Journal of the Operations Research
Society of Japan
Vol. 45, No. 4, December 2002

QUASIRANDOM TREE METHOD FOR PRICING AMERICAN STYLE DERIVATIVES

Hozumi Morohosi
National Graduate Institute for Policy Studies

Masanori Fushimi
Nanzan University

(Received September 27, 2001; Revised February 18, 2002)

Abstract Pricing American options is a difficult task due to the early exercise opportunities, and the higher the dimension, i.e. the number of underlying assets, is, the more complicated the problem is. Broadie and Glasserman proposed a Monte Carlo method for pricing American style options by using random trees. Their method has a merit that its computational complexity is linear in the dimension of the problem, but it often shows a slow convergence. In this paper, we propose a method making use of low-discrepancy sequences instead of random numbers to construct trees, based on which we obtain the estimate of the option price. We will present the detail of our "quasirandom tree method," and compare the random tree method with our method in pricing some American style options of high dimensionality. Our numerical experiments show a rapid convergence of our quasirandom tree method.

1. Introduction

Pricing American options is a current focus of financial engineering research. This problem contains several difficulties due to its early exercise opportunities. Let $\mathbf{S}(t)$ be the price vector of underlying assets, $h(\mathbf{S})$ be the payoff function, and r be the continuous discount factor, then the price of American option at time t is a function of the asset price $\mathbf{S}(t) = \mathbf{x}$ and given by

$$V_t(\mathbf{x}) = \max_{\tau} \mathrm{E}[\mathrm{e}^{-r(\tau-t)} h(\mathbf{S}(\tau)) | \, \mathbf{S}(t) = \mathbf{x}], \tag{1}$$

where maximum is taken over all stopping times $\tau < T$, and T is the maturity of the option.

Many works are devoted to find a good algorithm to compute the price $V_0(\mathbf{S}(0))$. In this paper we present an algorithm based on Broadie and Glasserman's random tree method [2]. Our modification to their method is introducing quasi-Monte Carlo method, which is recently used for many (non-American) option problems and found to be very effective (for the pioneering works of applying quasi-Monte Carlo methods to financial problems, see e.g. [7] or [9]). Random tree method has a nice property that it can provide a confidence interval of the price, but it has a disadvantage in its exponential growth of computational time with respect to the number of exercise opportunities. Although our modification cannot avoid this blow-up of the computational time, it can reduce the rate of the blow-up.

The outline of this paper is as follows. We explain the random tree method in Section 2, then we introduce a quasirandom tree method in Section 3. Some numerical examples are given in Section 4.

2. Random Tree Method

Let $\mathbf{S}(t) = (S_1(t), \ldots, S_n(t))^\top$ denote the prices of n assets at time t ($t = 0, \Delta t, 2\Delta t, \ldots$). In this paper we restrict the model to the discrete time case. Assume that $\mathbf{S}(t)$ follows a

Markov process with the initial state $\mathbf{S}(0)$:

$$\begin{aligned}\mathbf{S}(t+\Delta t) &= \exp[\Delta t(rI - \mathrm{diag}(\delta_1 + \sigma_1^2/2, \ldots, \delta_n + \sigma_n^2/2)) + \\ &\quad \sqrt{\Delta t}\,\mathrm{diag}(\sigma_1 W_1(t), \ldots, \sigma_n W_n(t))]\mathbf{S}(t), \end{aligned} \tag{2}$$

where I is the n-dimensional unit matrix, r is the risk-free interest rate, δ_i is the dividend rate for the i-th asset, σ_i is the volatility of the i-th asset, and $\mathbf{W}(t) = (W_1(t), \ldots, W_n(t))$ is a vector whose components are normal random variables with mean zero and correlation matrix $R = (\rho_{ij})$ $(\rho_{ii} = 1,\ i = 1, \ldots, n)$. The vectors $\mathbf{W}(t)$ and $\mathbf{W}(s)$ are independent if $t \neq s$.

Given a payoff function $h(\mathbf{S})$, the price of an American style option at time t with maturity T is equal to

$$V_t(\mathbf{x}) = \max_{\tau} \mathrm{E}\,[\mathrm{e}^{-r(\tau-t)} h(\mathbf{S}(\tau)) \,|\, \mathbf{S}(t) = \mathbf{x}], \tag{3}$$

where maximum is taken over all the stopping time τ taking the value in $\{0, \Delta t, 2\Delta t, \ldots, N\Delta t (= T)\}$. It is known that the price V_t satisfies the following Bellman equation

$$V_t(\mathbf{x}) = \max\left\{h(\mathbf{x}), \mathrm{E}\,[\mathrm{e}^{-r\Delta t} V_{t+\Delta t}(\mathbf{S}(t+\Delta t)) \,|\, \mathbf{S}(t) = \mathbf{x}]\right\}, \tag{4}$$

with the terminal condition $V_T(\mathbf{x}) = h(\mathbf{x})$. Our aim is to compute the price $V_0(\mathbf{x}_0)$ with the initial condition $\mathbf{S}(0) = \mathbf{x}_0$. The price $V_0(\mathbf{x}_0)$ can be obtained by solving the equation (4) recursively given the terminal condition $V_T(\mathbf{x}) = h(\mathbf{x})$. When we construct an algorithm for solving this problem, the crucial point is to find a good estimator of the successive conditional expectation $\mathrm{E}\,[\mathrm{e}^{-r\Delta t} V_{t+\Delta t}(\mathbf{S}(t+\Delta t)) \,|\, \mathbf{S}(t) = \mathbf{x}]$.

Broadie and Glasserman [2] proposed a method for evaluating American style options by simulation. Their method first generates a random tree, and then computes two estimators based on the tree. Their estimators are called the high estimator and the low estimator, because they are high biased and low biased, respectively. By generating a number of random trees and computing the high and low estimators, they compute the sample mean and variance of each estimator. Then they give the point estimate of the price of the option by the arithmetic mean of the high and low estimators, and the confidence interval based on the sample standard deviations of two estimators. Their method is summarized below.

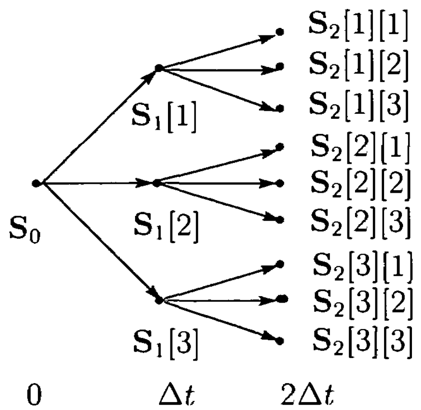


Figure 1: Random tree ($b = 3$)

Random Tree Method
To fix the idea we present the method for the case of American style call options on the maximum of n assets.

Input n: number of assets, N: number of exercise opportunities, Δt: time step, r: risk-free interest rate, δ_i $(i = 1, \ldots, n)$: dividend rate, σ_i $(i = 1, \ldots, n)$: volatility, $R = (\rho_{ij})$ $(i, j = 1, \ldots, n)$: correlation matrix, b: number of branches per node in the tree, $h(\mathbf{S})$: payoff function, in our case $h(\mathbf{S}) = (\max_{i=1,\ldots,n} S_i - K)^+$, where K is the strike price of the option and $(x)^+ = \max\{x, 0\}$.

Random Tree A random tree is a rooted tree and a realization of the Markov process (2). (See Figure 1.) The nodes in a random tree are associated with the arrays of the n-dimensional vectors. A random tree can be identified with a collection of vector arrays:

$$\{\mathbf{S}_0, \mathbf{S}_1[b], \ldots, \mathbf{S}_N \overbrace{[b] \ldots [b]}^{N}\}.$$

The indices of the array represent the generation process of the tree, and this point will be explained in detail below.

The n-dimensional vectors $\{\mathbf{S}_k[i_1] \ldots [i_k] | \, i_1, \ldots, i_k = 1, \ldots, b\}$ contain the prices of n assets at time $k\Delta t$. Each node in the tree has the same number of branches b. Each branch has a direction, and this direction represents the time transition. A node at the head of a branch is considered generated from the tail node. The first vector $\mathbf{S}_0$, associated to the root, contains the initial prices of n assets. Given the k-th vector $\mathbf{S}_k[i_1] \ldots [i_k] = (S_k[i_1] \ldots [i_k](j))_{j=1}^n$ at time $k\Delta t$ for fixed $i_1, \ldots, i_k$, the $(k+1)$-th vector $\mathbf{S}_{k+1}[i_1] \ldots [i_{k+1}] = (S_{k+1}[i_1] \ldots [i_{k+1}](j))_{j=1}^n$ $(i_{k+1} = 1, \ldots, b)$ is generated according to (2) under the condition $\mathbf{S}(k\Delta t) = \mathbf{S}_k[i_1] \ldots [i_k]$ at time $k\Delta t$ as follows.

$$S_{k+1}[i_1] \ldots [i_k][i](j) = S_k[i_1] \ldots [i_k](j) \exp\left[(r - \delta_j - \sigma_j^2/2)\Delta t + \sigma_j \sqrt{\Delta t} W_i(j)\right] \quad (5)$$

$(i = 1, \ldots, b;\ j = 1, \ldots, n)$, where $\mathbf{W}_i = (W_i(1), \ldots, W_i(n))$ are normal random vectors with the components having correlation matrix R. For each branch generation independent random vectors $\mathbf{W}_i$ are used.

One random tree produces a pair of high and low estimates by the procedure described below. In order to obtain a reliable estimate of the price, we must compute the sample mean of each estimate by independent replications of random trees.

High Estimator Compute the values of $\Theta_0, \Theta_1[b], \ldots, \Theta_N[b] \ldots [b]$ backward recursively as follows. First we set

$$\Theta_N[i_1] \ldots [i_N] = h\left(\mathbf{S}_N[i_1] \ldots [i_N]\right), \quad (6)$$

and then

$$\Theta_k[i_1] \ldots [i_k] = \max\left\{ h\left(\mathbf{S}_k[i_1] \ldots [i_k]\right), \frac{1}{b} \sum_{j=1}^{b} \mathrm{e}^{-r\Delta t} \Theta_{k+1}[i_1] \ldots [i_k][j] \right\} \quad (7)$$

for $k = N-1, N-2, \ldots, 0$. Let $\bar{\Theta}_0$ be the sample mean of independent replications of Θ_0, and $s(\Theta_0)$ be its sample standard deviation.

Low Estimator Compute the values of $\theta_0, \theta_1[b], \ldots, \theta_N[b] \ldots [b]$ backward recursively as follows. Let

$$\theta_N[i_1] \ldots [i_N] = h\left(\mathbf{S}_N[i_1] \ldots [i_N]\right). \quad (8)$$

Next define intermediate variables $\eta_k[i_1]\dots[i_k][j]$ as

$$\eta_k[i_1]\dots[i_k][j] = \tag{9}$$
$$\begin{cases} h(\mathbf{S}_k[i_1]\dots[i_k]) & \text{if} \quad h(\mathbf{S}_k[i_1]\dots[i_k]) \geq \frac{1}{b-1}\sum_{i=1,i\neq j}^{b} \mathrm{e}^{-r\Delta t}\theta_{k+1}[i_1]\dots[i_k][i] \\ \mathrm{e}^{-r\Delta t}\theta_{k+1}[i_1]\dots[i_k][j] & \text{otherwise} \end{cases}$$

for $j = 1, \dots, b$. Then let

$$\theta_k[i_1]\dots[i_k] = \frac{1}{b}\sum_{j=1}^{b} \eta_k[i_1]\dots[i_k][j], \tag{10}$$

for $k = N-1, \dots, 0$. Let $\bar{\theta}_0$ be the sample mean of independent replications of θ_0, and $s(\theta_0)$ be its sample standard deviation.

Point Estimate and Confidence Interval The point estimate for $V_0(\mathbf{S}_0)$ of an American style call option on the maximum of n assets is given by the mean of high and low estimators $(\bar{\Theta}_0 + \bar{\theta}_0)/2$. Its confidence interval is given by

$$\left[\max\{(S_0 - K)^+, \bar{\theta}_0 - \frac{z_{\alpha/2}s(\theta_0)}{\sqrt{M}}\}, \bar{\Theta}_0 + \frac{z_{\alpha/2}s(\Theta_0)}{\sqrt{M}}\right], \tag{11}$$

where $S_0 = \max_{1\leq j\leq n}\{S_0(j)\}$, and $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution, and M is the number of replications of random trees.

The random tree method has an advantage of easy implementation. Once an asset model and a payoff function are given, even if the model is very complicated, we can immediately apply this method to the model and obtain an estimate of the option price with a confidence interval. Nevertheless this method has its drawback in the aspect of computational complexity. The running time of the random tree algorithm is proportional to the number of nodes. Since the number of nodes in the tree is given by $\sum_{j=0}^{N} b^j = (b^{N+1}-1)/(b-1)$, the running time of the algorithm has the order $O(n^2 b^N)$ for one replication. The time complexity $O(n^2)$ is required for generating n correlated normal random numbers. This result shows that this algorithm becomes impractical as the number of exercise opportunities(N) or the number of branches per node(b) grows large.

When the number of exercise opportunities is given beforehand, our main efforts are put in reducing the number of branches. Since the estimates Θ and θ are the approximations for the conditional expectations, these computations are essentially approximate integrals in n-dimensional space, where n is the number of assets. As the dimension of the space grows large, numerical integration problems encounter a difficulty of increasing number of sample points. When we solve the model with many assets, we need a large number of branches to obtain the estimate with good accuracy. Recently quasi-Monte Carlo method is widely used for high-dimensional numerical integration problems, and achieves great successes. In the next section we propose to use low-discrepancy point sets to reduce the number of branches.

3. Quasirandomization

The random tree method uses n-dimensional normal random vectors $\{\mathbf{W}_k\}$ to generate the branches of the tree. Our basic idea is to replace random vectors with a low-discrepancy point set.

One problem that we should consider when we utilize a low-discrepancy point set is how to estimate the error of the calculated value. In the random tree method the confidence interval, which represents the error of the computed value, is given in terms of the sample standard deviations of the estimates, and the sample standard deviations are derived from replications of random trees. So we need several independent replications of trees to give error estimations. To this end, we utilize randomization techniques for quasi-Monte Carlo methods. These techniques allow us to produce a set of independent low-discrepancy sequences. More specifically, our procedure for generating "quasirandom trees" is as follows.

Quasirandom Tree Let $\{\mathbf{Z}_i | i = 1, \ldots, b\}$ be a low-discrepancy point set.

Given the k-th vector $\tilde{\mathbf{S}}_k[i_1]\ldots[i_k] = (\tilde{S}_k[i_1]\ldots[i_k](j))_{j=1}^n$ at time $k\Delta t$, then the $(k+1)$-th vector $\tilde{\mathbf{S}}_{k+1}[i_1]\ldots[i_{k+1}] = (\tilde{S}_{k+1}[i_1]\ldots[i_{k+1}](j))_{j=1}^n$ is determined by

$$\tilde{S}_{k+1}[i_1]\ldots[i_k][i](j) = \tilde{S}_k[i_1]\ldots[i_k](j)\exp\left[(r - \delta_j - \sigma_j^2/2)\Delta t + \sigma_j\sqrt{\Delta t}\tilde{W}_i(j)\right] \quad (12)$$

$(i = 1, \ldots, b;\ j = 1, \ldots, n)$. Here we introduce the vectors $\tilde{\mathbf{W}}(i) = (\tilde{W}_1(i), \ldots, \tilde{W}_n(i))^\top$ $(i = 1, \ldots, b)$ which are generated by using the Cholesky factorization $R = AA^\top$ of the correlation matrix and the inverse of standard normal distribution function ϕ^{-1} as

$$\tilde{\mathbf{W}}(i) = A(\phi^{-1}(\tilde{Z}_1(i)), \ldots, \phi^{-1}(\tilde{Z}_n(i)))^\top. \quad (13)$$

The vectors $\tilde{\mathbf{Z}}(i) = (\tilde{Z}_1(i), \ldots, \tilde{Z}_n(i))^\top$ $(i = 1, \ldots, b)$ are the key of our algorithm, and are generated from $\{\mathbf{Z}_i | i = 1, \ldots, b\}$ as

$$\tilde{\mathbf{Z}}(i) = \mathbf{Z}_i + \mathbf{U} \pmod{\mathbf{1}}, \quad (14)$$

where $\mathbf{U}$ is a uniform random vector in the unit cube $[0, 1)^n$ and (mod $\mathbf{1}$) means component-wise modulo one operation. For each branch generation we use an independent random vector $\mathbf{U}$.

In the above procedure we can produce independent quasirandom trees. The low and high estimators are computed in the same way as in the random tree method.

Remark In the following experiments, we use a (t, m, s)-net as a low-discrepancy point set. For the detailed definition of (t, m, s)-net refer to [6]. The randomization technique used above is originally proposed in [4] for good lattice point methods. In [5] we apply this randomly shifting technique to the error estimation of the integration by (t, m, s)-net, and compare it with the scrambling method proposed by Owen [8] from theoretical and experimental viewpoints. In [5] we described the error estimation method for the numerical integral of multidimensional function $f(\mathbf{x})$ as follows. First we select low-discrepancy point sets $\{\mathbf{x}_i^{(j)} | i = 1, \ldots, b\}$, $j = 1, \ldots, M$, independently, and compute the approximate integral value for each point set,

$$I^{(j)} = \frac{1}{b}\sum_{i=1}^{b} f(\mathbf{x}_i^{(j)}), \quad j = 1, \ldots, M, \quad (15)$$

then we calculate the estimate of the integral $\int f(\mathbf{x})d\mathbf{x}$ by

$$\hat{I} = \frac{1}{M}\sum_{j=1}^{M} I^{(j)}. \quad (16)$$

The standard error of $\hat{I}$ is estimated using the sample standard deviation of the evaluated values.

$$\hat{\sigma} = \sqrt{\frac{1}{M(M-1)}\sum_{j=1}^{M}(I^{(j)} - \hat{I})^2}. \quad (17)$$

It should be noted that a randomly shifted (t, m, s)-net is not necessarily a (t, m, s)-net, but it remains a low-discrepancy point set.

Recently Boyle et al. [1] proposed a low discrepancy mesh method for pricing American style options. Their method can be considered as a quasirandom version of the stochastic mesh method by Broadie and Glasserman [3]. The low discrepancy mesh method avoids the exponential growth of computational complexity with respect to the number of exercise opportunities by generating the fixed number of nodes at each exercise time point under one approximate probability distribution of the asset price. This method is appropriate for the case when we can easily obtain a good approximation to the probability distribution of the asset price at each time point. On the other hand our method does not require the each probability distribution explicitly.

4. Numerical Results

We performed numerical experiments on American style call options on the maximum of n assets. Recall that the payoff function of this option is given by

$$h(\mathbf{S}) = (\max_{i=1,\dots,n} S_i - K)^+,$$

where $(x)^+ = \max\{x, 0\}$ and K is the strike price. As a low-discrepancy point set $\{\mathbf{Z}_i\}$ we adopt the first b points of Faure sequence. By using M independent randomly shifted point sets $\{\tilde{\mathbf{Z}}^{(r)}(i)\}$ $(r = 1, \dots, M)$ instead of random vectors in the random tree method, we compute M pair of estimates $\Theta_0^{(r)}$ and $\theta_0^{(r)}$ $(r = 1, \dots, M)$ and set

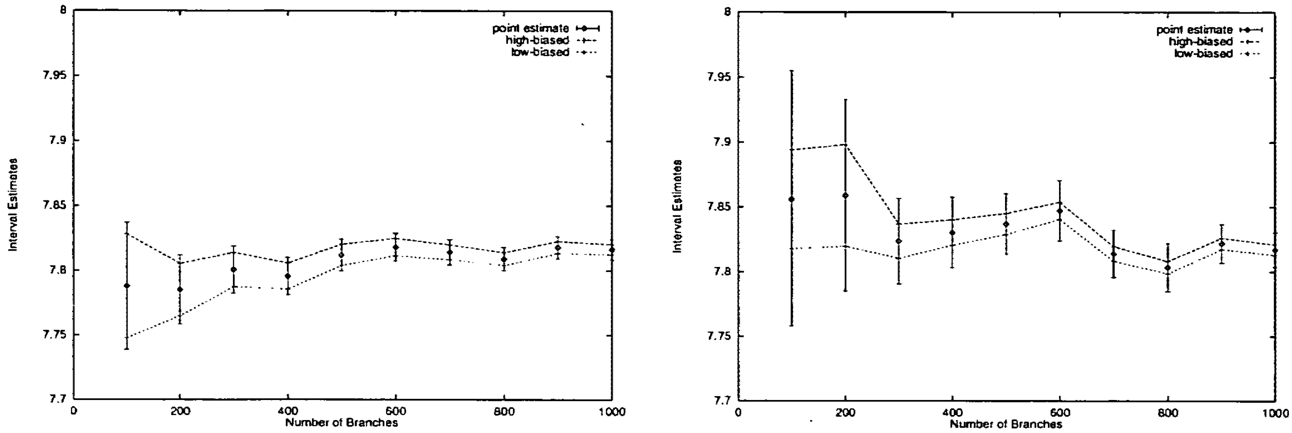
$$\bar{\Theta}_0 = \frac{1}{M}\sum_{r=1}^{M}\Theta_0^{(r)}, \quad \bar{\theta}_0 = \frac{1}{M}\sum_{r=1}^{M}\theta_0^{(r)}. \tag{18}$$

The point estimate of the price of the option at time 0 is given by $\bar{V}_0 = \frac{1}{2}(\bar{\Theta}_0 + \bar{\theta}_0)$. Let $s(\Theta_0)$ and $s(\theta_0)$ be the sample standard deviations of Θ_0 and θ_0, respectively. The confidence interval is defined as in (11).

Example 1 (five assets).

We choose $K = 100$, $r = 0.05$, $\delta_i = 0.1$ $(i = 1, \dots, n)$, $\sigma_i = 0.2$ $(i = 1, \dots, n)$, $\rho_{ij} = 0.3$ $(i, j = 1, \dots, n,\ i \neq j)$, $T = 1$, $N = 3$, and $n = 5$. These parameters are used in [2]. We set $\mathbf{S}_0 = (90, \dots, 90)$. We compute $\bar{\Theta}_0$ and $\bar{\theta}_0$ for the different number b of branches per node ranging between 100 and 1000. The confidence interval is computed based on 100 replications of trees $(M = 100)$. The result is shown in Figure 2. The result of the quasirandom tree method is shown in the left graph, and the random tree method is shown in the right. In the graph, the horizontal axis shows the number b of branches per node, and the vertical axis shows the estimated price; the point estimates of the price are shown by $\diamond$, high and low estimates are shown by dashed lines, and the vertical bars on the points shows the confidence intervals given by (11).

By comparing the graphs we find that the quasirandom tree method gives smaller confidence intervals than the random tree method (The width of the confidence interval of the random tree estimation is about 3 to 5 times that of the quasirandom tree estimation), and that for each fixed b the confidence interval of the quasirandom tree method is overlapped by that of the random tree method. All the confidence intervals of both methods are contained in the confidence interval [7.674, 8.069] shown by [2]. Moreover the estimates by the quasirandom tree method show more stable convergence than that by the random tree method. Although the true price of this option is not known, we consider these facts show



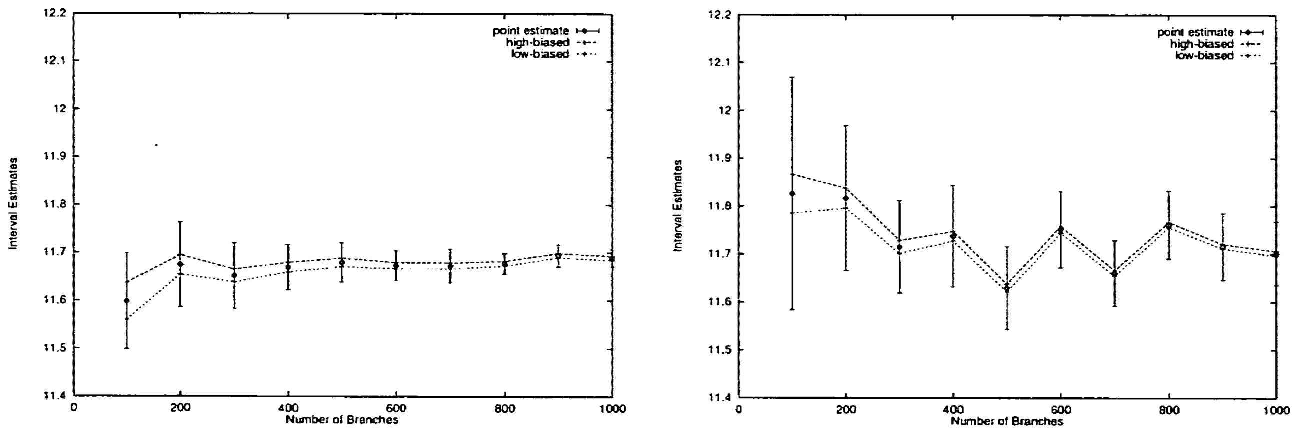
(a) Quasirandom tree method (b) Random tree method

Figure 2: Comparison of quasirandom and random methods (five assets)

that the quasirandom tree method gives reliable confidence intervals and estimates for the option price.

Example 2 (ten assets).

All the parameters are the same as in Example 1 except the number of assets $n = 10$. The result is shown in Figure 3. In this example we have a similar result as Example 1. The quasirandom tree method gives narrower confidence intervals and more stable convergence of estimates than random tree method.

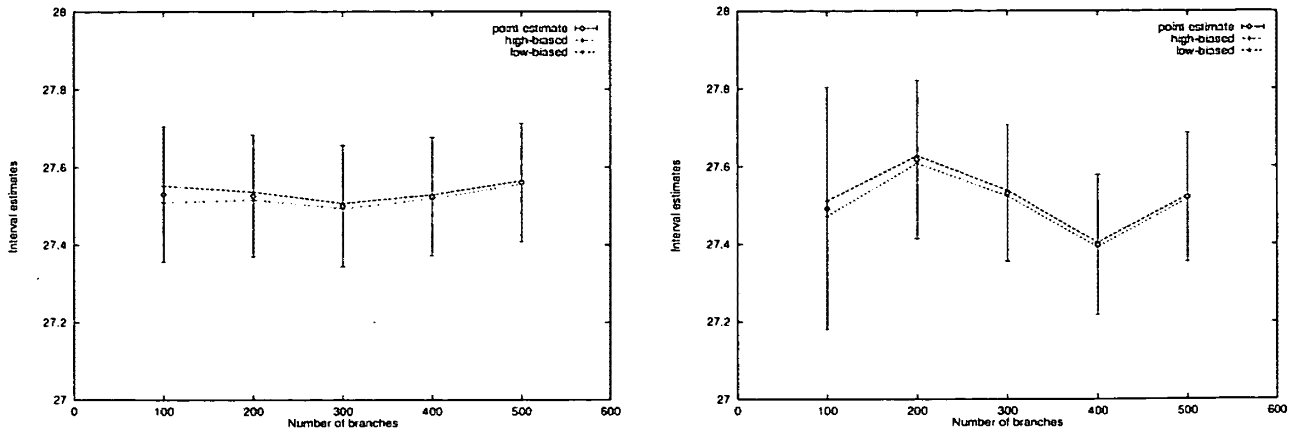


(a) Quasirandom tree method (b) Random tree method

Figure 3: Comparison of quasirandom and random methods (10 assets)

Example 3 (one hundred assets).

In this experiments the number of assets is $n = 100$. The number of replications is $M = 50$, and the number of branches per node b is in the range $100 \leq b \leq 500$ due to computational time limitation. In the experiment for 100 assets, the quasirandom tree method gives relatively smaller confidence interval than the random tree method and the intervals given by both methods overlap for each other. We consider the confidence intervals given by both methods are reliable. But the convergence of the confidence interval with respect to the increase of the number b of branches per node, seems to be very slow. It is



(a) Quasirandom tree method (b) Random tree method

Figure 4: Comparison of quasirandom and random methods (100 assets)

necessary to investigate the cases for larger values of b, which is almost impossible at present due to computational time limitation. This may be a future work.

5. Conclusion

We introduced a (randomized) quasirandom modification for the random tree method of Broadie and Glasserman, and compared the quasirandom and the random tree methods by numerical experiments. In all the experiments the quasirandom method gives smaller error estimates than the random method, and the running times of both methods are almost the same.

Although the quasirandom tree method can reduce the number of branches per node, it inherits from the random tree method a demerit of exponential growth of the computational time with respect to the number of exercise opportunities. This drawback limits the application of the methods. On the other hand the quasirandom tree method as well as the random tree method has a merit that we can easily implement them even for complicated stochastic assets models. For such models, we consider the quasirandom tree method can be a good choice for the pricing.

Acknowledgement

The authors are grateful to the anonymous referees for their valuable comments. The first author was supported by Grant-in-Aid for Encouragement of Young Scientists, Japan Society for the Promotion of Science, and the second author was supported by Pache Subsidiary I–A, Nanzan University.

References

[1] P. P. Boyle, A. W. Kolkiewicz, and K. S. Tan: Pricing American style options using low discrepancy mesh methods. working paper, Department of Statistics and Actuarial Science, University of Waterloo, 2000.

[2] M. Broadie, and P. Glasserman: Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, **21**(1997), 1323–1352.

[3] M. Broadie, and P. Glasserman: A stochastic mesh method for pricing high-dimensional American options. working paper, Columbia Business School, Columbia University, 1997.

[4] R. Cranley, and T. N. L. Patterson: Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, **13**-6(1976), 904–914.

[5] H. Morohosi, and M. Fushimi: A practical approach to the error estimation of quasi-Monte Carlo integrations. H. Niederreiter and J. Spanier(eds.): *Monte Carlo and Quasi-Monte Carlo Methods 1998* (Springer-Verlag, Berlin, Heidelberg, 2000), 377–390.

[6] H. Niederreiter: *Random Number Generation and Quasi-Monte Carlo Methods* (SIAM, Philadelphia, 1992).

[7] S. Ninomiya, and S. Tezuka: Toward real-time pricing of complex financial derivatives. *Applied Mathematical Finance*, **3**(1996), 1–20.

[8] A. B. Owen: Monte Carlo variance of scrambled net quadrature. *SIAM Journal on Numerical Analysis*, **34**-5(1997), 1884–1910.

[9] S. H. Paskov, and J. F. Traub: Faster valuation of financial derivatives. *Journal of Portfolio Management*, **22**-1(1995), 113–120.

Hozumi Morohosi
National Graduate Institute for Policy Studies
2-2 Wakamatsu-cho, Shinjuku-ku,
Tokyo 162-8677, Japan
E-mail: morohosi@grips.ac.jp