Bermudan options by simulation

L. C. G. Rogers

August 25, 2015

Abstract

The aim of this study is to devise numerical methods for dealing with very high-dimensional Bermudan-style derivatives. For such problems, we quickly see that we can at best hope for price bounds, and we can only use a simulation approach. We use the approach of Barraquand & Martineau [2] which proposes that the reward process should be treated as if it were Markovian, and then uses this to generate a stopping rule and hence a lower bound on the price. Using the dual approach introduced by Rogers [13] and Haugh & Kogan [7], this approximate Markov process leads us to hedging strategies, and upper bounds on the price. The methodology is generic, and is illustrated on eight examples of varying levels of difficulty. Run times are largely insensitive to dimension.

1 Introduction.

A general optimal stopping problem can be formulated as finding

$$\sup_{0 \le \tau \le T} E[\, Z_\tau \,] \tag{1.1}$$

where the termination time T is a fixed positive real, the process Z is a given process adapted to some filtration $(\mathcal{F}_t)_{t \ge 0}$, and τ is constrained to be an $(\mathcal{F}_t)$-stopping time. We shall in this paper assume that the reward process Z has the form

$$Z_t = g(t, X_t), \tag{1.2}$$

where X is some Markov process, and g is some measurable function of time and X_t. For what mainly concerns this paper - the analysis of Bermudan options - this formulation is already sufficiently general. Much of what follows is valid more widely, but at times we shall refer to properties that relate to a typical financial context; the aim is to provide bounds, and we need context in order to assess the effectiveness of these bounds.

Of course, it is well known how to solve an optimal stopping problem of the form given by (1.1), (1.2); we define the value function

$$V^*(t,x) = \sup_{t \leq \tau \leq T} E[\, g(\tau, X_\tau) \,|\, X_t = x \,] \tag{1.3}$$

and find V^* by dynamic programming[1]. When this can be done, this is everything we could ask for. However, explicitly-soluble examples are rare, and we soon have to reach beyond them. For example, the standard Bermudan put option, where

$$g(t,x) = e^{-rt}(K - \exp(x))^+ \tag{1.4}$$

and the process X is a Brownian motion, is a celebrated example where no closed-form solution is known. This has launched many studies in the last 25 years, and very efficient numerical schemes have been devised. But what if we have some vector $X_t = (X_t^1, \ldots, X_t^d)$ of correlated Brownian motions, and the reward function is

$$g(t,x) = e^{-rt}(\, K - \exp(\, \min_i x^i \,)\,)^+ \quad , \tag{1.5}$$

a so-called *min put*? If we look for the value function, we have to determine numerically[2] some function of d variables. The standard Bellman equation approach requires us to calculate recursively $V_t^*(\cdot) \equiv V^*(t, \cdot)$ as

$$V_t^*(x) = \max\{\; g(t,x),\; E[V_{t+1}^*(X_{t+1}) | X_t = x] \;\} \;; \tag{1.6}$$

but if d is large (fifty, say), how is V_{t+1}^* to be stored? How is the expectation on the right-hand side of (1.6) to be calculated or approximated? The more one thinks about these issues, the clearer it becomes that *calculating an approximation to the value function can only work in dimensions that are not too high, and will most likely rely heavily on the structure of the problem under study.* In other words, *any methodology that attempts to identify the value function will be of restricted applicability.*

So we must be content with less; but less may be enough. If we had calculated the value function, what use would we make of it? We would use it to *determine the optimal control*: at each time t, we would see the state

[1] For a Bermudan option, which has only finitely many possible times of exercise, the optimization is over a discrete set of times, though we generally think that the time of the underlying process runs continuously.

[2] If we can only solve the one-asset problem numerically, we can certainly only solve the d-asset problem numerically.

$X_t = x$ of the system, and if $V_t^*(x) > g(t, x)$ we would continue, otherwise we would stop. We would use it to determine a *fair price* to pay or ask for the derivative at time 0. We would use it to *delta-hedge* the derivative once it had been sold. The view taken in this paper is that the analysis of a Bermudan option requires just these:

- At each time, whatever the state, we are able to decide whether or not to stop;
- At each time, we are able to propose a hedge for the next period;
- We are able to provide reasonably close bounds for the price of the derivative at all times.

Exact knowledge of the value function would achieve all of these objectives, but can we attain them without knowing the value function or an approximation to the value function? The message of this paper is that this **can** be done. We present an approach to Bermudan options with the properties:

- The *only* information required about the Markov process is the ability to simulate a step of the process;
- The methodology is *generic* - the same code is used to do the calculations for all examples, changing only the specification of the Markov process X and the stopping reward g;
- Computational cost is *largely insensitive to dimension*, so a derivative written on hundreds of underlyings can be solved simply by changing the dimension parameter in the code, and takes only a little longer to run;
- Upper and lower bounds on the price differ by typically 5-10 percent (sometimes more, sometimes less);
- The stopping rules and hedging rules obtained are very simple to calculate and implement.
- Computational times depend on the problem, but a few tens of seconds usually suffices.

The claim is that the method offered here is an effective general method for dealing with any Bermudan option. In truth, the components used are already known in one form or another, and what is added here is the judicious combination of them, and a redefinition of the questions to be answered. The waypoints are the following:

- Any numerical scheme has to be a finite calculation, so the Markov process has to be coerced to a finite set of values;
- The finite coercion of the underlying Markov process X has to be tailored to the stopping reward - using the approach of Barraquand & Martineau [2];
- The finite coercion generates a stopping rule, and a hedging rule, using the dual approach of [13], [7], [1] which can then be evaluated by simulation.

The particular structure of a problem may suggest variants that improve on the performance, but in the examples studied any improvement from simple variants is not large.

2 The general methodology.

The general situation concerns a Markov process X taking values in a statespace $\mathcal{X}$, a stopping reward function g, and a finite set $\mathbb{T} \subseteq [0, T]$ of size N_T of possible times to stop[3]. The aim is to associate this problem with an optimal stopping problem for a (discrete-time) Markov process with a *finite* statespace, and the way this is done is simply a paraphrase of the method of Barraquand & Martineau [2].

Definition 2.1. *A (real-valued)* Markovian coercion *of X is specified by a measurable function $g : \mathbb{T} \times \mathcal{X} \mapsto \mathbb{R}$, an $N_T \times N$ matrix η of* bin values *, an $N_T \times (N-1)$ matrix y of* bin edges*, and an $(N_T - 1) \times N \times N$ array P with the properties:*

1. *for each $t \in \mathbb{T}$,*

$$\eta_t^{(1)} < y_t^{(1)} < \eta_t^{(2)} < y_t^{(2)} < \ldots < y_t^{(N-1)} < \eta_t^{(N)};$$

2. *for each $t \in \mathbb{T} \backslash \{T\}$, $P(t, \cdot, \cdot)$ is an $N \times N$ transition matrix.*

We can approximate the real-valued process $g(t, X_t)$, $t \in \mathbb{T}$, by using the matrix y of bin edges to define a partition of $\mathbb{R}$ for each $t \in \mathbb{T}$:

$$J_t^{(1)} = (-\infty, y_t^{(1)}] \, , \ldots , \; J_t^{(N)} = (y_t^{(N-1)}, \infty) \qquad (2.1)$$

[3] We will always assume that 0 and T are in $\mathbb{T}$.

and then the matrix η of bin values to define the approximating process

$$Y_t = \sum_{k=1}^{N} \eta_t^{(k)} \; I_{\{g(t,X_t) \in J_t^{(k)}\}} \; . \tag{2.2}$$

For the current purpose, we shall take g to be the function appearing in the definition (1.2) of the stopping reward.

The process Y takes only finitely many values, but will *not* in general be Markov. Nevertheless, the essential idea of the Barraquand-Martineau approach is that we **pretend that it is**, with transition probabilities given by the array P. We then solve the optimal stopping problem for this Markov coercion, and use the solution found to propose exercise and hedging strategies for X.

2.1 Implementation.

For the implementation, we shall assume that the state process X takes values in some (subset of) euclidean space $\mathbb{R}^d$. The dimension d of this space is unrestricted, and can be quite large. The numerical implementation consists of four stages:

1. use simulation to initialize, calculating the transition matrix array P, and the bin edges and values;

2. calculate the value and optimal stopping rule for the (presumed Markovian) process Y;

3. find a lower bound by evaluating the performance of the stopping rule from step 2 when applied to stopping the process X;

4. find an upper bound by evaluating the hedging rule derived from step 2 when applied to the actual process X.

We now give some more detail on each of these stages in turn.

Initialization. First choose some number N_{bins} of bins, and some number b of points in each 'half bin', so that in total we will simulate $N_{\text{sim}} = 2bN_{\text{bins}}$ sample paths. The simulation will fill up a $N_T \times d \times N_{\text{sim}}$ array, though since the process is Markovian, we can economize on space by just carrying along the current values, in effect a $d \times N_{\text{sim}}$ array, which gets overwritten with newly-calculated values.

So suppose that we have the $d \times N_{\rm sim}$ array x of the values[4] at time $t_i \in \mathbb{T}$. We now use the simulation of the Markov process to step these values forward to the next time t_{i+1}, creating a $d \times N_{\rm sim}$ array x'. Next apply the function g to give

$$y_j \equiv g(t_i, x[:, j]), \qquad y'_j \equiv g(t_{i+1}, x[:, j]) \qquad (j = 1, \ldots, N_{\rm sim}).$$

If we define $y_{(1)} < y_{(2)} < \ldots y_{(N_{\rm sim})}$ to be the sample (y_j) in increasing order, we define the bin edges at time t_i to be the values $y_{(2kb)}$, $k = 1, \ldots, N_{\rm bins} - 1$, and the bin values at time t_i to be the values $y_{(2kb-b)}$, $k = 1, \ldots, N_{\rm bins}$. We similarly calculate[5] the bin values and edges at time t_{i+1}. Now for each simulated path j we can see which bin that path was in at time t_i, and which bin that path moved into at time t_{i+1}; counting the number of paths which moved from bin ℓ at time t_i into bin m at time t_{i+1} gives us an estimate of the transition probability $P[i, \ell, m]$.

It is worth remarking that this step differs slightly from what Barraquand & Martineau do; we let the data tell us where the bin edges should be, and Barraquand & Martineau set the bin edges before any simulation takes place, as an *a priori* modelling choice.

Calculating the value function and optimal stopping rule. Now that we have made a Markov chain proxy which jumps at discrete times from one bin to another according to the transition probabilities stored in P, the calculation of the value V and stopping rule $\mathcal{S}$ is done by dynamic programming. The arrays V and $\mathcal{S}$ are both $N_T \times N_{\rm bins}$; the values in $\mathcal{S}$ are either 0 or 1, where $\mathcal{S}[i, k] = 1$ signifies that we should stop if at time t_i the Y-value is in bin k. The time taken to compute V and $\mathcal{S}$ is negligible. As a notational convenience, for $y \in \mathbb{R}$ we shall write

$$V(t, y) = V[i, k], \qquad \mathcal{S}(t, y) = \mathcal{S}[i, k] \tag{2.3}$$

when $t = t_i \in \mathbb{T}$ and $y \in J_t^{(k)}$, the kth bin at time t_i.

Lower bound. The computed stopping rule $\mathcal{S}$ is now used to provide a lower bound for the value of the option. We simply simulate a large number of paths of the process X, and for each path we stop the first time t_i for

[4] In accordance with Python notation, we denote the (i, j) element of an array z by $z[i, j]$, and the jth column of z by $z[:, j]$.

[5] We implicitly assume that the y-values are distinct. This is clearly not going to happen for (say) a put option, but we make this happen by replacing the reward $\max\{0, K - S\}$ by the reward $\max\{\varepsilon(K - S), (K - S)\}$ for some small ε. The error committed will be small compared to other errors. Likewise, we do not bother to locate the bin edges in between values of $y_{(j)}$ as perhaps we ought.

which $\mathcal{S}(t_i, g(t_i, X_{t_i})) = 1$. For each path, this gives a stopping value, and the lower bound is just the average over all paths.

Upper bound. To derive an upper bound, we need to recall some results about the dual approach from [13], [7], where it is shown that the value (1.1) of the optimal stopping problem has the alternative characterization

$$\sup_{0 \le \tau \le T} E[\, Z_\tau \,] = \min_{M \in \mathcal{M}_0} E[\, \sup_{0 \le t \le T} (Z_t - M_t) \,], \tag{2.4}$$

as a minimum over the space $\mathcal{M}_0$ of martingales vanishing at 0. The minimum on the right-hand side is attained when M is the martingale part of the Snell envelope process: see [13]. In this setting, the Snell envelope process is simply the value function $V^*(t, X_t)$ evaluated along the path, so the optimal martingale difference sequence would be

$$M^*_{t_{i+1}} - M^*_{t_i} = V^*(t_{i+1}, X_{t_{i+1}}) - E[V^*(t_{i+1}, X_{t_{i+1}}) \mid \mathcal{F}_{t_i}]. \tag{2.5}$$

Of course, we do not know V^*, but we have calculated and stored in the array V some approximation to V^*, so a natural approximation to M^* would be found by taking the martingale difference sequence

$$M_{t_{i+1}} - M_{t_i} = V(t_{i+1}, Y_{t_{i+1}}) - E[V(t_{i+1}, Y_{t_{i+1}}) \mid \mathcal{F}_{t_i}], \tag{2.6}$$

where Y is as defined at (2.2). The only issue with this is how we are to evaluate the conditional expectation on the right-hand side of (2.6). The function $V(t_{i+1}, \cdot)$ is a simple function, taking only N_{bins} values, but if we have simulated a sample path and we see $X_{t_i} = x$, how are we to calculate (or approximate) $E[V(t_{i+1}, Y_{t_{i+1}}) \mid X_{t_i} = x\ ]$? The approach adopted is to perform a subsimulation of some N_{sub} values of $X_{t_{i+1}}$ starting from $X_{t_i} = x$. It is generally considered a bad idea to perform subsimulations, because this will usually take a lot of time, and may not be very accurate, but in this application the approach is effective *because $Y_{t_{i+1}}$ is scalar.* The importance of this is that our subsimulations do not need to search out some high-dimensional space, they only need to search out the real line; in practice, if the payoff g is continuous, and the time-step not too large, most values of $g(t_{i+1}, X_{t_{i+1}})$ will be fairly close to $g(t_i, X_{t_i})$, so a relatively small number of subsimulations will suffice. In the examples reported later, we mainly used $N_{\text{sub}} = 25$, on 4000 simulated paths.

This explains how we construct candidate hedging martingales from the array V. In a financial context, we would want to be able to express such martingales in terms of traded assets; this could be done by calculating delta-hedges for the martingale differences arising from the approximation, but since this would be application-specific, we prefer not to go into the detail of how this would be done. The approach offered merely indicates a martingale to be used for hedging, not how exactly this is to be synthesised from marketed assets.

3 Examples.

Here we present a range of examples to illustrate the methodology, some familiar from the literature, others not. In the following examples, we will be considering a d-vector of log-Brownian assets whose prices S_t^i at time t evolve as

$$dS_t^i = S_t^i \left(\sum_{j=1}^{d} \sigma_{ij} \, dW_t^j + \mu_i \, dt \right), \tag{3.1}$$

where W is a d-dimensional Brownian motion, the σ_{ij} and μ_i are previsible processes which are assumed *constant*, except in Example (3.8). We write

$$\Sigma \equiv \sigma \sigma^T, \tag{3.2}$$

a positive-definite symmetric matrix. Since the focus is on derivative pricing, we shall assume that we are working in the risk-neutral measure, which amounts to the condition

$$\mu_i = r - \tfrac{1}{2} \Sigma_{ii}, \tag{3.3}$$

where r is the riskless rate of interest, assumed *constant*, except in Example (3.8).

It proves convenient in most of the examples to simulate the discounted log prices

$$x_t^i \equiv - \int_0^t r_s \, ds + \log S_t^i = \sum_j \sigma_{ij} W_t^j - \tfrac{1}{2} \Sigma_{ii} \, t + \log S_0^i. \tag{3.4}$$

Variants. If N is a positive martingale, $N_0 = 1$, then

$$\begin{aligned} \sup_{0 \le \tau \le T} E\Big[\, g(\tau, X_\tau) \, \Big] &= \sup_{0 \le \tau \le T} E\left[N_\tau \, \frac{g(\tau, X_\tau)}{N_\tau} \right] \\ &= \sup_{0 \le \tau \le T} E\left[N_T \, \frac{g(\tau, X_\tau)}{N_\tau} \right] \\ &= \sup_{0 \le \tau \le T} \tilde{E}\left[\frac{g(\tau, X_\tau)}{N_\tau} \right], \end{aligned}$$

where

$$\frac{d\tilde{P}}{dP} = N_T. \tag{3.5}$$

If the martingale N can be expressed as $N_t = \psi(t, X_t)$, then we can work in the new measure $\tilde{P}$ with the new reward $\tilde{g}(t,x) = g(t,x)/\psi(t,x)$, and it may be convenient sometimes to do this. This change of numeraire approach is reminiscent of Jamshidian's [8] version of dual American option valuation. We could similarly transform the stopping reward $g(t, X_t)$ to $g(t, X_t) - \varphi(t, X_t) + \varphi(0, X_0)$, where φ is some function for which $\varphi(t, X_t)$ is a martingale, for example, the value of the European option.

3.1 Min put.

In this example, the state variable is $X_t \equiv x_t$, the vector of discounted log prices, and the reward function for stopping at time t will be

$$g(t, X_t) = \left(Ke^{-rt} - \exp(\min_{1 \le i \le d} X_t^i) \right)^+. \tag{3.6}$$

This example was studied in [13], and the figures in the column MC price of Table 1 were taken from that paper; for d = 30, 60 no values are given in [13]. The columns headed low and high are sample means obtained from a simulation method, so there is simulation error in the values reported, but this error is quite small in relation to the gap.

d	low	MC price	high	gap(%)	time
2	24.77	25.16	26.63	6.98	5.00
3	31.26	31.76	33.00	5.40	5.74
4	35.57	36.28	37.48	5.08	6.82
5	39.11	39.47	40.22	2.74	7.66
10	47.95	48.33	49.06	2.26	12.42
15	51.97	52.14	52.86	1.68	17.34
30	57.66	-	58.23	0.98	35.86
60	62.10	-	62.62	0.82	68.32

Table 1: Min put prices. The d assets are independent, $S^i(0) = 100$. Other parameters are $K = 100$, $r = 0.06$, $T = 0.5$, $\sigma_{ii} = 0.6$.

3.2 Max call.

As with the min put, the state variable is $X_t \equiv x_t$, the vector of discounted log prices, but this time the reward function for stopping at time t will be

$$g(t, X_t) = \big(\exp(\max_{1 \le i \le d} X_t^i) \ - K e^{-rt} \big)^+. \tag{3.7}$$

In order to make the problem interesting, the assets will be assumed to pay dividends at a constant rate. This example was studied by Broadie & Glasserman[6] [5], and has been used as a test example in a number of other studies, including [6], [7], [1], [9]. The results shown in Table 2 are calculated using the average of the discounted assets as a numeraire, and took $N_{\text{sub}} = 125$. If we did not use the change of measure, the gaps between the lower and upper bounds are a little higher.

m	S_0	low	BG price	high	gap(%)	time
3	90	15.29	16.006	16.11	5.08	4.21
	100	24.41	25.284	25.35	3.71	4.20
	110	34.74	35.695	35.84	3.07	4.20
6	90	15.52	16.474	16.79	7.59	8.47
	100	24.63	25.92	26.50	7.04	8.57
	110	34.81	36.497	37.23	6.48	8.57
9	90	15.81	16.659	16.95	6.71	12.67
	100	24.78	26.158	26.75	7.35	12.82
	110	35.25	36.782	37.63	6.30	12.66

Table 2: Max call prices on 5 independent assets with common volatility $\sigma = 0.2\%$ and expiry $T = 3$. There are $m = 3, 6, 9$ exercise opportunities at times $iT/m,\ i = 0, \ldots, m$. Other parameters are $K = 100$, $r = 0.05$, $\delta = 0.1$.

3.3 Basket put.

This is an example studied in Kovalov, Linetsky & Marcozzi [10], and subsequently in Jin *et al* [9]. The state variable is again the vector of d discounted

[6] In the preprint version, the example presented in Table 2 is said to give results for common expiry $T = 3$ with 3, 6, 9 equally-spaced exercise opportunities, but the published version gives the same numerical values purportedly for end-of-year exercise opportunities with expiries 3, 6, 9 years. Our numerics show that in fact the problem solved is correctly stated in the preprint, and mis-stated in the published version.

log prices, and this time the stopping reward function is

$$g(t, X_t) = \big(\, Ke^{-rt} - d^{-1} \sum_{i=1}^{d} \exp(X_t^i)\, \big)^+. \tag{3.8}$$

All the stocks start at 100, the strike is $K = 100$, the riskless rate is 0.05, the expiry is $T = 0.25$, and the individual asset volatilities are all 0.2, but this time the assets are not supposed independent; there is constant correlation $\rho = 0.5$ between all the assets[7]. Kovalov *et al* use a numerical PDE approach, Jin *et al* use a simulation methodology, and both approaches appear to give better precision than the method we have used here. Nevertheless, as we shall soon see, the difference in precision is not practically relevant. We used $N_{\text{sub}} = 125$ in this calculation, and report the results in Table 3

d	low	KLM	high	gap(%)	time
2	3.05	3.14	3.14	6.11	27.20
3	2.74	2.94	2.94	6.80	36.66
4	2.64	2.84	2.84	7.03	45.56
5	2.57	2.77	2.77	7.24	53.85
6	2.54	2.72	2.72	6.73	63.01
12	2.43	-	2.61	7.07	131.8

Table 3: Basket put. All assets start at 100, $K = 100$, $T = 0.25$, $r = 0.05$. All assets have volatility 20%, and the correlation between assets is 0.5.

These values were computed assuming that the volatility parameter σ is equal to 20%. But are we sure of that? In any application, the volatility (assumed constant) would have to be estimated; are we really sure that the volatility is not 19%? Or 21%? Are we really sure that the volatility will remain constant at 20% until expiry of the option? Suppose we repeat the calculations of Table 3 for those values of the volatility parameter and see what ranges for the price result. The outcomes are recorded in Table 4, and what we see is that *there is virtually no overlap between the computed intervals for the price for the three values of σ*. In other words, *the uncertainty in the price arising from our simulation bounds is comparable to the uncertainty in price which would arise from the uncertainty in the input parameter values.*

[7] The specification of the various parameters in [10] is internally inconsistent, and not in agreement with the parameters quoted in [9]. For the reported prices to be correct, we find that the parameter values used must be those we have stated.

d	low (19%)	high (19%)	low (20%)	high (20%)	low (21%)	high (21%)
2	2.77	2.97	3.05	3.14	3.10	3.32
3	2.58	2.75	2.74	2.94	2.90	3.10
4	2.49	2.66	2.64	2.84	2.79	3.01
5	2.43	2.62	2.57	2.77	2.73	2.92
6	2.39	2.54	2.54	2.72	2.70	2.88

Table 4: Basket put. Parameters as for Table 3, except the volatility parameter which takes values 19%, 20%, 21%.

3.4 Fixed strike Bermudan-Asian call.

In this example, there is a single asset S, and the reward for stopping at time τ is

$$g(\tau, X_\tau) = e^{-r\tau}(A_\tau - K)^+, \tag{3.9}$$

where we define the average price

$$A_t = \frac{\int_{-\delta}^{t} S_u \, du}{t+\delta}. \tag{3.10}$$

Here, $\delta > 0$ is some initial window required to prevent wild oscillations. There is also some initial lock-out time $t^* \geq 0$ during which exercise of the option is forbidden. The state variable of the problem is $X_t = [S_t, A_t, t]$. The numerical results are reported in Table 5. The price gaps are generally in the range 4-9%, though some rows of the table required more work than others, and sometimes the gaps between the bounds were quite large. Any row where the reported time is 20s (an approximate figure) required runs with several different stories; in fact we used three different numeraires (the bank account, the stock, and the martingale $E_t[A_T]$), and used the given stopping reward, or adjusted the stopping reward by adding the martingale of the European put. The results are usable, but not particularly good, and this is really because this example is intrinsically two-dimensional, so any attempt to coerce it to one dimension is missing something essential. We cannot expect a stopping rule which only looks at A_t to do very well, because the current value of S_t has to be considered as well; if S_t is high enough, the value $g(t, X_t)$ is actually increasing, so we would certainly not stop at such a time. But a rule that only considers A_t would not understand that.

A_0	S_0	low	FD price	high	gap(%)	time
	80	0.919	0.949	0.991	7.33	4.12
	90	3.20	3.267	3.35	4.46	4.15
90	100	7.49	7.889	8.03	6.71	4.15
	110	13.68	14.538	14.89	8.16	4.12
	120	21.01	22.423	22.99	8.61	4.10
	80	1.052	1.108	1.147	8.28	4.13
	90	3.53	3.710	3.82	7.52	4.14
100	100	8.07	8.658	8.83	8.61	20
	110	14.95	15.717	16.22	7.83	20
	120	23.33	23.811	24.42	4.46	20
	80	1.21	1.288	1.33	9.02	20
	90	3.87	4.136	4.35	11.03	11.21
110	100	8.7	9.821	10.66	18.4	20
	110	15.76	17.399	18.34	14.1	20
	120	24.50	25.453	26.32	6.91	20

Table 5: Fixed strike Bermudan Asian call. Parameters are $\sigma = 0.2$, $K = 100$, $t^* = 0.25$, $\delta = 0.25$, $T = 2$.

3.5 Floating strike Bermudan Asian call

The story is very similar to Section 3.4, except that the reward is

$$g(\tau, X_\tau) = e^{-r\tau}(A_\tau - S_\tau)^+ \tag{3.11}$$

for stopping at time τ. This example is in fact much easier than the fixed strike, because the process A_t/S_t is a Markov process already. Scaling says we need only vary S_0 while keeping A_0 fixed, which is what we do. The results obtained when we fix $A_0 = 100$ are given in Table 6.

3.6 Fixed window lookback option.

This example illustrates the capacity of the methodology to handle high-dimensional problems. Here we suppose that the stock price is recorded at times which are multiples of some $h > 0$, and stopping at time $\tau = kh$ delivers reward

$$g(\tau, X_\tau) = \sup_{k-a \leq j \leq k} S_{jh}, \tag{3.12}$$

where a is some positive integer. This time, the state variable X has to record the last a values of the price, since the sup and inf are taken over

S_0	low	high	gap(%)	time
80	10.60	10.86	2.50	8.5
90	8.33	8.67	3.98	8.40
100	7.14	7.50	4.88	8.51
110	6.51	6.85	5.00	8.81
120	6.16	6.57	6.26	8.32

Table 6: Floating strike Bermudan Asian call. Parameters are $\sigma = 0.2$, $A_0 = 100$, $t^* = 0.25$, $\delta = 0.25$, $T = 2$.

a fixed window. The results are presented in Table 7. Notice that for this example the bounds are very close, getting slightly less good as the lookback parameter a rises. Equally noteworthy is the fact that the run times are essentially unchanged as we increase the lookback parameter; so in the final row of the table, the state variable is 25-dimensional. It should not be a surprise that there is so little variation in run times, because increasing a makes no difference to the simulation load; each period, we simulate one new value, all that is different is that we are storing more or fewer values from the past.

a	low	high	gap(%)	time
5	116.07	116.74	0.57	115
10	121.8	123.2	1.18	109
15	125.2	127.2	1.57	110
20	127.7	130.1	1.84	114
25	129.6	132.4	2.17	115

Table 7: Fixed window lookback option. Parameters were $T = 1$, $\sigma = 0.5$, $r = 0.05$, $S_0 = 100$, and the time interval was divided into 250 equal time steps. The calculations were done in the numeraire of the discounted asset price.

3.7 Fixed window range option.

This example is similar to the fixed window lookback option of Section 3.6, except that the reward for stopping at time $\tau = kh$ is

$$g(\tau, X_\tau) = \sup_{k-a\leq j\leq k} S_{jh} - \inf_{k-a\leq j\leq k} S_{jh}, \tag{3.13}$$

where a is some positive integer. The results are reported in Table 8. The gaps between the upper and lower bounds are higher than for the lookback example expressed as a percentage, but the arithmetic gaps are roughly comparable, with similar run times.

a	low	high	gap(%)	time
5	17.06	17.91	4.75	92
10	23.64	25.78	8.30	97
15	27.86	31.26	10.89	103
20	31.09	35.46	12.34	109
25	33.76	38.99	13.41	116

Table 8: Fixed window range option. Parameters were $T = 1$, $\sigma = 0.5$, $r = 0.05$, $S_0 = 100$, and the time interval was divided into 250 equal time steps. The calculations were done in the numeraire of the discounted asset price.

3.8 Min puts with stochastic volatility and interest.

In this example, we consider a situation where there are $d > 1$ assets, with stochastic volatility and interest rates. There are examples of such models in various places in the literature, for example, Medvedev & Scaillet [12], Boyarchenko & Levendorskii [4], Jin *et al.* [9]. Heston dynamics for the asset and the volatility are popular in theoretical work, but it is far from clear that the dependence of the volatility of volatility on level takes the square-root form postulated in the Heston model; and still less is it persuasive that the Cox-Ingersoll-Ross interest-rate model correctly describes the volatility of interest rates when those rates are low. As this is a simulation study, we are freed from any need to choose models that are theoretically tractable[8], so we may make some modelling assumptions that match observed behaviour better.

So our story supposes that there is some market Brownian motion W^M, and that log prices $x_t^i \equiv \log S_t^i$ evolve as

$$dx_t^i = \sigma_t^i \{ \rho_S \, dW_t^M + \rho_S' \, dW_t^{S,i} \} + (r_t - \tfrac{1}{2} (\sigma_t^i)^2) \, dt \qquad (3.14)$$

[8] Tractability is in any case illusory; we regard a model as tractable if there is a closed-form solution for *a small number* of derivative prices, overlooking the fact that for the majority of derivative prices there is no closed form solution.

where r is the riskless rate process, and $\rho_S \in (0,1)$ is the correlation[9] of the log prices with the market Brownian motion. The process W^S is a d-dimensional Brownian motion independent of W^M.

The volatility process σ appearing in (3.14) is represented as

$$\sigma^i_t = \bar{\sigma}^i \exp(\,\xi^i_t\,) \tag{3.15}$$

in terms of constants $\bar{\sigma}^i$ and a process ξ which is an OU process evolving as

$$d\xi_t = -\beta_\xi\, \xi_t\, dt + \sigma_\xi\, (\rho_\xi\, dW^M_t + \rho'_\xi\, dW^\xi_t\,). \tag{3.16}$$

Finally, our model for the interest-rate process r is just a Black-Karasinski model: we have $r_t = \bar{r}\exp(z_t)$, where

$$dz_t = -\beta_r\, dt + \sigma_r\, (\rho_r\, dW^M_t + \rho'_r\, dW^r_t) \tag{3.17}$$

for some constants $\beta_r > 0$, $\sigma_r > 0$ and ρ_r, which would typically be assumed positive since we expect that as the market rises the rate of interest should also rise.

Altogether then, this is a simple but sprawling model; even assuming (as we do here) that correlations are common across stocks, the parameter vector is

$$\theta = (\,\rho_S, \rho_\xi, \rho_r, (\bar{\sigma}^i), \bar{r}, \beta_\xi, \sigma_\xi, \beta_r, \sigma_r). \tag{3.18}$$

What would be reasonable values for these parameters? For the interest rate evolution, we shall be guided by Black & Karasinski [3] and take $\bar{r} = 0.06$, $\sigma_r = 0.12$, $\beta_r = 0.02$, and $\rho_r = 0.3$. Correlations between stocks are variable, but typically in the range 0.25-0.60; we shall take $\rho_S = 0.3$. For simplicity we assume all stocks have common volatility $\bar{\sigma}^i = 0.6$. Fluctuations in volatilities are of the order of tens of percent, so by comparing with the standard deviation of an OU process, we impose

$$\frac{\sigma_\xi}{\sqrt{2\beta_\xi}} = 0.1. \tag{3.19}$$

We shall set $\beta_\xi = 4.5$, so that the mean reversion time for volatility is of the order of three months, and this gives from (3.19) that $\sigma_\xi = 0.3$. Finally, we take $\rho_\xi = 0.3$.

For the rest of our discussion, we shall focus on the case where $d = 5$, that is: there are five assets. We will also restrict attention to min puts, taking strike $K = 100$ and all assets starting at 100 throughout. This will allow us to investigate the effects of varying initial values of r and σ, as well as various

[9] When ρ is a correlation coefficient, we use ρ' to denote $\sqrt{1-\rho^2}$.

parameters. If we make $\rho_S = 0$ and make the mean reversion parameters β_r, β_ξ very large, we have in effect got back to the situation with independent assets, constant volatilities and interest rate that was studied in Examples 3.1, 3.2. So we should see the same answer; and we do - the range from this calculation came out to be [39.06, 40.65], which is the same as we found in Table 1, to within sampling error.

Next, we can see what happens when we keep the volatility and interest rate constant, but allow ρ_S to vary; the results are in Table 9. What we see is that while the correlation is not too far from zero, there is no clear effect on the price, but as the correlation between the assets rises, the price of the min put falls. This is to be expected; the higher the correlation the more alike the assets are, so there will be less dispersion in the prices at any time, so the minimum will be higher.

ρ_S	low	high	gap (%)	time
-0.15	38.75	40.27	3.77	27
0.00	39.06	40.65	3.92	28
0.15	38.61	40.11	3.73	27
0.30	37.47	38.99	3.89	27
0.45	35.58	37.26	4.52	27
0.60	32.98	34.58	4.61	27

Table 9: Prices of min puts as ρ_S varies. Volatility is constant at $\bar{\sigma} = 0.6$, interest is constant at $r = 0.06$. 40 timesteps are used, $N_{\text{bins}} = 200$, $b = 50$. To establish the lower bound, 50000 paths are used, and for the upper bound, 4000 paths are used with $N_{\text{sub}} = 25$ representatives in each subsimulation.

Now we relax the assumption of constant interest rate, and let the interest rate evolve as in the Black-Karasinski specification (3.17), fixing ρ_S at its default value 0.3, and observing the effects of different initial values of the riskless rate. We hold the volatility constant at the default value 0.6. The results are reported in Table 10. As the initial interest rate rises[10], the price of the min put falls, as would be expected from the stronger discounting of the stopping reward; even allowing for the fact that we can only obtain an interval for the price, the effect of change of initial interest rate is discernible.

Having seen the effect of changing initial interest rate while volatility is held constant, we next hold the interest rate constant at its default value $\bar{r} = 0.06$, and allow the volatility to be stochastic. The impact of different

[10] The row of the table for $r_0 = 0$ was obtained by taking $\log r_0 = -15.6$.

r_0	low	high	gap (%)	time
0.00	40.31	41.74	3.43	27
0.025	39.10	40.59	3.66	27
0.06	37.47	39.17	4.35	27
0.10	35.63	37.52	5.01	27

Table 10: Prices of min puts as r_0 varies. Volatility is constant at $\bar{\sigma} = 0.6$, correlation between assets is $\rho_S = 0.3$. Parameters are $\rho_r = 0.3$, $\bar{r} = 0.06$, $\beta_r = 0.02$, $\sigma_r = 0.12$. 40 timesteps are used, $N_{\text{bins}} = 200$, $b = 50$. To establish the lower bound, 50000 paths are used, and for the upper bound, 4000 paths are used with $N_{\text{sub}} = 25$ representatives in each subsimulation.

initial levels of volatility is shown in Table 11, where it is assumed that the initial volatility is common across all the assets. Again, we see quite pronounced effect of the initial volatility on the price of the min put option; the price rises as the initial volatility increases, as one would expect.

σ_0	low	high	gap (%)	time
0.10	22.09	23.10	4.36	26
0.20	26.56	27.74	4.25	26
0.30	29.88	31.22	4.29	27
0.40	32.74	34.11	4.03	27
0.50	35.20	36.79	4.32	27
0.60	37.46	39.12	4.23	27

Table 11: Prices of min puts as σ_0 varies. Interest is constant at $\bar{r} = 0.06$, correlation between assets is $\rho_S = 0.3$. Parameters are $\rho_\xi = 0.3$, $\bar{\sigma}^i = 0.6$ for all i, $\beta_\xi = 4.5$, $\sigma_\xi = 0.3$, 40 timesteps are used, $N_{\text{bins}} = 200$, $b = 50$. To establish the lower bound, 50000 paths are used, and for the upper bound, 4000 paths are used with $N_{\text{sub}} = 25$ representatives in each subsimulation.

The final study considers the full model where both volatility and interest rate are stochastic. There are too many parameters to explore in a paper, so we content ourselves with holding the parameters fixed at their default values, and varying the initial riskless rate and initial volatility. We see the results in Table 12. Again the comparative statics behave as one would expect, with the magnitudes of the effects of changing initial values being big enough to show up even allowing for the fact that we have only got bounds on the prices.

r_0	σ_0	low	high	gap (%)	time
0.00	0.2	29.41	30.50	3.59	26
	0.4	35.53	36.76	3.35	27
	0.6	40.35	41.44	2.62	26
0.06	0.2	26.58	27.74	4.21	26
	0.4	32.66	34.10	4.23	27
	0.6	37.58	39.08	3.84	27

Table 12: Prices of min puts as σ_0 and r_0 vary. Parameters are $\rho_S = \rho_\xi = \rho_r = 0.3$, $\bar{\sigma}^i = 0.6$ for all i, $\bar{r} = 0.06$, $\beta_\xi = 4.5$, $\sigma_\xi = 0.3$, $\beta_r = 0.02$, $\sigma_r = 0.12$. 40 timesteps are used, $N_{\text{bins}} = 200$, $b = 50$. To establish the lower bound, 50000 paths are used, and for the upper bound, 4000 paths are used with $N_{\text{sub}} = 25$ representatives in each subsimulation.

4 Conclusions and discussion.

The aim of this paper has been to see to what extent we are able to solve the problem of pricing Bermudan options in very high dimensions. Once we accept that for such problems it is impossible that we can know the value function, we realize that in fact various approaches which have been developed in the last twenty or so years may be combined to provide a practical solution in many instances. Working entirely with Markovian problems, the key elements to the approach studied here are:

- pretend that the stopping reward process Z is itself Markovian, and by discretizing Z onto a suitably-chosen finite set of values we estimate the transition probabilities of this finite state Markov chain by simulation (this is the approach of Barraquand & Martineau [2]);
- solve the optimal stopping problem for this finite state Markov chain by dynamic programming;
- use the solution to generate a stopping rule whose performance is evaluated by simulation;
- use the dual characterization of the value of the problem (see [13], [7]) to find a hedging martingale.

This approach is a very general methodology that delivers upper and lower bounds on the price that are generally reasonably close, but more importantly it provides effective recipes for action. It is often said that the price a bank charges for a derivative is more to do with the cost of hedging that derivative than with any number that comes out of some model; and the approach we advocate here puts that into effect. Indeed, at each time the analysis we propose tells the seller of the option what hedge he should use - all he has to do is to hedge the approximate value at the next time step. That approximate value is a simple function of the stopping reward at the next time step. Similarly, the approach we use provides a compact solution for the buyer of the option; at each time, he calculates the value of immediate stopping, and stops if and only if this value is in some finite union of intervals.

This approach is remarkably successful in many of the examples we have studied, providing bounds which are often within 5% of each other. Since this sort of error is inherent in the estimates of input parameters, or in the assumption that those parameters are constant over time, there is really little benefit in getting the bounds very much tighter. We would ideally like to have methods which (if they cannot get the derivative price exactly) can give bounds which are apart by, say, 1 basis point. This is an industry-standard criterion ... but where does it come from? Does a bank *really* care if their calculation of the price of a derivative is out by 1 cent in $100? Of course not! The 1bp criterion really comes from the desire to delta-hedge the option; so we want to vary the prices of the underlyings by ±1% and then find the change in the price in order to put on the delta hedge, and at this point 1bp accuracy is a relevant requirement. But the approach here gives us the hedging strategy by a completely different route - there is *no* delta hedging, only the hedge that comes from the dual approach! Moreover, getting 1bp accuracy from a simulation method is already rather over-optimistic.

So what happens when the upper and lower bounds are further apart, as in the fixed-strike Bermudan Asian option? This is not problematic conceptually; the lower bound is what the buyer objectively thinks the option *on its own* is worth, the upper bound is objectively what the seller expects hedging this derivative *on its own* to cost him, but either side may move beyond the bound if by taking on the contract they lay off risk elsewhere in their portfolio - this is the most basic reason for a market in derivatives. There is nothing difficult or contradictory in a market where the bid price is below the ask price - this is normal! If we find a situation where the bounds are far apart, and we feel it is important to bring the bounds closer together, what can we do? There are four places where error enters into this approach:

1. the assumption that the reward process Z is a real-valued Markov process;

2. the error from discretizing Z values into bins, and deriving the transitions from simulation;

3. simulation error in evaluating the stopping strategy;
4. simulation error in evaluating the hedging strategy.

We can reduce the last two by doing many more simulations, and the second can also be addressed by taking more bins and doing more simulations, but *the first error is intrinsic* - we can do nothing to reduce it other than change the problem in some way. The fixed-strike Bermudan Asian option illustrates this well, because as we discussed in Example 3.4, the state variable for this particular problem really has to be the two-dimensional vector (S_t, A_t), and the crude notion that we can get a good approximation just by looking at A_t on its own is shown by our calculations to be wide of the mark. Now of course we could work to exploit the specific features of this problem to devise a problem-specific solution (and the finite-difference calculations in Longstaff & Schwartz [11] and Rogers & Shi [14] are instances), but this is contrary to the *generic* nature of the approach presented here. If we wanted to continue to use this approach, we might try to bin the values of the bivariate process (S_t, A_t), which is of course conceptually no different from the binning of the scalar-valued reward process, but we may expect that the coarser binning we can expect from a two-dimensional underlying process will raise the second type of error.

To summarize, then, the approach explored in this paper:

- is completely generic - the same code gets used with only changes in the Markov process and the reward function;
- always gives bounds, which are often within the range of error introduced by estimation or questionable modelling assumptions - and always within the profit margin required for an OTC product;
- is largely insensitive to dimension;
- gives simple and explicit exercise strategies, and hedging strategies;
- requires only the ability to simulate a step of the underlying Markov process.

The approach of this study puts together earlier discoveries from the last twenty years. While it may be premature to declare that Bermudan options in high dimensions are now a solved solved problem, what we have seen is that there is a general approach which proves to be at very least a good start on the problem, even if for a particular question we may want to dig deeper. Once we set the question in the context of estimation error and model uncertainty, it may indeed seem pointless to dig deeper in any case.

References

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

[2] J. Barraquand and D. Martineau. Numerical valuation of high dimensional multivariate American securities. *Journal of Financial and Quantitative Analysis*, 30(3):383–405, 1995.

[3] F. Black and P. Karasinski. Bond and option pricing when short rates are lognormal. *Financial Analysts Journal*, 47(4):52–59, 1991.

[4] S. Boyarchenko and S. Levendorskiĭ. American options in the Heston model with stochastic interest rate and its generalizations. *Applied Mathematical Finance*, 20(1):26–49, 2013.

[5] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7:35–72, 2004.

[6] M. C. Fu, S. B. Laprise, D. B. Madan, Y. Su, and R. Wu. Pricing American options: a comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.

[7] M. B. Haugh and L. Kogan. Pricing American options: a duality approach. *Operations Research*, 52:258–270, 2004.

[8] F. Jamshidian. Numeraire-invariant option pricing & American, Bermudan, and trigger stream rollover. Technical report, University of Twente, 2004.

[9] X. Jin, X. Li, H. H. Tan, and Z Wu. A computationally efficient state-space partitioning approach to pricing high-dimensional American options via dimension reduction. *European Journal of Operational Research*, 231(2):362–370, 2013.

[10] P. Kovalov, V. Linetsky, and M. Marcozzi. Pricing multi-asset American options: A finite element method-of-lines with smooth penalty. *Journal of Scientific Computing*, 33(3):209–237, 2007.

[11] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

[12] A. Medvedev and O. Scaillet. Pricing American options under stochastic volatility and stochastic interest rates. *Journal of Financial Economics*, 98(1):145–159, 2010.

[13] L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002.

[14] L. C. G. Rogers and Z. Shi. The value of an Asian option. *Journal of Applied Probability*, pages 1077–1088, 1995.