = (X_t^1, ..., X_t^d)$ satisfying:

$$dX_t = \mu(x_t)dt + \sigma(x_t) \cdot dW_t, \tag{1}$$

where $\sigma(X_t)$ is a $d \times e$ dimensional matrix, $\{W_t\}$ is a standard e-dimensional Wiener process on $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$. The filtration $\mathbb{F}$ satisfies the usual conditions, with $\mathcal{F}_0$ being trivial. μ, σ can be time dependent; we omit t for convenience.

We model the ability to schedule the start-up and shut-down order as a control process $u = (u_t)$. The control u is dynamically chosen and adapted to the information filtration $\mathcal{F}_t^X \triangleq \sigma\left(X_s : 0 \leq s \leq t\right)$. u can be represented by:

$$u = \left(\left(\xi_1, \tau_1\right), \left(\xi_2, \tau_2\right), ...\right), \quad \text{with} \quad \xi_l \in \mathbb{Z}_{M_d} \triangleq \{0, ..., M_d - 1\}, \quad \text{and} \quad 0 \leq \tau_{l-1} < \tau_l < T,$$

and u_t can be written as: $u_t = \sum_{\tau_l < T} \xi_l \mathbb{1}_{[\tau_l, \tau_{l+1})}(t)$. The total gain from t to T for a control u and scenario $\omega \in \Omega$ is:

$$H([t,T], x; u)(\omega) \triangleq \int_t^T \psi_{u_s(\omega)}(s, X_s(\omega))ds - \sum_{t \leq \tau_l < T} C_{u_{\tau_l^-}(\omega), u_{\tau_l}(\omega)}, \quad X_t = x.$$

Let $\mathcal{U}(t)$ be the set of all allowed u on the interval $[t, T]$. The operational flexibility problem may now be stated as calculating the value function:

$$J(t, x, i) = \sup_{u \in \mathcal{U}(t)} \mathbb{E}[H([t,T], x; u) | u_t = i], \tag{2}$$

where $\mathbb{E}$ denotes expectation with respect to a risk-neutral pricing measure $\mathbb{P}$.

To solve the problem 2, we introduce

$$\mathcal{U}^k(t) \triangleq \{u \in \mathcal{U}(t) : \tau_{k+1} = T\},$$

the set of all admissible strategies on $[t, T]$ with at most k switches, and

$$J^k(t, x, i) = \sup_{u \in \mathcal{U}^k(t)} J(t, x, i; u).$$

For convenience, we note J_i^k for $J^k(t = 0, X_0, i)$.

With the notation

$$\mathscr{S}_T^p \triangleq \{Z : \mathcal{F}_t^X - \text{adapted}, \mathbb{E}[\sup_{t \in [0,T]} |Z_t|^p] < \infty\}, \quad p \geq 1,$$

we make the following standing assumption:

Assumption 2.2. *For each $m = 0, 1, ..., M-1$, the reward function $\psi_m : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ is locally Lipschitz in (t,x) and $\psi_m(\cdot, X) \in \mathscr{S}^2_T$ uniformly in $X_0 = x$ restricted to bounded set.*

Theorem 2.3. *We can iteratively define $J^k(t,x,i)$ for $t \in [0,T]$, $i \in \mathbb{Z}_{M_d}$, by*

$$J^0(t,x,i) \triangleq \mathbb{E}[\int_t^T \psi_i(s, X_s^{t,x})ds], \tag{3}$$

$$J^k(t,x,i) \triangleq \sup_{\tau \in \mathcal{S}_t} \mathbb{E}[\int_t^\tau \psi_i(s, X_s^{t,x})ds + \mathcal{R}J^{k-1}(\tau, X_\tau^{t,x}, i)], \quad k \geq 1, \tag{4}$$

where $S_t \triangleq \{\tau \leq T : \mathcal{F} - \text{stopping time s.t. } t \leq \tau \quad a.s.\}$ is the set of all stopping times after t and the function operator $\mathcal{R}$ is defined by

$$\mathcal{R}\omega(t,x,i) \triangleq \max_{j \neq i}\{-C_{i,j} + \omega(t,x,j)\}, \quad i,j \in \mathbb{Z}_{M_d}.$$

Proof. See the paper of Carmona and Ludkovski [2]. □

Theorem 2.4.

$$\lim_{k \to \infty} J^k = J \quad \textit{pointwise.}$$

Proof. See the paper of Carmona and Ludkovski [2]. □

2.2 Simplified Situation

We now let $M_d = 2$ to simplify our problem.

Definition 2.5. *Inspired from (4), we can define the payoff at time t $(0 \leq t \leq T)$*

$$h_t^k(\overline{X}_t, i) = \begin{cases} \int_0^t \psi_i(X_s)\,ds - C_{i,j} + J^{k-1}(t, X_t, j), & i,j \in \{0,1\} \quad \textit{and} \quad i \neq j, \quad t \in [0,T), \\ \int_0^T \psi_i(X_s)\,ds \quad i \in \{0,1\} \quad t = T, \end{cases} \tag{5}$$

where $\overline{X}_t \triangleq \{X_s, 0 \leq s \leq t\}$.

Thus, we can regard the tolling agreement as an American option.

Definition 2.6. *The value of this "option" at time t is denoted by*

$$V_t^k(\overline{X}_t, i) = \sup_{\tau \geq t} \mathbb{E}_t\left(h_\tau^k(\overline{X}_\tau, i)\right), \tag{6}$$

where τ is a stopping time satisfying $0 \leq \tau \leq T$ with respect to $\mathcal{F}(X_t)$.

In order to do numerical simulation, we first transfer the continuous time to discrete time. Let $\mathcal{S}^\Delta = \{m\Delta t, m = 0, 1, ..., M\}$ be a discrete time grid with $\Delta t = T/M$. Mode switches are only allowed at grid points, i.e. $\tau_l \in \mathcal{S}^\Delta$.

Theorem 2.7. *(**Dual Representation**) The price of the tolling agreement with at most k switches is equal to*

$$V_0^k = \inf_{\mathcal{M} \in \mathcal{H}_0^1} \mathbb{E}[\sup_{0 \leq t \leq T}(h_t^k - \mathcal{M}_t)],$$

where $\mathcal{H}_0^1$ is the set of integrable martingales which are null at 0.

Moreover, if switches are only allowed at grid points, the the minimum is attained for the martingale $\mathcal{M}^*$ with $\mathcal{M}^*_{t=0} = 0$ and

$$\mathcal{M}^*_{m\Delta t} - \mathcal{M}^*_{(m-1)\Delta t} = V^k_{m\Delta t} - \mathbb{E}_{(m-1)\Delta t}[V^k_{m\Delta t}],$$

where $m \in \{1, 2, ..., M\}$.

Proof. See the papers of Rogers [6] and Aleksandrov and Hambly [1]. $\square$

In the following two numercial examples, we will introduce methods to compute upper bound for the price of tolling agreement, using Theorem 2.7.

3 First Numerical Example

Here we use the notion and the algorithms adopted by Carmona and Ludkovski [2]. According to (1), we use a simple one dimensional (X_t) and

$$\begin{aligned} dX_t &= 2\,(10 - X_t)\,dt + 2dW_t, \\ X_0 &= 10, \end{aligned} \tag{7}$$

with time horizon $T = 2$. We have two regimes with continuous reward rates of

$$\begin{aligned} \psi_0\,(X_t) &= 0, \\ \psi_1\,(X_t) &= 10\,(X_t - 10)\,, \end{aligned} \tag{8}$$

and the switching costs between them are: $C_{1,0} = C_{0,1} = 0.3$.

We note:

- M: number of discrete time epochs. Here we take $M = 400$.
- $\Delta t \triangleq \frac{T}{M}$.
- N_p: the number of paths to calculate the lower bound.
- k: total number of switching opportunities.
- $x^n_{m\Delta t}, m = 0, 1, ..., M, n = 1, 2, ..., N_p$: N_p paths of driving process with fixed initial condition $x^n_0 = X_0$.
- $B_j(X_t)$: j-th basis function for Regression Monte Carlo approach.
- $\alpha^{t,k,i}_j$: coefficient of basis function $B_j(X_t)$ at time t with k switching opportunities left and with initial regime mode $i \in \{0, 1\}$.
- N_N: total number of separate runs.
- σ: the standard deviation of all results after running the algorithm N_N times.

3.1 Lower Bound

We use the algorithm introduced in [2] to calculate the lower bound. The standard deviation is obtained by separately running the program N_N times.

3.1.1 Analysis of Basis Functions

First, our aim is to choose a suitable set of basis functions with which the lower bound computed is reasonably approximating the true value. Basis functions we will take are

$$\begin{aligned}
B_1 &= 1, \\
B_2 &= X_t, \\
B_3 &= \log\left(X_t\right), \\
B_4 &= X_t^2, \\
B_5 &= X_t \log\left(X_t\right), \\
B_6 &= X_t^2 \log\left(X_t\right), \\
B_7 &= \max(X_t - 10, 0), \\
B_8 &= \exp(X_t - 10), \\
B_9 &= X_t^3.
\end{aligned}$$

After the computation of the lower bound, we also draw the curves of coefficients against time and analyze their smoothness, however, the smoothness of these coefficients is hard to define. For example, in the left panel of Figure 2, the dashed line seems smooth when time is larger than $100\Delta t$. But when we "zoom in" it turns out to be rough. Another reason is that the orders of values of these basis functions are not the same; $B_2 = X_t \sim 10$, $B_4 = X_t^2 \sim 100$ and $B_9 = X_t^3 \sim 1000$ for instance. So the smoothness we define here is more subjective than objective. In this paper, our main method is to compare the performance of the same basis function between different sets of basis functions.

With $N_p = 16000, N_N = 40$, total switching opportunities $k = 10$, we write results in the following table using notions S (stable), U (unstable) and SU (severely unstable) to mark the smoothness of coefficients. Note that the true value obtained by using a pde solver is 5.93 [2] . The value in the parentheses is the standard deviation. We also define by $\alpha_j^{t,k,i}$ the coefficient of B_j and the initial regime is mode i. In our paper, we use $\alpha_j^{t,k}$ instead of $\alpha_j^{t,k,1}$ for convenience.

Basis functions	Smoothness	Lower bound(σ)	Figure
B_1, B_2	S	5.538(0.053)	
B_1, B_2, B_3	S	5.566 (0.055)	
B_1, B_2, B_4	S	5.538 (0.053)	
B_1, B_2, B_5	S	5.539 (0.053)	
B_1, B_2, B_7	S	5.554 (0.057)	
B_1, B_2, B_8	S	5.584 (0.052)	
B_5, B_7, B_8	S	5.317 (0.046)	
B_1, B_2, B_3, B_4	U	5.725 (0.055)	Fig 1
B_1, B_2, B_3, B_5	U $-$	5.720(0.055)	
B_1, B_2, B_5, B_7	S $-$	5.553(0.053)	
B_1, B_2, B_5, B_8	S	5.687(0.054)	
B_1, B_2, B_4, B_8	S	5.687(0.057)	
B_1, B_5, B_7, B_8	S $-$	5.634(0.057)	
B_1, B_2, B_5, B_7, B_8	U $+$	5.694(0.052)	
B_1, B_2, B_4, B_5, B_8	U	5.726(0.053)	
B_1, B_2, B_4, B_7, B_9	S $-$	5.772(0.058)	Fig 2
B_1, B_2, B_4, B_8, B_9	S $--$	5.777(0.057)	
$B_1, B_2, B_3, B_4, B_5, B_6$	SU	5.860(0.057)	Fig 3
$B_1, B_2, B_3, B_4, B_5, B_8$	SU	5.865(0.051)	
$B_1, B_2, B_4, B_5, B_7, B_8$	SU	5.778(0.055)	
$B_1, B_2, B_4, B_5, B_8, B_9$	SU	5.878(0.057)	

Table 1: The smoothness of coefficients against time and the lower bound calculated with different sets of basis functions. $N_p = 16000, N_N = 40, k = 10$. Notions: S (stable), U (unstable) and SU (severely unstable).

Figure 1 shows that $\alpha_1^{t,k}$ and $\alpha_3^{t,k}$ are not stable when we use $\{B_1, B_2, B_3, B_4\}$ as basis functions. In addition, at the beginning ($0 \leq m \leq 50$), coefficients vary sharply. This is a common phenomenon for almost all sets of basis functions. The reason is that when t is small ($0 < t < 0.25$), X_t is close to $X_0 = 10$ for nearly all paths, causing higher variability of regression. When we use $\{B_1, B_2, B_4, B_7, B_9\}$ as the basis functions, we can see from Figure 2 that except the $\alpha_1^{t,k}$, other coefficients seems relatively stable against time. Note that $\alpha_7^{t,k}$ stays most of the time in $[1, 2]$; we will explain it in the following part of this subsection. Overall, $\{B_1, B_2, B_4, B_5, B_6\}$ is the set of basis functions having the best results for the value function, but $\alpha_1^{t,k}, \alpha_2^{t,k}, \alpha_3^{t,k}$ and $\alpha_5^{t,k}$ are very large (being on the order of 10^4) and change fast, and as the Figure 3 shows, the coefficients are severely unstable.

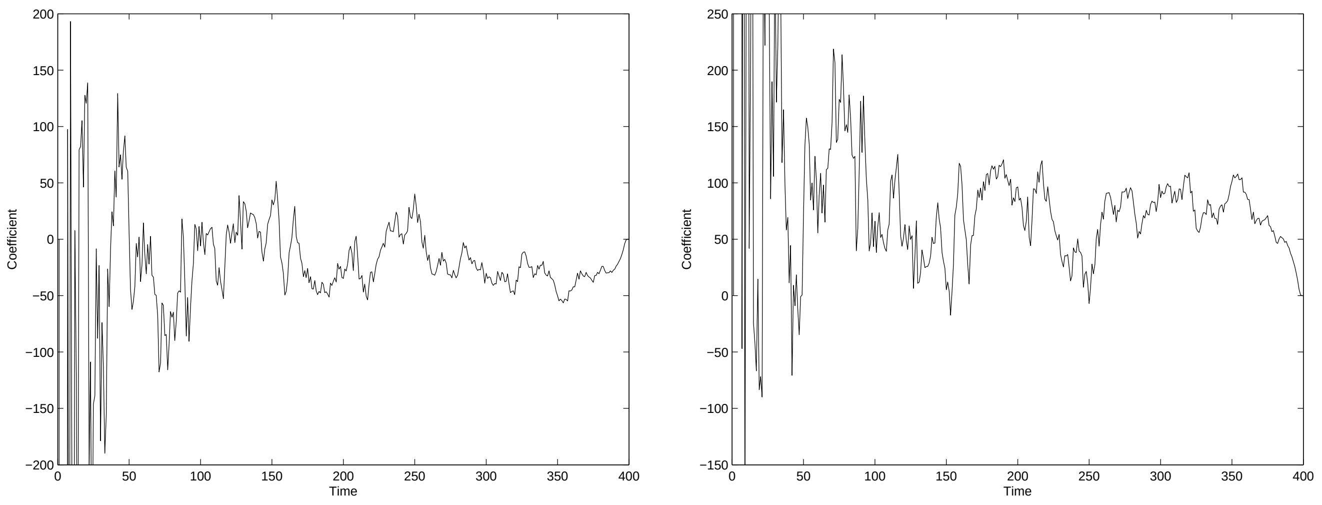

Figure 1: The graph of regression coefficients against time, using the set of basis functions $\{B_1, B_2, B_3, B_4\}$ with $N_p = 16000, k = 10$. Left is the graph of $\alpha_1^{t,k}$ (coefficient of B_1); right $\alpha_3^{t,k}$.

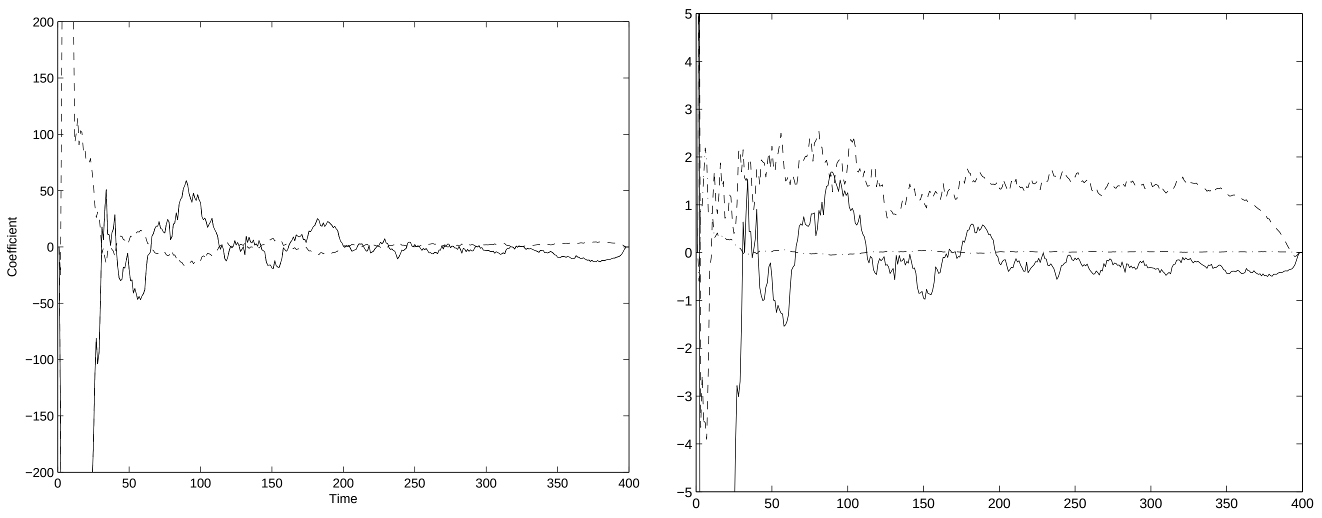

Figure 2: The graph of coefficients, using $\{B_1, B_2, B_4, B_7, B_9\}$ as the basis functions, with $N_p = 16000, k = 10$. Left is the graph of $\alpha_1^{t,k}$ (solid line) and $\alpha_2^{t,k}$(dashed line); Right $\alpha_4^{t,k}$ (solid line), $\alpha_7^{t,k}$ (dashed line) and $\alpha_9^{t,k}$ (dash-dot line).

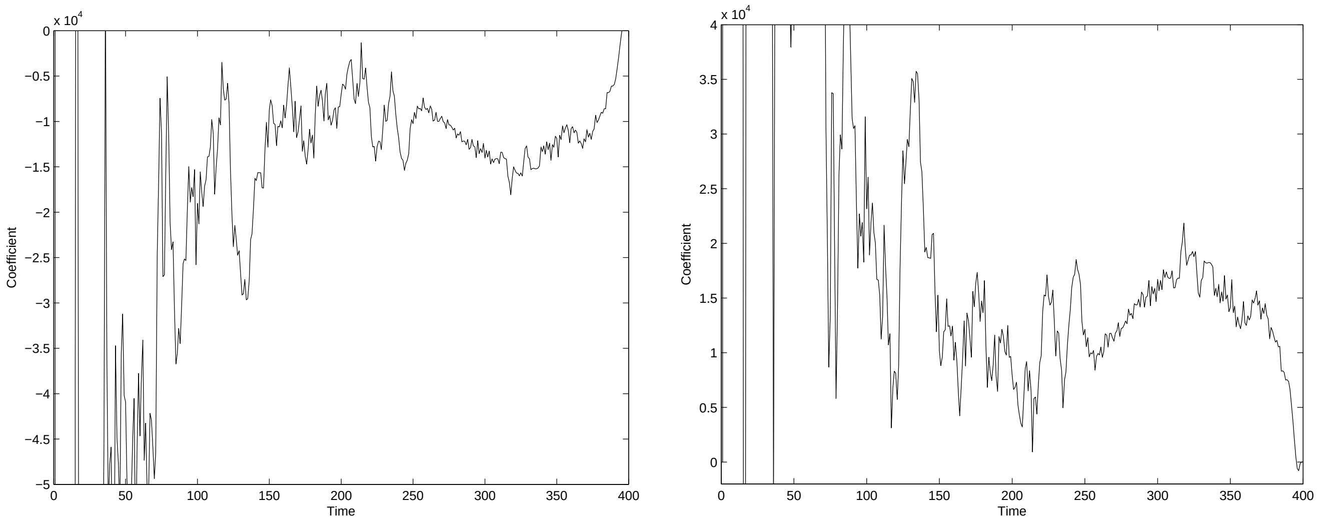

Figure 3: The graph of coefficients, $\alpha_1^{t,k}$ (left), $\alpha_2^{t,k}$ (right) with $N_p = 16000, k = 10$, basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$.

We also analyze the "remaining value" of this tolling agreement at time t, $J^{10}(t, X_t, 1)$, as a function of X_t (from 5 to 15) at time t ($0 \leq t \leq T$), which is shown in Figure 4. It was computed by simulating the lower bound of $J^{10}(t, X_t, 1)$ at time t, with different initial $X_t \in \{5, 5.1, 5.2, ..., 15\}$. The computation is similar to that of the lower bound of $J^{10}(0, X_0, 1)$ with the horizon of $T^* = T - t$. We can see that $J^{10}(t, X_t, 1)$ is increasing and convex. The curves are more or less similar to the curve $max(X_t - 10, 0)$ with the slope $k_s \in [3, 5]$ when $X_t > 10$. This partly explains the curve of $\alpha_7^{t,k}$ in Figure 2.

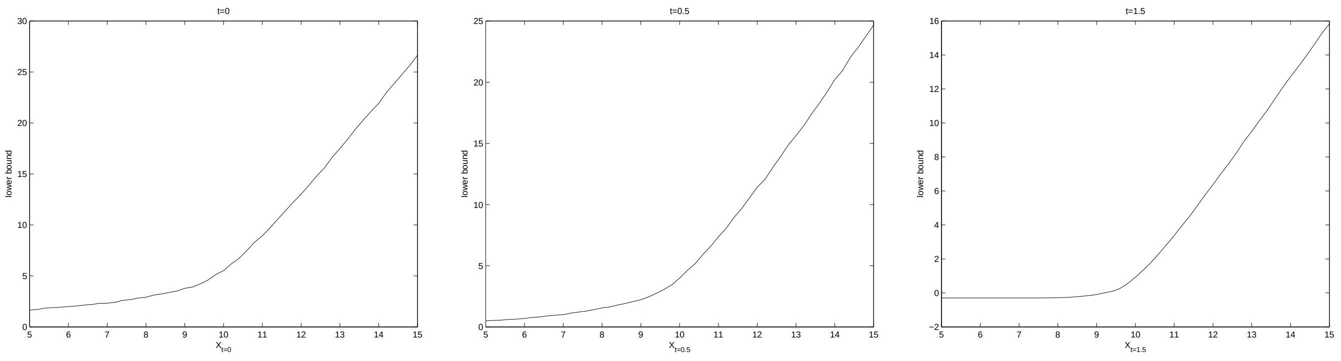

Figure 4: The graph of $J^{10}(t, X_t, 1)$ at time t as function of X_t with $t = 0, t = 0.5$ and $t = 1.5$ (T=2), using $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ as basis functions, with $N_p = 16000$, $k = 10$.

3.1.2 Switching Boundary

The large number of basis functions might also cause the complicated behavior of $\hat{J}$, and finally cause multi-solutions in calculation of the switching boundary.

We calculate switching boundaries with k switching opportunities at time t by:

$$\begin{cases} 10(X_t - 10) \cdot \frac{T}{M} + \sum_i \alpha_i^{t,k-1,1} B_i(X_t) - C_{0,1} = \sum_i \alpha_i^{t,k,0} B_i(X_t), & \text{boundary for } u_{t-1} = 0 \text{ to } u_t = 1; \\ 10(X_t - 10) \cdot \frac{T}{M} + \sum_i \alpha_i^{t,k,1} B_i(X_t) = \sum_i \alpha_i^{t,k-1,0} B_i(X_t) - C_{1,0}, & \text{boundary for } u_{t-1} = 1 \text{ to } u_t = 0. \end{cases} \quad (9)$$

Figure 5 illustrates the results. The left panel is the boundary calculated normally, i.e. we find the solution of (9) nearest to $X_t = 10$. But other solutions often exist as well, as shown in the right panel of Figure 5. The existence of vertical lines means that at some time points, we can only find 1 or 2 solutions (not 3 solutions). From common sense, we know that the switching boundary from mode 1 to mode 0 is definitely below 10 and there will be only one such boundary.

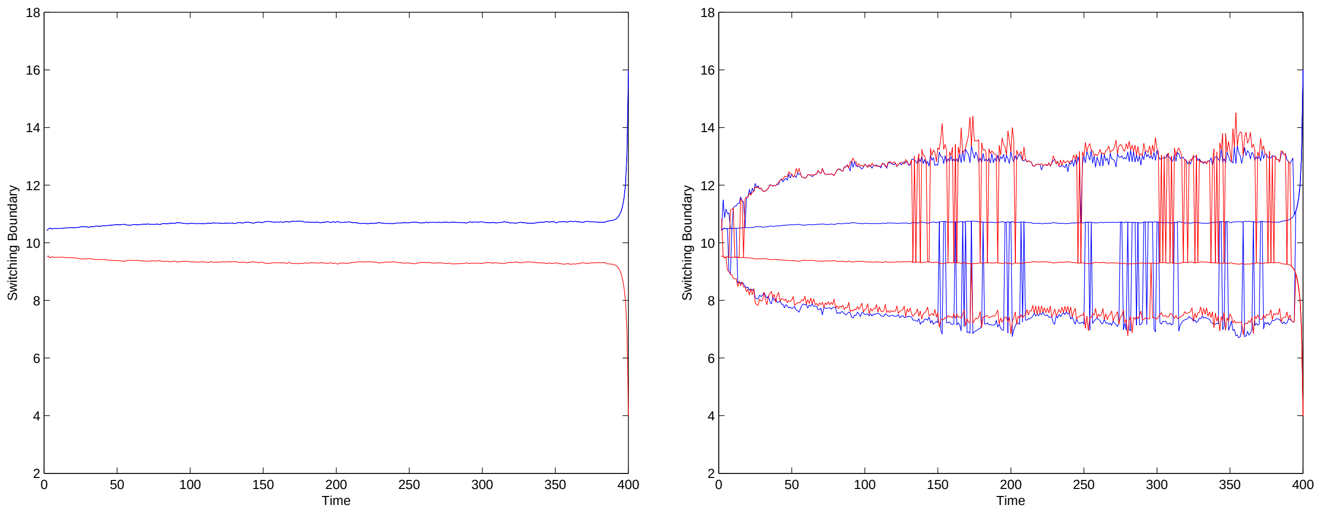

Figure 5: The graph of the switching boundary at different times, using $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ with $N_p = 16000, k = 10$. Left panel: the boundary solution nearest to 10, upper curve (in blue) is the boundary for mode 0 to mode 1, lower curve (in red) is the boundary for mode 1 to mode 0. Right panel: we can find three solutions of equation of (9) at some time points and vertical lines link the same type of boundary (there are two types of boundary: mode 0 to 1 (in blue) and mode 1 to 0 (in red)).

As we can see from Figure 6, the problem resides in the regression. In the right panel of Figure 6, $J^{10}(t = 0.5, X_{t=0.5}, 0)$ and $J^{10}(t = 0.5, X_{t=0.5}, 1)$ are calculated by computing the lower bound at time $t = 0.5$ with different X_t for $t = 0.5$. The basis functions are $\{B_1, B_2, B_3, B_4, B_5, B_6\}$. As we can see from Table 3, the lower bound and the true value are very close, so we expect that these two curves are very close to the real ones. In the left panel, $\hat{L}^{10}(t = 0.5, X_{t=0.5}, 0)$ and $\hat{L}^{10}(t = 0.5, X_{t=0.5}, 1)$ are calculated using coefficients and basis functions at time t, i.e.

$$\begin{aligned}\hat{L}^{10}(t = 0.5, X_{t=0.5}, 1) &= \sum_i \alpha_i^{t,k,1} B_i(X_t) + 10(X_t - 10)T/M, \\ \hat{L}^{10}(t = 0.5, X_{t=0.5}, 0) &= \sum_i \alpha_i^{t,k,0} B_i(X_t).\end{aligned} \tag{10}$$

Comparing (9) and (10), together with the left panel of Figure 6, we see that the deviation in the regression leads to the multiple solutions for the switching boundary.

$\hat{J}^{10}(t = 0.5, X_{t=0.5}, 0)$ and $\hat{J}^{10}(t = 0.5, X_{t=0.5}, 1)$ are calculated by

$$\begin{aligned}\hat{J}^{10}(t = 0.5, X_{t=0.5}, 0) &= \max\{\hat{L}^{10}(t = 0.5, X_{t=0.5}, 1) - C_{0,1}, \hat{L}^{10}(t = 0.5, X_{t=0.5}, 0)\}, \\ \hat{J}^{10}(t = 0.5, X_{t=0.5}, 1) &= \max\{\hat{L}^{10}(t = 0.5, X_{t=0.5}, 1), \hat{L}^{10}(t = 0.5, X_{t=0.5}, 0) - C_{0,1}\}.\end{aligned} \tag{11}$$

As we can see from the middle panel and the right panel of Figure 6, we can conclude that $\hat{J}^{10}(t = 0.5, X_{t=0.5}, 0)$ and $\hat{J}^{10}(t = 0.5, X_{t=0.5}, 1)$ are actually above the true value when X_t is far from 10.

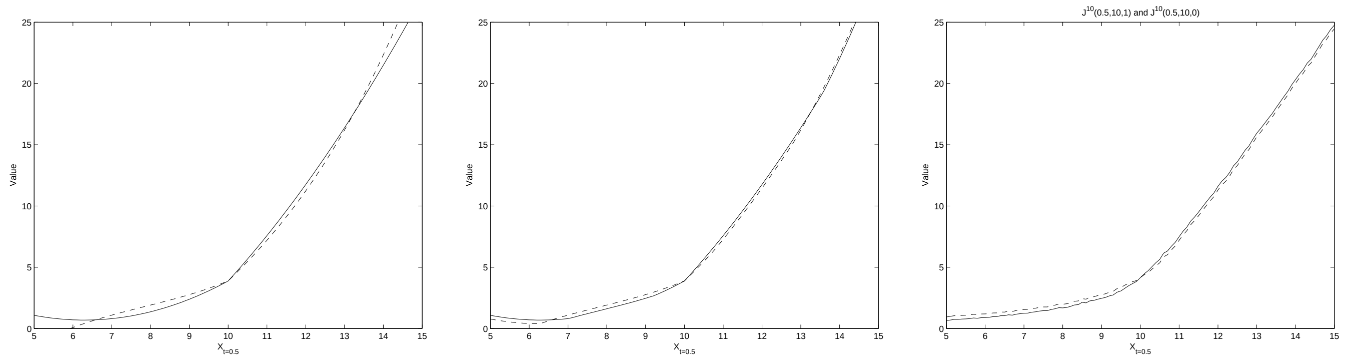

Figure 6: Left panel: the graph of $\hat{L}^{10}(t = 0.5, X_{t=0.5}, 0)$ (dashed line), $\hat{L}^{10}(t = 0.5, X_{t=0.5}, 1)$ (solid line) with basis functions $\{B_1, B_2, B_4, B_7, B_9\}$ and $N_p = 16000$; middle panel: $\hat{J}^{10}(t = 0.5, X_{t=0.5}, 0)$ (dashed line), $\hat{J}^{10}(t = 0.5, X_{t=0.5}, 1)$ (solid line) with basis functions $\{B_1, B_2, B_4, B_7, B_9\}$ and $N_p = 16000$; right panel: $J^{10}(t = 0.5, X_{t=0.5}, 0)$ (dashed line), $J^{10}(t = 0.5, X_{t=0.5}, 1)$ (solid line). $J^{10}(t = 0.5, X_{t=0.5}, 0)$ and $J^{10}(t = 0.5, X_{t=0.5}, 1)$ are calculated by computing the lower bound at time $t = 0.5$ with different $X_{t=0.5}$ with basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and $N_p = 16000$.

.

We have tried different sets of basis functions to test if they have more than one switching boundary. Table 2 tells us that most sets of basis functions have more than one boundary. In addition, if the number of basis functions is more than 5, it was hard to avoid having multi-solutions.

Basis function	Double solution? (Percentage)
B_1, B_2, B_5	No (0%)
B_1, B_2, B_8	Yes ($\sim$ 10%)
B_1, B_2, B_4, B_8	Yes ($\sim$ 13%)
B_1, B_2, B_5, B_8	Yes
B_1, B_2, B_4, B_8, B_9	Yes
$B_1, B_2, B_3, B_4, B_5, B_6$	Yes
$B_1, B_2, B_4, B_5, B_8, B_9$	Yes

Table 2: The occurrence of double boundary with different sets of basis functions with $N_p = 16000, k = 10$. The percentage in the column "Double solution" indicates the proportion of time when there are more than one boundary solution. If the percentage is not indicated, it means that the proportion is larger than 50%.

3.1.3 Influence of N_p

Here we choose two sets of basis functions to simulate the lower bound. The results with different N_p are showed in Table 3. Comparing this table and Table 1, we can conclude that N_p, the number of paths to simulate, only has influence on the standard deviation when N_p is larger than 4000, while the choice of basis function have a bigger influence on the value of the lower bound computed.

low bound(σ)	$N_p = 4000$	$N_p = 8000$	$N_p = 16000$	$N_p = 32000$
B_1, B_2, B_4, B_7, B_9	5.775(0.116)	5.743(0.073)	5.744(0.065)	5.754(0.035)
$B_1, B_2, B_3, B_4, B_5, B_6$	5.884(0.104)	5.898(0.063)	5.860(0.057)	5.873(0.037)

Table 3: The lower bound computed with different N_p and different sets of basis functions. $k = 10, N_N = 40$.

As seen from our extensive discussion, it is not possible to find a perfect set of basis functions. From now on, we will limit our choice of basis functions to two sets: $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and $\{B_1, B_2, B_4, B_7, B_9\}$ — first one with stable coefficients, and second one with better simulated lower bounds. To circumvent the problem of multiple boundaries, in the lower bound algorithm, we will forbid the transfer from mode 1 to mode 0 if $X_t > 10$, and from mode 0 to mode 1 if $X_t < 10$.

3.2 Quasi-upper Bound

Consider the payoff:

$$\hat{h}^k_t(\overline{X}_t, i) := \int_0^t \psi_i\left(X_s\right) ds - C_{i,j} + \hat{J}^{k-1}\left(t, X_t, j\right). \tag{12}$$

We regard this tolling agreement as an American option. The value of this "option" at time t is denoted by $\hat{V}^k_t(\overline{X}_t, t, i)$ (see (6)).

Remark 3.1. *This algorithm has a flaw: $\hat{h}^k(\overline{X}_t, i)$ is not an strict upper estimate. That is why we call this subsection "quasi-upper bound", since the simulated value cannot be guaranteed to be an upper bound on the value function.*

3.2.1 First Algorithm

In this algorithm, we see $\hat{h}^k_t(\overline{X}_t, i)$ as the payoff of an American option and use the algorithm introduced in [1]. Notice that in the paper of Aleksandrov [1], the payoff $h(X_t)$ only depends on X_t while in our example it depends on $\overline{X}_t = (X_s, s \in [0,t])$. Since the regression of $h(\overline{X}_t)$ over basis functions of X_t creates much deviation from the true value, we introduce a basis function depending on $(X_s, s \in [0,t])$ to avoid this problem. According to the expression of $\hat{h}^k(\overline{X}_t, i)$ in (12), we add the following basis function

$$B_a\left(\overline{X}_t\right) = \int_0^t \psi_1(X_s) ds.$$

First, we still use $\{B_1, B_2, ..., B_6\}$ to calculate $\hat{h}^k_t(\overline{X}_t, i)$, then we use $\{B_1, B_2, ..., B_6, B_a\}$ to calculate (13), (14) and the upper bound:

- We calculate the coefficients of marginal continuation value, using the regression and seven basis functions. With the algorithm of Aleksandrov [1] we have:

  $$\hat{E}_t\left(\hat{V}^k_{t+1}(\overline{X}_{t+1}, t+1, i)\right) = \sum_p \beta^{t,k,i}_p B_p(X_t), \tag{13}$$

  where i is the regime at time t, $p \in \{1, 2, 3, 4, 5, 6, a\}$ and k the the number of switching opportunities remain. Note that here $\beta^{t,k,i}_p$ is not the same as $\alpha^{t,k,i}_p$, since $V^k_t(\overline{X}_t, t, i) \neq J^k(X_t, t, i)$.

- We calculate the martingale $\mathcal{M}$ using again the algorithm of Aleksandrov [1] with

  $$\mathcal{M}^k_{m\Delta t} = \sum_{p=1}^m \left(\hat{V}^k_{p\Delta t} - \hat{E}_{(p-1)\Delta t}\left(\hat{V}^k_{p\Delta t}\right)\right), \tag{14}$$

  where $m \in \{1, 2, ..., M\}$.

- We use this $(\mathcal{M}_t^k)$ instead of the optimal dual martingale to obtain an upper bound

$$^{\uparrow}\hat{V}_t^k\left(X_t,t,i\right)=\mathbb{E}[\sup_{u\in\mathcal{T}}\left(\hat{h}_u^k-\mathcal{M}_u^k\right)],$$

where $\mathcal{T}=\{m\Delta t, m\in\{0,1,...,M\}\}$.

We note:

- $n1$: the number of paths to calculate (13), using the algorithm of Alekesandrov [1].
- N : the number of paths for the martingale $(\mathcal{M}_t^k)$ approximation (14).
- N_K : the number of inner paths for the martingale $(\mathcal{M}_t^k)$ approximation (14).

With $N_p=32000, n1=10000, N=100, N_K=50$ and $k=10$, we have an upper bound 6.473 and the standard deviation over 50 separate runs is 0.114. These results are not good, being 10% higher than the lower bound. The cause might be that we have regressed two times and it was over regressed.

3.2.2 Second Algorithm

Notice that we can skip (13) and directly work with equation (14). Since both $\hat{V}_t^k$ and $\mathbb{E}_{t-1}[\hat{V}_t^k]$ contain the term $\int_0^{t-1}\psi_i\left(X_s\right)ds$, we can re-use the results obtained in the computation of the lower bound:

$$\begin{aligned}\mathcal{M}_{m\Delta t}^k &= \sum_{p=1}^m\left(\hat{V}_{p\Delta t}^k-\hat{E}_{(p-1)\Delta t}\left(\hat{V}_{p\Delta t}^k\right)\right) \qquad (15)\\ &= \sum_{p=1}^m\left(\hat{J}^k(p\Delta t,X_{p\Delta t},i)-\hat{\mathbb{E}}_{(p-1)\Delta t}(\hat{J}^k(p\Delta t,X_{p\Delta t},i))\right),\end{aligned}$$

where $m\in\{1,2,...,M\}$.

The results for the quasi-upper bound with initial mode 1 are shown in Table 4.

Basis functions	N_p	k	1	2	9	10
A_1	8000	$\hat{J}_1^k$	3.739(0.070)	5.073(0.073)	5.868(0.066)	5.868(0.066)
		V_1^k	4.163(0.062)	5.364(0.053)	6.155(0.058)	6.155(0.058)
	16000	$\hat{J}_1^k$	3.742(0.057)	5.087(0.059)	5.869(0.055)	5.869(0.055)
		V_1^k	4.158(0.060)	5.336(0.050)	6.104(0.049)	6.104(0.049)
	32000	$\hat{J}_1^k$	3.744(0.036)	5.079(0.035)	5.863(0.030)	5.863(0.030)
		V_1^k	4.150(0.056)	5.301(0.032)	6.063(0.028)	6.063(0.028)
A_2	32000	$\hat{J}_1^k$	3.717(0.044)	5.010(0.043)	5.747(0.038)	5.747(0.038)
		V_1^k	4.159(0.049)	5.318(0.039)	6.041(0.032)	6.041(0.032)
A_1	32000	$\hat{J}_0^k$	3.736(0.036)	5.079(0.030)	5.862(0.029)	5.862(0.029)
		V_0^k	4.190(0.073)	5.3114(0.041)	5.996(0.034)	5.996(0.034)

Table 4: Simulation of the upper bound with basis functions $A_1\triangleq\{B_1,B_2,B_3,B_4,B_5,B_6\}$ and $A_2\triangleq\{B_1,B_2,B_4,B_7,B_9\}$ with $N_N=40$, $N_K=100$. Here V_1^k (V_0^k) is the upper bound with k switching opportunities and initial mode 1 (resp., 0); $\hat{J}_1^k$ $(\hat{J}_0^k)$ is the lower bound with k switching opportunities and initial mode 1 (0).

The set of basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ has better results than $\{B_1, B_2, B_4, B_7, B_9\}$ considering the smaller difference between its lower bound and upper bound. Increasing N_N will also decrease the difference between the upper and lower bounds.

The results are not so good, so we trace the graph of $\{X_t | t = \arg\left(\sup_{u\in\mathcal{T}}\left(\hat{h}_u^k - \mathcal{M}_u\right)\right)\}$ to investigate the problem. Since $N = 1000$, we have 1000 points. These points indicate the time t and the value X_t when we make the first transfer of regime. Therefore, they should be near the switching boundary from mode 1 to mode 0. As we can see from the left panel in Figure 7, first switch is made mainly before $t = 100\Delta t$ and most points are less than 10, however, some points are anomalous, being much larger than 10. This arises due to the multi-solutions of the switching boundary as we can see that the anomalous points are near the second switching boundary from mode 1 to mode 0. There are also some points at the end of period, suggesting that we have not made any switches. These abnormal points and those that are far from the true switching boundary are the principal cause of the bad estimation of the quasi-upper bound.

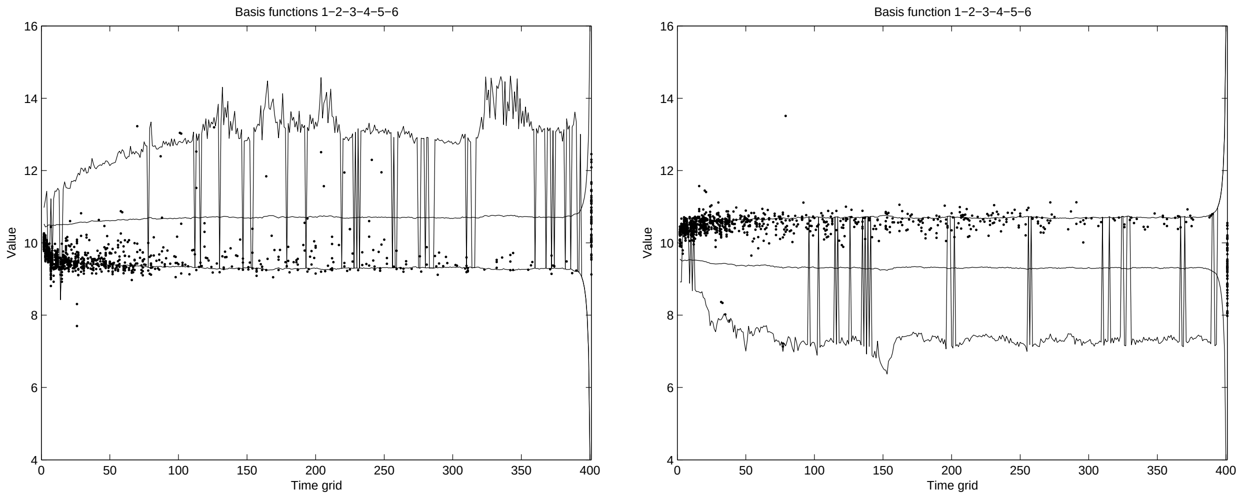

Figure 7: Left panel: points are $\{X_t | t = \arg\left(\sup_{u\in\mathcal{T}}\left(h_u - \mathcal{M}_u\right)\right)\}$ with $N_p = 16000, N = 1000, N_K = 100, k = 10$, basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and initial regime 1. There is a total of 1000 points. The lowest curve is the switching boundary from mode 1 to mode 0, the middle is the switching boundary from mode 0 to mode 1, and the uppermost curve is the second solution of the switching boundary from mode 1 to mode 0. Right panel: points are $\{X_t | t = \arg\left(\sup_{u\in\mathcal{T}}\left(h_u - \mathcal{M}_u\right)\right)\}$ with $N_p = 16000, N = 1000, N_K = 100, k = 10$, basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and initial regime 0. Again there is a total of 1000 points. The middle curve is the switching boundary from mode 1 to mode 0, the upper curve is the switching boundary from mode 0 to mode 1, and the lowest curve is the second solution of the switching boundary from mode 0 to mode 1.

When we do the simulation of upper bound with initial regime mode 0, we find the graph of $\{X_t | t = \arg\left(\sup_{u\in\mathcal{T}}\left(\hat{h}_u^k - \mathcal{M}_u\right)\right)\}$ is better (points are closer to the switching boundary), as the left panel of Figure 7 shows. The numerical results are shown in Table 5. The difference between the upper and lower bound is also much less than in Table 4.

N_p	swtch oppts	1	2	9	10
32000	$\hat{J}_0^{10}$	3.736(0.036)	5.079(0.030)	5.862(0.029)	5.862(0.029)
	upper bound	4.190(0.073)	5.3114(0.041)	5.996(0.034)	5.996(0.034)

Table 5: Simulation of upper bound with $N_p = 8000, M = 400, k = 10, N = 1000, N_k = 100$ and basis functions $\{B_1, B_2, B_3, B_4, B_5, B_6\}$. Initial regime is mode 0.

The difference between the right and the left panel of Figure 7 is not so surprising, since the simulation of $h_t^k(\overline{X}_t, 1) = \int_0^t 10(X_s - 10)ds - C_{1,0} + J^{k-1}(t, X_t, 0)$ will make more deviation than that of $h_t^k(\overline{X}_t, 0) = -C_{0,1} + J^{k-1}(t, X_t, 1)$.

3.3 Upper bound

3.3.1 Third Algorithm

To solve the problem of remark 3.1, we introduce the third algorithm, which changes the algorithm of lower bound to calculate a strict upper bound $\widetilde{J}^k(t, X_t, i)$for $J^k(t, X_t, i)$. First, we refer to equations (32) and (33) in the paper of Carmona and Ludkovski [2]:

$$H^k(m\Delta t, x^n_{m\Delta t}, i) = \begin{cases} \psi_i(m\Delta t, x^n_{m\Delta t})\Delta t + H^k\left((m+1)\Delta t, x^n_{(m+1)\Delta t}, i\right), \text{no switch;} \\ -C_{i,j} + H^{k-1}(m\Delta t, x^n_{m\Delta t}, j), \text{switch to } j. \end{cases} \tag{16}$$

$$\begin{aligned} \hat{j}(m\Delta t; i) = \arg\max_j \Big(&-C_{i,j} + H^{k-1}(m\delta t, x^n_{m\Delta t}, j), \\ &\psi_i(m\Delta t, x^n_{m\Delta t})\Delta t + (\hat{\mathbb{E}})_{m\Delta t}[H^k((m+1)\Delta t, \cdot, i)](x^n_{m\Delta t})]\Big). \end{aligned} \tag{17}$$

We now change equation (17) to:

$$\begin{aligned} \widetilde{j}(m\Delta t; i) = \arg\max_j \Big(&-C_{i,j} + H^{k-1}(m\delta t, x^n_{m\Delta t}, j), \\ &\psi_i(m\Delta t, x^n_{m\Delta t})\Delta t + H^k((m+1)\Delta t, \cdot, i)](x^n_{m\Delta t})\Big). \end{aligned} \tag{18}$$

Thus, we choose the switching time by "**precognition**" — assuming we know $(X_t)_{s<t<T}$ already at time s. $\widetilde{J}^k(t, X_t, i)$ is therefore an upper estimate of $J^k(t, X_t, i)$. Then we do regression on $\widetilde{J}^k(t, X_t, i)$, and set

$$\widetilde{h}_t^k(\overline{X}_t, i) = \int_0^t \psi_i(X_s)\, ds - C_{i,j} + \widetilde{J}^{k-1}(t, X_t, j).$$

Finally, we can use the second algorithm to calculate the upper bound based on payoff functions determined by $\widetilde{h}^k$. Results are shown in Table 6 below. For convenience we denote $\widetilde{J}_1^k$ for $\widetilde{J}^k(t = 0, X_0 = 10, 1)$. Observe that there are no guarantees that the upper bound computed in this way is larger than $\widetilde{J}_1^k$. The upper bound calculated here is higher than quasi-upper bound calculated with the second algorithm, except for the case when $k = 1$ (we have one only switching opportunity).

N_p	switch oppts	1	2	9	10
32000	$\widetilde{J}_1^k$	5.297(0.032)	6.208(0.030)	6.422(0.030)	6.422(0.030)
	upper bound	4.001(0.049)	5.900(0.029)	6.483(0.028)	6.483(0.028)

Table 6: Results of the third algorithm. $N_N = 40, N_p = 32000, N = 1000, N_K = 100$. The initial regime is mode 1.

4 Second Numerical Example

Now we suppose that the operator's actions impact the price and cost, i.e. the evolution of X_t. As an illustration, we change the model in the first numerical example to

$$\begin{cases} dX_t = 2(9.5 - X_t)dt + 2dW_t, & \text{if } u_t = 1, \\ dX_t = 2(10.5 - X_t)dt + 2dW_t, & \text{if } u_t = 0, \end{cases} \quad X_0 = 10. \tag{19}$$

When the operator decides to run the plant, the price of electricity will decrease, so the "balance point" is $X_t = 9.5$, lower than $X_t = 10.5$ when we shut down the plant. The payoff remains the same as in (8). According to the algorithm introduced by [2], we change (16) and (17) to:

$$H^k\left(m\Delta t, x^n_{m\Delta t}, i\right) = \begin{cases} \psi_i\left(m\Delta t, x^n_{m\Delta t}\right)\Delta t + H^k\left(\left(m+1\right)\Delta t, x^n_{(m+1)\Delta t}, i\right) \text{ no switch;} \\ -C_{i,j} + \psi_j\left(m\Delta t, x^n_{m\Delta t}\right)\Delta t + \hat{\mathbb{E}}_{m\Delta t}[H^{k-1}\left((m+1)\Delta t, \cdot, i\right)](x^n_{m\Delta t}), \text{switch to } j. \end{cases} \tag{20}$$

and

$$\hat{j}(m\Delta t; i) = \arg\max_j \Big(-C_{i,j} + \psi_j\left(m\Delta t, x^n_{m\Delta t}\right)\Delta t + \hat{\mathbb{E}}_{m\Delta t}[H^{k-1}\left((m+1)\Delta t, \cdot, j\right)](x^n_{m\Delta t}), \tag{21}$$
$$\psi_i(m\Delta t, x^n_{m\Delta t})\Delta t + (\hat{\mathbb{E}})_{m\Delta t}[H^k\left((m+1)\Delta t, \cdot, i\right)](x^n_{m\Delta t})\Big).$$

4.1 Lower Bound

As mentioned in [2], we separately simulate (X_t) under the 2 regimes with mode 0 and mode 1, and use the LS scheme over each set of paths $x^{n,i}_{m\Delta t}, i \in 0, 1, m \in 1, 2, ..., M$. The results are given in Table 7.

switch oppts	1	2	9	10
$\hat{J}_1^{10}$	1.255(0.028)	3.366(0.062)	4.674(0.047)	4.675(0.047)
$\hat{J}_0^{10}$	2.958(0.068)	4.181(0.046)	4.935(0.046)	4.936(0.047)

Table 7: Simulated lower bound with $X_0 = 10, N_N = 40, N_p = 16000, N = 1000, N_K = 100$. $\hat{J}_1^{10}$ denotes the lower bound with initial regime mode 1, $\hat{J}_0^{10}$ the lower bound with initial regime mode 0

With $X_0 = 10$, it is better to begin with mode 0, especially when the number of switching opportunities k is small. Similar to (9), we can draw the switching boundary in Figure 8. It is not so symmetric compared with the switching boundary of the first numerical example.

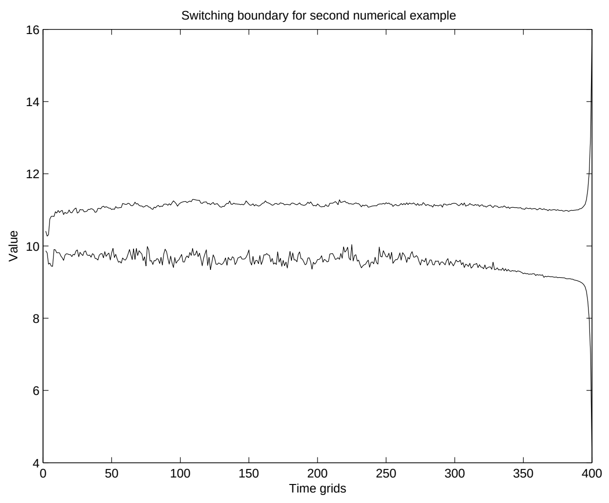

Figure 8: The graph of the switching boundary with basis functions $\{B_1, B_2, B_4, B_7, B_9\}$, $N_p = 16000, k = 10$. The upper curve is the switching boundary from mode 0 to mode 1, the lower curve is theX switching boundary from mode 1 to mode 0.

Remark 4.1. *As we can see from (*20*), this algorithm for the lower bound for the second numerical example is based on this deduction:*

if (property): $\hat{J}((m+1)\Delta t, X_{m\Delta t}, 1)$ *and* $\hat{J}((m+1)\Delta t, X_{m\Delta t}, 0)$ *are lower bounds for any value of* $X_{m\Delta t}$*, where* $m \in \mathbb{N}$*,*

then: we prove the same property at time $(m-1)\Delta t$*.*

But in the first numerical example, the justification of the algorithm for the lower bound did not require this property, in view of the fact that (16*) does not include the term* $\hat{\mathbb{E}}_{m\Delta t}[H^k\left((m+1)\Delta t, \cdot, i\right)](x^n_{m\Delta t})$*.*

Figure 6 *tells us that we can not assure this property, i.e. for same points* X_t*, we can have* $\hat{J}((m+1)\Delta t, X_{m\Delta t}, 1)$ *and* $\hat{J}((m+1)\Delta t, X_{m\Delta t}, 0)$ *larger than true value, especially when* X_t *is far from 10. So the results offered here may not be good (the lower bound here might be even higher than the true value). The lower bound algorithm in the second numerical example requires therefore much more attention in the select of basis function to make sure the the lower bound computed have this property.*

4.2 Quasi-upper Bound

We then use the second algorithm from Section 3.2.2 to calculate the quasi-upper bound.

switch oppts	1	2	9	10
$\hat{J}_1^{10}$	1.255(0.028)	3.366(0.062)	4.674(0.047)	4.675(0.047)
upper bound	1.800(0.090)	3.619(0.063)	4.747(0.048)	4.747(0.048)

Table 8: Quasi-upper bound with $X_0 = 10, N_N = 40, N_p = 16000, N = 1000, N_K = 100$, using the second algorithm and basis functions $\{B_1, B_2, B_4, B_7, B_9\}$. The initial regime is mode 1.

The difference between the quasi-upper bound and the lower bound is smaller than the first numerical example. If we use the lower bound algorithm introduced in the second numerical example to calculate the first numerical example, i.e. we see the first numerical example as

$$\begin{cases} dX_t = 2(10 - X_t)dt + 2dW_t, & \text{if } u_t = 1, \\ dX_t = 2(10 - X_t)dt + 2dW_t, & \text{if } u_t = 0, \end{cases} \quad X_0 = 10, \tag{22}$$

and calculate the lower bound in a similar way with (19)

we find that (21) and (20) may increase the lower bound computed, as was shown in Table 9. This may be one of the reasons why the difference in the second numerical example is much less than that of the first example. The "improvement" of the lower bound using the algorithm of second numerical example may be explained by Remark 4.1.

switch oppts	1	2	9	10
Algorithm in Example 1	3.704(0.084)	4.996(0.067)	5.743(0.070)	5.744(0.070)
Algorithm in Example 2	3.724(0.080)	5.031(0.084)	5.821(0.093)	5.820(0.096)

Table 9: The result of quasi-upper bound with $X_0 = 10, N_N = 40, N_p = 8000$, using the second algorithm and basis functions $\{B_1, B_2, B_4, B_7, B_9\}$. The initial regime is mode 1.

5 Conclusion

In this paper we have extended a new method used in pricing American options to investigate the numerical solution of operational flexibility problems. Our main theoretical proposal is trying to represent this kind of problems as a solution of a dual minimization problem. Due to the complexity of our problem, we can not compute a tight upper bound. However, we find a way to calculate a tight quasi-upper bound, which is not strictly an upper bound but higher than true value in our example. We have also considered the case when the operator's action may impact the price of the commodity. In the future, a better numerical solution and a better set of basis functions are needed to compute a tight lower and upper bound for this sort of problems.

6 Acknowledgment

It is a great pleasure to thank Michael Ludkovski for his stimulating ideas and his careful reading of early drafts of this paper.

References

[1] N. Aleksandrov and B.M. Hambly. A duel approach to multiple exercise option problems under constraints. 2009.

[2] René Carmona and Michael Ludkovski. Pricing asset scheduling flexibility using optimal switching. *Applied Mathematical Finance*, 15(6):405–447, 2008.

[3] M. Davis and I. Karatzas. A deterministic approach to optimal stopping, with applications. *Probability, Statistics and Optimization: A Tribute to Peter Whittle*, pages 455–466, 1994.

[4] Leonid Kogan Martin B. Haugh. Pricing american options: A duality approach. *Operations Research*, 52(2):258–270, 2001.

[5] N. Meinshausen and B.M. Hambly. Monte carlo methods for the valuation of multiple exercise options. *Math. Finance*, 14:557–583, 2004.

[6] L.G.G Rogers. Monte carlo valuation of american options. *Math. Finance*, 12:271–286, 2002.

7 Appendix: selected program code for our algorithms

We give the code of the algorithm (in Matlab) for the computation of the upper bound using the second algorithm in the first numerical example.

```
clear;

NNN=40; %Number of separate runs
N_p=16000;   % Number of paths to simulate the lower bound
N=1000;  % Number of paths to determine Martingale to calculate the upper bound
N_K=100;  %Number of inner paths to determine Martingale
T=2;  %Expired time
M=400;  %The division of time  (0,1,2,...M)*delta_t=T
K=11;   %the nomber of opportunity of switch
l=5; % The number of basis functions
J0_K=zeros(NNN,K);%The lower bound  with different switching
                  %opportunity with initial regime mode 0, when K=1,
                  %we can not change,and we note J0_K(:,1)=0 for convenience
J1_K=zeros(NNN,K); %The lower bound  with different
                   %switching opportunity with initial regime mode 1, when K=1,
                      %we can not change,and we note J1_K(:,1)=0 for convenience
Average=zeros(NNN,K); %The upper bound for different runs (NNN)
                      %and different switching opportunities (k=K-1)
Maxtimes=zeros(N,K,NNN);  %To record the time when the (h-M) is the largest
Max_X=zeros(N,K,NNN);%To record the largest (h-M)
cost=0.3;
k_mont=5;
for iiii=1:NNN

    Pre_Simu_X  = 10*ones(N_p,M+1);  %Matrix of N_p paths of (X_t) at
                                    %time grids   (t=0~T) -> Pre_Simu_X (:,t+1)

    medi_x=10*ones(N_p,1);  %This data is used to calculate Pre_Simu_X.  and X_0=10
    for i_Time=2:1:(M+1)
        for i_k_mont=1:1:k_mont
            Pre_Simu_W  = randn(N_p,1);  %Matrix of simulation results of W   FIN
            medi_x= medi_x + 2*(10-medi_x)*T/k_mont/M + 2*Pre_Simu_W* sqrt(T/M/k_mont);
        end
        Pre_Simu_X(:,i_Time)=medi_x;
    end

    clear  Pre_Simu_W

    %%                              REGRESSION                           %%
    % look up for the Page 16 of Carmona and Ludkovski, 2007
    H_post=zeros(N_p,K,2);%H^k((m+1)\delta t,x_{(m+1)\delta t} ^{n},  i),  1<=n<=N_p  ,
                          %i=initial regime 0 or 1 two status   at time: i_time+1
                           %H_post(:,:,1): initial regime is mode 0;
                           %%H_post(:,:,2): initial regime is mode 1;
    H_anti=zeros(N_p,K,2);%H^k(m\delta t,x_{m\delta t} ^{n},  i),
    Expt_H=zeros(N_p,K,2);
               %\hat{E}_{m\delta t,}[H^k((m+1)\delta t,x_{(m+1)\delta t} ^{n},  i)]
               % 2 status for   ::1 -> initial status 0;  ::2 ->initial status 1
    Coeff_0=zeros(l,K,M+1);  % coefficients of basis functions for beginning with mode 0
    Coeff_1=zeros(l,K,M+1);   %coefficients of basis functions for beginning with mode 1
    Psi=zeros(l,N_p);  %The value of basis functions (l)
                       %at different time for different path (N_p)
    for i_time=-(-M:1:-2)   %implement the algorithm to calculate the lower bound
        for i_K=1:1:K
            for i_N_p=1:1:N_p
                h00=Expt_H(i_N_p,i_K,1); %if we remain on mode 0 (h00 continuation value)
                h11=Expt_H(i_N_p,i_K,2)+10*(Pre_Simu_X(i_N_p,i_time)-10) *  (T/M);
                                %if we remain on mode 1  (h11 continuation value)

                if i_K>1 && Pre_Simu_X(i_N_p,i_time)>10
                    h01=-cost+
                         10*(Pre_Simu_X(i_N_p,i_time)-10)*(T/M)+
                               Expt_H(i_N_p,i_K-1,2);
                        %if we change from mode 0 to mode 1 (h01 switching value)
                else
                    h01=-inf;
                      %if k<1 and X_t<10,
                      %the transfer from mode 0 to mode 1 is forbidden
                end
                if i_K>1 &&  Pre_Simu_X(i_N_p,i_time)<10
                                 %if we change from mode 1 to mode 0
                    h10=-cost+Expt_H(i_N_p,i_K-1,1); %if k<1 and X_t>10,
                      %the transfer from mode 1 to mode 0
                      %is forbidden (h10 switching value)
                else
                    h10=-inf;
                end

                if h00 >= h01    % now, we compare the obtained continuation
                                %value against the regressed switching value
                    H_anti(i_N_p,i_K,1) = H_post(i_N_p,i_K,1);
                else %h00 < h01
                    H_anti(i_N_p,i_K,1)= -cost+
                          10*(Pre_Simu_X(i_N_p,i_time)-10)*(T/M)+H_post(i_N_p,i_K-1,2);
                end
                if h11>=h10
                    H_anti(i_N_p,i_K,2) = H_post(i_N_p,i_K,2)+
                          10*(Pre_Simu_X(i_N_p,i_time)-10)*(T/M);
                else
                    H_anti(i_N_p,i_K,2) = H_post(i_N_p,i_K-1,1)-cost;
                end
            end
        end

            Psi(1,:)=ones(N_p,1)';
```

```
            Psi(2,:)=Pre_Simu_X(:,i_time-1)';
            Psi(3,:)=Pre_Simu_X(:,i_time-1)'.^2;
            Psi(4,:)=max(Pre_Simu_X(:,i_time-1)'-10,0);
            Psi(5,:)=Pre_Simu_X(:,i_time-1)'.^3;

%             Psi(1,:)=ones(N_p,1)';
%             Psi(2,:)=Pre_Simu_X(:,i_time-1)';
%             Psi(3,:)=log(Pre_Simu_X(:,i_time-1)');
%             Psi(4,:)=Pre_Simu_X(:,i_time-1)'.^2;
%             Psi(5,:)=log(Pre_Simu_X(:,i_time-1)').*Pre_Simu_X(:,i_time-1)';
%             Psi(6,:)=Pre_Simu_X(:,i_time-1)'.^2.*log(Pre_Simu_X(:,i_time-1)');
%             Psi(7,:)=max(Pre_Simu_X(:,i_time-1)'-10,0);
%             Psi(8,:)=exp(Pre_Simu_X(:,i_time-1)'-10);
%             Psi(9,:)=Pre_Simu_X(:,i_time-1)'.^3;

        for i_K=1:1:K  %regression

            Coeff_0(:,i_K,i_time-1)=Psi'\H_anti(:,i_K,1);  %  l*1= N*l  \  N*1
            Coeff_1(:,i_K,i_time-1)=Psi'\H_anti(:,i_K,2) ; %  l*1= N*l  \  N*1
            Expt_H(:,i_K,1)= Psi'*Coeff_0(:,i_K,i_time-1);
            Expt_H(:,i_K,2)= Psi'*Coeff_1(:,i_K,i_time-1);
        end
        H_post=H_anti;  %prepare for the (m-1)\Delta t
        H_anti=zeros(N_p,K,2);
    end

    for KK=2:1:K
        J0_K(iiii,KK)=sum(H_post(:,KK,1))/N_p;
            % the value of this product at time t=0 with initial mode=0   NNN*K
        J1_K(iiii,KK)=sum(H_post(:,KK,2))/N_p;
            % the value of this product at time t=0 with initial mode=1   NNN*K
    end
    J0_K
    J1_K

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%% The Calculate of Upper bound %%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    Pre_Simu_X  = 10*ones(N,M+1);
         %Matrix of simulation results of X_t  (t=0~T) -> Pre_Simu_X (:,t+1)  INI
    h_avant=zeros(N,(M+1)); % Calculate the \int_0^t \psi(X_t,i) ds
    h_mm=zeros(N,1);  % Aid to  calculate  h_avant
    medi_x=10*ones(N,1);
    for i_Time=2:1:(M+1)
        for i_k_mont=1:1:k_mont
            Pre_Simu_W  = randn(N,1);  %Matrix of simulation results of W   FIN
            h_mm=h_mm+10*(medi_x-10)*T/M/k_mont;
            medi_x=medi_x+2*(10-medi_x)*T/k_mont/M+2*Pre_Simu_W* sqrt(T/M/k_mont);
        end
        h_avant(:,i_Time)= h_mm;
        Pre_Simu_X(:,i_Time)=medi_x;
    end

    %%BASIS FUNCTION                                           %%
    Psi=zeros(N,(M+1),l);
    Psi(:,:,1)=ones(N,(M+1));
    Psi(:,:,2)=Pre_Simu_X;
    Psi(:,:,3)=Pre_Simu_X.^2;
    Psi(:,:,4)=max(Pre_Simu_X-10,0);
    Psi(:,:,5)=Pre_Simu_X.^3;

    Psi=shiftdim(Psi,2);  %l,N_p,(M+1)

%     Psi(:,:,1)=ones(N,(M+1));
%     Psi(:,:,2)=Pre_Simu_X;
%     Psi(:,:,3)=log(Pre_Simu_X);
%     Psi(:,:,4)=Pre_Simu_X.^2;
%     Psi(:,:,5)=Pre_Simu_X.*log(Pre_Simu_X);
%     Psi(:,:,6)=Pre_Simu_X.*Pre_Simu_X.*log(Pre_Simu_X);
%     Psi(:,:,7)=max(Pre_Simu_X-10,0);
%     Psi(:,:,8)=exp(Pre_Simu_X-10);
%     Psi(:,:,9)=Pre_Simu_X.^3;


    clear  Pre_Simu_W



    %%   APPROXIMATIONS TO THE OPTIMAL MARTINGALE %%

    Diff=zeros(N,K); %Matrix the largest difference for every path (N)
                     %and every switching opportunities (K).

    for i_N=1:1:N


        h_minus_M=zeros(M+1,K);%We will calculate the difference $ h-\mathcal{M}$ i
                              %n this matrix for all time grid (M+1),
                              %and switch oppts(K)
        h=zeros(1,K);  %h for every time grid

        h_minus_M(1,:)=-inf*ones(1,K);  %suppose that at t=0, we do not  change the mode
        M_m_m=zeros(1,K);  %The approximation of Martingal M and
                          %it changes as i_time changes. At time t=0, M=0


        for i_time=2:1:M+1
```

```
            %% CALCUL  EXPT_Delta_V%%
            EXPT_Delta_V=zeros(1,K); %E_{t-1}(V_t)
            Simu_X_K= Pre_Simu_X(i_N,i_time-1)*ones(N_K,1); %N_k*1  inner paths
            Simu_W_K=randn(N_K,k_mont);

            h_mmm=zeros( N_K,1);  %from t-1 to t, V_t and E_{t-1}(V_t) have
                                  %different \int_{t-1}^{t}\psi(X_s).
                                  %Here h_mmm= \int_{t-1}^{t}\psi(X_s)
            for i_k_mont=1:1:k_mont
                h_mmm=10*(Simu_X_K-10)*T/M/k_mont+h_mmm;
                Simu_X_K=Simu_X_K+2*(10-Simu_X_K)*T/M/k_mont+
                      2*Simu_W_K(:, i_k_mont)*sqrt(T/M/k_mont);
            end

            Psi_N_K=[ones(N_K,1) Simu_X_K Simu_X_K.^2 max(Simu_X_K-10,0) Simu_X_K.^3];
                                         %  Basis function for inner paths

%            1  ones(N_K,1)
%            2  Simu_X_K
%            3  log(Simu_X_K)
%            4  Simu_X_K.^2
%            5  Simu_X_K.*log(Simu_X_K)
%            6  (Simu_X_K.^2).*log(Simu_X_K)
%            7  max(Simu_X_K-10,0)
%            8  exp(Simu_X_K-10)
%            9  Simu_X_K.^3

            for i_N_K=1:1:N_K
                Psi_K=Psi_N_K(i_N_K,:);
                if i_time<=M
                    mmd=max(10*(Simu_X_K(i_N_K)-10)*T/M+Psi_K*Coeff_1(:,:,i_time),
                                     Psi_K*Coeff_0(:,:,i_time)-cost);%J^{k}(t,X_t,1)
                else
                    mmd=0; %when i_time=M+1, J^{k}(t,X_t,1)=0
                end
                EXPT_Delta_V=EXPT_Delta_V+mmd+h_mmm(i_N_K);
                            %+10*(Simu_X_K(i_N_K)-10)*T/M; Coeff: l,K,M+1
            end
            EXPT_Delta_V=EXPT_Delta_V/N_K;  % E_{t-1}(V_t) - \int_0^{t-1} \psi(X_s) ds

            %%   CALCUL   Delta_V    %%
            if i_time<=M  %to calculate  V_t
                mmd=max(10*(Pre_Simu_X(i_N,i_time)-10)*T/M
                       +Psi(:,i_N,i_time)'*Coeff_1(:,:,i_time),
                               Psi(:,i_N,i_time)'*Coeff_0(:,:,i_time)-cost);
            else
                mmd=0;
            end

            Delta_V=mmd ;  %+10*(Pre_Simu_X(i_N,i_time)-10)*T/M ;
            M_m_m=M_m_m+Delta_V+(h_avant(i_N,i_time)-h_avant(i_N,i_time-1))-
                                EXPT_Delta_V; %calculate the Martingale at time t
            h=zeros(1,K); %for the beginning, two states(:,:,1)->psi_0(:,:,2)->psi_1
            if i_time<=M  %to calculate h_t
                for KK=2:1:K
                    h(1,KK)=h_avant(i_N,i_time)-cost+
                            Psi(:,i_N,i_time)'*Coeff_0(:,KK-1,i_time);
                end
            else
                h(1,:)=h_avant(i_N,i_time); %if k=1, no opportunities to change
            end



            h_minus_M(i_time,:)=h-M_m_m;

        end

        [Diff(i_N,:) Maxtimes(i_N,:,iiii)]=max(h_minus_M);

         Max_X(i_N,:,iiii)=Pre_Simu_X(i_N,Maxtimes(i_N,:,iiii));%Pre_Simu_X: N,M+1

    end
    Average(iiii,:)=sum(Diff)/N

    clear Psi
end
J0_K_average=sum(J0_K)/NNN
J0_K_sigma=sqrt(var(J0_K))
J1_K_average=sum(J1_K)/NNN
J1_K_sigma=sqrt(var(J1_K))
average=sum(Average)/NNN
sigma=sqrt(var(Average))
%Max_X=shiftdim(Max_X,2);  %NNN,N,KK
%Maxtimes=shiftdim(Maxtimes,2);  %NNN,N,KK
%plot(reshape(Maxtimes(:,:,KK),[N*NNN,1]),reshape(Max_X(:,:,KK),[N*NNN,1]),'.');
% plot(Maxtimes(:,KK),Max_X(:,KK),'.');
```