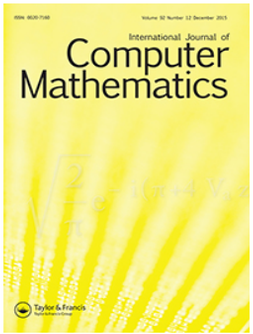

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: http://www.tandfonline.com/loi/gcom20

Pricing Bermudan options under Merton jump-diffusion asset dynamics

F. Cong & C.W. Oosterlee

To cite this article: F. Cong & C.W. Oosterlee (2015) Pricing Bermudan options under Merton jump-diffusion asset dynamics, International Journal of Computer Mathematics, 92:12, 2406-2432, DOI: 10.1080/00207160.2015.1070838

To link to this article: http://dx.doi.org/10.1080/00207160.2015.1070838

Accepted author version posted online: 20 Jul 2015.
Published online: 14 Aug 2015.

International Journal of Computer Mathematics, 2015
Vol. 92, No. 12, 2406–2432, http://dx.doi.org/10.1080/00207160.2015.1070838

Pricing Bermudan options under Merton jump-diffusion asset dynamics

F. Cong[a*] and C.W. Oosterlee[a,b]

[a]*TU Delft, Delft Institute of Applied Mathematics, Delft, The Netherlands;* [b]*CWI-Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*

(*Received 12 September 2014; revised version received 9 January 2015; second revision received 21 May 2015; accepted 25 May 2015*)

In this paper, a recently developed regression-based option pricing method, the Stochastic Grid Bundling Method (SGBM), is considered for pricing multidimensional Bermudan options. We compare SGBM with a traditional regression-based pricing approach and present detailed insight in the application of SGBM, including how to configure it and how to reduce the uncertainty of its estimates by control variates. We consider the Merton jump-diffusion model, which performs better than the geometric Brownian motion in modelling the heavy-tailed features of asset price distributions. Our numerical tests show that SGBM with appropriate set-up works highly satisfactorily for pricing multidimensional options under jump-diffusion asset dynamics.

Keywords: Monte Carlo simulation; least-squares regression; jump-diffusion process; Bermudan option; high-dimensional problem

2010 AMS Subject Classifications: 65C05; 80M31; 93E24; 60E75; 41A63

1. Introduction

Pricing high-dimensional Bermudan options is a challenging topic. For this type of problem, the traditional methods based on solving partial differential equations or on Fourier transformation may fail, because the complexity of these techniques grows exponentially as the dimensionality of the problem increases. Pricing methods based on simulation generally do not suffer from the curse of dimensionality and, therefore, have become increasingly attractive for high-dimensional pricing problems.

Simulation-based pricing for Bermudan options took off in 1993 when Tilley [22] introduced a bundling algorithm to estimate the continuation values of the option at intermediate time steps. In 1996, an option pricing method based on regression was introduced by Carriere [7]. The basic idea was to estimate the option's continuation values at all time points by projections of the future option values on finite-dimensional subspaces spanned by pre-selected basis functions. Depending on the procedure of generating basis functions, regression methods can be categorized into two types: Regress-Now and Regress-Later, as in [13]. More details of these two methods will be discussed in Section 3. Following Carriere's work [7], many papers discussing

*Corresponding author. Email: f.cong@tudelft.nl

regression methods based on the Regress-Now feature appeared, for example [17,23]. However, the investigation on Regress-Later methods is not abundant.

The Stochastic Grid Bundling Method (SGBM), a newly developed method introduced in [14], belongs to the type of Regress-Later approaches. In SGBM, both 'bundling' and 'regression' are utilized to estimate the continuation values. Similar to [5], SGBM produces two estimators: one biased high and the other biased low, which, respectively, correspond to the 'value function approximation' and the 'stopping time approximation' discussed in [21]. Compared to the well-known least-square method (LSM), introduced in [17], for pricing Bermudan options, SGBM typically yields estimates with significant lower variances, according to [14,15]. In our numerical tests, we obtain similar results: for achieving comparable accuracy, many more paths and higher computational times are required in LSM compared to SGBM. Moreover, according to [14], SGBM generates upper and lower bounds for the option price and also accurate sensitivities or Greeks of the option price, while the original LSM is only applicable for calculating the lower bound of the option price.

In this paper, we extend the discussion of SGBM in four directions. First, we gain insight into the essential components of SGBM. According to our analysis, it is sufficient to choose the basis functions of polynomial type, which ensures that conditional expectations of the basis functions can be calculated exactly. Second, in the error analysis, we explicate that the number of bundles used is a 'trade-off' factor of two types of biases in SGBM. Third, we combine SGBM with control variates to reduce the variance of the biased low estimator. We implement the traditional control variates and an improved approach proposed in [19]. According to the tests, the improved control variates work uniformly better in the one-dimensional case, but for higher dimensional problems, the cost of calculating the improved control variates is significant and the traditional control variates appear favourable. Instead of considering plain geometric Brownian motion, we focus our discussion on assets with their dynamics following the Merton jump-diffusion (MJD) process for high-dimensional Bermudan option pricing.

This paper is organized as follows. Section 2 gives the formulation of the problem. In Section 3, we compare SGBM with the standard regression method (SRM). In Section 4, we focus on the features of SGBM and explain how we can configure SGBM. In Section 5, the sources of errors in SGBM are compared to those in the SRM. Section 6 discusses traditional control variates and the improved versions. In Section 7, the MJD model is introduced and in Section 8, the corresponding numerical results are presented.

2. Problem formulation: Bermudan option pricing

This section describes the Bermudan option pricing problem mathematically and sets up the notations used in this paper. We assume that the financial market is defined on a complete filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0\le t\le T}, \mathbb{P})$ with finite time horizon $[0, T]$. Here the state space Ω is the set of all realizations of the financial market within time horizon $[0, T]$, $\mathcal{F}$ is the sigma algebra of events at time T, that is, $\mathcal{F} = \mathcal{F}_T$. We assume that the filtration $\{\mathcal{F}_t\}_{0\le t\le T}$ is generated by the price processes of the financial market and augmented with the null sets of $\mathcal{F}$. The probability measure $\mathbb{P}$ is defined on $\mathcal{F}$ and we assume that a risk-neutral measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ exists under which the asset prices are martingales with appropriate numeraire. The Bermudan option considered can be exercised within a set of prescribed time points $\mathbb{T} = [t_0 = 0, \ldots, t_m, \ldots, t_M = T]$. The d-dimensional state of economy is represented by an $\mathcal{F}_t$-adapted Markovian process $\boldsymbol{S}_t = (S_t^1, \ldots, S_t^d) \in \mathbb{R}^d$, where $t \in \mathbb{T}$. Let $h(\boldsymbol{S}_t)$ be the intrinsic value of the option, that is, the holder of the option receives pay-off $g(\boldsymbol{S}_t) = \max(h(\boldsymbol{S}_t), 0)$ if the option is exercised at time t. With the money savings account process $\beta_t = \exp(\int_0^t r_s \, \mathrm{d}s)$, where r_s denotes the instantaneous

risk-free rate of return, we define the discounting process as

$$D_{t_m} = \frac{\beta_{t_m}}{\beta_{t_{m+1}}}.$$

For simplicity, we consider the special case where r_s is equal to a constant r. The problem of valuing a Bermudan option is to find the optimal exercise strategy (or equivalently the optimal stopping time, $\tau \in \mathbb{T}$) and calculating the expected discounted pay-off following this strategy, that is:

$$V_0(\boldsymbol{S}_0) = \sup_{\tau \in \mathbb{T}} \mathbb{E}\left[\frac{h(\boldsymbol{S}_\tau)}{\beta_\tau} \middle| \mathcal{F}_0 \right]. \tag{1}$$

The expectation $\mathbb{E}[\cdot]$ is computed under the risk-neutral measure $\mathbb{Q}$. Here we write the option value in the form $V_0(\boldsymbol{S}_0)$ to emphasize that when the asset dynamics are fixed the option value is uniquely determined by the initial asset value.

The optimal exercise strategy can be determined via a recursive process, by which the option values, $V_{t_m}(\boldsymbol{S}_{t_m})$, at intermediate time steps can be computed correspondingly. The value of the Bermudan option at maturity state[1] $(T, \boldsymbol{S}_T)$ is equal to its pay-off, that is,

$$V_T(\boldsymbol{S}_T) = g(\boldsymbol{S}_T) = \max(h(\boldsymbol{S}_T), 0). \tag{2}$$

In the recursive process, the conditional continuation value $Q_{t_m}(\boldsymbol{S}_{t_m})$ associated with state $(t_m, \boldsymbol{S}_{t_m})$, that is, the discounted expected option value at time t_{m+1} conditioned on filtration $\mathcal{F}_{t_m}$, is given by[2]

$$Q_{t_m}(\boldsymbol{S}_{t_m}) = D_{t_m} \mathbb{E}[V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}}) \mid \boldsymbol{S}_{t_m}]. \tag{3}$$

The option value at state $(t_m, \boldsymbol{S}_{t_m})$ is then given by taking the maximum of its continuation value and the direct exercise value,

$$V_{t_m}(\boldsymbol{S}_{t_m}) = \max(Q_{t_m}(\boldsymbol{S}_{t_m}), g(\boldsymbol{S}_{t_m})). \tag{4}$$

We are interested in finding the option value at initial state $(t_0, \boldsymbol{S}_{t_0})$, using either Equation (1) or the recursive process as mentioned above.

3. Regression methods for Bermudan option pricing

We consider the Bermudan option with M equally distributed exercise opportunities before maturity T, that is, the option can be exercised at time $t_m = m\Delta t$, where $m = 1, \ldots, M$ and $\Delta t = T/M$. When the Monte Carlo generation for the sample of N paths is done and the function values $h(\cdot)$ are determined, we find the option value associated with each path at maturity directly via Equation (2). Similarly for the ith path, we obtain the direct exercise value $g(\boldsymbol{S}_{t_m}(i))$ at each exercise time t_m. The remaining problem is to calculate the conditional continuation value $Q_{t_m}(\boldsymbol{S}_{t_m})$ as in Equation (3). To settle this problem, regression methods are employed.

As mentioned, regression methods can be classified into two categories: Regress-Now and Regress-Later approaches. In the remaining part of this section, we consider the SRM, which resembles the method introduced in [7], as a typical case of Regress-Now methods and the SGBM as a representative of Regress-Later methods.

3.1 *Standard regression method*

The classical SRM, described in [11], has been widely discussed for pricing Bermudan options. The pricing procedure can be described as follows:

Step I: Get the option value at maturity time for each path:

$$V_{t_M}(\boldsymbol{S}_{t_M}(i)) = \max(h(\boldsymbol{S}_{t_M}(i)), 0), \quad i = 1, \ldots, N.$$

Recursively moving backward in time from maturity time t_M, the following steps are performed at time t_m, $m \leq M$.

Step II: Regression step.

For all paths, we get the regression parameters $\{\alpha_k\}_{k=1}^K$ by regressing the option values $\{V_{t_m}(\boldsymbol{S}_{t_m}(i))\}_{i=1}^N$ on basis functions $[\phi_1(\boldsymbol{S}_{t_{m-1}}(i)), \ldots, \phi_K(\boldsymbol{S}_{t_{m-1}}(i))]_{i=1}^N$, that are constructed using the asset values at time t_{m-1}. Linear regression offers us an approximation of the option value for any specified $S_{t_{m-1}}$, that is,

$$V_{t_m}(\boldsymbol{S}_{t_m}) \approx \sum_{k=1}^{K} \alpha_k \phi_k(\boldsymbol{S}_{t_{m-1}}). \tag{5}$$

Step III: Calculate the continuation value and the option value at time t_{m-1} for the ith path:

$$\begin{aligned} Q_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i)) &= D_{t_{m-1}} \mathbb{E}[V_{t_m}(\boldsymbol{S}_{t_m}) | \boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i)] \\ &\approx D_{t_{m-1}} \mathbb{E}\left[\sum_{k=1}^{K} \alpha_k \phi_k(\boldsymbol{S}_{t_{m-1}}) \,\middle|\, \boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i) \right] \\ &= D_{t_{m-1}} \sum_{k=1}^{K} \alpha_k \phi_k(\boldsymbol{S}_{t_{m-1}}(i)). \end{aligned} \tag{6}$$

The first equality is immediate from the definition of the continuation value in Equation (3). The approximation is supported by Equation (5). The second equality is valid based on a property of conditional expectations. The option value $V_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i))$ can be computed as follows:

$$V_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i)) = \max(Q_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i)), g(\boldsymbol{S}_{t_{m-1}}(i))). \tag{7}$$

3.2 *Stochastic grid bundling method*

The SGBM introduced in [14] belongs to the category of Regress-Later approaches. After generating all paths by Monte Carlo simulation, the algorithm of SGBM can be described as follows:

Step I: Get the option value at maturity for each path:

$$V_{t_M}(\boldsymbol{S}_{t_M}(i)) = \max(h(\boldsymbol{S}_{t_M}(i)), 0), \quad i = 1, \ldots, N.$$

The following steps are subsequently performed at time t_m, $m \leq M$.

Step II: Bundle paths at time t_{m-1}.

With a specified bundling criterion, we bundle all paths at time t_{m-1} into $\mathcal{B}_{t_{m-1}}(1), \ldots, \mathcal{B}_{t_{m-1}}(b), \ldots, \mathcal{B}_{t_{m-1}}(B)$ non-overlapping partitions. Figure 1 illustrates how bundling is performed in the one-dimensional case. The details of the bundling technique are discussed in the section to follow.

Step III: Regression step.

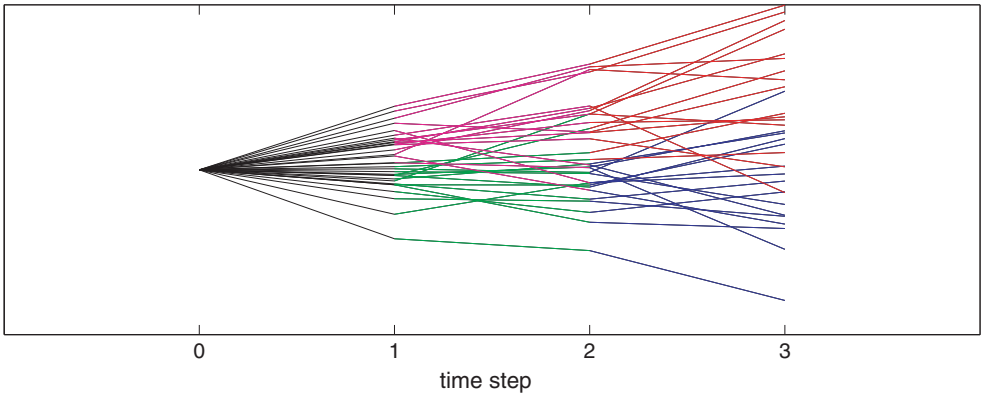

Figure 1. Paths from initial state to terminal time step 3. At the first backward recursion step, paths are bundled according to their state at time step 2, giving the 'red' and 'blue' paths in two bundles. At the next recursion step, the paths are bundled according to their state at time step 1, giving the 'magenta' and 'green' paths.

Assume that there are $N(b)$ paths in bundle $\mathcal{B}_{t_{m-1}}(b)$ and denote their asset values at time t_m as $\{\boldsymbol{S}_{t_m}^{(b)}(i)\}_{i=1}^{N(b)}$ and the option values as $\{V_{t_m}^{(b)}(i)\}_{i=1}^{N(b)}$. For these paths, we get the bundle regression parameters $\{\alpha_k(b)\}_{k=1}^{K}$ by regressing the option values $\{V_{t_m}^{(b)}(\boldsymbol{S}_{t_m}(i))\}_{i=1}^{N(b)}$ on the basis functions $[\phi_1(\boldsymbol{S}_{t_m}^{(b)}(i)),\ldots,\phi_K(\boldsymbol{S}_{t_m}^{(b)}(i))]_{i=1}^{N(b)}$, which are constructed using the asset values at time t_m. For assets whose values $\boldsymbol{S}_{t_{m-1}} = [S_{t_{m-1}}^1,\ldots,S_{t_{m-1}}^d]$ are covered by bundle $\mathcal{B}_{t_{m-1}}(b)$, the corresponding option value at time t_m can be approximated by[3]

$$V_{t_m}(\boldsymbol{S}_{t_m}) \approx \sum_{k=1}^{K} \alpha_k(b)\phi_k(\boldsymbol{S}_{t_m}). \tag{8}$$

At each time step, the regression is repeated for all bundles. In each bundle, the same basis functions $[\phi_1(\cdot),\ldots,\phi_K(\cdot)]$ are utilized.

Step IV: Calculate the continuation value and the option value at time t_{m-1} for the ith path.

Assume that the ith path at time t_{m-1} belongs to bundle $\mathcal{B}_{t_{m-1}}(b)$. The continuation value at time t_{m-1} associated with this path is given by

$$\begin{aligned} Q_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i)) &= D_{t_{m-1}}\mathbb{E}[V_{t_m}(\boldsymbol{S}_{t_m})|\boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i)] \\ &\approx D_{t_{m-1}}\mathbb{E}\left[\sum_{k=1}^{K}\alpha_k(b)\phi_k(\boldsymbol{S}_{t_m})\,\middle|\,\boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i)\right] \\ &= D_{t_{m-1}}\sum_{k=1}^{K}\alpha_k(b)\mathbb{E}[\phi_k(\boldsymbol{S}_{t_m})\,|\,\boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i)]. \end{aligned}$$

Note that, compared to Equation (6), the last equation contains conditional expectations of the basis functions, which is typical for Regress-Later approaches.

The motivation for the equality and approximation signs above is the same as for Step III of SRM. To obtain a closed-form expression for $Q_{t_{m-1}}(\boldsymbol{S}_{t_{m-1}}(i))$, we need analytic conditional expectations of the basis functions, $\mathbb{E}[\phi_k(\boldsymbol{S}_{t_m})|\boldsymbol{S}_{t_{m-1}} = \boldsymbol{S}_{t_{m-1}}(i)]$, $k = 1,\ldots,K$, which are achievable when the basis functions $\{\phi_k(\boldsymbol{S}_{t_m})\}_{k=1}^{K}$ are chosen appropriately. The option value can be computed via Equation (7).

4. Configuration of SGBM

There are basically two distinct features between the algorithms of SGBM and SRM:

- The basis functions in SGBM are required to have explicit analytic moments so that there is no error introduced in the last step of the algorithm. For SRM, the basis functions can be chosen freely.
- At each time step, the regression in SRM is done for all paths, while the regression in SGBM is done separately within each bundle. By the bundling technique in SGBM, the global fitting problem reduces to a local fitting problem.

Based on these two points, we will explain how to configure SGBM to make it feasible and robust for different scenarios.

4.1 *Choice of basis functions*

The special requirement for the basis functions in SGBM may complicate the application of this pricing algorithm for some involved options. For example, in [14], the powers of the maximum of asset values are chosen as the basis functions for pricing max-on-call options. Since the moments of these basis functions are not analytically available, they need to be approximated by Clark's algorithm [8]. Because of the inaccuracy of this numerical approximation, the duality method is required. This procedure makes the pricing algorithm less tractable.

We find that if the following conditions are satisfied, it is not necessary to choose 'max' or 'min' function as the basis functions.

CONDITION 4.1 *The transition probability density function* $f(s, \boldsymbol{S}_s; t, \boldsymbol{S}_t)$, *which denotes the probability density function from state* $(s, \boldsymbol{S}_s)$ *to state* $(t, \boldsymbol{S}_t)$, *is continuous with respect to* $\boldsymbol{S}_s$.

CONDITION 4.2 *The option's direct exercise value* $g(\boldsymbol{S}_t)$ *is continuous with respect to* $\boldsymbol{S}_t$.

With these conditions, we can prove the following theorem.

THEOREM 4.3 *At each exercise time, the option value* $V_{t_m}(\boldsymbol{S}_{t_m})$ *can be uniformly approximated by polynomials formed by* $\boldsymbol{S}_{t_m}$.

Proof Consider the backward pricing process of Bermudan options. At maturity time we have $V_T(\boldsymbol{S}_T) = g(\boldsymbol{S}_T)$, which is continuous with respect to $\boldsymbol{S}_T$. This follows directly from Condition 4.2. We then use backward induction. Assuming that $V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}})$ is continuous with respect to $\boldsymbol{S}_{t_{m+1}}$, we have

$$\begin{aligned} Q_{t_m}(\boldsymbol{S}_{t_m}) &= D_{t_m}\mathbb{E}[V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}}) \mid \boldsymbol{S}_{t_m}] \\ &= D_{t_m}\int_{\mathbb{R}^d} V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}}) f(t_m, \boldsymbol{S}_{t_m}; t_{m+1}, \boldsymbol{S}_{t_{m+1}})\, \mathrm{d}\boldsymbol{S}_{t_{m+1}} \\ &\approx D_{t_m}\int_{H} V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}}) f(t_m, \boldsymbol{S}_{t_m}; t_{m+1}, \boldsymbol{S}_{t_{m+1}})\, \mathrm{d}\boldsymbol{S}_{t_{m+1}}. \end{aligned}$$

The second equality is from the definition of conditional expectation and assuming that the dimension of $\boldsymbol{S}_{t_{m+1}}$ is d. The approximation sign is because of truncation of the integral from $\mathbb{R}^d$ to H. Without loss of generality, we assume that H is a compact subspace of $\mathbb{R}^d$.

Since $V_{t_{m+1}}(\boldsymbol{S}_{t_{m+1}})$ is continuous with respect to $\boldsymbol{S}_{t_{m+1}}$ on the compact domain H, it is bounded. With Condition 4.1, we can prove that $Q_{t_m}(\boldsymbol{S}_{t_m})$ is continuous with respect to $\boldsymbol{S}_{t_m}$.

The option price $V_{t_m}(\boldsymbol{S}_{t_m})$ is constructed by taking the maximum of the continuation value and the direct exercise value:

$$V_{t_m}(\boldsymbol{S}_{t_m}) = \max(Q_{t_m}(\boldsymbol{S}_{t_m}), g(\boldsymbol{S}_{t_m})) \tag{9}$$

and both $Q_{t_m}(\boldsymbol{S}_{t_m})$ and $g(\boldsymbol{S}_{t_m})$ are continuous with respect to $\boldsymbol{S}_{t_m}$. So, the option price $V_{t_m}(\boldsymbol{S}_{t_m})$ is also continuous with respect to $\boldsymbol{S}_{t_m}$.

We conclude the proof by using the generalized Stone–Weierstrass theorem on the space H. ■

Conditions 4.1 and 4.2 generally hold in option pricing. The continuous transition density functions associated with the commonly implemented models, such as the geometric Brownian motion and the jump-diffusion model, satisfy Condition 4.1 directly. The direct exercise value of a call or a put option is continuous with respect to the values of underlying assets.

Theorem 4.3 tells us that it is not necessary to include the 'max' or 'min' of underlying assets as a basis function. We choose here to only use polynomials as the basis functions in SGBM for multidimensional problems.

4.2 *Bundling*

A good 'bundling' technique should make the regression within the bundle easier, or, more precisely, make the regression less biased even though only a few paths are inside the bundle. This gives us a hint for bundling: if we bundle the paths such that paths in one bundle have similar option values, we expect that regression in this bundle would be easier.

The instruction that paths inside one bundle should have similar option values is not directly under our control, since bundling is done at time t_m but the option values considered in regression are from time t_{m+1}. However, the option value at time t_{m+1} should be to some degree related to its intrinsic value at time t_m. For example, considering the max-on-call option, if one path has a large intrinsic value at time t_m, which means that one asset associated with this path has a large value, we expect that the option value of this path at time t_{m+1} would still be large. In other words, if dramatic changes in the asset values are rarely to happen within the time interval $(t_{m+1} - t_m)$, paths, whose intrinsic values at time t_m are almost identical, are supposed to have similar option values at time t_{m+1}.

'Bundling' is not new in the field of Bermudan option pricing. Tilley [22] initiated the technique for pricing Bermudan options by Monte Carlo simulation using a simple bundling algorithm, which is however only applicable for a one-dimensional problem. Tilley's bundling algorithm can be described as a two-step method: 'reordering' and 'partitioning'. In the 'reordering' step, all paths are sorted according to their asset values. Then in the 'partitioning' step, the reordered paths are partitioned into distinct bundles of N_b paths each. The first N_b paths are assigned to the first bundle, the second N_b paths to the second bundle and so on.

Tilley's [2,10] bundling is extended to high-dimensional scenarios. The technique in [10], where multidimensional max options are dealt with using bundling, is to first reduce the multidimensional bundling problem to one dimension by choosing one single asset as representative for the multidimensional function. All paths are then bundled by applying Tilley's algorithm on the one-dimensional data. Within each bundle, a next bundling step is done by choosing another single asset as the new representative and again applying Tilley's algorithm. These newly generated bundles are called the 'sub-bundles'. The bundling can be done recursively within each sub-bundle until a prescribed number of bundles is reached.

Inspired by bundling in [2,10,22], we define our bundling algorithm as a two-step method. For reordering the paths in the multidimensional case, we first transform the multidimensional problem to a single-dimensional problem. Mathematically, it is equivalent to mapping the vector

$\boldsymbol{S}_t = (S_t^1, \ldots, S_t^d)$ to a number by specifying a function $R(\cdot)$, such that $R : \mathbb{R}^d \to \mathbb{R}$. In this paper, we call the variable $R(\boldsymbol{S}_t)$ the 'bundling reference'. Sometimes, we need more than one bundling reference as shown in [10]. In that case, we denote the bundling references subsequently as $R_1(\boldsymbol{S}_t)$, $R_2(\boldsymbol{S}_t)$ and so on.

In [10,22], the bundling is done to make each bundle cover the same number of paths so that we call it 'equal-size bundling'. This is different from the bundling in the original SGBM in [14], which we call 'equal-range bundling'.[4] In this paper, we will perform 'equal-size bundling'. According to our tests, there is no clear advantage on accuracy of either bundling scheme over the other. However, 'equal-size bundling' is more robust than 'equal-range bundling', because we always keep enough paths within each bundle to support the regression. If we choose the latter, the number of paths within some bundles may be so small that the estimation in those bundles is highly biased. The necessity of having enough paths inside one bundle will be further discussed in the next section.

Our bundling algorithm for the paths with asset values $\{\boldsymbol{S}_t(i)\}_{i=1}^N$, where $\boldsymbol{S}_t(i) = (S_t^1(i), \ldots, S_t^d(i))$, can be described as follows:

Step I: Reordering

(1) Based on the type of option, choose mapping functions $R_1(\cdot), \ldots, R_P(\cdot)$, by which the bundling references can be generated.
(2) Start with bundling reference $R_1(\boldsymbol{S}_t)$, bundle all paths equally into n_1 partitions following Tilley's bundling. Record the index of the bundle $b_1(i)(b_1(i) \in \{1, 2, \ldots, n_1\})$, where the ith $(i \in \{1, 2, \ldots, N\})$ path is located in.
(3) With reference $R_2(\boldsymbol{S}_t)$, divide the paths in a sub-bundle generated in the previous step into n_2 partitions. Again record the index of the bundle $b_2(i)(b_2(i) \in \{1, 2, \ldots, n_2\})$, where the ith $(i \in \{1, 2, \ldots, N\})$ path is located in.
(3) Repeat the process above with each bundling reference inside a sub-bundle. For the ith path, we get the vector recording its location $(b_1(i), \ldots, b_P(i))$, see Figure 2 for an example of recording the location of a single path.
(4) Construct the global bundling reference for the ith path as

$$R(\boldsymbol{S}_t(i)) = b_1(i) \cdot N^{P-1} + b_2(i) \cdot N^{P-2} + \cdots + b_{P-1}(i) \cdot N + b_P(i), \quad i = 1, \ldots, N.$$

(5) Reorder the paths according to the global bundling reference $R(\boldsymbol{S}_t)$.

Step II: Partitioning

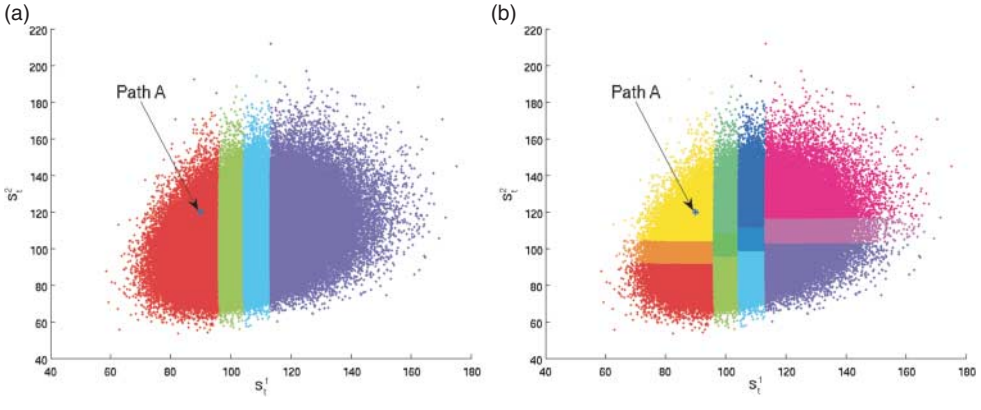

Figure 2. Obtaining the location of Path A with two bundling references. (a) Start bundling according to $R_1(\boldsymbol{S}_t) = S_t^1$ with $n_1 = 4$. Record the location of Path A as $(1, \cdot)$. (b) Within each bundle perform sub-bundling according to $R_2(\boldsymbol{S}_t) = S_t^2$ with $n_2 = 3$. Record the location of Path A as $(1, 3)$.

Partition the sorted paths into $\prod_{p=1}^{P} n_p$ bundles each of $N/\prod_{p=1}^{P} n_p$ paths, where $\prod_{p=1}^{P} n_p$ is an integer factor of N.

The following examples demonstrate that some common bundling schemes fit into our generalized bundling technique.

Example 4.4 For a one-dimensional problem, we choose the bundling reference $R(S_t) = S_t$. So the bundling algorithm covers the simplest one-dimensional case.

For a basket option of assets $\boldsymbol{S}_t = (S_t^1, \ldots, S_t^d)$, if we choose bundling references, respectively, equal to the value of each individual asset, we will get 'bundling on the original state space', as termed in [14].

4.2.1 *Choosing the bundling reference.*

After we have specified the basis functions of polynomial type, the performance of SGBM depends on whether we can choose an accurate bundling reference. For example, when we consider the geometric basket option with underlying assets following multidimensional geometric Brownian motion, an accurate bundling reference is the geometric mean of the asset values. This is supported by the fact that the geometric average of (jointly) log-normal random variables is still log-normal. This implies that when dealing with the geometric basket option, an optimal bundling reference is the geometric mean of the asset values. Moreover, although there is no representation technique for the arithmetic basket option, our tests suggest that the arithmetic mean of asset values is a preferred bundling reference for arithmetic basket options.

For options whose pay-off functions are related to the 'max' or 'min' of asset values, choosing the intrinsic value alone as the bundling reference is not sufficient, as shown in Example 4.5. Inspired by this example, we should separate paths whose option values are related to only one asset, from paths whose option values are affected by each asset. This gives us another bundling reference: the difference between the asset values. In Section 8.2, we can see that combining them offers us a much better result than using any of them individually and this combination also outperforms other possible combinations of the bundling references.

Example 4.5 If we consider a two-dimensional put-on-min option with assets $\boldsymbol{S}_t = (S_t^1, S_t^2)$ and strike $K = 2$, following the instructions in the previous subsection we choose basis functions as

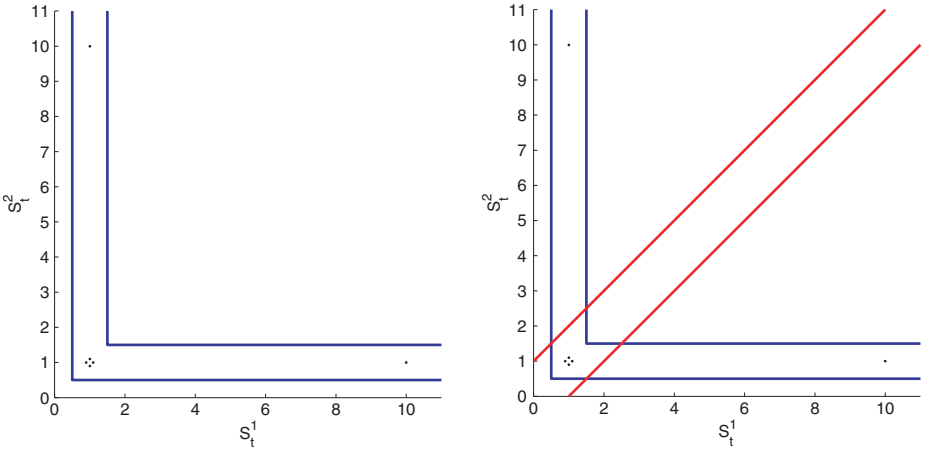

Figure 3. Bundling for pricing the two-dimensional put-on-min option: (a) one bundling reference and (b) two bundling references.

$[1, S_t^1, S_t^2]$. Assume that we have six paths, respectively, with assets $\boldsymbol{S}_t(1) = (1, 10)$, $\boldsymbol{S}_t(2) = (10, 1)$, $\boldsymbol{S}_t(3) = (1, 0.9)$, $\boldsymbol{S}_t(4) = (1, 1.1)$, $\boldsymbol{S}_t(5) = (0.9, 1)$ and $\boldsymbol{S}_t(6) = (1.1, 1)$. Their option values are recorded as $[1, 1, 1.1, 1, 1.1, 1]$.

If we bundle these paths in the same partition based on their intrinsic values (Figure 3(a)), then the approximated option values will be $[1, 1, 1.05, 1.05, 1.05, 1.05]$. If we introduce one more bundling reference (Figure 3(b)) so that the first two points are separated from the other ones, then the approximated option values for the last four paths will be given by $[1.1, 1, 1.1, 1]$.

5. Error analysis: comparing SRM and SGBM

In this section, we will compare the errors of SRM and SGBM when estimating the conditional continuation value $Q_{t_{m-1}}(\hat{\boldsymbol{S}})$, where we denote $\hat{\boldsymbol{S}}$ as a realization of $\boldsymbol{S}_{t_{m-1}}$. Here we consider the estimation error in one backward pricing step, so the option value $V_{t_m}(\boldsymbol{S}_{t_m})$ at time t_m is assumed to be known exactly. In the following discussions, we will write $\pi(\boldsymbol{S}_{t_m})$ as the density function of $\boldsymbol{S}_{t_m}$ conditioned on $\boldsymbol{S}_{t_{m-1}} = \hat{\boldsymbol{S}}$. With these notations, the analytic continuation value $Q_{t_{m-1}}(\hat{\boldsymbol{S}})$ reads[5]

$$Q_{t_{m-1}}(\hat{\boldsymbol{S}}) = \mathbb{E}[V_{t_m}(\boldsymbol{S}_{t_m})|\boldsymbol{S}_{t_{m-1}} = \hat{\boldsymbol{S}}] = \mathbb{E}^{\pi}[V_{t_m}(\boldsymbol{S}_{t_m})], \tag{10}$$

where $\mathbb{E}^{\pi}[\cdot]$ indicates that the expectation is computed with $\pi(\boldsymbol{S}_{t_m})$ as the density function of $\boldsymbol{S}_{t_m}$.

5.1 *Error in the optimal regression-based approach*

Let us start with a trivial problem where we perform sub-simulation to calculate $Q_{t_{m-1}}(\hat{\boldsymbol{S}})$. In the framework of Monte Carlo pricing, we simulate the realizations $\{\hat{\boldsymbol{S}}_{t_m}(i)\}_{i=1}^N$ with the dynamics associated with the density function $\pi(\boldsymbol{S}_{t_m})$.[6] We denote their empirical density function as $\hat{\pi}(\boldsymbol{S}_{t_m})$, which can be defined by[7]

$$\hat{\pi}(\boldsymbol{S}_{t_m}) := N^{-1} \sum_{i=1}^{N} \delta(\boldsymbol{S}_{t_m}(i) - \boldsymbol{S}_{t_m}),$$

where $\delta(\cdot)$ indicates a kernel density function. With a suitable simulation technique, we assume that the empirical density $\hat{\pi}(\boldsymbol{S}_{t_m})$ resembles its theoretical counterpart $\pi(\boldsymbol{S}_{t_m})$ when the number of simulation trajectories goes to infinity.

Since the function $V_{t_m}(\boldsymbol{S}_{t_m})$ is assumed to be known, we find realizations $\{V_{t_m}(\hat{\boldsymbol{S}}_{t_m}(i))\}_{i=1}^N$ of the exact option value. If we estimate $Q_{t_{m-1}}(\hat{\boldsymbol{S}})$ by regression instead of simply by taking the average of $\{V_{t_m}(\hat{\boldsymbol{S}}_{t_m}(i))\}_{i=1}^N$, we can regress $\{V_{t_m}(\hat{\boldsymbol{S}}_{t_m}(i))\}_{i=1}^N$ on $\{\phi_1(\hat{\boldsymbol{S}}_{t_m}(i)), \ldots, \phi_K(\hat{\boldsymbol{S}}_{t_m}(i))\}_{i=1}^N$ and obtain the regression parameters $\{\hat{\alpha}_k\}_{k=1}^K$, which minimize the sum of the squared errors of the samples:

$$\sum_{i=1}^{N} \left(V_{t_m}(\hat{\boldsymbol{S}}_{t_m}(i)) - \sum_{k=1}^{K} \hat{\alpha}_k \phi_k(\hat{\boldsymbol{S}}_{t_m}(i)) \right)^2.$$

Since $\{\hat{\boldsymbol{S}}_{t_m}(i)\}_{i=1}^N$ follows the empirical distribution of density $\hat{\pi}(\boldsymbol{S}_{t_m})$, the regression parameter $\{\hat{\alpha}_k\}_{k=1}^K$ also minimizes the mean square error:

$$\mathbb{E}^{\hat{\pi}} \left[\left(V_{t_m}(\boldsymbol{S}_{t_m}) - \sum_{k=1}^{K} \hat{\alpha}_k \phi_k(\boldsymbol{S}_{t_m}) \right)^2 \right].$$

If we denote the regression error $\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})$ as

$$\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m}) = V_{t_m}(\boldsymbol{S}_{t_m}) - \sum_{k=1}^{K} \hat{\alpha}_k \phi_k(\boldsymbol{S}_{t_m}),$$

the least-squares linear regression guarantees

$$\mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})] = 0. \tag{11}$$

The approximated continuation value $\hat{Q}_{t_{m-1}}(\hat{\boldsymbol{S}})$ can be computed as

$$\begin{aligned}\hat{Q}_{t_{m-1}}(\hat{\boldsymbol{S}}) &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{K} \hat{\alpha}_k \phi_k(\boldsymbol{S}_{t_m})\right] = \mathbb{E}^{\pi}[V_{t_m}(\boldsymbol{S}_{t_m}) + \epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})] \\ &= Q_{t_{m-1}}(\hat{\boldsymbol{S}}) + (\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})]).\end{aligned}$$

The first equality is directly from the regression-based approximation scheme. The second equality is valid because we rewrite the approximated option value as the true option value plus regression error, and the last step is supported by Equation (11). Since we can simulate a large number of realizations of $\boldsymbol{S}_{t_m}$, the empirical distribution function $\hat{\pi}(\boldsymbol{S}_{t_m})$ resembles $\pi(\boldsymbol{S}_{t_m})$. Moreover, using the Cauchy–Schwarz inequality we have

$$\begin{aligned}|\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})]| &= \left|\int_H \epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})\pi(\boldsymbol{S}_{t_m})\,\mathrm{d}\boldsymbol{S}_{t_m} - \int_H \epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})\hat{\pi}(\boldsymbol{S}_{t_m})\,\mathrm{d}\boldsymbol{S}_{t_m}\right| \\ &\le \left(\int_H (\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m}))^2\,\mathrm{d}\boldsymbol{S}_{t_m}\right)^{1/2} \cdot \left(\int_H (\pi(\boldsymbol{S}_{t_m}) - \hat{\pi}(\boldsymbol{S}_{t_m}))^2\,\mathrm{d}\boldsymbol{S}_{t_m}\right)^{1/2},\end{aligned}$$

where the integral domain H is defined in Section 4.1 as a truncated subspace of $\mathbb{R}^d$ and we assume that the regression error is bounded on this domain. When the sample size is sufficiently large, $|\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\boldsymbol{S}_{t_m})]|$ will be close to 0 and therefore $\hat{Q}_{t_{m-1}}(\hat{\boldsymbol{S}})$ will be an accurate approximation of $Q_{t_{m-1}}(\hat{\boldsymbol{S}})$.

However, the above-mentioned process is not achievable in a real application, since we cannot afford sub-simulation for every state $(t_{m-1}, \boldsymbol{S}_{t_{m-1}})$. As feasible alternatives, we have the cross-path regression methods, for example, SRM and SGBM.

5.2 *Error in SRM*

In the regression step of SRM, we regress the option values $\{V_{t_m}(\boldsymbol{S}_{t_m}(i))\}_{i=1}^{N}$ on the basis functions $[\phi_1(\boldsymbol{S}_{t_{m-1}}(i)), \ldots, \phi_K(\boldsymbol{S}_{t_{m-1}}(i))]_{i=1}^{N}$. Since all paths are generated from the same initial state, we denote the theoretical density function of $\boldsymbol{S}_{t_m}$ by $\pi_G(\boldsymbol{S}_{t_m})$ and the empirical density function, which is represented by the samples $\{\boldsymbol{S}_{t_m}(i)\}_{i=1}^{N}$, by $\hat{\pi}_G(\boldsymbol{S}_{t_m})$. Since the regression is done with respect to the samples $\{\boldsymbol{S}_{t_m}(i)\}_{i=1}^{N}$, we have

$$\mathbb{E}^{\hat{\pi}_G}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})] = 0. \tag{12}$$

After we determine the regression parameters $\{\alpha_k^G\}_{k=1}^{K}$, the approximated continuation value $\hat{Q}^G_{t_{m-1}}(\hat{\boldsymbol{S}})$ can be generated as

$$\begin{aligned}\hat{Q}^G_{t_{m-1}}(\hat{\boldsymbol{S}}) &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{K} \alpha_k^G \phi_k(\boldsymbol{S}_{t_{m-1}})\right] = \mathbb{E}^{\pi}[V_{t_m}(\boldsymbol{S}_{t_m}) + \epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})] \\ &= Q_{t_{m-1}}(\hat{\boldsymbol{S}}) + (\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})]) + (\mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}_G}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})]).\end{aligned}$$

The last equality is found by writing $\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})]$ in the form of a telescopic sum and eliminating the last term based on Equation (12).

When there are enough samples, we have $\hat{\pi}_G(\boldsymbol{S}_{t_m}) \approx \pi_G(\boldsymbol{S}_{t_m})$, which leaves the approximation bias in SRM merely determined by

$$\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\boldsymbol{S}_{t_m})].$$

Since $\pi(\boldsymbol{S}_{t_m})$ stands for the analytic density function of $\boldsymbol{S}_{t_m}$ conditioned on $\boldsymbol{S}_{t_{m-1}} = \hat{\boldsymbol{S}}$ and $\pi_G(\boldsymbol{S}_{t_m})$ for the analytic density function of $\boldsymbol{S}_{t_m}$ conditioned on $\boldsymbol{S}_{t_0}$, they are obviously not identical. This makes the path-wise bias in SRM uncontrollable no matter how we change the set-up of simulation.

5.3 *Error in SGBM*

In SGBM, we consider the paths originating from the same bundle $\mathcal{B}_{t_{m-1}}(b)$, which covers the state $(t_{m-1}, \hat{\boldsymbol{S}})$, and regress the option values $\{V_{t_m}^{(b)}(\boldsymbol{S}_{t_m}^{(b)}(i))\}_{i=1}^{N(b)}$ on the basis functions $[\phi_1(\boldsymbol{S}_{t_m}^{(b)}(i)), \ldots, \phi_K(\boldsymbol{S}_{t_m}^{(b)}(i))]_{i=1}^{N(b)}$. Again we denote the theoretical density function of $\boldsymbol{S}_{t_m}$, whose previous state $(t_{m-1}, \boldsymbol{S}_{t_{m-1}})$ is within the spreading of bundle $\mathcal{B}_{t_{m-1}}(b)$, by $\pi_B(\boldsymbol{S}_{t_m})$ and the empirical density function of $\{\boldsymbol{S}_{t_m}^{(b)}(i)\}_{i=1}^{N(b)}$ by $\hat{\pi}_B(\boldsymbol{S}_{t_m})$.

With similar arguments as in SRM, we obtain the approximated continuation value $\hat{Q}_{t_{m-1}}^{B}(\hat{\boldsymbol{S}})$ by SGBM as

$$\hat{Q}_{t_{m-1}}^{B}(\hat{\boldsymbol{S}}) = Q_{t_{m-1}}(\hat{\boldsymbol{S}}) + (\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})]) + (\mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})]).$$

Different from SRM, the set-up in SGBM can help us to control the bias. When we increase the number of bundles, the spreading of any individual bundle will reduce. In the limiting case where the bundle covers only the state $(t_{m-1}, \hat{\boldsymbol{S}})$, we will have $\pi_B(\boldsymbol{S}_{t_m}) = \pi(\boldsymbol{S}_{t_m})$. However, in a simulation-based approach, if we do not increase the total sample size, increasing the number of bundles will cause a decrease in the number of paths per bundle, which makes the empirical density function $\hat{\pi}_B(\boldsymbol{S}_{t_m})$ different from the analytic density function $\pi_B(\boldsymbol{S}_{t_m})$.

To summarize, if we regard $\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})]$ as the 'distribution bias' and $\mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})] - \mathbb{E}^{\hat{\pi}_B}[\epsilon^{\hat{\pi}_B}(\boldsymbol{S}_{t_m})]$ as the 'sample bias', the number of bundles is a 'trade-off' between these two types of biases. To make a balance, we should choose the number of bundles neither too small nor too large so that both the biases are controlled.

Based on the analysis above, we can conclude that the path-wise estimation error of regression methods comes from two parts: the regression error and the sample bias. By choosing suitable basis functions, we reduce the impact of the first part. By introducing 'bundling', we control the sample bias and also simplify the problem of global regression to that of local regression.

6. Variance reduction for path estimator

6.1 *Path estimator*

From the backward pricing algorithm of SGBM, we will get a biased high estimator $\bar{V}_0(\boldsymbol{S}_0)$ of the initial option value $V_0(\boldsymbol{S}_0)$. We call this estimator the direct estimator. Once we obtain the regression parameters for any bundle at any time step, the approximated continuation value $\hat{Q}_{t_m}(\boldsymbol{S}_{t_m})$ of the option at the given state $(t_m, \boldsymbol{S}_{t_m})$ can be calculated. Relying on this approximation, we can decide either to exercise the option or to hold it at the specified state. Based on this

exercise policy and some fresh simulated paths, we can develop a biased low estimator $\underline{V}_0(\boldsymbol{S}_0)$ of the option value $V_0(\boldsymbol{S}_0)$. We call this estimator the path estimator. The procedure of calculating the path estimator can be described as

Step I Simulate a new sample of paths $\{\boldsymbol{S}_0(i), \ldots, \boldsymbol{S}_{t_M}(i)\}$, $i = 1, \ldots, N_p$.

Step II Based on the approximation of the continuation value, determine the optimal exercise time $\hat{\tau}(i)$ for the ith path:

$$\hat{\tau}(i) = \min(t_m : h(\boldsymbol{S}_{t_m}(i)) \geq \hat{Q}_{t_m}(\boldsymbol{S}_{t_m}(i)),\ m = 1, \ldots, M). \tag{13}$$

Step III Compute the path estimator:

$$\underline{V}_0(\boldsymbol{S}_0) = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{h(\boldsymbol{S}_{\hat{\tau}(i)}(i))}{\beta_{\hat{\tau}(i)}}.$$

The proof of convergence and the bias of the path estimator are shown in [14].

6.2 *Variance reduction: control variates*

When estimating the option value via Monte Carlo simulation, we not only desire a precise point estimate but also pursue a reasonable interval estimate, which is constructed in the form of the point estimate plus-or-minus its standard error multiplied by the confidence factor. A simulation-based method should provide us a narrow interval estimate, which implies that the method is valid even in extreme cases.

Within the framework of the general simulation-based estimation, the standard error of the point estimate is believed to be proportional to the reciprocal of the square root of the sample size. Therefore, to reduce the range of the interval estimate by a factor of 10, the sample size should increase by a factor of 100.

Variance reduction methods offer us an alternative approach to reduce the standard error of the estimation. A commonly used variance reduction method is the control variate method, which has been implemented for American-style option pricing, for example, in [4,6,19].

As the first choice for control variates for pricing Bermudan options one would consider the corresponding European options, whose values can be easily computed. From the perspective of optimal exercise, we never exercise a single asset Bermudan call option before its maturity, so using the European option will provide us a zero-variance control. As concluded by Rasmussen in [19], for pricing the Bermudan option 'a good control variate' should have the following two properties: it should be highly correlated with the pay-off of the option in question and its conditional expectation should be easy to compute.

For simplicity, we restrict the following discussion to using only one control variate. For the generalized control variate method, where multi-controls are involved, we refer the reader to [1].

6.2.1 *Path estimator with control variates*

To improve, in particular, the path estimator based on crude Monte Carlo simulation by control variates, the Bermudan option value $h(\boldsymbol{S}_{\hat{\tau}(i)}(i))/\beta_{\hat{\tau}(i)}$ for the ith path will be replaced by

$$Z_{\hat{\tau}}(i) = \frac{h(\boldsymbol{S}_{\hat{\tau}(i)}(i))}{\beta_{\hat{\tau}(i)}} + \theta(Y(i) - \mathbb{E}[Y]),$$

where $Y(i)$ denotes the control variate for the ith path and $\mathbb{E}[Y]$ is its analytic expectation with filtration $\mathcal{F}_0$. The weighting parameter θ can be chosen freely, since the new estimate $Z_{\hat{\tau}}$ is always

unbiased to the original estimate $h(\boldsymbol{S}_{\hat{\tau}})/\beta_{\hat{\tau}}$. A reasonable choice would be

$$\theta = \frac{\text{Cov}(h(\boldsymbol{S}_{\hat{\tau}})/\beta_{\hat{\tau}}, Y)}{\text{Var}(Y)},$$

which controls the variance of $Z_{\hat{\tau}}$ to the minimum value:

$$\text{Var}\left(\frac{h(\boldsymbol{S}_{\hat{\tau}})}{\beta_{\hat{\tau}}}\right)(1-\rho^2),$$

where $\rho = \text{Corr}(h(\boldsymbol{S}_{\hat{\tau}})/\beta_{\hat{\tau}}, Y) = \text{Cov}(h(\boldsymbol{S}_{\hat{\tau}})/\beta_{\hat{\tau}}, Y)/\sqrt{\text{Var}(h(\boldsymbol{S}_{\hat{\tau}})/\beta_{\hat{\tau}})\text{Var}(Y)}$. In [19], the ratio $1/(1-\rho^2)$ is called the 'speed-up factor', which indicates that utilizing control variates is equivalent to amplifying the sample size in a crude Monte Carlo by a factor of $1/(1-\rho^2)$.

6.2.2 *Traditional control variates*

As mentioned before, the conditional expectations of the control variates should be easy to calculate. For the Bermudan option with multidimensional underlying assets $\boldsymbol{S}_t = (S_t^1, \ldots, S_t^\delta, \ldots, S_t^d) \in \mathbb{R}^d$, the traditional choice of control variate is the discounted pay-off of the European option measured at maturity with the single underlying asset S_t^δ, that is,

$$Y = \frac{g(S_T^\delta)}{\beta_T},$$

and the Monte Carlo estimate with control variate for the ith path is

$$Z_{\hat{\tau}}(i) = \frac{h(\boldsymbol{S}_{\hat{\tau}(i)}(i))}{\beta_{\hat{\tau}(i)}} + \theta\left(\frac{g(S_T^\delta(i))}{\beta_T} - \mathbb{E}\left[\frac{g(S_T^\delta)}{\beta_T}\right]\right).$$

Here $\mathbb{E}[g(S_T^\delta)/\beta_T]$ indicates the value of the single asset European option starting at state $(0, S_0^\delta)$ and maturing at time T with the same strike as the discussed Bermudan option.

6.2.3 *Improved control variates*

Monte Carlo pricing with the option pay-off measured at maturity as control variate is quite cheap, because after the simulation of the paths, we get $\{g(S_T^\delta(i))\}_{i=1}^N$ immediately. However, empirical tests indicate that this choice of control variate is not always efficient. An alternative introduced in [19] is to replace $g(S_T^\delta(i))/\beta_T$ by $W_{\hat{\tau}}(i)$ which is defined as

$$W_{\hat{\tau}}(i) = \frac{1}{\beta_{\hat{\tau}(i)}}\mathbb{E}_{\hat{\tau}(i)}\left[\frac{g(S_T^\delta(i))}{\beta_{T-\hat{\tau}(i)}}\right].$$

The stopping time $\hat{\tau}(i)$ for the ith path is defined in Equation (13). $\mathbb{E}_{\hat{\tau}(i)}[\cdot]$ indicates the conditional expectation with filtration $\mathcal{F}_{\hat{\tau}(i)}$ and $\mathbb{E}_{\hat{\tau}(i)}[g(S_T^\delta(i))/\beta_{T-\hat{\tau}(i)}]$ denotes the single asset European option value associated with the ith path starting at state $(\hat{\tau}(i), S_{\hat{\tau}(i)}^\delta(i))$ and maturing at time T with the same strike as the Bermudan option. The expectation $\mathbb{E}[W_{\hat{\tau}}]$ is identical to the single asset European option value $\mathbb{E}[g(S_T^\delta)/\beta_T]$. Rasmussen [19] shows that this new choice of control variate makes variance reduction more efficient. However, we notice that $\{W_{\hat{\tau}}(i)\}_{i=1}^N$ is not directly available any more, because for the ith path, it is the discounted single asset European option value measured at time $\hat{\tau}(i)$. For one-dimensional pricing problems, fast pricing algorithms that help us calculate the path-wise control variates $\{W_{\hat{\tau}}(i)\}_{i=1}^N$ efficiently exist. We will implement the COS method introduced in [9] for the one-dimensional pricing.

In our numerical test, we find that in some situations applying improved control variates is much more efficient than increasing the sample size. However, for the geometric basket option and the arithmetic basket option, the effect of using improved control variates is not so obvious that we still prefer to use traditional control variates and increase the sample size for reducing the standard error of estimation.

7. MJD process

7.1 *Motivation of jump-diffusion model*

Despite the wide use of the geometric Brownian motion to model the movement of asset prices, the almost instantaneous asset price change cannot be captured well. Such rapid price variations are sometimes modelled by a 'jump'. It is stated in [16] that the jump model behaves better in modelling the leptokurtic feature of the asset price distribution and the empirical phenomenon 'volatility smile' in option markets.

Jump-diffusion models essentially contain a Brownian component punctuated by jumps at random intervals. Compared to their counterparts 'infinite activity Lévy processes' in jump models, 'finite activity jump-diffusion models' are easier to simulate. In this paper, we will consider an elementary jump model, the MJD model,[8] which was introduced in [18].

7.2 *Model formulation*

We consider the MJD model with contagious jumps on each asset, that is, the jumps in the dynamics of each asset arrive following the same Poisson process. Under this model, the d-dimensional asset prices follow

$$\mathrm{d}S_t^i = S_t^i((r - \delta_i - \lambda\kappa_i)\,\mathrm{d}t + \mathrm{d}W_t^i + (\mathrm{e}^{Z^i} - 1)\,\mathrm{d}\Gamma_t), \quad i = 1, \ldots, d,$$

where $\kappa_i = \mathbb{E}[\mathrm{e}^{Z^i} - 1]$, $\mathrm{d}W_t^i\,\mathrm{d}W_t^j = \sigma_i\sigma_j\rho_{ij}\,\mathrm{d}t$, r the risk-free rate, δ_i the dividend rate, σ_i the volatility of diffusion, Γ_t a Poisson process with mean arrival rate λ, $\boldsymbol{Z} = [Z^1, \ldots, Z^d]'$ the multivariate normally distributed jumps with mean $\boldsymbol{\mu}^J = [\mu_1^J, \ldots, \mu_d^J]'$ and covariance matrix Σ^{J} with elements $\Sigma_{ij}^J = \sigma_i^J\sigma_j^J\rho_{ij}^J$.

The analytic formulas for the dynamics read

$$S_t^i = S_0^i \exp((r - \delta_i - \lambda\kappa_i)t + W_t^i) \exp\left(\sum_{m=1}^{N(t)} Z_m^i\right), \quad i = 1, \ldots, d, \tag{14}$$

where $\boldsymbol{S}_0 = (S_0^1, \ldots, S_0^d)$ is the initial state, $\boldsymbol{W}_t = [W_t^1, \ldots, W_t^d]'$ the diffusion component, $\boldsymbol{Z}_m = [Z_m^1, \ldots, Z_m^d]'$ the jump component and $N(t)$ the number of Poisson jumps within time interval t with mean arrival rate λ. The diffusion component $\boldsymbol{W}_t$ follows multivariate normal distribution with mean 0 and covariance matrix Σ with elements $\Sigma_{ij} = \sigma_i\sigma_j\rho_{ij}t$ and the jump component $\boldsymbol{Z}_m$ with mean $\boldsymbol{\mu}^J = [\mu_1^J, \ldots, \mu_d^J]'$ and covariance matrix Σ^J with elements $\Sigma_{ij}^J = \sigma_i^J\sigma_j^J\rho_{ij}^J$.

The log-process $\boldsymbol{X}_t = (X_t^1, \ldots, X_t^d)$, where $X_t^i = \log(S_t^i), i = 1, \ldots, d$, has an analytic form:

$$\boldsymbol{X}_t = \boldsymbol{X}_0 + \boldsymbol{\mu} \cdot t + \boldsymbol{W}_t + \sum_{m=1}^{N(t)} \boldsymbol{Z}_m,$$

where $\boldsymbol{\mu} = [r - \delta_1 - \lambda\kappa_1 - \sigma_1^2/2, \ldots, r - \delta_d - \lambda\kappa_d - \sigma_d^2/2]$.

7.3 *Dimension reduction: geometric average of MJD assets*

Within the framework of geometric Brownian motion, the dynamics of the geometric average of multidimensional assets can be formulated as a one-dimensional problem. This technique is also applicable for the MJD model.[9] Suppose that the equivalent one-dimensional MJD model has an analytic formula:

$$\tilde{S}_t = \tilde{S}_0 \exp((r - \tilde{\delta} - \lambda\tilde{\kappa})t + \tilde{W}_t) \exp\left(\sum_{m=1}^{N(t)} \tilde{Z}_m\right),$$

where $\tilde{W}_t$ is normally distributed with mean 0 and variance $\tilde{\sigma}^2 t$ and $\tilde{Z}_m$ normally distributed with mean $\tilde{\mu}^J$ and variance $\tilde{\sigma}^J$. To make it represent the geometric mean of the assets with dynamics shown in Equation (14), we need

$$\tilde{S}_0 = \left(\prod_{i=1}^{d} S_0^i\right)^{1/d}, \quad \tilde{\mu}^J = \frac{\sum_{i=1}^{d} \mu_i^J}{d},$$
$$\tilde{\sigma}^J = \frac{\sqrt{\sum_{i,j} \sigma_i^J \sigma_j^J \rho_{ij}^J}}{d}, \quad \tilde{\sigma} = \frac{\sqrt{\sum_{i,j} \sigma_i \sigma_j \rho_{ij}}}{d},$$
$$\tilde{\kappa} = \exp\left(\tilde{\mu}^J + \frac{(\tilde{\sigma}^J)^2}{2}\right) - 1, \quad \tilde{\delta} = \frac{\sum_{i=1}^{d} (\delta_i + \sigma_i^2/2 + \lambda\kappa_i)}{d} - \frac{\tilde{\sigma}^2}{2} - \lambda\tilde{\kappa}.$$

Remark 7.1 This dimension reduction technique also works on the geometric basket containing assets following geometric Brownian motion and assets following the MJD model, because geometric Brownian motion can be regarded as an MJD model with zero jump size.

7.4 *Analytic moments of basis functions in the MJD model*

For the model with dynamics shown in Equation (14), we have the conditional expectations of polynomial basis functions in closed form. The conditional moments of the original stock prices, $\mathbb{E}[(S_t^i)^k | S_0^i]$ $(i = 1, \ldots, d;\ k = 1, 2, \ldots)$, read

$$\mathbb{E}[(S_t^i)^k | S_0^i] = (S_0^i)^k \exp(k\hat{\mu}_i t + \tfrac{1}{2}k^2\sigma_i^2 t + \lambda t(\exp(k\mu_i^J + \tfrac{1}{2}k^2(\sigma_i^J)^2) - 1)), \tag{15}$$

where

$$\hat{\mu}_i = r - \delta_i - \frac{\sigma_i^2}{2} - \lambda\left(\exp\left(\mu_i^J + \frac{(\sigma_i^J)^2}{2}\right) - 1\right).$$

The conditional moments of the geometric mean of the asset prices $\{S_t^i\}_{i=1}^d$ can be calculated by first presenting the dynamics of the geometric mean in one dimension as shown in Section 7.3 then using Equation (15).

Since there exists no general form of the conditional expectation of the log-stock prices $\mathbb{E}[\log(S_t^i)^k|S_0^i]$ $(i = 1,\ldots,d;\ k = 1,2,\ldots)$, we present the first three moments as follows:

$$\begin{aligned}
\mathbb{E}[\log(S_t^i)|S_0^i] &= \log(S_0^i) + \hat{\mu}_i t + \lambda t \mu_i^J,\\
\mathbb{E}[\log(S_t^i)^2|S_0^i] &= (\log(S_0^i) + \hat{\mu}_i t)^2 + \sigma_i^2 t + 2(\log(S_0^i) + \hat{\mu}_i t)(\mu_i^J \lambda t)\\
&\quad + (\lambda^2 t^2 + \lambda t)(\mu_i^J)^2 + \lambda t(\sigma_i^J)^2,\\
\mathbb{E}[\log(S_t^i)^3|S_0^i] &= (\log(S_0^i) + \hat{\mu}_i t)^3 + 3(\log(S_0^i) + \hat{\mu}_i t)(\sigma_i^2 t) + 3((\log(S_0^i) + \hat{\mu}_i t)^2 + \sigma_i^2 t)(\mu_i^J \lambda t)\\
&\quad + 3(\log(S_0^i) + \hat{\mu}_i t)((\lambda^2 t^2 + \lambda t)(\mu_i^J)^2 + \lambda t(\sigma_i^J)^2)\\
&\quad + (\lambda^3 t^3 + 3\lambda^2 t^2 + \lambda t)(\mu_i^J)^3 + 3(\lambda^2 t^2 + \lambda t)(\mu_i^J(\sigma_i^J)^2).
\end{aligned}$$

The conditional expectation of the cross-product term $\mathbb{E}[\log(S_t^i)\log(S_t^j)|S_0^i, S_0^j](i \neq j)$ reads

$$\begin{aligned}
\mathbb{E}[\log(S_t^i)\log(S_t^j)|S_0^i, S_0^j] &= (\log(S_0^i) + \hat{\mu}_i t)(\log(S_0^j) + \hat{\mu}_j t)\\
&\quad + \sigma_i\sigma_j\rho_{ij} t + (\log(S_0^j) + \hat{\mu}_j t)(\mu_i^J \lambda t) + (\log(S_0^i) + \hat{\mu}_i t)(\mu_j^J \lambda t)\\
&\quad + (\lambda^2 t^2 + \lambda t)\mu_i^J \mu_j^J + \lambda t \sigma_i^J \sigma_j^J \rho_{ij}^J.
\end{aligned}$$

8. Numerical experiments

In this section, we perform several numerical experiments to test the performance of SGBM for pricing different types of Bermudan options with assets following the MJD process. We compare different choices of bundling references, basis functions and variance reduction approaches for options on multidimensional assets.

The one-dimensional MJD model is furnished with three different choices of model parameters, which, respectively, indicate 'common' jump, 'intensive' jump and 'rare' jump. The multidimensional tests are conducted for various options: geometric basket, arithmetic basket, put-on-min and call-on-max. For some scenarios, we get the benchmark value directly from the literature. However, in case of the absence of references we generate the benchmark ourselves. For the geometric basket option, the representation discussed in Section 7.3 is implemented and the one-dimensional problem is solved by the COS method. For the remaining scenarios, we implement the LSM method [17] to generate the reference values. MATLAB R2011b is used and the computations are performed on Intel(R) Core(TM) i5 3.33 GHz processor with 16 GB RAM.

The parameter sets used for the tests are listed in Table 1.

8.1 *SGBM and tuning parameters*

We start testing the performance of SGBM on single asset options under geometric Brownian motion dynamics, which gives a general insight how SGBM performs with its tuning parameters. The performance of SGBM is influenced by three parameters:[10]

(1) N and N_p: the number of simulated paths,
(2) n: the number of bundles and
(3) M: the number of exercise opportunities.

In the following tests, the default set-up is as follows: $n = 16, M = 20, N = 2^{17}, N_p = 2^{-2} \cdot N$. The model parameters are chosen from Set I in Table 1 without the jump component. We use

Table 1. Parameter settings used in the test.

Set I(a): 'common' jump
$S_0 = 40, K = 40, r = 0.06, \delta = 0, \sigma = 0.2, \lambda = 3, \mu^J = -0.2, \sigma^J = 0.2, T = 1, M$[a] $= 20$.
Set I(b): 'intensive' jump
$S_0 = 40, K = 40, r = 0.06, \delta = 0, \sigma = 0.2, \lambda = 8, \mu^J = -0.2, \sigma^J = 0.2, T = 1, M = 20$.
Set I(c): 'rare' jump
$S_0 = 40, K = 40, r = 0.06, \delta = 0, \sigma = 0.2, \lambda = 0.1, \mu^J = -0.9, \sigma^J = 0.45, T = 1, M = 20$.
Set II:
$\boldsymbol{S}_0 = [100, 100]', K = 100, r = 0.05, \delta = 0, \sigma = [0.12, 0.15]', \rho_{ij} = 0.3, \lambda = 0.6,$
$\boldsymbol{\mu}^J = [-0.1, 0.1]', \boldsymbol{\sigma}^J = [0.17, 0.13]', \rho_{ij}^J = -0.2, T = 1, M = 8$.
Set III:
$\boldsymbol{S}_0 = [100, 100]', K = 100, r = 0.05, \delta = 0.1, \sigma = [0.2, 0.2]', \rho_{ij} = 0, T = 3, M = 9$.
Set IV :
$\boldsymbol{S}_0 = [100, 100, 100, 100, 100]', K = 100, r = 0.05, \delta = 0, \sigma = [0.15, 0.15, 0.15, 0.15, 0.15]', \rho_{ij} = 0.3, \lambda = 0.5,$
$\boldsymbol{\mu}^J = [-0.3, -0.2, -0.1, 0.1, 0.2]', \boldsymbol{\sigma}^J = [0.1, 0.1, 0.1, 0.1, 0.1]', \rho_{ij}^J = -0.2, T = 1, M = 8$.

[a] M denotes the number of early-exercise opportunities, which are equidistantly distributed in T years.

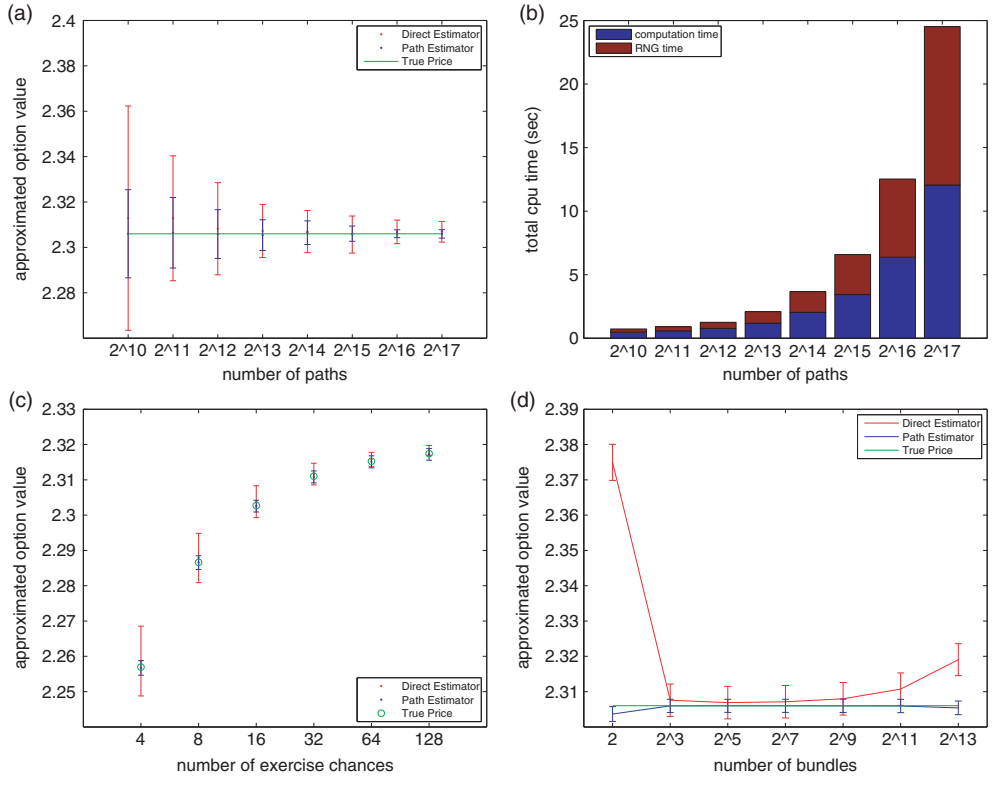

Figure 4. Performance of SGBM with different tuning parameters. The set-up is described in Section 8.1. The reference option price is generated by the COS method [9]: (a) fix M and n, vary N, (b) fix M and n, vary N, (c) fix N and n, vary M and (d) fix N and M, vary n.

the improved control variates for variance reduction on the path estimator, and perform three tests, respectively, by changing n, M and N. The test results are plotted in Figure 4.

Figure 4(a) shows that as the sample size increases, the standard error of the estimators in SGBM decreases by the order $N^{-1/2}$. As shown in Figure 4(b), the total computational time increases in order N. When we increase the number of exercise opportunities up to $M = 128$, both the direct and the path estimator of SGBM are satisfactory. If we keep doubling the number

of exercise opportunities, the performance of the direct estimator in SGBM decreases while the path estimator remains reliable. The poor performance of the direct estimator in case of many exercise opportunities is mainly caused by the 'distribution bias' as explained in Section 5. However, in any scenario the path estimator is approving, as in LSM. Figure 4(d) displays the trade-off between the 'distribution bias' and the 'sample bias'. When the number of bundles is small, the 'distribution bias' is the dominant part that makes the direct estimator highly biased. As the number of bundles increases, SGBM exhibits highly satisfactory performance. However, when the number of bundles increases further, the 'sample bias' forms a problem and the direct estimator in SGBM becomes unsatisfactory.

8.2 *Choice of bundling reference*

In this section, we consider different bundling schemes while testing and focusing on pricing the put-on-min option with assets $\boldsymbol{S}_t = (S_t^1, S_t^2)$ following the two-dimensional MJD model. The model parameters are chosen from Set II in Table 1. The basis functions are fixed as follows: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$. Seven different ways for bundling are included in the test. They are as follows:

(1) bundling according to one reference:

- bundling reference A: $R^{\mathrm{A}}(\boldsymbol{S}_t) = \min(S_t^1, S_t^2)$,
- bundling reference B: $R^{\mathrm{B}}(\boldsymbol{S}_t) = S_t^1$,
- bundling reference C: $R^{\mathrm{C}}(\boldsymbol{S}_t) = S_t^1 - S_t^2$.

(2) bundling according to two references:

- bundling reference D: $R_1^{\mathrm{D}}(\boldsymbol{S}_t) = S_t^1$, $R_2^{\mathrm{D}}(\boldsymbol{S}_t) = S_t^2$,
- bundling reference E: $R_1^{\mathrm{E}}(\boldsymbol{S}_t) = \min(S_t^1, S_t^2)$, $R_2^{\mathrm{E}}(\boldsymbol{S}_t) = S_t^1 - S_t^2$,
- bundling reference F: $R_1^{\mathrm{F}}(\boldsymbol{S}_t) = \min(S_t^1, S_t^2)$, $R_2^{\mathrm{F}}(\boldsymbol{S}_t) = S_t^1$,
- bundling reference G: $R_1^{\mathrm{G}}(\boldsymbol{S}_t) = S_t^1 - S_t^2$, $R_2^{\mathrm{G}}(\boldsymbol{S}_t) = S_t^1$.

According to Figure 5, when pricing the put-on-min options, we should not limit ourselves to bundling with a single reference, which is outperformed by any bundling scheme with two references. Among the two-reference bundling schemes, the one involving the intrinsic value of the option and the difference between the asset values is the best choice. For the geometric basket option and the arithmetic basket option, we do not present our test results. However, for both of them bundling simply with the option's intrinsic value yields highly satisfactory results.

8.3 *Choice of basis functions*

One aspect influencing the efficiency of regression methods for option pricing is the choice of basis functions. Although we claim that it is sufficient to get convergent results for various option contracts by simply choosing polynomials as basis functions, in some situations we have alternative choices. For example, for the geometric basket option, it is recommended in [14] to choose the powers of the geometric mean of asset prices as basis functions. We compare three choices of basis functions for pricing the geometric basket option with assets following the two-dimensional MJD process:

- Basis A: the intrinsic value of option

$$1, \sqrt{S_t^1 S_t^2}, S_t^1 S_t^2, \sqrt{S_t^1 S_t^2}^3, (S_t^1 S_t^2)^2.$$

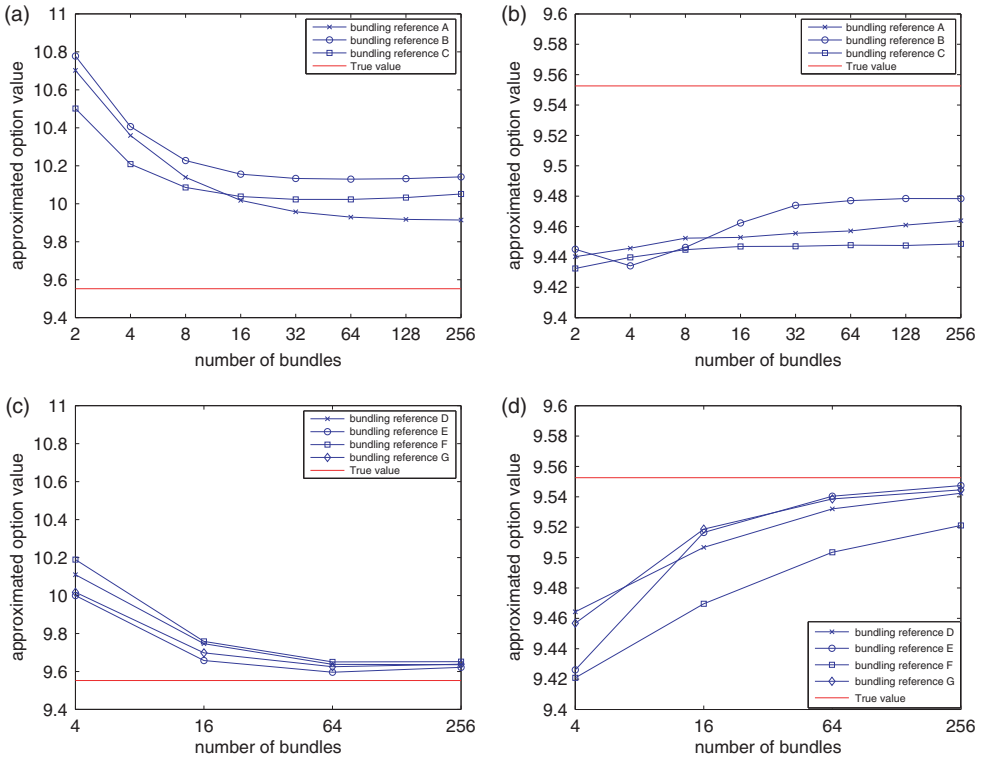

Figure 5. Comparison of different bundling schemes for pricing two-dimensional put-on-min option. The basis functions are fixed as follows: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$. When the bundling is done according to two references, the number of bundles with respect to each reference is the square root of the 'number of bundles'. The sample size for the direct estimator is 2^{17} and the sample size for the path estimator is 2^{18}. The reference option price is collected from [20]: (a) direct estimator (one bundling reference), (b) path estimator (one bundling reference), (c) direct estimator (two bundling reference) and (d) path estimator (two bundling reference).

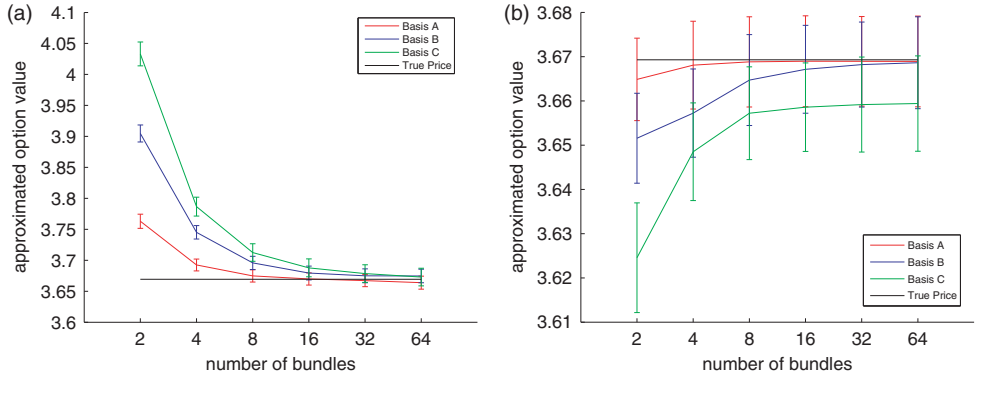

Figure 6. Comparing different choices of basis functions in SGBM for pricing two-dimensional geometric basket option. The parameters of the MJD model are chosen from Set II in Table 1. The sample size for the direct estimator is 2^{17}. The path estimator with sample size 2^{18} is controlled by traditional control variates. (a) Direct estimator in SGBM, (b) Path estimator in SGBM.

- Basis B: polynomial terms of asset prices:

$$1, \log(S_t^1), \log(S_t^1)^2, \log(S_t^2), \log(S_t^2)^2, \log(S_t^1)\log(S_t^2).$$

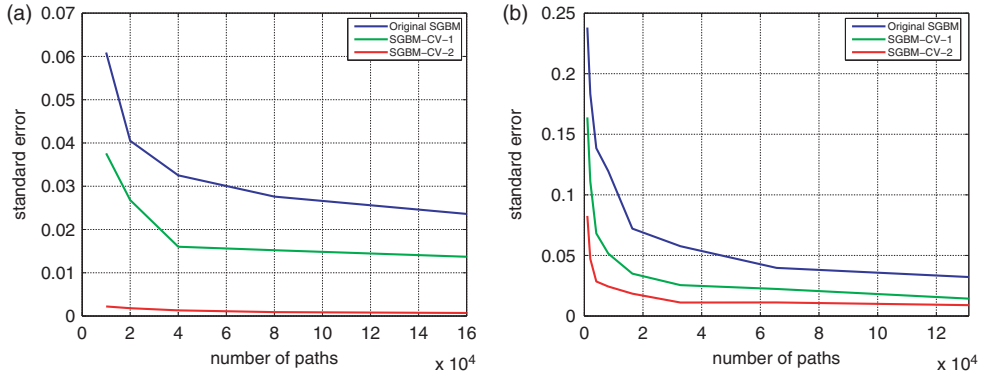

Figure 7. Comparison of two variance reduction approaches. The standard errors of the path estimators following three different algorithms are presented. 'Original SGBM' means standard SGBM algorithm without control, 'SGBM-CV-1' stands for SGBM with the traditional control variates and 'SGBM-CV-2' for SGBM with the improved control variates. For all three algorithms, the basis functions, the bundling reference and the number of the paths for the direct estimator are identical. The x-axis indicates the sample size for the path estimator. (a) Using control variates for a put option with asset following a one-dimensional MJD model and (b) using control variates for a put-on-min option with assets following a two-dimensional MJD model.

- Basis C: polynomial terms of asset prices, without the cross-product term:

$$1, \log(S_t^1), \log(S_t^1)^2, \log(S_t^2), \log(S_t^2)^2.$$

Figure 6 shows that different choices of basis functions in SGBM have an impact on the option price estimates. When the number of bundles is small, Basis A performs best. When the number of bundles is sufficiently large, the final results of SGBM with Basis A and Basis B are very similar. On the other hand, we notice that although Basis C does not appear satisfactory compared to the other two choices, the confidence intervals of the associated direct and the path estimator cover the true option values. For truly high-dimensional problems, including the cross-product terms into basis functions will lead to a quadratic increase in the number of basis functions, which further requires an exponential increase in the sample size to make the regression accurate [12]. In this case, we prefer to form basis functions by the polynomials without cross-product terms.

So, when dealing with low-dimensional problems, we choose the ordinary polynomials as basis functions since their conditional expectations are always available. When the dimensionality of the problem is high, we consider polynomials without cross-product terms, or, if possible, we use polynomials of the option's intrinsic values as the basis functions. In Section 8.7, we can see that these choices of basis functions provide us highly satisfactory results.

8.4 *Efficiency of using control variates*

One problem with SGBM [14] is that the standard error of the path estimator is usually larger than that of the direct estimator. Here we test the variance reduction methods introduced in Section 6 for reducing the standard error of the path estimator. The test is performed under one-dimensional and two-dimensional MJD models. Figure 7(a) and 7(b) show that using the control variates helps to reduce the standard error of the path estimator. In the one-dimensional case using the improved control variates is extremely efficient with a speed-up factor around 900, while for the two-dimensional model the improved control variate is less efficient with speed-up factor around 14. In both scenarios, applying the traditional control variates provides us a variance reduction with speed-up factor around 3.

Table 2. Comparing two variance reduction approaches in SGBM pricing for different types of two-dimensional options.

	Geometric basket		Arithmetic basket		Put-on-min		
N_p^a	2^{16}	2^{17}	2^{16}	2^{17}	2^{16}	2^{18}	2^{19}
Control variates[b]	CV-2	CV-1	CV-2	CV-1	CV-2	CV-1	No
S.e.[c]	0.0123	0.0097	0.0118	0.0107	0.0093	0.0106	0.0151
RNG time[d]	6.2051	11.4671	6.2051	11.4671	6.2051	23.7172	48.8246
Computation time	20.3588	2.9489	20.3984	3.2454	21.8977	5.3324	10.1195
Total time	26.5639	14.4160	26.6035	14.7125	28.1028	29.0496	58.9441
Speed-up factor	3.7284	2.1081	2.7834	1.8016	13.8211	3.9224	1

Note: The sample size for the direct estimator is always fixed as 2^{17}.
[a]The sample size for the path estimator.
[b]'CV-1' stands for SGBM with the traditional control variates and 'CV-2' for SGBM with the improved control variates.
[c]The standard error of the path estimator.
[d]The time for generating the sample of the path estimator. Its unit is seconds.

The traditional control variates is free of additional cost and therefore we should treat the traditional control variates as an alternative to the improved control variates. As shown in Table 2, for geometric and arithmetic basket options, the improved control variates is not effective regarding its cost. In these cases, we will use the traditional control variates and increase the sample size for variance reduction.

8.5 *One-dimensional problem*

We start systematic testing under the one-dimensional MJD process. Three different types of jumps are considered as follows: common jump, intensive jump and rare jump. Their model parameters are, respectively, Set I(a), Set I(b) and Set I(c) in Table 1. The basis functions in SGBM are chosen as $1, \log(S_t), \log(S_t)^2, \log(S_t)^3$. We choose improved control variate for the one-dimensional case.

In Figure 8, we see that the path estimator is always an accurate lower bound estimate to the true option price: its standard error is small and it is consistently smaller than the true option price. The direct estimator also converges to the true option value as the number of bundles grows. The convergence is rapid for the MJD process with common jump and for that with intensive jump. In the rare jump case although the convergence is not satisfactory, the exercise strategy associated with the direct estimator is accurate since the relevant path estimator is very close to the true value. The estimation error of the direct estimator is the curse of rare events. For the MJD model with the rare jump, the sample distribution of the paths within one bundle is quite likely to be biased to their analytic distribution and consequently SGBM may be inaccurate according to our discussion in Section 5.

Using improved control variates provides efficient variance reduction in the one-dimensional case. Even though the path estimator has a sample size of only one-tenth of the direct estimator, the standard error of the path estimator is smaller. Moreover, even when the number of bundles is small, the path estimator is much closer to the true value than the direct estimator.

8.6 *Two-dimensional problem*

In this section, we test SGBM on three different types of two-dimensional options. We fix the basis functions as $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$ and perform the bundling based on the bundling references recommended earlier: for the geometric basket option and the arithmetic basket option, the paths are bundled according to the option's intrinsic value;

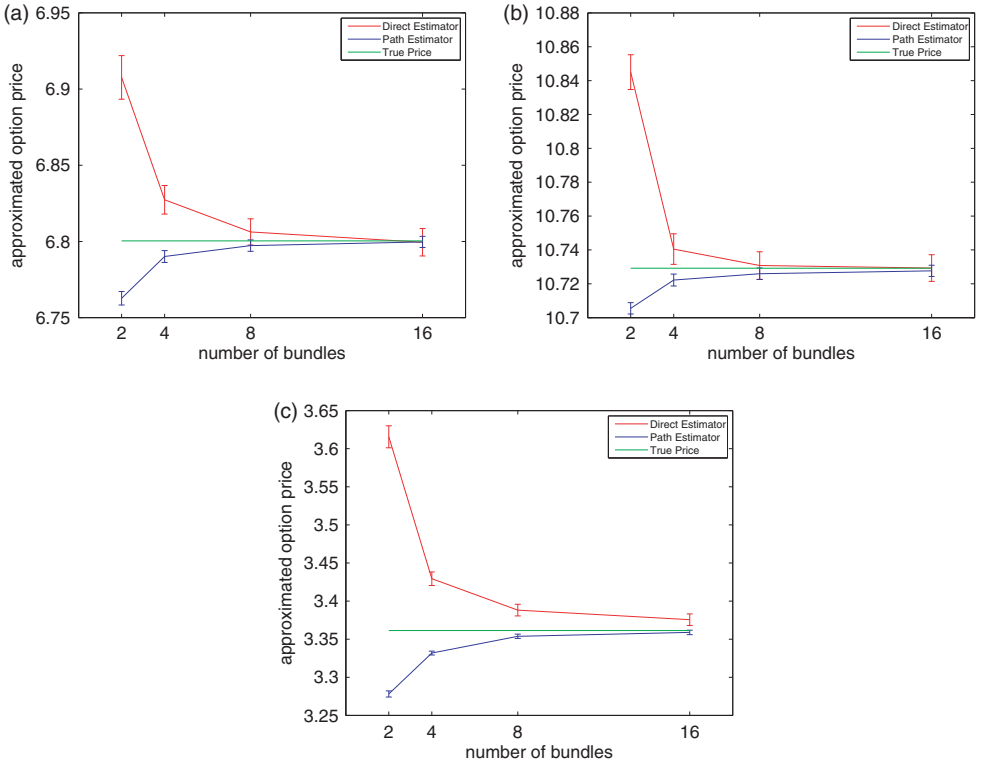

Figure 8. SGBM with the improved control variates is implemented for pricing options with underlying assets following three different one-dimensional MJD processes. The number of the paths for estimating direct estimator is 200,000, while the number of the paths for estimating path estimator is only 20,000. The reference price is generated using the COS method [9]: (a) common jump, (b) intensive jump and (c) rare jump.

Table 3. Test of SGBM for two-dimensional max-on-call options.

$\boldsymbol{S}_0$	Reference upper bound[a] (s.e.)	Reference lower bound (s.e.)	SGBM[b] upper bound[c] (s.e.)	SGBM lower bound (s.e.)	Reference value[d]
(90,90)	8.105	8.067	8.075	8.072	8.075
	(0.086)	(0.020)	(0.011)	(0.008)	
(100,100)	13.906	13.898	13.907	13.897	13.902
	(0.035)	(0.023)	(0.017)	(0.012)	
(110,110)	21.339	21.338	21.352	21.338	21.345
	(0.023)	(0.022)	(0.022)	(0.015)	

[a]This upper bound is provided in [14], where duality approach is applied to generate this.
[b]The sample size for the direct estimator of SGBM is 2^{17} and that for the path estimator is 2^{16}. The single asset European option values measured at the exercise time are used as the control variates for the path estimator. The bundling done in 'SGBM' is based on two references: the maximum of assets' prices and the difference between assets' prices. According to each bundling reference, 16 bundles are constructed. This leads to 256 bundles in total. The basis functions are as follows: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$.
[c]This upper bound is just the direct estimator of SGBM.
[d]The reference value is obtained from [14].

for the min option, the paths are bundled based on the option's intrinsic value and the difference between asset prices.

Besides the test on the MJD model, we also consider pricing the max-on-call option with assets following two-dimensional geometric Brownian motion. We compare our results to those of the same test in [14]. The model parameters are presented in Set III of Table 1. The test results are shown in Table 3. We find that with proper choice of bundling reference SGBM performs

Table 4. Comparing SGBM and LSM for pricing three different types of options for the two-dimensional MJD model.

	SGBM			LSM[a]		
	DE (s.e.)	PE (s.e.)	Computation time[b] (RNG time[c])	Option value (s.e.)	Computation time (RNG time)	Reference value
Geometric basket option	3.6747 (0.0055)	3.6686 (0.0053)	8.9260 (34.3233)	3.6682 (0.0063)	2.4796 (46.4791)	3.6693
Arithmetic basket option	3.3904 (0.0056)	3.3812 (0.0055)	8.8582 (34.3233)	3.3813 (0.0066)	1.9655 (46.4791)	3.3825
Put-on-min option	9.5960 (0.0098)	9.5404 (0.0181)	22.0341 (18.1747)	9.5075 (0.0158)	2.2155 (46.4791)	9.5526

[a]The sample size for LSM is 2^{19}.
[b]The time for backward recursive calculation. The unit is second.
[c]The time for simulating paths. The unit is second.

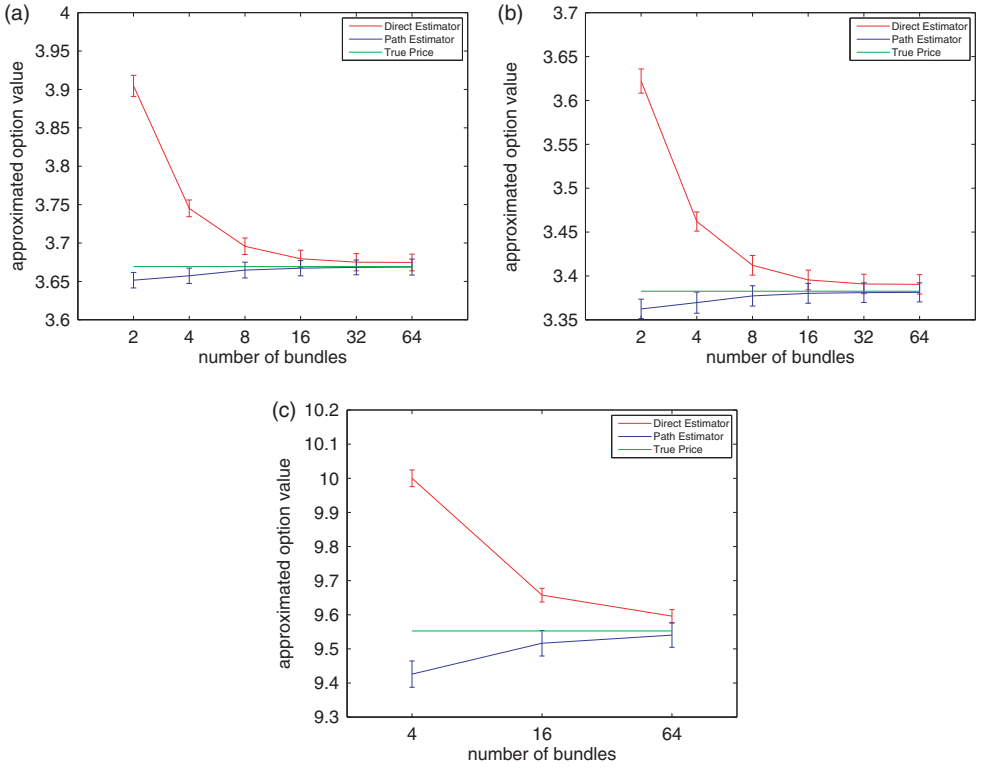

Figure 9. SGBM with control variates for pricing three different types of options with assets following two-dimensional MJD process. The sample size for the direct estimator is always fixed as 2^{17}. For the geometric basket option and the arithmetic basket option, we cast the traditional control variates on the path estimator with sample size 2^{18}. For the put-on-min option, the path estimator with sample size 2^{16} is controlled by the improved control variates. We always consider two controls equal to the European option value of the single asset. The reference option value for the put-on-min option is acquired from [20]: (a) geometric basket option, (b) arithmetic basket option and (c) put-on-min option.

highly satisfactorily for pricing max options even though the maximum of the asset values is not included in the basis functions.

In Table 4, we compare the results of SGBM, presented in Figure 9, to those of LSM. The parameters of the MJD model are chosen from Set II in Table 1. The estimated option values using 64 bundles are chosen to stand for the reference results of SGBM. We see that for the mean

Table 5. SGBM for pricing five-dimensional geometric basket options.

K	Reference price[a]	Algorithm	Direct est. (s.e.)	Path est. (s.e.)	Path est. with CV[b] (s.e.)	Computation time[c] (seconds)
90	0.5564	SGBM-A[d]	0.5567(0.0010)	0.5572 (0.0033)	0.5575 (0.0029)	12.4011
		SGBM-B	0.5588(0.0026)	0.5564 (0.0035)	0.5567 (0.0030)	9.3775
		LSM		0.5563 (0.0031)		1.4305
100	3.1231	SGBM-A	3.1233(0.0037)	3.1220 (0.0097)	3.1226 (0.0071)	12.2884
		SGBM-B	3.1228(0.0052)	3.1198 (0.0096)	3.1204 (0.0073)	9.4389
		LSM		3.1238 (0.0063)		1.8754
110	9.8020	SGBM-A	9.8025(0.0075)	9.8014 (0.0108)	9.8018 (0.0101)	12.2220
		SGBM-B	9.8055(0.0080)	9.7986 (0.0104)	9.7990 (0.0102)	9.3486
		LSM		9.8045 (0.0103)		2.4182

Notes: The parameters of the model are chosen from Set IV. The sample sizes for both the direct estimator and the path estimator in SGBM are 2^{17}. The sample size for LSM is 2^{18}. Sixty-four bundles are constructed in SGBM with the intrinsic value of option as the bundling reference.

[a]The reference price is generated by using the technique introduced in Section 7.3 to reduce the high-dimensional problem to one dimension and pricing the one-dimensional option by the COS method.

[b]The traditional control variates are used here.

[c]The computation time includes the time to compute the direct estimator and the path estimator. However, it does not cover the simulation time, which is around 45 seconds.

[d]'SGBM-A' stands for SGBM with Basis A and 'SGBM-B' for SGBM with Basis B.

basket options the result of LSM is similar to the path estimator of SGBM. However, for the put-on-min option, SGBM gives a better estimate than LSM.

Remark 8.1 In Figure 9(c), we see that the direct estimator of SGBM is not satisfactory. The main cause for this bias is the volatility of the jump size. High volatility of the jump size implies difficulty of having unbiased samples. The impact of the jump intensity is similar but much smaller than that of the volatility of jump size.

The direct estimator has higher bias than the path estimator. This observation is consistent to the conclusion in [21], as the direct estimator can be viewed as the 'value function approximation' and the path estimator as the 'stopping time approximation' [21].

8.7 *Five-dimensional problem*

According to our two-dimensional tests, LSM does not perform well at pricing min options with jump assets. For the five-dimensional case, since there is no reliable reference price for the min or max option with jump assets, we restrict our discussion to the geometric and the arithmetic basket options. In Table 5, we compare LSM with SGBM for pricing five-dimensional geometric basket options. In-the-money, at-the-money and out-of-the-money options are included and two different types of basis functions are investigated:

- Basis A: $\phi_1(\boldsymbol{S}_t) = 1$, $\phi_{k+1}(\boldsymbol{S}_t) = (\prod_{i=1}^{5} S_t^i)^{k/5}$ $k = 1, 2, 3, 4$;
- Basis B: $\phi_1(\boldsymbol{S}_t) = 1$, $\phi_{2i}(\boldsymbol{S}_t) = \log(S_t^i)$, $\phi_{2i+1}(\boldsymbol{S}_t) = \log(S_t^i)^2$ $i = 1, 2, 3, 4, 5$.

In Table 5, we can see that Basis A offers slightly better results than Basis B. However, both are close to the reference value.

Table 6 contains the results for pricing five-dimensional arithmetic basket options. For this type of option, since the conditional expectation of the power of its intrinsic value has a complicated form, we will not consider them as basis functions. According to our test, SGBM with Basis B still performs well with a small difference between the direct and the path estimator, which means that the true option value is located in an interval with sharp bounds.

Table 6. SGBM for pricing five-dimensional arithmetic basket options.

K	Algorithm	Direct est. (s.e.)	Path est. (s.e.)	Path est. with CV[a] (s.e.)	Computation time[b] (seconds)
90	SGBM-B	0.3516(0.0017)	0.3504 (0.0030)	0.3506 (0.0025)	9.0072
	LSM		0.3508 (0.0024)		0.8488
100	SGBM-B	2.5783(0.0048)	2.5745 (0.0078)	2.5748 (0.0060)	9.3762
	LSM		2.5795 (0.0059)		1.1445
110	SGBM-B	9.4683(0.0082)	9.4604 (0.0101)	9.4606 (0.0100)	9.0477
	LSM		9.4675 (0.0091)		2.4182

Notes: The parameters of the model are chosen from Set IV. The sample sizes for both the direct estimator and the path estimator in SGBM are 2^{17}. The sample size for LSM is 2^{18}. Sixty-four bundles are constructed in SGBM with the intrinsic value of option as the bundling reference.

[a]The traditional control variates are used here.

[b]The computation time includes the time to compute the direct estimator and the path estimator. However, it does not cover the simulation time, which is around 45 seconds.

9. Conclusion

We have discussed the SGBM, which is a hybrid of regression-based and bundling-based Monte Carlo methods. SGBM was compared to the SRM and its configuration is thoroughly discussed, including how to choose basis functions for regression and how to partition the bundles. We conducted error analysis on the regression-based pricing methods, especially focusing on the features of SGBM. Traditional and improved control variate methods were introduced for variance reduction in SGBM. Numerical examples on the MJD model were presented for problems up to five dimensions.

Bundling has a significant impact on the accuracy of SGBM. For the arithmetic and geometric basket options, it is sufficient to choose the intrinsic value of the option as the bundling reference, but for 'min' or 'max' options introducing more than one bundling reference is preferred. Control variates work well for reducing the variance of the path estimator in SGBM. In the one-dimensional case, using an improved control variate is highly efficient. When the dimension of problem grows, the cost for implementing the improved control variates increases while its effect decreases. As a result, we favour the traditional control variates in the high-dimensional case.

We have shown that it is sufficient to choose the basis functions in SGBM as polynomials to get convergent results. The outcome of our tests suggests that sometimes it is not necessary to include all terms as the basis functions and in some situations choosing the basis functions determined by the type of option contract could be more effective. When the dimensionality of the problem is not large, it appears feasible to use SGBM with basis functions of polynomial type. According to our experience, choosing polynomials without cross-product terms as basis functions works even for 10-dimensional problems. However, when the dimensionality of problem surges up, we need alternatives to polynomials as basis functions to release ourselves from the corresponding demand for the huge sample size. As mentioned in [14], polynomials of the intrinsic value of option are promising choices in this scenario.

Notes

1. It denotes the realization at time T with the values of the option's underlying assets equal to $\boldsymbol{S}_T$.
2. We write $\boldsymbol{S}_{t_m}$, which means that the stock price at time t_m is equal to $\boldsymbol{S}_{t_m}$. In the following discussions, the condition of the expectation may also be formulated as $\boldsymbol{S}_{t_m} = \hat{\boldsymbol{S}}$ to emphasize that the stock price at time t_m is known as a realization $\hat{\boldsymbol{S}}$.
3. The authors of [3] show that Regress-Later is fundamentally different from Regress-Now, noticing that the former does not introduce a projection error between two time steps in the regression stage. As a result, Regress-Later achieves a faster convergence rate than Regress-Now in terms of the sample size.

4. If we want to bundle the paths into two parts, the partition point for 'equal-size bundling' is the median of the asset prices while that for 'equal-range bundling' is the mean of the asset prices.
5. For simplicity, we neglect the discounting term $D_{t_{m-1}}$.
6. Since $\pi(\boldsymbol{S}_{t_m})$ is defined by the density function of $\boldsymbol{S}_{t_m}$ conditioned on $\boldsymbol{S}_{t_{m-1}} = \hat{\boldsymbol{S}}$, the simulation of $\boldsymbol{S}_{t_m}$ with respect to this density function can be treated as a sub-simulation from the unique state $\boldsymbol{S}_{t_{m-1}} = \hat{\boldsymbol{S}}$.
7. In the following sections, the other empirical density functions can be defined in a similar fashion.
8. SGBM is also feasible for another well-known jump-diffusion model, the Kou model [16]. Since the only distinction between the MJD model and the Kou model is the distribution of jump sizes, all discussions in this paper about SGBM can be extended to the Kou model as well.
9. The jumps in the dynamics of each asset should, however, follow the same Poisson process.
10. In fact, the number of basis functions is also a tuning parameter, but large number of basis functions may result in an over-fitting problem in the regression step. We therefore choose the number of basis functions just three or four in the one-dimensional case.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] S. Asmussen and P.W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, Vol. 57, Springer, New York, 2007.
[2] J. Barraquand and D. Martineau, *Numerical valuation of high dimensional multivariate American securities*, J. Financ. Quant. Anal. 30 (1995), pp. 383–405.
[3] E. Beutner, J. Schweizer, and A. Pelsser, *Fast convergence of Regress-Later estimates in least squares Monte Carlo*, preprint (2013). Available at arXiv:1309.5274.
[4] P. Boyle, M. Broadie, and P. Glasserman, *Monte carlo methods for security pricing*, J. Econ. Dyn. Control 21 (1997), pp. 1267–1321.
[5] M. Broadie and P. Glasserman, *Pricing American-style securities using simulation*, J. Econ. Dyn. Control 21 (1997), pp. 1323–1352.
[6] M. Broadie and P. Glasserman, *A stochastic mesh method for pricing high-dimensional American options*, J. Comput. Finance 7 (2004), pp. 35–72.
[7] J.F. Carriere, *Valuation of the early-exercise price for options using simulations and nonparametric regression*, Insur. Math. Econ. 19 (1996), pp. 19–30.
[8] C.E. Clark, *The greatest of a finite set of random variables*, Oper. Res. 9 (1961), pp. 145–162.
[9] F. Fang and C.W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput. 31 (2008), pp. 826–848.
[10] M.C. Fu, S.B. Laprise, D.B. Madan, Y. Su, and R. Wu, *Pricing American options: A comparison of Monte Carlo simulation approaches*, J. Comput. Finance. 4 (2001), pp. 39–88.
[11] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Vol. 53, Springer, New York, 2004.
[12] P. Glasserman and B. Yu, *Number of paths versus number of basis functions in American option pricing*, Ann. Appl. Probab. 14 (2004), pp. 2090–2119.
[13] P. Glasserman and B. Yu, *Simulation for American Options: Regression Now or Regression Later? Monte Carlo and Quasi-Monte Carlo Methods 2002*, Springer, Berlin Heidelberg, 2004, pp. 213–226.
[14] S. Jain and C.W. Oosterlee, *The stochastic grid bundling method: Efficient pricing of bermudan options and their greeks*, preprint (2013). Available at SSRN 2293942.
[15] P. Karlsson, S. Jain, and C.W. Oosterlee, *Counterparty credit exposures for interest rate derivatives using the stochastic grid bundling method*, preprint (2014). Available at SSRN 2538173.
[16] S.G. Kou, *A jump-diffusion model for option pricing*, Manage. Sci. 48 (2002), pp. 1086–1101.
[17] F.A. Longstaff and E.S. Schwartz, *Valuing American options by simulation: A simple least-squares approach*, Rev. Financial Stud. 14 (2001), pp. 113–147.
[18] R.C. Merton, *Option pricing when underlying stock returns are discontinuous*, J. Financ. Econ. 3 (1976), pp. 125–144.
[19] N.S. Rasmussen, *Control variates for Monte Carlo valuation of American options*, J. Comput. Finance 9 (2005), pp. 83–118.
[20] M.J. Ruijter and C.W. Oosterlee, *Two-dimensional Fourier cosine series expansion method for pricing financial options*, SIAM J. Sci. Comput. 34 (2012), pp. B642–B671.
[21] L. Stentoft, *Value function approximation or stopping time approximation: a comparison of two recent numerical methods for American option pricing using simulation and regression*, J. Comput. Finance 18 (2014), pp. 1–56.
[22] J.A. Tilley, *Valuing American options in a path simulation model*, Trans. Soc. Actuaries 45 (1993), pp. 55–67.
[23] J.N. Tsitsiklis and B. Van Roy, *Regression methods for pricing complex American-style options*, IEEE Trans. Neural Netw. 12 (2001), pp. 694–703.