Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

EFFICIENT ESTIMATION OF OPTION PRICE AND PRICE SENSITIVITIES VIA STRUCTURED DATABASE MONTE CARLO (SDMC)

Gang Zhao

Manufacturing Engineering Department
15 Saint Mary's Street
Boston University
Brookline, MA 02446, U.S.A.

Tarik Borogovac

Electrical & Computer Engineering Department
8 Saint Mary's Street
Boston University
Boston, MA 02215, U.S.A.

Pirooz Vakili

Manufacturing Engineering Department
15 Saint Mary's Street
Boston University
Brookline, MA 02446, U.S.A.

ABSTRACT

We describe how to develop generic efficient simulation algorithms for estimating price and price sensitivities (the Greeks) of financial options using the Structured Database Monte Carlo (SDMC) approach. These algorithms are based on stratification, control variate and a combination of the two in an SDMC setting. Experimental results and some discussion of the effectiveness of the approach are provided. The algorithms also serve as illustrations of the basic approach of developing variance reduction algorithms in an SDMC setting that are not necessarily limited to stratification and control variate techniques.

1 INTRODUCTION

Most statistical estimation problems via Monte Carlo, including those of estimating price and price sensitivities of financial options, can be viewed as estimating the integral of a function, say f, on a unit hyper-cube, say $[0,1]^d$. Crude or basic Monte Carlo corresponds to sampling randomly in $[0,1]^d$ and averaging the function values at sampled points. Note that there is no link between the sampling strategy of crude Monte Carlo (choice of points at which f is sampled) and the function f. Most variance reduction techniques attempt to create such a link by bringing some auxiliary information to bear on the estimation problem; most often this information reflects some feature of the function f albeit not in a very direct fashion. Consider, for example, the control variate technique. A control variate carries some (linear) information about the output of the crude Monte Carlo ($f(U)$, $U \sim U[0,1]^d$) and hence f and it is most effective when this linear link is very strong. The source of the auxiliary information, i.e., the choice of the control, and the manner in which the information is used by variance reduction techniques are highly problem dependent: in each case a new source of information needs to be *discovered* and a method devised for its effective utilization. In other words, there are no generic recipes for designing effective variance reducing algorithms. The point of departure of the recently introduced approach of Structured Database Monte Carlo (SDMC) is an attempt to devise a generic method for designing such algorithms.

We begin by noting that quite often, as when pricing options or estimating price sensitivities, we use Monte Carlo to "solve" many instances of a single problem that differ only in parameter values. In such a context SDMC utilizes information obtained at a nominal parameter value, say θ_0 that may be a scalar, a vector, or a more general index, to design effective variance reduction algorithms at neighboring parameters (see Zhao et al. 2006 for an introduction and Zhao et al. 2007 for a more complete discussion). Therefore, in the SDMC approach $f(.,\theta_0)$ is the generic source of information. Our first use of this information is to induce a linear order (structure) on the underlying probability space (database) using values at the nominal parameter, i.e., $f(.,\theta_0)$. In many problems this structure is approximately maintained when the parameter is perturbed. This feature enables the design of generic variance reducing algorithms using the known techniques of variance reduction such as stratification, control variate, and importance sampling for estimation at the perturbed parameter value. Therefore, in contrast to the common practice of *discovering* a source of information about $f(\theta)$, when using SDMC, the information is *extracted* from $f(\theta_0)$ where θ_0 is a neighboring parameter of θ.

In this paper we focus on designing variance reducing algorithms based on two variance reduction techniques of stratification, control variate, and a combination of the two. In the case of the stratification technique, the SDMC setting allows for a generic and effective stratification approach. Once the strata are defined, the choice of sample allocation to strata follows the standard techniques without much modification. As for the control variate technique, the SDMC approach allows for the selection of effective and generic control variates. Combining both approaches can add an additional degree of variance reduction that goes beyond the contribution of each approach in isolation. As we point out in the paper, stratification can deal effectively with nonlinearity in f (as a function of elements of the database) and control variate can effectively remove the variance due to the linear "component" of f (see, Glasserman 2004, Section 4.3 for similar observations).

It is worth noting that a (possibly significant) setup cost is associated with the construction and structuring of the database in the SDMC setting. This cost can be viewed as an investment whose dividends are recuperated through efficiency gains in each future estimation exercise that uses the database. In the context of option pricing we show that the efficiency gains are not limited to price estimations at neighboring parameters but also extend to estimating price sensitivities at these parameters. As is well known, estimating price sensitivities is an indispensable part of risk management and hedging in trading options. Hence the efficiency gains in this domain are of particular practical interest.

In addition to the well-known references to variance reduction techniques, of which Glasserman (2004) is an excellent representative providing a comprehensive coverage of Monte Carlo methods applied to computational finance, the literature on *information based complexity* and *optimal recovery* is also quite relevant to the work in this paper (see, e.g., Traub et al. 1988, and Novak 1988). The work on information based complexity and optimal recovery, starting with formulations and results as in Kiefer (1957), seeks optimal algorithms (under different notions of optimality) for, among others, estimating the integral of an unknown function f. The estimation relies on two sources of information: (a) a priori information represented by $f \in \mathscr{F}$ where $\mathscr{F}$ is a specific class of functions (for example monotone functions), and (b) sampled values of f. A critical question then becomes optimal sampling of f, given $f \in \mathscr{F}$. In the SDMC context the database can be conveniently represented/approximated by $[0,1]$ and the function f is monotone/approximately monotone on the database; hence, our estimation problem can be framed as one considered in the context of information based complexity and the question of optimal sampling in that context becomes intimately related to stratifying the database in the SDMC setting. More detailed discussion along these lines are provided in the paper.

The rest of the paper is organized as follows. We begin with a preliminaries section in which we define the estimation problems that will be addressed and briefly review SDMC, stratification, and control variate techniques. In Section 3 we consider an idealized setting that in spite of its apparent simplicity provides significant insight into the design and performance of the algorithms we propose. Section 4 considers price and price sensitivity estimation of a path-dependent option and includes experimental results. Brief conclusions and directions for future research are provided in Section 5.

2 PRELIMINARIES

In this section we describe the estimation problems in a general context, we give a short introduction to the SDMC approach, and briefly review variance reduction techniques of stratification and control variate. This section also serves to establish some notation for the rest of the paper.

2.1 The Estimation Problem

Consider the problem of estimating

$$J(\theta) = E[f(\omega;\theta))] = \int_{\Omega} f(\omega;\theta)P(d\omega) \tag{1}$$

where ω is a random object (number, vector, graph, path, etc.) belonging to a probability space (Ω, F, P) and $f(\ .\ ;\theta)$ is a real-valued function on Ω for all $\theta \in \Theta$. As we pointed out earlier, the dependence on a parameter θ is critical for the SDMC approach in the following sense: we assume that the estimation problem is to be solved for different and possibly a large number of $\theta \in \Theta$.

In some problems it is the probability measure that depends on the parameter of interest θ. In such cases one can find an equivalent estimation problem where the parameter in the measure is "pushed" to a transformed f function. Think, for example, of a stochastic model where the parameter of interest is a distributional parameter of its random "inputs." Since samples of random inputs may be generated from standard uniforms $U(0,1)$, the underlying probability space can be defined as the product space of a number of uniforms that are independent of the distributional parameters. Therefore, no generality is lost in the above formulation.

2.2 Sensitivity Estimation

Dynamic hedging in finance requires evaluating price sensitives. These are quantities that are not market observable and Monte Carlo simulation is a key tool for their estimation. In many instances, the estimation of such sensitivities can be formulated as the estimation of a functional on the paths of the underlying stochastic process, namely as an estimation problem of the kind we specified above.

Specifically, let $g(X(\omega);\theta)$ denote a sample performance defined on a path of X. Fixing ω makes g a deterministic function of θ. Therefore, formally we can take its derivative with respect to θ and obtain $\nabla_\theta g(X(\omega);\theta)$. In those cases where

$$\nabla_\theta E[g(X(\omega);\theta)] = E[\nabla_\theta g(X(\omega);\theta)]$$

we can set $f(\omega;\theta) = \nabla_\theta g(X(\omega);\theta)$ and the sensitivity estimation problem becomes an estimation problem of type (1).

We now turn to a short introduction to the SDMC method.

2.3 SDMC

The following are the basic steps of the SDMC algorithm. First, the primitives of the simulation (members of Ω) need to be selected. In the type of problems discussed in this paper, a convenient set of primitives are paths of the standard Brownian motion or vectors of 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. However, sampling according to a more general user defined measure is possible, and sometimes desirable.

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

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

where X is a random element of Ω_N selected uniformly. For large N, $J_1(\theta)$ approximates $J(\theta)$ closely. In this paper, we completely bypass the question of what an acceptable Ω_N is and how best to generate it. This question will be a separate research question to be addressed in the future. We take Ω_N and the above construction as a given.

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). As we will see, if the sample performance is continuous with respect to θ, the homogeneity induced survives when θ is perturbed.

Figure 1 describes the basic steps of the SDMC approach.

---

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,

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

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.)

Figure 1: Structured Database Monte Carlo simulation.

We now briefly review the two variance reduction techniques of stratification and control variate that will be used in this paper. For a more complete discussion of the techniques see Glasserman (2004), Chapter 4.

2.4 Stratification & Control Variate Methods

As noted earlier, to gain efficiency, most variance reduction techniques rely on utilizing *additional* information about the random variable whose expectation is to be estimated. Their effectiveness depends on the relevance of the information and on how it is used.

2.4.1 Stratification

The stratification method involves partitioning the probability space into a finite number, say k, of strata. Then, the original estimation problem turns into that of k estimation subproblems. The relevant and additional information is the "size" of the strata (their probabilities). This information allows one to assemble the subproblem estimators to construct an estimator for the original problem without introducing additional variance.

More precisely, assume we aim to estimate $\mu = E[Y]$ where $Y = f(X)$, X is a random element of Ω, and f is a real-valued function. 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 $\sigma_i^2 = \mathsf{Var}[Y_i] = Var[Y|X \in A_i]$. Given p_i and $\hat{\mu}_i$, an estimator of μ_i for $i = 1, \cdots k$, the stratified estimator of μ is

$$\hat{\mu}_{st} = p_1\hat{\mu}_1 + \cdots + p_k\hat{\mu}_k.$$

It is easy to see that the variance of this estimator is

$$\mathsf{Var}(\hat{\mu}_{st}) = \sum_{i=1}^{k} p_i \mathsf{Var}(\hat{\mu}_i) = E[\mathsf{Var}(Y|X \in A_i)] \leq \mathsf{Var}(Y).$$

In other words, stratification is always beneficial. The magnitude of the benefit depends on the choice of stratification. However, strata definition, i.e., the appropriate partitioning of Ω, is problem dependent and is left to the creativity of the user. No generic prescription for "optimal" partitioning of Ω is provided in the literature. The user has some flexibility in the allocation of samples to strata.

Given a fixed partition, it is well known that the optimal allocation of samples is according to quantities q_i

$$q_i = \frac{p_i\sigma_i}{\sum_{j=1}^{k} p_j\sigma_j},$$

i.e., the number of samples out of n allocated to stratum A_i, denoted by n_i is given by $n_i = \lfloor n * q_i \rfloor$. The minimum variance is given by

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

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.

2.4.2 Control Variates

In the method of control variates the additional information is furnished via one or more random variables, called control variates. The controls are *correlated* with Y and the method of control variate uses the standard techniques of linear regression to extract the information and construct "correction" term to be added to the "raw" estimator of $E[Y]$.

More specifically, let $V_1, \ldots, V_k$ be a set of controls with *known* means $E[V_1], \cdots, E[V_k]$. Let

$$Z = Y - \sum_{i=1}^{k} \beta_i \cdot (V_i - E[V_i])$$

Then, for all vectors $(\beta_1, \cdots, \beta_k)$, Z is an unbiased estimator of $\mu = E[Y]$.

The problem of finding the vector $\beta^* = (\beta_1^*, \ldots, \beta_k^*)$ that minimizes the variance of Z can be formulated as the problem of projecting Y onto the linear subspace spanned by the $V_i - E[V_i]$'s. The optimal solution is given by

$$\beta^* = \Sigma_V^{-1}\Sigma_{VY}$$

where V is a vector of the V_i's and Σ with subscript denotes the appropriate covariance vector or matrix. The variance of the optimal estimator is:

$$\sigma_Z^2 = \sigma_Y^2 - \Sigma_{VY}'\Sigma_V^{-1}\Sigma_{VY} = (1 - R^2)\sigma_Y^2.$$

Note that the optimal choice of β's requires knowledge of the covariance matrix of V and Y, which is rarely available and most often needs to be estimated.

Again there are no generic prescriptions for selecting controls and this selection has been highly problem specific.

3 DESIGNING VARIANCE REDUCTION ALGORITHMS

In this section we consider an idealized estimation setting, namely we assume that the problem is to estimate μ defined as

$$\mu = E[f(U)] = \int_0^1 f(u)du.$$

where f is an increasing function on $[0,1]$ and U is uniformly distributed on $[0,1]$ (in what follows we use increasing to mean more generally non-decreasing). Note that if $i = 1, \cdots, N$ denote the indices of the elements of a large and monotone database, then the correspondence $i \to i/N$ maps the database into $[0,1]$ and the monotonicity is preserved. The above integral then is an idealized form of the summation of the elements of the database. This idealized setting provides significant insight for design and performance evaluation of estimation algorithms in the SDMC setting.

Assume that the only known information about f is that it is increasing and no other regularity properties are assumed about f. The key question we would like to address is the following: how can we use the information that f is increasing to design effective stratification and control variate algorithms or algorithms that combine the two techniques? In what follows we use, with some modifications, the formulation and terminology of information based complexity. Due to lack of space we give a general review of key concepts and results and present the algorithms that will be used in the following section.

Let $\mathscr{F} = \{f : [0,1] \to R; f \text{ increasing}\}$ be the set of increasing functions on $[0,1]$. For each element f of $\mathscr{F}$ the fact that f is increasing is an apriori information about it. Additional information can be obtained by sampling f, i.e., by evaluating f at points in $[0,1]$. We assume that $f(x)$ can be evaluated precisely (without estimation noise). Let $x_1, \cdots, x_k$ be k distinct points in $[0,1]$. Let $I(f; x_1, \cdots, x_k) = ((x_1, f(x_1)), \cdots, (x_k, f(x_k)))$ represent the new information about f based on sampling. To simplify notation, we use the above notation for non-adaptive or adaptive and deterministic or stochastic sampling. In other words, x_i may be random variables and the choice of x_i may depend on previous samples. Moreover, to further simplify the notation, we often write $I(f;k)$ or simply I to denote this information.

To simplify the discussion, consider the deterministic sampling case. Let

$$N(I) = N(I;k) = \{f' \in \mathscr{F}; I(f';k) = I(f;k)\}.$$

$N(I)$ represents the uncertainty associated with the information I and it is the set of functions that are indistinguishable from f given the information I. Let $S : \mathscr{F} \to R$ be the integration operator, i.e., $S(f) = \int_0^1 f(u)du$. Let $c \in R$ denote an estimate of μ based on I. Then for any $f' \in N(I)$, $e(f';c) = |s(f') - c|$ is the absolute estimation error. We seek an *optimal* estimate of μ, denoted by $\phi(I)$, in the following *worst case* sense

$$\phi(I) = \text{argmin}_c\{\sup\{e(f';c); f' \in N(I)\}\}.$$

$e(I) = \sup\{e(f'; \phi(I)); f' \in N(I)\}$ is the worst case estimation error and can be used as estimation error bound.

The following are known from the literature on information based complexity: (i) in the deterministic case, assuming k samples are to be selected it is optimal to set $x_i = (i-1)/k-1$, for $i = 1, \cdots, k$ $(k \geq 2)$ (see Kiefer 1957, Section 5), (ii) assume, without loss of generality, that $x_1 < x_2 < \cdots x_k$, $\delta_i = x_{i+1} - x_i$ and $\delta f_i = f(x_{i+1}) - f(x_i)$ $(i = 1, \cdots, k-1)$. Assume k points are already selected and we would like to sequentially select the $k+1$th sample. Then it is optimal (in a sense specified in Sukharev 1987) to select the $k+1$th point as the midpoint of the subinterval for which $\delta_i \cdot \delta f_i$ is maximum; and finally, (iii) in the stochastic setting (error is defined as $E[|s(f) - c|]$ where expectation is with respect to probability measure induced by stochastic sampling) adaptive stochastic sampling is superior to non-adaptive stochastic sampling and adaptive and non-adaptive deterministic sampling (in the sense of superior asymptotic rate of convergence) (see Novak 1992). Moreover, in this case the sampling scheme is very similar to the stratification scheme we suggest below.

3.1 Stratification

We consider the following stratification scheme. Assume $x_1 = 0 < x_1 < \cdots < x_k = 1$ are given and δ_i and δf_i are given as above. Then

- Find the subinterval (stratum) such that $\delta_i \cdot \delta f_i$ is maximum. (In case of ties select any subinterval that maximizes $\delta_i \cdot \delta f_i$.)
- Divide the subinterval into two equal subintervals (strata).

A few comments are in order. In the SDMC context one possibility (and one that we utilize in the experiments reported in the next section) is to use $f(., \theta_0)$, i.e., the function values at the nominal parameter, as the monotone function to define the strata using the above scheme. In the above scheme (applied to a monotone function) the worst rate of convergence (to zero) of the error bound corresponds to $f(x) = x$ (or any other linear function). In other words, the linear functions are in some sense the worst functions for the above scheme.

3.2 Control Variate

Let U be a random (uniform) sample on $[0,1]$ and $Y = f(U)$ in the above setting where f is assumed to be increasing. An immediate and natural control is $Z = h(U) = U$. Given that f and h are both monotone it is easy to verify that $Cov(f(U), h(U)) \geq 0$, i.e., Y and Z are positively correlated. The maximum efficiency of using U as a control is gained when f is a linear function of U in which case the control removes the estimation variance completely. Therefore, we propose the following control variate.

- Use $Z = U$ as a control to estimate $Y = f(U)$.

In the SDMC setting the indices of the ordered database play the role of U and can be used as the control. Moreover, in the SDMC setting another natural control is $Z = f(U, \theta_0)$ for estimating $Y = f(U, \theta)$. We will point out in the next section that this latter control is quite effective when θ is sufficiently close to θ_0.

3.3 Stratification + Control Variate

The fact that the control $Z = U$ is a very effective control when f is linear while a linear f is most challenging for stratification suggests that combining the two algorithms can be effective. As we achieve finer and finer stratifications

Table 1: Variance reduction ratios for $f(x) = x^a$

Method	$a = 0.1$	$a = 1$	$a = 2$	$a = 5$
Control Variate	5	∞	160	3
Stratification	33	640	1176	300
Combined	247	∞	101950	1558

we expect that f on each stratum can be more closely approximated by a linear function. Hence we consider the following algorithm.

- Use the above stratification algorithm.
- Let $[x_i, x_{i+1})$ be the ith stratum. Use $Z_i = U_i = U([x_i, x_{i+1})$ as a control for $Y_i = f(U_i)$.

Table 1 provides some variance reduction results (relative to random sampling) for $f(x) = x^a$ for different values of a and gives an idea about the contributions of each algorithm and their combination.

4 ESTIMATING OPTION PRICE AND PRICE SENSITIVITIES

We illustrate how the algorithms can be used to estimate price and price sensitivities of financial options by describing their application to the estimation of price and price sensitivities of a lookback option where the interest rate is a mean-reverting CIR process (see, e.g., Glasserman 2004, Chapter 3). There is no closed-form solution for the price of this option and simulation is a solution method of choice.

Specifically, we consider the following stochastic differential equations,

$$\begin{aligned} \frac{dS_t}{S_t} &= r_t dt + \sigma dW_t \\ dr_t &= \alpha(\bar{r} - r_t)dt + \sigma_r \sqrt{r_t} dW_t' \end{aligned}$$

where W_t and W_t' are two independent Brownian motions, α is the speed of mean reversion, and $\bar{r}$ is the long run mean drift rate. Given a discretized path, $\{S_{t_1}, S_{t_2}, \cdots, S_{t_m} = S_T\}$, the present value of the payoff of the lookback call option is given by

$$C_T = e^{r_0 T} \max\{0, S_T - \min_{\tau \in \{t_0, \cdots, t_m = T\}} S_\tau\}$$

The option price (of the discretized process) is given by $E[C_T]$, its sensitivity to the initial asset price, i.e., its *Delta*, is given by $dE[C_T]/dS(0)$ and its sensitivity to asset volatility, i.e., its *Vega*, by $dE[C_T]/d\sigma$. Let τ^* be the time step at which the asset price S_t attains its (discretized) minimum value. Then, the path derivatives of Delta and Vega are given by

$$\frac{dC_T}{dS_0} = e^{-r_0 T} I\{S_T > S_{\tau^*}\} * \left(\frac{S_T - S_{\tau^*}}{S_0}\right) = \frac{C_T}{S(0)},$$

and

$$\frac{dC_T}{d\sigma} = e^{-r_0 T} I\{S_T > S_{\tau^*}\} * \left[\frac{dS_T}{d\sigma} - \frac{dS_{\tau^*}}{d\sigma}\right]$$

where, for all t:

$$\frac{dS_t}{d\sigma} = \left(\log\left(\frac{S_t}{S_0}\right) - \int_0^t r_u + \frac{\sigma^2}{2} du\right).$$

Estimating the option price and option price sensitivities to $S(0)$ and σ then corresponds to estimating $E[C_T]$, $E[\frac{dC_T}{dS_0}]$ and $E[\frac{dC_T}{d\sigma}]$.

The following parameters were considered: $S(0) = 100$, $r(0) = 0.1$, $\sigma_S = 0.3$, $\bar{r} = 0.1$, $\sigma_r = 0.1$, $\alpha = 0.5$, $T = .25$, and $m = 60$. A database of $100,000$ elements were generated where each element of the database corresponds to a two-dimensional vector of discretized standard Brownian process simulated for $m = 60$ time steps. The database was then linearly ordered based on sample option prices at the above nominal parameter values. In the figures to follow the ordered database is represented by the horizontal axis scaled to $[0, 1]$.

We begin by providing graphical representations of the impact of perturbing some parameter values. Figure 2 illustrates the effect of perturbing σ, from its nominal value 0.3 to the perturbed value 0.4, on the order of sample option values. The dotted line traces the sampled values C_T at the nominal parameters while the solid line traces C_T at the perturbed parameter values. The conclusion is that the order of the database is approximately maintained at the perturbed parameters and this order carries useful information for estimation at perturbed parameter values.

We next look at the relationship between the order on the database based on sample option price C_T and the values of path derivatives, i.e., sample values of Delta and Vega. In other words, the question is whether the order imposed on the database carried any useful information for estimating Delta and Vega. Given the path estimate of Delta, it is clear that the path derivatives at the nominal parameter value will be perfectly ordered and at the perturbed parameter values will have a behavior similar to sample option prices. Therefore, in this case, the order of the database carries valuable information for estimating Delta. Figure 3 illustrates the values of the path derivatives (samples of Vega) at the nominal parameter values as a function of the ordered database. As can be seen, a behavior similar to a perturbation of a parameter value is exhibited, namely the order of the database is approximately maintained and

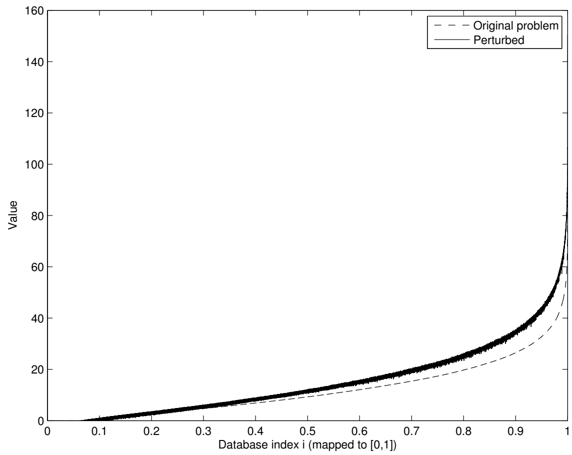

Figure 2: The order is preserved for the lookback option when the stock volatility σ is perturbed from 0.3 to 0.4.

therefore it can be used to gain efficiency in estimating Vega.

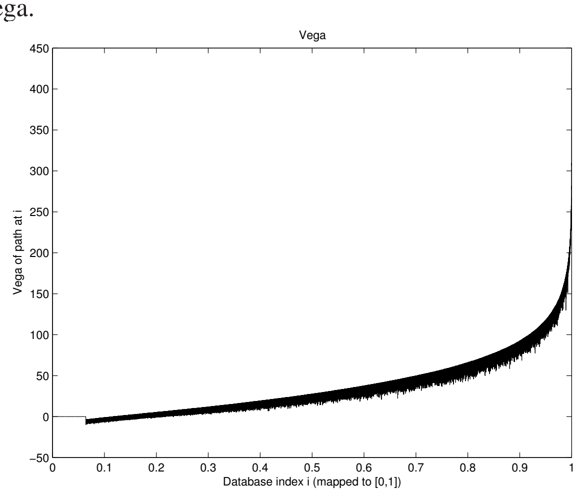

Figure 3: Sample Vega (path derivative) of the lookback option.

We are now prepared to present some quantification of the variance reduction based on stratification, control variate, and a combination of the two. The results are for illustration purposes; they give a general idea of the order of magnitudes of variance reductions that can be expected and provide some insight about the comparative efficacy of different algorithms. No attempt was made to "optimize" the algorithms and there are no claims that they are the best that can be devised in this context.

The number of strata were arbitrarily preselected to be 10 and the total number of samples were also arbitrarily preselected to be 100. Variance and variance reduction ratio estimates reported are based on outer replications of the price or price sensitivity estimators. The following estimators are

Table 2: Estimating option value at original parameter: $\sigma = 0.3$.

Estimator	Estimate	Variance	VRR
Crude	11.32	1.10	1
UCV	11.50	$1.58E-1$	6.98
DBCV	11.91	0	∞
STRAT	12.06	$1.99E-2$	55.33
STRAT CV	11.87	$3.20E-4$	$3.45E3$

Table 3: Estimating option value using perturbed parameter $\sigma = 0.4$.

Estimator	Estimate	Variance	VRR
Crude	15.66	2.06	1
UCV	15.36	$3.71E-1$	5.55
DB CV	15.34	$3.67E-3$	560.46
STRAT	15.09	$3.55E-2$	57.95
STRAT CV	15.23	$1.74E-3$	$1.18E3$

Table 4: Estimating option Vega using path derivatives.

Estimator	Estimate	Variance	VRR
Crude	32.91	15.28	1
UCV	33.14	3.25	4.70
DB CV	34.82	$2.04E-1$	74.98
STRAT	34.70	$3.29E-1$	46.48
STRAT CV	35.11	$1.00E-1$	152.24

used: (i) crude Monte Carlo (Crude), (ii) control variate with the index of the database as the control (UCV), (iii) control variate with sample prices at the nominal value as control (DBCV), (iv) stratified estimator (STRAT), (v) stratified + control variate estimator (STRAT CV). Variance reduction ratios (VRR) are relative to crude Monte Carlo.

To begin, and as a baseline for comparison, we report the performance of the algorithms at nominal parameters (in particular $\sigma = 0.3$) in Table 2. Table 3 gives the performance of the algorithms for estimating the option price when σ is perturbed to 0.4 and Table 4 gives the performance of the algorithms when estimating Vega of the option.

Not surprisingly, the most efficient algorithms are DBCV and STRAT CV. One can also speculate that for smaller perturbations of the parameter DBCV may perform better than STRAT CV but this needs to be verified by further experimentation and may in fact be highly problem dependent.

5 CONCLUSIONS

In this paper we presented algorithms based on stratification, control variate and a combination of the two in a Structured Database Monte Carlo (SDMC) setting and showed how they can be used for efficient estimation of option price and option price sensitivities. We argued that one of the features of these algorithms is that they are generic and can be applied in a generic manner in all parametric settings. The effectiveness of the algorithms (their efficiency gains), however, depends critically on whether the order imposed on the database is maintained when parameters are perturbed. We expect this to be the case for many problems of practical interest, while there are also important cases when the order is severely disrupted with the perturbations of the parameter values. In such cases specific and non-generic remedies need to be devised.

SDMC is a fairly new direction of investigation for gaining computational efficiency in Monte Carlo and there are many unanswered questions that need to be investigated. Some of these, in particular questions related to “optimal” algorithms as formulated in Section 3 of the paper, constitute one of the directions of our future research.

ACKNOWLEDGMENTS

Research supported in part by the National Science Foundation grants CMMI-0620965 and DGE-0221680.

REFERENCES

Glasserman, P. 2004. *Monte carlo methods in financial engineering*. Springer Verlag.

Kiefer, J. 1957. Optimal sequential search and approximation methods under minimum regularity assumptions. *J. Soc. Indust. Appl. Math.* 5 (3): 105–136.

Novak, E. 1988. *Deterministic and stochastic error bounds in numerical analysis*. Lecture Notes in Mathematics. Springer-Verlag.

Novak, E. 1992, May. Quadrature formulas for monotone functions. *Proceedings of the American Mathematical Society* 115 (1): 59–68.

Sukharev, A. 1987. The conept of sequential optimality for problems in numerical analysis. *Journal of Complexity* 3:347–357.

Traub, J. F., H. Wozniakowski, and G. W. Wasilkowski. 1988. *Information-based complexity*. Academic Press.

Zhao, G., T. Borogovac, and P. Vakili. 2007. Structured database monte carlo: a new strategy for efficient simulation. Technical report, Department of Manufacturing Engineering, Boston University, Boston, Massachusetts.

Zhao, G., Y. Zhou, and P. Vakili. 2006. A new efficient simulation strategy for pricing path-dependent options. In *Proceedings of the 2006 Winter Simulation Conference*, ed. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 703–710.

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⟩.

TARIK BOROGOVAC is a Ph.D. student of Systems Engineering at Boston University. He holds an M.S. degree in Communications and Computer Networks and a B.S. degree in Computer Engineering, both from Boston University. His current research interests include efficient Monte Carlo simulation and approximate dynamic programming. His e-mail address is ⟨tarikb@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⟩