Available online at www.sciencedirect.com

ScienceDirect

Mathematical and Computer Modelling 47 (2008) 484–494

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

On pricing discrete barrier options using conditional expectation and importance sampling Monte Carlo

Giray Ökten*, Emmanuel Salta, Ahmet Göncü

Department of Mathematics, Florida State University, Tallahassee FL 32306-4510, United States

Received 10 April 2007; accepted 11 May 2007

Abstract

Estimators for the price of a discrete barrier option based on conditional expectation and importance sampling variance reduction techniques are given. There are erroneous formulas for the conditional expectation estimator published in the literature: we derive the correct expression for the estimator. We use a simulated annealing algorithm to estimate the optimal parameters of exponential twisting in importance sampling, and compare it with a heuristic used in the literature. Randomized quasi-Monte Carlo methods are used to further increase the accuracy of the estimators.

Keywords: Barrier options; Quasi-Monte Carlo; Importance sampling; Conditional expectation

1. Introduction

A barrier option is an option that becomes alive or is killed depending on whether the option crosses a barrier before expiry. In this paper, we will consider discrete down-and-in barrier options, and discuss Monte Carlo methods for pricing these options based on two variance reduction techniques: conditional expectation and importance sampling. Simple modifications of these methods can be used to price other types of barrier options. The application of the conditional expectation technique to barrier options was considered by Boyle, Broadie and Glasserman [2] and Ross and Shanthikumar [13]. The authors of these papers derived the conditional expectation estimator for the price of a barrier option. In Section 2, we will show that their formulas for the estimator contain errors, and we will derive the correct expression for the conditional expectation estimator.

The application of importance sampling in pricing barrier options was first described by Boyle et al. [2]. The resulting estimator involves two “twisting” parameters that need to be determined. A heuristic to compute these parameters was introduced in Boyle et al. [2]. In Section 7, we will use numerical results to compare the parameters obtained from this heuristic with the parameters obtained from a simulated annealing algorithm.

In Section 4 we will combine the conditional expectation and importance sampling estimators, as was done by Ross and Shanthikumar [13]. We point to another error in a formula in Ross and Shanthikumar [13]. We will then investigate the use of randomized quasi-Monte Carlo methods to obtain further error reduction.

* Corresponding author.
E-mail addresses: okten@math.fsu.edu (G. Ökten), esalta@math.fsu.edu (E. Salta), agoncu@math.fsu.edu (A. Göncü).

0895-7177/$ - see front matter
doi:10.1016/j.mcm.2007.05.001

2. Conditional expectation Monte Carlo

Consider a down-and-in European call option written on a stock $S(t)$, whose prices are observed at discrete time steps $0 = t_0 < t_1 < \cdots < t_m = T$. The barrier price will be denoted by H, the exercise price by K, and the expiry by T. The option becomes a vanilla European call option when the stock "crosses" the barrier, i.e. if $S(t_i) < H$ for some $t_i < T$. We will assume that $S(0) > H$. The time the stock crosses the barrier is t_τ, where τ is a nonnegative random variable.

The payoff of the down-and-in European call option at time T is the product

$$\mathbf{1}(\tau < m) \cdot (S(T) - K, 0)^+, \tag{1}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function and $(x, y)^+$ denotes the maximum of x and y. The price of the option at time 0 is the discounted expected payoff

$$I = \mathbf{E}[\mathrm{e}^{-rT}\mathbf{1}(\tau < m)(S(T) - K, 0)^+] \tag{2}$$

where r is the risk-free interest rate and the expectation is conditioned on the initial stock price $S(0)$.

The crude Monte Carlo estimates I by simulating N stock price paths, averaging over the values

$$\begin{cases} (Y - K, 0)^+ & \text{if } \tau < m \text{ and } S(T) = Y \\ 0 & \text{otherwise} \end{cases}$$

and then discounting this average by multiplying by e^{-rT}.

The conditional expectation Monte Carlo is based on the following observation: if it is possible to price the down-and-in European call option analytically when it crosses the barrier, then one should use this analytical formula instead of simulating the price path until expiry. An example is the lognormal model for $S(t)$, for which the Black–Scholes–Merton formula can be used to price the European call option, when the stock crosses the barrier. To proceed, we condition the expectation (2) on τ and $S(t_\tau)$, and consider the case when $\tau < m$:

$$\mathbf{E}[(S(T) - K, 0)^+] = \mathbf{E}[\mathbf{E}[(S(T) - K, 0)^+ \mid \tau, S(t_\tau)]]. \tag{3}$$

To compute the inner expectation, we assume the lognormal model for $S(t)$, and obtain

$$\begin{aligned} \mathbf{E}[(S(T) - K, 0)^+ \mid \tau = k, S(t_k)] &= \mathbf{E}[(S(T) - K, 0)^+ \mid S(t_k)] \\ &= \mathrm{e}^{r(T-t_k)}\mathrm{BSM}(S(t_k), t_k, T) \end{aligned}$$

where $\mathrm{BSM}(S(t_k), t_k, T)$ is the Black–Scholes–Merton formula for the price of a European call option at time t_k, with expiry T, and initial stock price $S(t_k)$:

$$\begin{aligned} \mathrm{BSM}(S(t_k), t_k, T) &= \mathbf{E}[\mathrm{e}^{-r(T-t_k)}(S(T) - K)^+ \mid S(t_k) = \mathcal{S}] \\ &= \mathcal{S}\Phi(d_1) - K\mathrm{e}^{-r(T-t_k)}\Phi(d_2) \end{aligned}$$

with

$$d_1 = d_2 + \sigma\sqrt{T - t_k} = \frac{\log(\mathcal{S}/K) + (r + \sigma^2/2)(T - t_k)}{\sigma\sqrt{T - t_k}}.$$

Therefore, the inner expectation in (3) is

$$\mathbf{E}[(S(T) - K, 0)^+ \mid \tau, S(t_\tau)] = \mathrm{e}^{r(T-t_\tau)}\mathrm{BSM}(S(t_\tau), t_\tau, T)$$

and (3) simplifies to

$$\mathbf{E}[(S(T) - K, 0)^+] = \mathbf{E}[\mathrm{e}^{r(T-t_\tau)}\mathrm{BSM}(S(t_\tau), t_\tau, T)].$$

So, the conditional Monte Carlo estimator simulates N paths and averages over the values

$$\begin{cases} \mathrm{e}^{r(T-t_k)}\mathrm{BSM}(S(t_k), t_k, T) & \text{if } \tau = k < m \\ 0 & \text{otherwise.} \end{cases}$$

Discounting this average by e^{-rT} gives an unbiased estimate for the option value I. In other words, we estimate the option value by

$$e^{-rT}\left(\frac{1}{N}\sum_{i=1}^{N} e^{r(T-t_\tau)}\mathrm{BSM}(S^{(i)}(t_\tau), t_\tau, T)1(\tau < m)\right)$$
$$= \frac{1}{N}\sum_{i=1}^{N} e^{-rt_\tau}\mathrm{BSM}(S^{(i)}(t_\tau), t_\tau, T)1(\tau < m) \quad (4)$$

where $S^{(i)}$ is the ith stock price path, and $S^{(i)}(t_\tau)$ is the price observed at the time when the path crosses the barrier. If the barrier is not crossed by the expiry, the estimator value is zero.[1]

3. Importance sampling

The use of importance sampling in pricing down-and-in options was first introduced in Boyle et al. [2]. The idea is to change the probability distribution of the stock so that the probability that the stock crosses the barrier and the probability that the stock goes above the strike price are increased. In this way, we can gather useful information from more of the stock price paths.

Consider a general stock price model of the following form:

$$S(t_n) = S(0)\exp(L_n)$$

where $L_n = \sum_{i=1}^{n} X_i$ and $X_1, X_2, \ldots$ are i.i.d. random variables with the common density $f(x)$. We put $L_0 = 0$. Importance sampling uses two new density functions, say $f_1(x)$ and $f_2(x)$, to simulate the stock price. The random variables $X_1, \ldots, X_\tau$ are simulated using $f_1(x)$, and $X_{\tau+1}, \ldots, X_m$ using $f_2(x)$. From the importance sampling identity, we have

$$\tilde{\mathbf{E}}\left[h(S(t_1), \ldots, S(t_m))\prod_{i=1}^{\tau}\frac{f(X_i)}{f_1(X_i)}\prod_{i=\tau+1}^{m}\frac{f(X_i)}{f_2(X_i)}\right] = \mathbf{E}\left[h(S(t_1), \ldots, S(t_m))\right] \quad (5)$$

provided the independence of X_i is preserved under f_1 and f_2, and the expectation is finite. The expectation on the left-hand side of the equation is taken under the new probability measure imposed by the density functions f_1 and f_2.

A standard technique called exponential twisting (or, tilting) (see Glasserman [5], Ross [12]) chooses $f_1 \equiv f_{t^-}$ and $f_2 \equiv f_{t^+}$ where

$$f_{t^-}(x) = \frac{e^{t^- x} f(x)}{M(t^-)}, \qquad f_{t^+}(x) = \frac{e^{t^+ x} f(x)}{M(t^+)}$$

and $M(t)$ is the moment generating function of the density $f(x)$. With these choices, the "likelihood ratio" in (5) simplifies to

$$\prod_{i=1}^{\tau}\frac{f(X_i)}{f_1(X_i)}\prod_{i=\tau+1}^{m}\frac{f(X_i)}{f_2(X_i)} = \frac{M(t^-)^\tau}{\exp(t^-(X_1+\cdots+X_\tau))}\frac{M(t^+)^{m-\tau}}{\exp(t^+(X_{\tau+1}+\cdots+X_m))}$$
$$= \left(\frac{M(t^-)}{M(t^+)}\right)^\tau M(t^+)^m \exp\left((t^+ - t^-)L_\tau - t^+ L_m\right). \quad (6)$$

The parameters t^- and t^+ need to be determined in such a way that the probability of the "desired" events, i.e. the events that the stock crosses the barrier and the stock goes above the strike price, are increased. In Boyle et al. [2], a heuristic is used to compute these parameters. Consider a uniform discretization of time so that $t_n = nh$. If we specify

[1] This formula was erroneously written with the discount factor e^{-rT} in Boyle et al. [2] (p. 1289), and with no discounting factor in Ross and Shanthikumar [13] (Eq. (2), p. 321).

our model as the lognormal model, and thus let $f(x)$ be the normal density with mean $(r-\sigma^2/2)h$ and variance $\sigma^2 h$, then the heuristic of Boyle et al. [2] gives the following solution for t^- and t^+:

$$t^- = \left(\frac{1}{2}-\frac{r}{\sigma^2}\right)-\left(\frac{2b+c}{m\sigma^2 h}\right), \qquad t^+ = \left(\frac{1}{2}-\frac{r}{\sigma^2}\right)+\left(\frac{2b+c}{m\sigma^2 h}\right) \tag{7}$$

where $b=\log(S(0)/H)$ and $c=\log(K/S(0))$. In particular, these choices simplify the likelihood ratio (6) to

$$M(t^+)^m \exp\left((t^+-t^-)L_\tau - t^+ L_m\right).$$

The importance sampling Monte Carlo estimator simulates N paths and averages over the values

$$\begin{cases} M(t^+)^m \exp\left((t^+-t^-)L_\tau - t^+ L_m\right)(S(T)-K,0)^+ & \text{if } t_\tau < T \\ 0 & \text{otherwise.} \end{cases}$$

Discounting this average by e^{-rT} gives an unbiased estimate for the option value I.

4. Combination of conditional expectation and importance sampling

Ross and Shanthikumar [13] observe that the conditional expectation and importance sampling methods can be combined. Importance sampling can be applied to the stock price path until the path crosses the barrier, and when the barrier is crossed, the conditional expectation estimator can be used to compute the option price. Since we need to simulate the stock price path until it crosses the barrier, there is only one twisting parameter, $t := t^-$, that needs to be determined. The likelihood ratio simplifies to

$$\frac{M(t)^\tau}{\exp(t(X_1+\cdots+X_\tau))} = \frac{M(t)^\tau}{\exp(tL_\tau)}.$$

The estimate for the option price given by the combined estimator is

$$\frac{1}{N}\sum_{i=1}^{N} \mathrm{e}^{-rt_\tau} M(t)^\tau \exp(-tL_\tau)\mathrm{BSM}(S^{(i)}(t_\tau), t_\tau, T)1(\tau < m) \tag{8}$$

where $S^{(i)}$ is the ith stock price path.

If we assume the lognormal model for the stock price process, i.e., let f be the normal density with mean $(r-\sigma^2/2)h$ and variance $\sigma^2 h$, then f_t becomes the normal density with mean $(r-\sigma^2/2)h-\mathfrak{b}$ and variance $\sigma^2 h$, where $\mathfrak{b}=-\sigma^2 ht$. If t is obtained using the heuristic (7), then we get[2]

$$\mathfrak{b} = \left(r-\sigma^2/2\right)h + \frac{2\log(S(0)/H)+\log(K/S(0))}{m}. \tag{9}$$

5. Simulated annealing

It is possible to use simulated annealing to compute the twisting parameters needed by the importance sampling and combined estimators. We consider a particular simulated annealing algorithm designed for discrete stochastic optimization, introduced by Alrefaei and Andradóttir [1]. The algorithm computes

$$\min_{w\in\Omega} \mathbf{E}[h(w, Y_w)] \tag{10}$$

where Ω is a discrete set, h is a deterministic function, and Y_w is a random variable that depends on the parameter w. The temperature in the simulated annealing algorithm is constant, and the expectation in (10) is estimated by the sample mean $\bar{h}(w) = \frac{1}{L_k}\sum_{i=1}^{L_k} h(w, Y_w(i))$ at the kth step of the algorithm, where L_k is a an unbounded sequence

[2] This formula was erroneously reported as $\mathfrak{b} = rh - \frac{2\log\left(\frac{S(0)}{H}\right)+\log\left(\frac{K}{S(0)}\right)}{m}$ in Ross and Shanthikumar [13] (p. 321), where h is replaced by $1/N$, and m is replaced by n.

of positive integers. The random variables $Y_w(i)$, $i = 1, \ldots, L_k$ are i.i.d. realizations of Y_w. The algorithm constructs a Markov chain with state space Ω. If at step k, the Markov chain is in state w, then the next state is chosen from a set of predetermined "neighbours" of w in the following way: if a neighbour of w, say w', is a better state than w, i.e. $\bar{h}(w') \leq \bar{h}(w)$, then the next state of the Markov chain is chosen as w'. If $\bar{h}(w') > \bar{h}(w)$, the Markov chain moves to state w' (aptly called a hill-climbing move) with small probability $\exp(-(\bar{h}(w') - \bar{h}(w), 0)^+/\mathcal{T})$; otherwise it stays in state w. The constant $\mathcal{T}$ is the fixed temperature of the algorithm. After running the algorithm for $\mathcal{K}$ steps, the final estimate of the global minimum (10) is chosen as the state with the lowest average $\bar{h}(w)$.

In our option pricing problem, our objective is to find the twisting parameters that minimize the simulation error. The parameter w corresponds to the twisting parameters (t^-, t^+) in importance sampling, and to the single twisting parameter t in the combined estimator. The function h computes the corresponding simulation error. In the numerical results of Section 7, we will consider a "test" problem where the price of the option can be computed very accurately by another method. We will use these "accurate" option prices as the "true" solutions, and let h be the relative square error of the option estimate for a fixed sample size. In other words, if $Y_w(1), \ldots, Y_w(L_k)$ are option estimates based on N paths (obtained from any of the estimators discussed in the paper) then

$$h(w, Y_w(i)) = \left(\frac{Y_w(i) - True}{True} \right)^2.$$

The simulated annealing algorithm searches for the best parameters that minimize the mean of the relative square errors of the option estimates.

6. Randomized quasi-Monte Carlo

Quasi-Monte Carlo (QMC) methods use the so-called low-discrepancy sequences (other names used in the literature are, uniformly distributed mod 1 sequences, (t, s)-sequences, and QMC sequences) in simulating the underlying model instead of the pseudorandom numbers used by Monte Carlo. The QMC convergence rate is faster than the Monte Carlo method. Therefore, in recent years, QMC methods have been used successfully in many applications, including option pricing.

There is, however, a drawback of the QMC method: using a low-discrepancy sequence one can obtain only a single estimate and there is no known practical way of measuring the actual error of the estimate. In Monte Carlo, in contrast, one can generate several independent and unbiased estimates and use statistics to analyse error. The randomized quasi-Monte Carlo (RQMC) methods are introduced to address this drawback. In a sense, they combine the best of two worlds: by appropriately "randomizing" a low-discrepancy sequence, they make it possible to obtain several independent unbiased estimates so that statistics can be used to assess estimation error. Moreover, since each randomized sequence is still a low-discrepancy sequence, the better accuracy of the QMC method continues to apply. A survey of RQMC methods and their use in option pricing is given by Ökten and Eastman [10].

The Faure sequence is an example of a *digital* (t, s) *sequence*. The digital sequences comprise a big portion of low-discrepancy sequences used in practice. Consider an s-dimensional digital sequence $q_1, q_2, \ldots,$ in the s-dimensional unit hypercube $[0, 1)^s$. To compute the nth term $q_n = (q_n^{(1)}, \ldots, q_n^{(s)})$ of the sequence, we first write n in its base p expansion: $n = a_l p^{l-1} + \cdots + a_2 p + a_1$. Here p is called the base of the digital sequence. In the case of the Faure sequence, the base is the smallest prime number greater than or equal to s. Let $d_n = (a_1, \ldots, a_l)^{\mathrm{T}}$ be the (column) vector of the digits of n. Similarly, let $\phi(q_n^{(k)})$ denote the (column) vector of the digits of the real number $q_n^{(k)} \in [0, 1)$ in base p. For example, if the base p expansion of a real number is $\tilde{a}_{\tilde{l}}/p^{\tilde{l}} + \cdots + \tilde{a}_2/p^2 + \tilde{a}_1/p$ then $\phi(\cdot) = (\tilde{a}_1, \ldots, \tilde{a}_{\tilde{l}})^{\mathrm{T}}$. The kth component of the nth term q_n is defined through its digit vector:

$$\phi(q_n^{(k)}) = \mathbf{C}^{(k)} d_n, \quad k = 1, \ldots, s \tag{11}$$

where $\mathbf{C}^{(k)}$ is the kth *generator matrix* of the digital sequence, and the matrix–vector multiplication is done in modular arithmetic mod p.

The generator matrix for the Faure sequence is

$$\mathbf{C}^{(k)} = \mathbf{P}^{k-1}, \quad k = 1, \ldots, s \tag{12}$$

where $\mathbf{P}$ is the $l \times l$ Pascal matrix[3] mod p: this is an upper triangular matrix where the ijth entry is the binomial term $\binom{j-1}{i-1}$ mod p. For example, the 3×3 Pascal matrix is

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Powers of the Pascal matrix can be computed without matrix multiplication as follows. Define $\mathbf{P}(a)$ as the upper triangular matrix with the ij th entry equal to $a^{j-i}\binom{j-1}{i-1}$. Then, we have the following useful identity: $\mathbf{P}^k = \mathbf{P}(k)$ (see Call and Velleman [4]) for any integer k (adopting the convention $0^0 = 1$, so that $\mathbf{P}^0 = \mathbf{P}(0) = \mathbf{I}$).

We now describe how to obtain randomly scrambled "copies" of the Faure sequence. Generate s lower triangular square matrices $\mathbf{L}^{(k)}$, $k = 1, \ldots, s$, whose non-diagonal entries are random integers between 0 and $p-1$, and diagonal entries are random integers between 1 and $p-1$. Also generate s vectors $g^{(k)}$ whose entries are random integers between 0 and $p-1$. The dimension of the square matrices and the vectors is equal to l; the dimension of the vector d_n. A scrambled Faure sequence is obtained by replacing the generator matrices of the Faure sequence (12) by

$$\mathbf{C}^{(k)} = \mathbf{L}^{(k)}\mathbf{P}^{k-1}, \quad k = 1, \ldots, s \tag{13}$$

and replacing (11) by

$$\phi(q_n^{(k)}) = \mathbf{C}^{(k)} d_n + g^{(k)}, \quad k = 1, \ldots, s. \tag{14}$$

As before, all operations are done in mod p. This scrambling method is called linear scrambling by Matoušek [8], who discusses this as well as other scrambling techniques. A detailed discussion of scrambled sequences and their implementation is also given by Hong and Hickernell [6].

7. Numerical results

In this section our objective is to numerically compare the three Monte Carlo estimators discussed earlier. We will also investigate the potential benefits of the simulated annealing approach to compute the twisting parameters against the heuristic, and the RQMC implementation of the Monte Carlo algorithms against the traditional pseudorandom implementation. Our Monte Carlo algorithms are general-purpose algorithms that can be used in any model for the stock price, as long as the price of the option can be computed exactly when the underlying reaches the barrier. This assumption was used in deriving the conditional and combined estimators.

A commonly used approach to compare different Monte Carlo estimators is to apply them to a test problem where the exact solution is known, so that the actual error of the estimators can be compared. We will follow this approach, and concentrate on the lognormal model for the stock price, where an algorithm introduced by Tse, Li and Ng [14] can be used to compute the price of a discrete barrier option very accurately. In Exhibit 2 of Tse et al. [14], the authors give the price of a down-and-out call option for a number of parameters. They report that these prices have accuracy of at least 10 decimal places. In the numerical results that will follow, we will consider the same option parameters used in Tse et al. [14], and use their estimates as true solutions when we compare the accuracy of our Monte Carlo estimators in the lognormal setting.

7.1. Comparing the three estimators

Assuming the lognormal model for the stock price process, we consider two down-and-in call options. The options have the following common parameters: $\sigma = 0.3$, $r = 0.1$, $S(0) = 100$, $K = 100$, $T = 0.2$, and $m = 50$. In Table 1, $H = 95$, and the option price is 1.4373238784. In Table 2, $H = 91$, and the option price is 0.3670447223. These prices are computed from the down-and-out call prices given in Exhibit 2 of Tse et al. [14]. The twisting parameters used in importance sampling and combined estimators are computed using the heuristic (7) and (9). The tables display

[3] In other literature such as combinatorics, the Pascal matrix is the transpose of the matrix defined here, without the mod p reduction.

Table 1
$H = 95$, price $= 1.4373238784$

N	CondExp	ImpSamp-h	Combined-h
$5K$	1.36×10^{-4}	3.33×10^{-4}	6.66×10^{-5}
$10K$	8.40×10^{-5}	1.64×10^{-4}	3.48×10^{-5}
$50K$	1.65×10^{-5}	4.29×10^{-5}	1.25×10^{-5}
Cross barrier	61%	84%	84%
Cross exercise	n/a	43%	n/a

Table 2
$H = 91$, price $= 0.3670447223$

N	CondExp	ImpSamp-h	Combined-h
$5K$	4.66×10^{-4}	3.56×10^{-4}	9.78×10^{-5}
$10K$	4.17×10^{-4}	1.74×10^{-4}	5.11×10^{-5}
$50K$	6.98×10^{-5}	4.90×10^{-5}	1.66×10^{-5}
Cross barrier	40%	86%	86%
Cross exercise	n/a	42%	n/a

the relative mean square error (MSE) of fifty option estimates computed using the conditional expectation (CondExp), importance sampling (ImpSamp-h), and combined (Combined-h) Monte Carlo estimators. (The letter "h" shows that the heuristic is used in computing the twisting parameters.) Three sample sizes are considered: $N = 5000, 10,000$, and 50,000. The fifth row of the tables ("cross barrier") gives the percentage of price paths that cross the barrier, and the next row displays the percentage of paths that cross the exercise price among those that crossed the barrier. The pseudorandom number generator used in the simulations is tt800 (see Matsumoto and Kurita [9]).

Conclusions

(1) The Combined-h estimator gives lower relative MSE for all cases considered.
(2) The CondExp estimator gives lower relative MSE than the ImpSamp-h estimator in Table 1, where the barrier price is 95, and 61% of price paths reach the barrier. When the barrier price is lowered to 91 in Table 2, and thus only 40% of paths reach the barrier, we observe that the ImpSamp-h estimator becomes better than the CondExp estimator. Note that 84% and 86% of paths reach the barrier when importance sampling is used in Tables 1 and 2, respectively. The improvements given by CondExp over ImpSamp-h can be amplified by taking the barrier closer to the initial price, and conversely, ImpSamp-h can be made much better than CondExp by taking the barrier price farther from the initial price.

7.2. Comparing the heuristic with simulated annealing

In this section we investigate the quality of the twisting parameters obtained from the heuristic (7) and examine whether better results can be obtained by the use of simulated annealing to compute these parameters. We first introduce new variables, (d, u), which are related to the twisting parameters (t^-, t^+) as follows:

$$\begin{aligned} d &= (r - \sigma^2/2)h + \sigma^2 h t^- \\ u &= (r - \sigma^2/2)h + \sigma^2 h t^+. \end{aligned} \tag{15}$$

We introduce these variables for their intuitive meaning: d and u are the means of the tilted densities $f_{t^-}(x)$ and $f_{t^+}(x)$, respectively. The quantity $(r - \sigma^2/2)h$ in (15) is the mean of the original density $f(x)$. Clearly, one expects $\sigma^2 h t^-$ to be negative, so that the stock is "pulled down" to the barrier faster than before, and $\sigma^2 h t^+$ to be positive, so that the stock is "pushed up" to the barrier faster than before.

For the option parameters of Table 1, the heuristic gives the following parameters for the importance sampling estimator: $u = 0.0021$ and $d = -u$. We want to see how close $(-0.0021, 0.0021)$ is to the point at which the mean

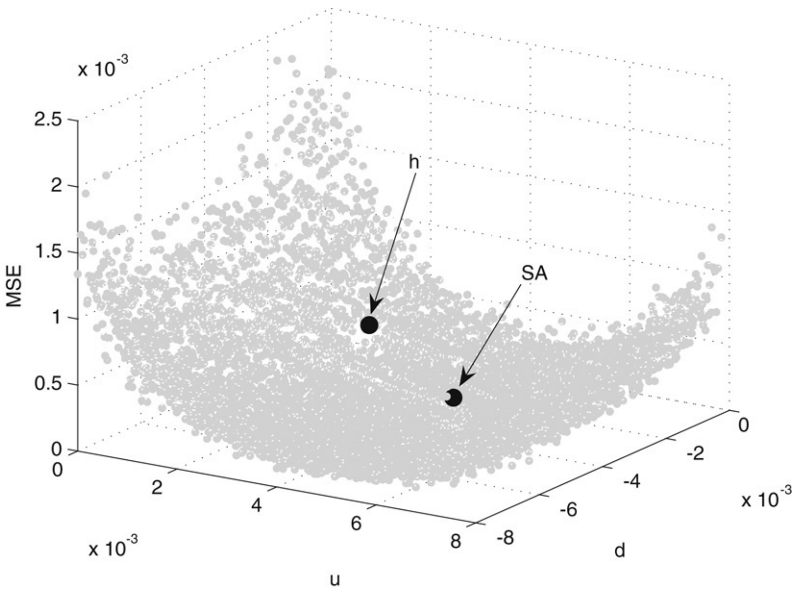

Fig. 1. MSE surface for importance sampling estimator.

Table 3
$H = 95$, price = 1.4373238784

N	ImpSamp-h	ImpSamp-SA	Combined-h	Combined-SA
Parameters	$u = 0.0021$ $d = -u$	$u = 0.005$ $d = -0.004$	$\mathfrak{b} = 0.0023$	$\mathfrak{b} = 0.0055$
$5K$	3.33×10^{-4}	2.85×10^{-4}	6.66×10^{-5}	3.34×10^{-5}
$10K$	1.64×10^{-4}	1.55×10^{-4}	3.48×10^{-5}	2.63×10^{-5}
$50K$	4.29×10^{-5}	9.99×10^{-5}	1.25×10^{-5}	1.44×10^{-5}
Cross barrier	84%	95%	84%	98%
Cross exercise	43%	79%	n/a	n/a

square error (MSE) of the option estimate attains its minimum. This will be done numerically, using Monte Carlo simulation.

Consider the grid, $(d, u) \in [-0.008, 0] \times [0, 0.008]$, with an increment size of 10^{-4}. For each (d, u) of the grid, we run a Monte Carlo simulation using the importance sampling estimator, and compute the MSE of fifty option estimates where each estimate is based on 10,000 price paths. We use the option parameters of Table 1, and use the option price obtained from Tse et al. [14] as the true value in the MSE calculations. Fig. 1 plots the surface of MSE values over this grid.

The point labelled by "h" on the surface has the MSE value of 4.67×10^{-4} and it corresponds to the heuristic $(-0.0021, 0.0021)$. Simulated annealing, on the other hand, gives $(-0.004, 0.005)$ as the best parameters; the corresponding MSE is 3.18×10^{-4}. This point is labelled by "SA" on the surface. The minimum MSE read from the surface is 0.5×10^{-4}. Simulated annealing gives better parameters than the heuristic in this problem. We will next present additional numerical results to investigate potential benefits of simulated annealing.

In the next two tables, we compare the estimators Combined-h with Combined-SA, and ImpSamp-h with ImpSamp-SA, by the relative MSE of the fifty option estimates they produce. Tables 3 and 4 use the option parameters of Tables 1 and 2, respectively. The second row of the tables give the (d, u) parameters for the ImpSamp estimator, and the $\mathfrak{b}$ parameter for the combined estimator, when they are computed using the heuristics (7) and (9), and the simulated annealing algorithm.

The simulated annealing based estimators give less error than the heuristic-based estimators when $N = 5K, 10K$. The opposite is true for $N = 50K$. The differences, however, are very small. Together with the simplicity of

Table 4
$H = 91$, price $= 0.3670447223$

N	ImpSamp-h	ImpSamp-SA	Combined-h	Combined-SA
Parameters	$u = 0.0038$ $d = -u$	$u = 0.0065$ $d = -0.006$	$\mathfrak{b} = 0.0040$	$\mathfrak{b} = 0.0065$
$5K$	3.56×10^{-4}	2.58×10^{-4}	9.78×10^{-5}	4.35×10^{-5}
$10K$	1.74×10^{-4}	1.46×10^{-4}	5.11×10^{-5}	2.83×10^{-5}
$50K$	4.90×10^{-5}	9.70×10^{-5}	1.66×10^{-5}	1.76×10^{-5}
Cross barrier	86%	97%	86%	98%
Cross exercise	42%	77%	n/a	n/a

computing the heuristic parameters, these results suggest that the heuristic is an effective strategy to compute the twisting parameters.

7.3. *Randomized quasi-Monte Carlo versus Monte Carlo*

In our option pricing problem, we want to generate fifty unbiased estimates for the option price. One unbiased estimate is obtained by generating m (the number of time steps) random matrices and vectors (see (13) and (14)), and using the resulting scrambled Faure sequence in simulation.[4] Subsequent unbiased estimates are obtained by independently generating new sets of random matrices and vectors.

In numerical results below, we consider two options. The common parameters for the options are: $r = 0.1, \sigma = 0.3, T = 0.2, K = 100, S(0) = 100$. The first option is the one considered in Table 1, and it has $m = 50$, $H = 95$, price $= 1.4373238784$. The second option has $m = 5$, $H = 93$, price $= 0.3443581039$. One important difference between these options is the number of time steps, m. This parameter determines the dimension of the underlying randomized QMC sequence. It is well known that the advantages of (randomized) QMC sequences over Monte Carlo diminish as the dimension grows (see Ökten, Tuffin and Burago [11]). We will examine the effects of increasing dimension in our problem by comparing these options.

We will also investigate the effects of different normal random variable generation methods on the accuracy of the option estimates. Joy, Boyle and Tan [7] stated that it was incorrect to use the Box-Muller method in transforming Faure vectors, and doing so could increase the estimation error. Numerical results contrary to this observation were obtained in Ökten and Eastman [10] who compared the Box-Muller and inverse transformation methods by the statistical accuracy of their estimates, and found that the Box-Muller method actually gave better results in selected problems from European and Asian option pricing. We will make a similar comparison here, using the first option problem where $m = 50$ is an even number. Note that the Box-Muller method uses two uniform variables to produce two normally distributed variables, whereas the inverse transformation method uses only one uniform variable to produce one normally distributed variable. Note also that if the dimension is even, then both methods use the same RQMC sequence. However, if the dimension m is odd, then the Box-Muller method requires the generation of an $m + 1$ dimensional RQMC sequence, which might result in higher error than the inverse transformation method that uses an m dimensional RQMC sequence. This would be a consequence of the increased dimension of the underlying RQMC sequence.

Table 5 displays the relative MSE when scrambled Faure sequences are used in simulating the price paths. Details of the computations are similar to the previous tables. There are two numbers in each cell of Table 5: the one in parenthesis is the relative MSE when the inverse transformation method is used in transforming the uniformly distributed numbers to numbers from the standard normal distribution. The particular inverse transformation method we used is by B. Moro, and it is described in Joy et al. [7]. The other number is the relative MSE when the Box-Muller method is used in the transformation. The Box-Muller method gives a lower relative MSE in all except for one case ($50K$ for Combined-h).

[4] The ith stock price path $S^{(i)}(t_1), \ldots, S^{(i)}(t_m)$ is generated using the components of the ith (m-dimensional) vector of the scrambled sequence, $q_i = (q_i^{(m)}, \ldots, q_i^{(m)})$, similar to the way m pseudorandom numbers are used to generate a path. See [10] for details.

Table 5
RQMC results

N	CondExp	ImpSamp-h	Combined-h
$5K$	$3.37 \times 10^{-5}(1.27 \times 10^{-4})$	$1.24 \times 10^{-4}(4.32 \times 10^{-4})$	$2.57 \times 10^{-5}(1.01 \times 10^{-4})$
$10K$	$2.00 \times 10^{-5}(4.44 \times 10^{-5})$	$6.43 \times 10^{-5}(1.22 \times 10^{-4})$	$1.57 \times 10^{-5}(3.09 \times 10^{-5})$
$50K$	$5.22 \times 10^{-6}(7.07 \times 10^{-6})$	$2.20 \times 10^{-5}(4.92 \times 10^{-5})$	$3.67 \times 10^{-6}(3.07 \times 10^{-6})$

$m = 50$, $H = 95$, price $= 1.4373238784$.

Table 6
MC/RQMC ratios

N	CondExp	ImpSamp-h	Combined-h
$5K$	4	3	3
$10K$	4	3	2
$50K$	3	2	3

$m = 50$.

Table 7
RQMC results

N	CondExp	ImpSamp-h	Combined-h
$5K$	2.75×10^{-5}	9.16×10^{-5}	1.19×10^{-5}
$10K$	9.16×10^{-6}	4.97×10^{-5}	5.87×10^{-6}
$50K$	1.09×10^{-6}	6.69×10^{-6}	6.78×10^{-7}

$m = 5$, $H = 93$, price $= 0.3443581039$.

Table 8
MC results

N	CondExp	ImpSamp-h	Combined-h
$5K$	6.02×10^{-4}	1.03×10^{-3}	2.37×10^{-4}
$10K$	1.93×10^{-4}	4.53×10^{-4}	1.04×10^{-4}
$50K$	7.89×10^{-5}	8.26×10^{-5}	3.59×10^{-5}

$m = 5$, $H = 93$, price $= 0.3443581039$.

Table 9
MC/RQMC ratios

N	CondExp	ImpSamp-h	Combined-h
$5K$	22	11	20
$10K$	21	9	18
$50K$	72	12	53

$m = 5$.

The parameters used in Table 5 are the same as the parameters of Table 1, where the estimators were compared by their relative MSE values using Monte Carlo. Table 6 displays the ratio of Monte Carlo relative MSE values of Table 1 to RQMC–Box Muller relative MSE values of Table 5. RQMC provides improvements between factors of 2 and 4.

We now consider the second option problem with lower dimension, $m = 5$. We use the inverse transformation method instead of Box-Muller, for reasons mentioned earlier. Tables 7–9 display the RQMC-relative MSE, MC-relative MSE, and their ratios. In this problem, we see improvements as high as factor of 72.

The accuracy of the (R)QMC implementation of the combined estimator for this low-dimensional problem ($m = 5$) fares very well when compared with the lattice methods. Broadie, Glasserman and Kou [3] used a trinomial procedure to compute the price of the down-and-out option that had the same parameters with the down-and-in option considered in Tables 7 and 8. They obtained 6.000 for the price of the down-and-out option. The corresponding European call

Table 10
QMC absolute error

N	$10K$	$20K$	$30K$	$40K$	$50K$
Abs error	1.55×10^{-3}	8.09×10^{-5}	3.91×10^{-4}	4.06×10^{-4}	5.15×10^{-4}
N	$60K$	$70K$	$80K$	$90K$	$100K$
Abs error	2.43×10^{-4}	7.83×10^{-6}	1.45×10^{-4}	7.72×10^{-5}	3.60×10^{-5}

$m = 5$.

price is 6.3441134633, which implies 0.344 for the trinomial estimate for the price of the down-and-in option. Tse et al. [14] report a more accurate price of 5.9997553594 for the same down-and-out option, and thus 0.3443581039 for the price of the down-and-in option. The absolute error of the trinomial estimate, using 5.9997553594 as the exact value, is 2.45×10^{-4}. In Table 10, we use the Faure sequence (not scrambled) to compute the price of the down-and-in option using the combined estimator, and display the absolute error of the estimates based on N samples, when $N = 10K, 20K, \ldots, 100K$.

The QMC absolute error stays much lower than the trinomial absolute error after $N = 60K$. It took about one minute to compute the estimates reported in Table 10 using Matlab (35 s up to $N = 60K$) in a 2 GHz Pentium, 2 GB RAM, PC. In contrast, the trinomial procedure used by Broadie et al. [3] required 80,000 time steps, and was reported to be quite computationally intensive.

8. Conclusion

Our numerical results confirm that the combined estimator, which makes use of both the importance sampling and conditional expectation techniques, provides the best error reduction in all option problems considered. If there is no available analytical formula to price the option at the time it crosses the barrier, however, one is left with the importance sampling estimator.

The use of simulated annealing to compute the twisting parameters provided lower error than the heuristic for the majority of the cases considered, but the improvements were very small. The small improvements, together with the simplicity of computing the heuristic values, suggest the heuristic is an efficient strategy to compute the twisting parameters.

Significant error reduction is obtained by the use of (randomized) quasi-Monte Carlo methods, especially in small dimensions. Although we did not attempt to give a direct and detailed numerical comparison of the lattice methods with the quasi-Monte Carlo method in this paper, the numerical results presented in Section 7.3 suggest such a study might be of value.

References

[1] M.H. Alrefaei, S. Andradóttir, A simulated annealing algorithm with constant temperature for discrete stochastic optimization, Management Science 45 (1999) 748–764.
[2] P. Boyle, M. Broadie, P. Glasserman, Monte Carlo methods for security pricing, Journal of Economic Dynamics and Control 21 (1997) 1267–1321.
[3] M. Broadie, P. Glasserman, S. Kou, A continuity correction for discrete barrier options, Mathematical Finance 7 (1997) 325–349.
[4] G.S. Call, D.J. Velleman, Pascal's matrices, The American Mathematical Monthly 100 (1993) 372–376.
[5] P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer-Verlag, 2004.
[6] H.S. Hong, F.J. Hickernell, Algorithm 823: Implementing scrambled digital sequences, ACM Transactions on Mathematical Software 29 (2003) 95–109.
[7] C. Joy, P.P. Boyle, K.S. Tan, Quasi-Monte Carlo methods in numerical finance, Management Science 42 (1996) 926–938.
[8] J. Matoušek, On the L_2-discrepancy for anchored boxes, Journal of Complexity 14 (1998) 527–556.
[9] M. Matsumoto, Y. Kurita, Twisted GFSR generators II, ACM Transactions on Modeling and Computer Simulation 4 (1994) 254–266.
[10] G. Ökten, W. Eastman, Randomized quasi-Monte Carlo methods in pricing securities, Journal of Economic Dynamics and Control 28 (2004) 2399–2426.
[11] G. Ökten, B. Tuffin, V. Burago, A central limit theorem and improved error bounds for a hybrid-Monte Carlo sequence with applications in computational finance, Journal of Complexity 22 (2006) 435–458.
[12] S.M. Ross, Simulation, 2nd ed., Academic Press, 1997.
[13] S.M. Ross, J.G. Shanthikumar, Pricing exotic options: Monotonicity in volatility and efficient simulation, Probability in the Engineering and Informational Sciences 14 (2000) 317–326.
[14] W.M. Tse, L.K. Li, K.W. Ng, Pricing discrete barrier and hindsight options with the tridiagonal probability algorithm, Management Science 47 (2001) 383–393.