Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

A NEW EFFICIENT SIMULATION STRATEGY FOR PRICING PATH-DEPENDENT OPTIONS

Gang Zhao
Yakun Zhou

Center for Information & Systems Engineering
Department of Manufacturing Engineering
Boston University
Boston, MA 02446, U.S.A.

Pirooz Vakili

Center for Information & Systems Engineering
Department of Manufacturing Engineering
Boston University
Boston, MA 02446, U.S.A.

ABSTRACT

The purpose of this paper is twofold. First, it serves to describe a new strategy, called Structured Database Monte Carlo (SDMC), for efficient Monte Carlo simulation. Its second aim is to show how this approach can be used for efficient pricing of path-dependent options via simulation. We use efficient simulation of a sample of path-dependent options to illustrate the application of SDMC. Extensions to other path-dependent options are straightforward.

1 INTRODUCTION

The purpose of this paper is twofold. First, it serves to describe a new strategy, called Structured Database Monte Carlo (SDMC), for efficient Monte Carlo simulation. The word "strategy" is deliberately chosen to emphasize that the approach is not designed to address a specific problem, using a specific method, but rather a new way of designing and implementing MC algorithms. Its second aim is to show how this approach can be used for efficient pricing of path-dependent options via simulation. We use efficient simulation of a sample of path-dependent options to illustrate the application of SDMC. Extensions to other path-dependent options are straightforward.

In Monte Carlo simulation, efficient strategies seek to reduce the variance of the MC estimator and they are generally referred to as variance reduction techniques. It is interesting and noteworthy that the small and delightful 1964 book of Hammersley and Handscomb (Hammersley and Handscomb 1964) has remained a basic reference on Monte Carlo methods and variance reduction techniques to this date and is regularly referred to in research publications. This points to the fact that many of the main strategies for improving efficiency and for variance reduction date far back. Many of the results to follow have consisted of adaption of these techniques to specific domains of application. It is well established that "the greatest gains in efficiency from variance reduction techniques result from exploiting specific features of a problem, rather than from generic applications of generic methods." (Glasserman 2004) Most methods rely on discovering such specific features for each problem, one problem at a time, and depend heavily on the ingenuity of the user of the technique.

The point of departure of SDMC is an attempt to develop effective variance reduction techniques that are in fact generic methods and can be generically applied. We rely on basic techniques of variance reduction such as *stratification*, *control variate*, and *importance sampling*, to name a few. The novelty of SDMC is in providing generic methods for defining strata, control variates, and new sampling measures when the above techniques are used. As will be described in the paper, SDMC relies on information obtained from sample prices (sample performances more generally) at one parameter value, say θ_0, to define strata (assume stratification is being used) at all "neighboring" θ. The method is not demanding on user ingenuity.

The idea of using information from paths at a neighboring parameter value to estimate a quantity at another parameter value is at the heart of the method of Perturbation Analysis (PA) used to derive performance sensitivity information (see, e.g., Ho and Cao 1991 and Glasserman 1991). SDMC, while different in its objective and application from PA, uses a similar principle and is indebted to developments in PA. Perturbation Analysis, closely related to the variance reduction technique of common random numbers, pairs paths whose "input" variables are only slightly different (have small perturbations). In the SDMC approach paths at two parameter values are also paired via their reference to the same element of the database. The goal of SDMC, however is to obtain information about the global dependence of the sample performance on sample paths while PA seeks to obtain local information about the dependence of the performance on parameter values.

The rest of the paper is organized as follows. A review of efficient estimation strategies is provided in Section 2 in order to provide a context for positioning the SDMC approach. SDMC method is introduced in Section 3. Section

1-4244-0501-7/06/$20.00 ©2006 IEEE

4 describes how stratification techniques can be used in the context of a structured database. Section 5 provides experimental results results when SDMC is applied to a number of path dependent options.

2 A REVIEW OF EFFICIENT ESTIMATION STRATEGIES

In general, estimation (approximation/evaluation) via MC simulation corresponds to estimating the expected value of an appropriately defined random variable. Specifically, let $(\Omega, \mathcal{F}, P)$ be a probability space, X a random element of Ω corresponding to the probability measure P (i.e., for all $A \in \mathcal{F}$, $P(X \in A) = P(A)$), and $f(.;\theta) : \Omega \to R$ a parametric family of functions defined on Ω ($\theta \in \Theta$). Let $Y(\theta) = f(X, \theta)$ and define $J(\theta)$ as

$$\begin{aligned} J(\theta) &= E[Y(\theta)] = E[f(X;\theta)] \\ &= \int_\Omega f(X;\theta)dP = \int_\Omega Y(\theta)dP. \end{aligned}$$

The objective is to evaluate $J(\theta)$ efficiently.

Assume that Ω is the path space of a stochastic process, $f(.;\theta)$ is a function that assigns a real number to each path, say the sample payoff, and θ is a parameter of the problem. In simulation there is a chain of transformations that takes place before a path is generated: A set of i.i.d. uniform (0,1) random variables are sampled, these are transformed into a set of more general non-uniform variates from which the path, X, is generated and finally $Y = f(X;\theta)$ is evaluated. Let us explicitly write out these transformations. Let $[0,1]^d$ be the d-dimensional hypercube where the uniform[0,1]'s reside (theoretically, d may be ∞). Let $\Omega^{'}$ denote the space of non-uniforms. Then we have

$$[0,1]^d \xrightarrow{T} \Omega^{'} \xrightarrow{K} \Omega \xrightarrow{f(.;\theta)} R.$$

$J(\theta)$ can be written as

$$\begin{aligned} J(\theta) &= \int_{[0,1]^d} f_1(U;\theta)dP_1 \\ &= \int_{R^d} f_2(W;\theta)dP_2 = \int_\Omega f(X;\theta)dP. \end{aligned}$$

where the random elements U and W and measures P_1 and P_2 are appropriately defined.

Our purpose for specifying the above equivalent integrals is to point out that different methods that seek to estimate $J(\theta)$ do not adopt the same domain as their basic domain of integration. Efficient estimation strategies/methods depend critically on exploiting the structure of the function to be integrated and/or the structure of the underlying domain. As will be seen, adopting the elements of one of the above domains as the primitives of the method has important implications as to which function is supposed to provide the structure to be utilized. Primitives for some methods are the underlying uniforms, for some they are the non-uniform samples, and for others, they are the final paths. As will be discussed below, the primitives in the SDMC approach in most cases will be the paths.

We briefly review some of the main existing methods of approximation/estimation. Specifically we consider *numerical integration* (see, e.g., Krommer and Ueberhuber 1998), *basic Monte Carlo* (see, e.g., Bratley, Fox, and Schrage 1987, or Glasserman 2004), *Quasi-Monte Carlo* (see, e.g., Niederreiter 1992, or Fox 1999) and three basic *variance reduction techniques* for Monte Carlo (see, e.g., Bratley, Fox, and Schrage 1987, or Glasserman 2004).

With some abuse of notation let us use the same set of notations for all methods in order to highlight the similarity between their algorithms and be able to see the difference between the strategies they utilize. Let

$$J = \int_\Omega f(X)dP,$$

and $X_1, \cdots, X_n$ be a random sample from Ω, set $Y_i = f(X_i)$. Let $\underline{X} = (X_1, \cdots, X_n)$ and $\underline{Y} = (Y_1, \cdots, Y_n)$.

All methods use an estimator of the following form

$$\hat{Y}(n) = w_1 Y_1 + \cdots + w_n Y_n = \sum_{i=1}^{n} w_i Y_n.$$

The difference between the methods is in how they select the samples $X_1, \cdots, X_n$ in Ω and in their choice of weights $\underline{w} = (w_1, \cdots, w_n)$. We now comment on each of the methods, their choice of weights and samples, and their effectiveness.

1. *Numerical integration.* It is very effective when $\Omega = [0,1]$ (or a subset of R). The X_i's are constrained to deterministic subintervals of $[0,1]$, i.e., they are carefully selected. Selection of w_i's are determined by different functional approximations to f and depend heavily on the sampled values of f, namely on $\underline{Y}$. This method has the fastest rate of convergence for low dimensional d but rapidly becomes inefficient (or infeasible) as d increases. In this case the structure of the domain $[0,1]^d$ plays an important role; the method attempts to cover $[0,1]^d$ closely while at the same time adapting to f. The tension between these two goals and, in particular, the desire to cover $[0,1]^d$ closely makes this strategy ultimately inefficient for large d.
2. *Standard Monte Carlo.* This method is available for very general Ω. X_i's are randomly selected from Ω. $w_i = \frac{1}{n}$ are independent of $\underline{Y}$. The rate

of convergence is the slow rate of $O(\frac{1}{\sqrt{n}})$. In this case there is no attempt to cover all regions of Ω closely or any attempt to adapt the sampling to the important areas, i.e., no attempt to take the structure of f into account. It is these deficiencies of the standard Monte Carlo that the variance reduction techniques seek to remedy.

3. *Quasi-Monte Carlo.* The domain of Quasi-MC is in an important way restricted to $[0,1]^d$. X_i's are carefully selected from $\Omega = [0,1]^d$ so as to form a low discrepancy sequence. This is an approach to ensure that $[0,1]^d$ is closely sampled. $w_i = \frac{1}{n}$ are independent of $\underline{Y}$. The rate of convergence of Quasi-Monte Carlo deteriorates as d increases. The innovation in this method is its strategy to cover $[0,1]^d$ intelligently and with a smaller number of samples (compared to numerical integration). It does not adapt itself to the function f and in fact, similar to the standard Monte Carlo, it ignores f completely.
4. *Variance Reduction techniques.* There are quite a number of techniques for reducing the variance of the standard MC. We limit ourselves to discussing three of the basic techniques. Most of these techniques can be used in conjunction with each other; they are also intimately related to each other where, from an appropriate perspective, one can be viewed as a subset of the other.
   - *Control variate.* X_i's are randomly selected from Ω. w_i's depend on $\underline{Y}$ and samples of a secondary variable Z called the control variate. The effectiveness of the technique depends entirely on the correlation between Z and Y (higher correlation, more effective). Utilizing a control variate Z can be viewed as an indirect attempt to capture some of the underlying structure of f. The selection of X_i's and hence the corresponding Z_i's are not controlled. There is no generic method for selecting effective control variates. User ingenuity is the key here.
   - *Stratified sampling.* X_i's are sampled in such a way that a specified number of samples from each stratum is selected (hence this method applies some control over the choice of X_i's, similar to numerical integration). $w_i = \frac{1}{n}$ if standard stratified sampling is used; the weights are adjusted by a discrete likelihood ratio if a sampling scheme other than proportional sampling induced by the probability measure P is used. Straightforward stratification has some similarity with the strategy of numerical integration in its desire to cover the domain. On the other hand, it is significantly more flexible, both in aiming for a far coarser coverage, and in being able to adapt to the structure of f. Again, the choice of the strata is problem dependent and at the discretion of the user.
   - *Importance sampling.* X_i's are randomly selected according to a different probability measure Q. The probability measure Q can be viewed as an indirect way of controlling the choice of X_i's so as to take the underlying structure of f into account. w_i's are likelihood ratios and are used to remove the bias due to sampling from Q. Importance sampling is one of the most sophisticated and effective variance reduction techniques. However, using Q can be viewed as an indirect way of tilting/biasing the sampling towards the most "important" samples. The effectiveness and the degree of control over sampling important regions can be improved if it is used in conjunction with Stratified Sampling.

Having reviewed the above techniques we are prepared to introduce the SDMC method.

3 STRUCTURED DATABASE MONTE CARLO

SDMC aims to capture/identify the "structure" of the function f at a nominal parameter value θ_0 and to use this knowledge in designing more effective variance reduction techniques when estimating $J(\theta)$ at "neighboring" values of θ.

The following are the basic steps of the algorithm. First, the primitives of the simulation need to be selected. (In examples to follow, primitives are paths of the standard Brownian motion). Next a large database of the primitives needs to be generated. The most straightforward approach is to generate the primitives from the given probability measure defined on the set of primitives. (Sampling according to a more general user defined measure is possible.)

Let us denote the database by Ω_N where $N = |\Omega_N|$ is the size of the database. The probability space $(\Omega_N, 2^{\Omega_N}, P_N)$ where P_N is the uniform measure is now the basic probability space of our estimation problem. Note that we have changed the estimation problem to an approximate version of its original form. Namely, we are now interested in estimating

$$J_1(\theta) = E[f(X_1;\theta)] = \frac{1}{N}\sum_{i=1}^{N} f(\omega_i;\theta).$$

where X_1 is a random element of Ω_N selected uniformly. For large N, $J_1(\theta)$ approximates $J(\theta)$ closely.

Once the database of primitives is generated, $f(.,\theta_0)$ is used to "structure" the database. The appropriate structure may depend on the method of variance reduction to be used. In what follows, we impose a linear order on the database using $f(.,\theta_0)$ or a function closely related to $f(.,\theta_0)$. This linear order induces some homogeneity of function values, i.e., values that have close database indices ($|i-j| < k$ for "small" k), have "close" function values ($|f(\omega_i;\theta_0) - f(\omega_j;\theta_0)| < a$ for "small" a). If the sample performance is continuous with respect to θ, the homogeneity induced survives when θ is perturbed.

The last step of SDMC is to use the imposed structure to design effective variance reduction techniques. The implementation of this step depends on the variance reduction technique being used. In this paper we illustrate one possible implementation of this step when stratification technique is used.

The above basic steps of the approach are summarized in the following:

1. **Data base generation**: Generate a "large" set of samples (paths) from Ω according to the probability measure P. Let $\{\omega_1, \cdots, \omega_N\}$ denote the set of paths generated. From now on we refer to this finite set of paths as the database and denote it by DB.
2. **Structuring the database DB**: Induce a linear order on the database DB according to the values $f(\omega,\theta_0)$. In other words,

$$\omega_i \leq \omega_j \Leftrightarrow f(\omega_i,\theta_0) \leq f(\omega_j,\theta_0).$$

3. **Simulation/sampling at** $\theta \neq \theta_0$: Sample from the database DB, taking into account the structure of the database. (We expect that the structure remains approximately unperturbed if θ is close to θ_0.)

Before making some general comments about SDMC strategy we provide an example by way of a graph to illustrate what we hope to gain from this approach.

This example refers to pricing an arithmetic Asian option. At this point, however, the specifics of the problem are less important. One can view the problem as follows. 25000 sample paths are generated. They are ordered based on the payoff of the option at the volatility parameter $\sigma = 0.2$. The points on the horizontal axis correspond to sample paths. Now assume that we wish to solve the estimation problem at $\sigma = 0.1$ or $\sigma = 0.3$. In this case, as can be seen from Figure 2, the structure that is induced on the database is to a great extent maintained for $\sigma = 0.1$ and $\sigma = 0.3$. While the value of the path payoffs when $\sigma = 0.1$ or $\sigma = 0.3$ are not known (before sampling), by viewing the samples at $\sigma = 0.2$ we already know which paths are important for $\sigma = 0.1$ and $\sigma = 0.3$ (those to the right of the axis). Moreover, the monotonicity induced by paths at $\sigma = 0.2$ is in some sense maintained for $\sigma = 0.1$ and $\sigma = 0.3$. This additional information about the underlying domain, as we will see shortly, can be exploited very effectively to design very efficient stratification algorithms. It is important to note that in this example, to generate each path, 64 random variables are used. Therefore, the estimation problem can be viewed as the evaluation of a 64-dimensional integral. The ordering induced by the paths at $\sigma = 0.2$ has in effect turned the problem into that of integrating two single variable functions whose graphs are depicted above.

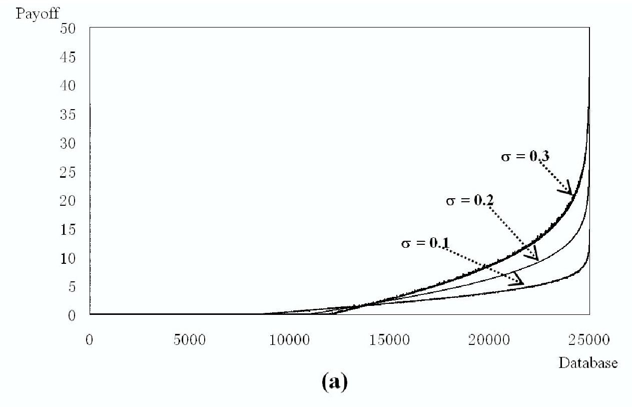

Figure 1: Database Ordered for $\sigma = 0.2$, Payoff for $\sigma = 0.1, 0.2, 0.3$

We make the following general comments about the SDMC strategy.

1. As mentioned earlier "the greatest gains in efficiency from variance reduction techniques result from exploiting specific features of a problem, rather than from generic applications of generic methods." Most methods rely on discovering such specific features for each problem, one problem at a time. We rely on the problem itself to reveal the structure via inducing an order on the database.
2. The SDMC strategy implies two setup costs that may not be minimal. The first is that of generating the database and the second is that of ordering it. There are important classes of problems for which the database can be generated once and for all. Consider classes of stochastic processes that are driven by vectors of Brownian motion. For example, many models in mathematical finance and many in statistical physics fall into this category. The cost of ordering and reordering the database can not be avoided. The issue to be explored is the extent of utility of a database ordered at θ_0 as

θ deviates from θ_0. One expects the answer to be problem dependent. It should be stressed that in many instances the reordering of the database can take place "off-line" and during the "downtime" of the estimation problem.

3. It is worth noting that the perspective of SDMC is closer to the perspective of *Lebesgue integration* than that of *Riemann integration* prevalent in all the methods we reviewed above. This difference is a key point of departure. Let us clarify this critical contrast. A recent book on computational integration states "Many mathematical disciplines (such as probability theory, statistics, or functional analysis) rely heavily on the concept of Lebesgue integration. However, the definition of Lebesgue integrals is inherently nonconstructive, which is why Lebesgue integration is important only in mathematical theory. The concept relevant to computational practice is the more restricted, but constructive concept of Riemann integration," (see, Krommer and Ueberhuber 1998). We contend that the Lebesgue perspective can, in fact, be computationally very beneficial.

   In Lebesgue integration it is the range of the function that provides the structure and there is less emphasis on the topology of the domain. The pullback by the function f of the Borel sets in the range, i.e., the well behaving real line R, are the relevant sets in the domain. In a similar fashion, SDMC structures the domain (orders the set of paths) using the values of the function $f(.;\theta_0)$. As we will see, this approach can lead to significant benefits.

4 SDMC & STRATIFICATION

In this section we discuss how the monotonicity or approximate monotonicity of the database can be used to design very efficient variance reduction algorithms. We limit ourselves to the stratification technique. It is not difficult to see how similar advantages can be gained when other variance reduction techniques are used.

We briefly review the stratification technique (for a thorough description of the technique see Glasserman 2004, Section 4.3).

Assume $\{A_1, \cdots A_k\}$ is a partition of Ω. Let $p_i = P(A_i)$, $\mu_i = E[Y_i] = E[Y|X \in A_i]$ and $\sigma_i^2 = Var[Y_i] = Var[Y|X \in A_i]$.

Direct or proportional stratified sampling selects $n_i = \lfloor n * p_i \rfloor$ samples randomly from A_i and uses the following estimator

$$\hat{Y}(n,k) = \sum_{i=1}^{k} p_i \cdot \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}.$$

It can be shown easily that this estimator is unbiased. However, this is not the best one can do. In other words, proportional sampling (i.e., $n_i = \lfloor n * p_i \rfloor$) is not necessarily the best allocation possible; note that this allocation completely disregards the structure of f, unless the partitioning of Ω into $A_1, \cdots, A_k$ has taken this structure into account. Given a fixed partition, it is well known that the optimal allocation of samples is according to quantities q_i (i.e., $n_i^{'} = \lfloor n * q_i \rfloor$) where

$$q_i = \frac{p_i \sigma_i}{\sum_{j=1}^{K} p_j \sigma_j}.$$

The estimator in this case needs to be adjusted to

$$\hat{Y'}(n,k) = \sum_{i=1}^{k} \frac{p_i}{n_i'} \sum_{j=1}^{n_i'} Y_{ij} = \frac{1}{n} \sum_{i=1}^{k} \frac{p_i}{q_i} \sum_{j=1}^{n_i'} Y_{ij},$$

and the minimum variance is given by

$$\sigma^{*2} = (\sum_{i=1}^{K} p_i \sigma_i)^2.$$

In general, strata definition, i.e., the appropriate partitioning of Ω, is problem dependent and is left to the creativity of the user. Once a partition is selected, optimal sampling within strata requires knowing σ_i's or estimating them. In almost all cases, these values are not known in advance and need to be estimated via pilot runs.

The key difficulty in both steps of (a) strata definition and (b) optimal allocation of samples is the fact that the structure of f is not known in advance. This difficulty is to a great extent removed in an appropriately structured database. In what follows we describe one possible strata definition approach in the context of SDMC. In the SDMC context:

1. The problem of partitioning the database Ω_N is transformed into that of partitioning a linearly ordered set (similar to a subinterval of R) over which the function $f(.;\theta)$ is monotone (precisely or "approximately"). In what follows, assume it is monotone.
2. Assuming monotonicity of $f(.;\theta)$ over the database, any function evaluation provides a great deal of relevant information in the following sense. Assume $f(\omega_0;\theta)$ is evaluated. Then for all $\omega < \omega_0$ we know $f(\omega;\theta) \leq f(\omega_0;\theta)$ and for all $x > x_0$

we know $f(x;\theta) \geq f(\omega_0;\theta)$. This information has significant implications for strata construction.

3. Consider two elements of the database a and b where $a < b$. If the points between a and b are sampled randomly from the database (think of $[a, b]$ as a stratum), then the standard deviation of the random function values are bounded by a multiple of $(f(b)-f(a))$. In this case, $(f(b)-f(a))$ can be used as substitute for σ_i. Therefore, again, a limited number of function evaluations from the database provides a significant amount of information that can be used for designing appropriate strata.

Given the form of the minimum variance when optimal sample allocation is used (i.e., $\sigma^{*2} = (\sum_{i=1}^{K} p_i\sigma_i)^2$), we use the following algorithm:

1. **Initialization**: Let $\omega_{[0]}$ and $\omega_{[N]}$ be the smallest and the largest values of the database (note that the database is linearly ordered). Select the point $\omega_{[N/2]}$ at the midpoint of the database. $[\omega_{[0]}, \omega_{[N/2]}]$ and $[\omega_{[N/2]}, \omega_{[N]}]$ form a partition of the database into two (equal size) strata. (If N is even choose $\omega_{[(N-1)/2]}$ as the "midpoint.")
2. **Iteration**: Assume the database is partitioned into n strata. For each stratum (say $[\underline{\omega}_i, \overline{\omega}_i]$ evaluate $\hat{p_i}\hat{\sigma_i} = [f(\overline{\omega}_i) - f(\underline{\omega}_i)](\overline{\omega}_i - \underline{\omega}_i)$. Select the stratum with the maximum index $\hat{p_i}\hat{\sigma_i}$ and divide that stratum into two equal size strata. (In case of ties select any of the strata to subdivide.)

This partitioning algorithm is not in general the optimal stratification algorithm but it has the useful property of requiring only one additional partition point in order to go from k to $k+1$ strata. In other words, the partitioning sequence goes through a refinement of already existing strata rather than defining a whole new set of strata. Similar choices are made in numerical integration and in Quasi-Monte Carlo.

Taking some liberty with precise definitions we offer the following connection between the above stratification approach and methods of adaptive subdivision for numerical univariate integration (see Krommer and Ueberhuber 1998, Chapter 8). Let $\Omega_N = \{\omega_1 < \cdots < \omega_N\}$. For large N, the mapping $\omega_i \to i/N$ defines a "one-to-one" map between the ordered database and the unit interval. Our estimation problem in this case is equivalent to evaluating

$$\int_0^1 f(x;\theta)dx.$$

The above stratification approach is the same as an adaptive subdivision approach for calculating the above integral under the assumption that $f(.;\theta)$ is monotone. In this case (and in our estimation problem) at each step of stratification a precise (deterministic) error bound to the estimation problem is available and repeated subdivision (further stratification) leads to more accurate estimates. In other words, if we know that $f(.;\theta)$ is monotone, no sampling is needed and one can have an efficient deterministic algorithm for estimating the desired expected value. In most cases however, $f(.;\theta_0)$ is approximately monotone and random sampling a la stratification is needed.

We next give a few examples and computational results to illustrate the effectiveness of the approach.

5 PRICING PATH-DEPENDENT OPTIONS

For illustrating examples we consider pricing of simple Asian, lookback, and hindsight options (See, e.g., Grant, Vora, and Weeks 1997, Vázquez-Abad and Dufresne 1998, Glasserman, Heidelberger and Shahabuddin 1999, Ross and Shanthikumar 2000, and Glasserman and Staum 2001 for a sample of efficient simulation methods applied to pricing path-dependent options). In all the cases we consider, the sample payoffs are continuous with respect to parameters of the model. Discontinuous cases such as digital or barrier options require modified algorithms and are not included here. See Zhao, Zhou, and Vakili 2006 for a discussion of these cases.

Let $S(t)$ be the price of an asset/security at time t ($t \in [0,T]$) and assume it follows a geometric Brownian motion (GBM) with constant drift and volatility (Black Scholes model), i.e.,

$$dS(t) = \mu dt + \sigma dW(t)$$

where $\{W(t); t \geq 0\}$ is a Brownian motion (BM). In a risk neutral setting the asset price follows the following stochastic differential equation (with an abuse of notation we use $S(t)$ for the asset price in this case as well).

$$dS(t) = rdt + \sigma dW(t)$$

where r is the risk free rate. (SDMC applies equally well to other more general asset price models.)

Let a discrete set of monitoring instances be given by $\{0 < t_1 < \cdots < t_k = T\}$. Pricing of an Asian option at a discrete set of points in time is one of the simplest option pricing problems that requires simulation even when the asset price follows a geometric Brownian motion. Let

$$\bar{S}_A = \frac{1}{n}\sum_{i=1}^{k} S(t_i).$$

Then the Asian option payoff is given by

$$L = [\bar{S}_A - K]^+$$

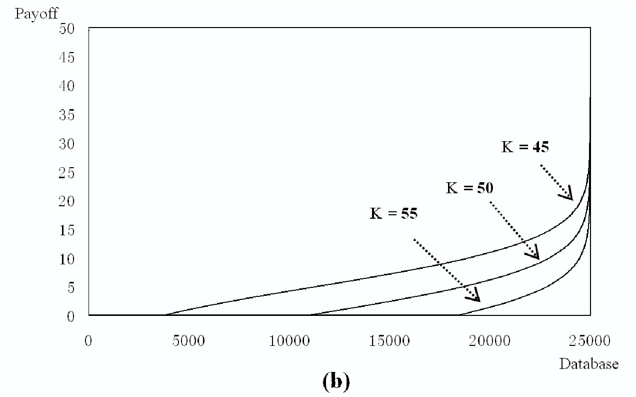

Figure 2: Database Ordered for $K = 50$, Payoff for $K = 45, 50, 55$

where x^+ $\max\{x, 0\}$. Let

$$S_M = \max\{S(t_1), \cdots, S(T)\}.$$

The payoff of a look-back option is

$$L = S_M - S_T,$$

and that of the hindsight option is

$$L = [S_M - K]^+.$$

The pricing problem is therefore equivalent to the estimation of the following expected value

$$J(r, K, \sigma) = E[e^{-rT} L].$$

The database we consider consists of 10^5 standard Brownian paths randomly generated. The database is ordered based on $\bar{S}_A$ for Asian option, based on $S_M - S_T$ for lookback option, and based on S_M for hindsight option. In each case with $r = 0.05$, $\sigma = 0.2$ were used. We perturb these parameters (as well as the strike price K) as follows (the nominal values are given in bold):

r	0.01	**0.05**	0.10
σ	0.1	**0.2**	0.3
K	45	**50**	55

Note that the perturbations in this context *are not small* perturbations. Figures 2, 4, and 5 show how the ordered payoff of the Asian option (ordered based on the nominal values) is perturbed as a result of parameter perturbations.

In the case of the strike price the monotonicity is retained strictly. In the case of the risk free rate (drift parameter) and volatility (σ) after perturbations, local monotonicity is lost; however, in some sense some global monotonicity is retained. This feature forms the basis of variance reduction that we achieve (see Tables 1 -3).

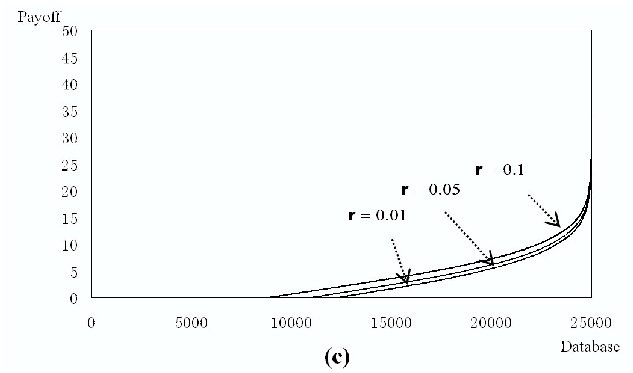

Figure 3: Database Ordered for $r = 0.05$, Payoff for $r = 0.01, 0.05, 0.1$

In the case of adaptive strata definition, the strata and the number of samples in each stratum were determined based on the payoff at the nominal parameter values and the strata and the number of samples were not changed when the parameters were perturbed. In all cases the database was partitioned into 20 stratum, estimators were based on 1000 samples. Variance estimates are based on estimates from 100 replications.

Table 1: Asian Option, $\sigma = 0.1$, r=0.05, K=55

Method	Mean & Variance
Standard Monte Carlo	$0.1816 \ldots 4.66e - 004$
SDMC & equal strata	$0.1821 \ldots 1.10e - 004$
SDMC & adaptive strata	$0.1822 \ldots 1.60e - 006$

Table 2: Lookback Option, $\sigma = 0.3$

Method	Mean & Variance
Standard Monte Carlo	$10.8119 \ldots 6.57e - 02$
SDMC & equal strata	$10.8081 \ldots 1.20e - 03$
SDMC & adaptive strata	$10.8049 \ldots 9.35e - 04$

Table 3: Hindsight Option, $\sigma = 0.2$, r=0.1, K=50

Method	Mean & Variance
Standard Monte Carlo	$11.0078 \ldots 7.00e - 02$
SDMC & equal strata	$11.0414 \ldots 2.76e - 03$
SDMC & adaptive strata	$11.0333 \ldots 9.82e - 04$

REFERENCES

Bratley, P., B. L. Fox, and L. Schrage. 1987. *A Guide to Simulation*. 2nd ed. New York: Springer-Verlag.

Fox, B. L. 1999. *Strategies for Quasi-Monte Carlo*. Boston, Mass: Kluwer Academic Publishers.

Glasserman, P. 1991. *Gradient Estimation via Perturbation Analysis*. Norwell, Mass: Kluwer Academic Publishers.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. New York: Springer Verlag.

Glasserman, P., P. Heidelberger, and P. Shahabuddin. 1999. Asymptotically Optimal Importance Sampling and Stratification for Path-dependent Options. *Mathematical Finance*. 9: 117–152.

Glasserman, P., and J. Staum. 2001. Conditioning on One-Step Survival for Barrier Option Simulations. *Operations Research*. 49 (6):923–937.

Grant, D., G. Vora, and D. Weeks. 1997. Path-dependent options: Extending the Monte Carlo simulation approach. *Management Science*. 43 (11): 1598.

Hammersley, J. M., and, D. C. Handscomb. 1964. *Monte Carlo Methods*. John Wiley.

Ho, Y. C. 1991. *Perturbation Analysis of Discrete Event Systems*. Kluwer Academic Publishers.

Krommer, A. R., and C. W. Ueberhuber. 1998. *Computational Integration*. The Society for Industrial and Applied Mathematics.

Niederreiter, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics.

Ross, S. M., and J. G. Shanthikumar. 2000. Pricing Exotic Options: Monotonicity in Volatility and Efficient Simulation. *Probability in Engineering and Information Sciences*. 14: 317–326.

Vázquez-Abad, F. J., and D. Dufresne. 1998. Accelerated Simulation for Pricing Asian Options. *Proceedings of the 1998 Winter Simulation Conference*. 1493–1500.

Zhao, G., Y. Zhou, and P. Vakili. 2006. Structured Database Monte Carlo: A New Strategy for Efficient Simulation. Working paper.

AUTHOR BIOGRAPHIES

GANG ZHAO is a Ph.D. student of Systems Engineering at Boston University. He received M.S. degrees in Electrical Engineering from The University of California at San Diego and Peking University, a B.S. in Electrical Engineering from Tsinghua University, China. His research interests include novel strategies for Monte-Carlo simulation, stochastic optimization, American style financial derivative pricing . His email address is <gzhao@bu.edu>.

YAKUN ZHOU is a Ph.D. student of Systems Engineering at Boston University. He received his M.S. degree in Mechanical Engineering, and a B.S. degree in Mechanical Engineering, both from Shanghai Jiao Tong University in Shanghai, China. His research interests include efficient Monte-Carlo simulation and risk management. His email address is <ykzhou@bu.edu>.

PIROOZ VAKILI is an Associate Professor in the Department of Manufacturing Engineering at Boston University. His research interests include Monte Carlo simulation, optimization, computational finance, and bioinformatics. His email address is <vakili@bu.edu>.