MONTE CARLO SIMULATIONS FOR STOCHASTIC OPTIMIZATION

1. Introduction

Many important real-world problems contain stochastic elements and require optimization. Stochastic programming and simulation-based optimization are two approaches used to address this issue. We do not explicitly discuss other related areas including stochastic control, stochastic dynamic programming, and Markov decision processes. We consider a stochastic optimization problem of the form

\[ z^* = \min_{x \in X} Ef(x, \xi), \quad (SP) \]

where \( x \) is a vector of decision variables with deterministic feasible region \( X \subset \Re^d \), \( \xi \) is a random vector, and \( f \) is a real-valued function with finite expectation, \( Ef(x, \xi) \), for all \( x \in X \). We use \( x^* \) to denote an optimal solution to \((SP)\). Note that the decision \( x \) must be made prior to observing the realization of \( \xi \).

A wide variety of types of problems can be expressed as \((SP)\) depending on the definitions of \( f \) and \( X \). Two of the most commonly-used approaches are rooted in mathematical programming and in discrete-event simulation modeling.

In a two-stage stochastic linear program with recourse [6, 14], \( X \) is a polyhedral set and \( f \) is defined as the optimal value of a linear program, given \( x \) and \( \xi \), i.e.,

\[ f(x, \xi) = cx + \min_{y \geq 0} qy \]

\[ \text{s.t.} \quad Wy = Tx + h. \quad (1) \]

Here, \( \xi \) is the vector of random elements from \( h, q, T, \) and \( W \). A prototypical problem of this nature is a capacity allocation model under uncertain demand and/or capacity availabilities. \( x \) is a strategic decision allocating resources while \( y \) represents an operational recourse decision that is made after observing the demand and availabilities. Example applications of this type include capacity expansion planning in an electric power system [16] and in a telecommunications network [60]. The two-stage model generalizes to a more dynamic, multi-stage model (see, e.g., [10]) in which decisions are made, and random events unfold, over time. For multi-stage applications in asset-liability management see [13] and in hydro-electric scheduling see [38].

In the context of a simulation model, \( f(x, \xi) \) could represent a performance measure under a design specified by \( x \). For example, \( f(x, \xi) \) might represent the number of hours in a workday that a critical machine is blocked in a queueing network model of a manufacturing system in which buffer sizes are determined by \( x \). In another application, Plambeck et al. [52] allocate constrained processing rates to unreliable machines with buffers in a fluid serial queueing network in order to maximize steady-state throughput. In nonterminating simulations, the expectation in \( Ef(x, \xi) \) is typically with respect to a steady-state distribution.

Note that \( Ef(x, \xi) \) can capture objectives not usually thought of as a “mean.” For example, if \( c \) represents random rates of return and \( x \) investment amounts, we might want to maximize the probability of exceeding a return threshold, \( T \). We can write \( P(cx \geq T) = EI(cx \geq T) \) where \( I(\cdot) \) is the indicator function that takes value one if its argument is true and zero otherwise. For more on probability maximization models (and generalizations of \((SP)\) in which \( X \) contains probabilistic constraints) see Prékopa [53]. See King et al. [44] for a discussion of risk modeling in stochastic optimization.

A more general model than \((SP)\) allows the distribution of \( \xi \) to depend on \( x \). Some simple types of dependencies can effectively be captured in \((SP)\) via modeling tricks, such as the \( x \) scaling random elements of \( T \) in \((1)\). General dependencies, however, are difficult to handle. For work on decision-dependent distributions when there are a finite number of possibilities see [25, 39].

Regardless of whether it is defined as the expected value of a mathematical program or as a long-run average performance measure of a discrete-event simulation model, it is usually impossible to calculate \( Ef(x, \xi) \) exactly—even for a fixed value of \( x \). When the dimension of
the random vector $\xi$ is relatively low, one approach is to obtain deterministic approximations of $Ef(x, \xi)$ using numerical quadrature or related ideas. In stochastic programming, this corresponds to generating and refining bounds on $Ef(x, \xi)$ within a sequential approximation algorithm [19, 23, 42]. For problems in which $\xi$ is of moderate-to-high dimension and is continuous or has a large number of realizations, Monte Carlo simulation is widely regarded as the method of choice for estimating $Ef(x, \xi)$. As a result, it is not surprising that Monte Carlo techniques play a fundamental role in solving (SP).

In recent years, considerable progress has been made in solving realistically-sized problems with a significant number of stochastic parameters and decision variables. The telecommunications model considered by Sen et al. [60] has 86 random point-to-point demand pairs and 89 links on which capacity may be installed. Plambeck et al. [52] study queueing networks with up to 50 nodes. Each node represents a machine with random failures and has a decision variable denoting its assigned cycle time. Plambeck et al. also solve a stochastic PERT (program evaluation and review technique) problem with 70 nodes and 110 stochastic arcs. The arcs model the times required to complete activities and a decision variable associated with each arc influences (parameterizes) the distribution of the random activity duration. These problems contain objectives with high-dimensional expectations and all were solved using Monte Carlo methods.

In this article we discuss: (i) several types of Monte Carlo-based solution procedures that can be used for solving (SP); (ii) methods for testing the quality of a candidate solution $\hat{x} \in X$; (iii) variance reduction techniques used in stochastic optimization; and (iv) theoretical justification for using sampling.

2. Solution Procedures

Monte Carlo methods for approximately solving stochastic optimization problems can typically be classified on the basis of whether the sampling is external to, or internal to, the optimization algorithm. Solution procedures of both types are driven by estimates of objective function values and/or gradients. Before turning to solution procedures we briefly discuss gradient estimation.

In stochastic programming, gradient (or subgradient) estimates of $Ef(x, \xi)$ are typically available via duality. In simulation-based optimization, the primary methods for obtaining gradient estimates are finite differences, the likelihood ratio (LR) method (also called the score function method) [28, 56], and infinitesimal perturbation analysis (IPA) [26, 34]. Finite-difference approximations require minimal structure, needing only estimates of $Ef(x, \xi)$; however, they result in solution procedures that can converge slowly. The LR method is more widely applicable than IPA, but when both apply the IPA approach tends to produce estimators with lower variance. See, for example, Glynn [27] for a discussion of these issues.

In the simplest form of “external sampling” (also called “sample-path optimization” [54] and the “stochastic counterpart” method [56]) we generate independent and identically distributed (i.i.d.) replicates $\xi^1, \ldots, \xi^n$ from the distribution of $\xi$ and form the approximating problem

$$
\hat{z}_n = \min_{x \in X} \frac{1}{n} \sum_{i=1}^{n} f(x, \xi^i).
$$

Even when it is possible to construct $(SP_n)$ using i.i.d. variates, it may be preferable to use another sampling scheme in order to reduce the
variance of the resulting estimators. Moreover, in nonterminating simulation models, generating i.i.d. replicates from a stationary distribution is often impossible (for exceptions see recent work on exact sampling, e.g., [3, 20]), but under appropriate conditions we may run the simulation for a length $n$ and replace the objective function in $(SP_n)$ with a consistent estimate of the desired long-run average performance measure.

After constructing an instance of $(SP_n)$ we employ a (deterministic) optimization algorithm to obtain a solution $x^*_n$. In the case of stochastic linear programming, $(SP_n)$ is a large-scale linear program. The cutting-plane algorithm of Van Slyke and Wets [64], its variant with a quadratic proximal term [57], and its multi-stage version [7, 9] are powerful tools for solving such problems. A cutting-plane algorithm with a proximal term and IPA-based gradients is used in an external sampling method for solving the queueing network problem in [52]. See Birge [8] for a recent survey of computational methods for stochastic programming instances of $(SP_n)$.

Intuitively, we might expect solutions of $(SP_n)$ to more accurately approximate solutions of $(SP)$ as $n$ increases. We discuss results supporting this in Section 5. In addition, after having solved $(SP_n)$ to obtain $x^*_n$ it would be desirable to know whether $n$ was “large enough.” More generally, we would like to be able to test the quality of a candidate solution (such as $x^*_n$). This is discussed in Section 3.

We now turn to solution procedures based on internal sampling. These algorithms adapt deterministic optimization algorithms by replacing exact function and gradient evaluations with Monte Carlo estimates. The sampling is internal because new observations of $\xi$ are generated on an as-needed basis at each iteration of the algorithm. We briefly discuss stochastic adaptations of steepest-descent and cutting-plane methods.

A deterministic steepest-descent algorithm for $(SP)$ forms iterates $\{x^\ell\}$ using the recursion

$$x^{\ell+1} \leftarrow \Pi_X \left[ x^\ell - \rho^\ell \nabla Ef(x^\ell, \xi) \right].$$

$\Pi_X$ performs a projection onto $X$ and $\{\rho^\ell\}$ are step lengths. It is usually impossible to calculate $\nabla Ef(x, \xi)$ exactly and it must be estimated. Stochastic approximation (SA) and stochastic quasigradient (SQG) algorithms are stochastic variants of a steepest-descent search. The Keifer-Wolfowitz SA method uses unbiased estimates of $Ef(x, \xi)$ to form finite-difference approximations of the gradient. The Robbins-Monro SA procedure requires unbiased estimates of $\nabla Ef(x, \xi)$. SQG methods do not require that $Ef(x, \xi)$ be differentiable and work under more general assumptions concerning the estimates of (sub)gradients of $Ef(x, \xi)$. In particular, the estimates need not be unbiased but the bias must effectively shrink to zero as the algorithm proceeds. For convergence properties of SA methods see [48] and for SQG procedures see [22].

Cutting-plane methods are applicable when $Ef(x, \xi)$ is convex. The iterates $\{x^\ell\}$ are found by solving a sequence of optimization problems of the form

$$\min_{x \in X} \max_{\ell = 1, \ldots, L} Ef(x^\ell, \xi) + \nabla Ef(x^\ell, \xi)(x - x^\ell),$$

where $L$ grows as the algorithm proceeds. At each iteration a first-order Taylor approximation of $Ef(x, \xi)$, i.e., a cutting plane, is computed at the current iterate $x^\ell$ and is used to refine the piecewise-linear outer approximation of $Ef(x, \xi)$. The key idea is that this approximation need only be accurate in the neighborhood of an optimal solution. For stochastic linear programs, Dantzig, Glynn, and Infanger...
[15, 36, 37] and Higle and Sen [31, 33] have developed Monte Carlo-based cutting-plane methods by using statistical estimates for the cut intercepts and gradients. Dantzig, Glynn, and Infanger use separate streams of observations of $\xi$ to estimate each cut. The stochastic decomposition algorithm of Higle and Sen uses common random number streams to calculate each cut and employs an updating procedure to ensure that the statistical cuts are asymptotically valid (i.e., lie below $Ef(x, \xi)$). Relative to SA and SQG methods, cutting-plane procedures avoid potentially difficult projections and, in practice, have a reputation for converging more quickly, particularly when $X$ is high dimensional.

**Grid search** and optimization of meta-models are two common approaches to optimizing system performance in discrete-event simulation models. In grid search, $X$ is replaced by a “grid” of points $X_m = \{x^1, \ldots, x^m\}$ and sample-mean estimates $\bar{f}_n(x) = \frac{1}{n} \sum_{i=1}^{n} f(x, \xi^i)$ are formed at each $x \in X_m$. $(SP)$ is then approximately solved by $z_n^* = \min_{x \in X_m} \bar{f}_n(x)$ with $z_n^*$ being the associated minimizer. Grid search is attractive because it requires minimal structure, but in implementing this procedure, we must exercise care in selecting $m$ and $n$. With independent sampling at each grid point, Ensor and Glynn [21] consider the rate at which $n$ must grow relative to $m$ in order to achieve consistency and they also discuss the method’s limiting behavior when the rate of growth is at (and slower than) the critical rate.

A meta-model can be used to approximate a more complex simulation model which, in turn, is an approximation of the real system. In such a meta-model, estimates of $Ef(x, \xi)$ are formed at each point in a set specified by an experimental design, and the parameters of the postulated response surface are fit to these observed values. The resulting function is then optimized with respect to $x$. For more on meta-models see, e.g., [11, 46]. The review in [24] includes optimization using response surfaces, and meta-modeling has also been applied in stochastic programming [5].

The grid-search and meta-model approaches are classified as external sampling procedures if the procedure is executed once. However, it may be desirable to refine the grid (or the region covered by the experimental design) in the neighborhood of promising values of $x$ and repeat the methodology. When it is adaptively repeated in this fashion the procedure is classified as an internal sampling method.

We have not explicitly discussed approaches for when $X$ is discrete. These range from methods for selecting the best design in simulation to those for solving stochastic integer programming models. Finally, sampling-based procedures for multi-stage stochastic programs have been proposed in [17].

### 3. Establishing Solution Quality

Establishing solution quality is a key concept when using an approximation scheme to solve an optimization problem. When applying Monte Carlo techniques to $(SP)$, the best we can expect are probabilistic quality statements. In the context of external sampling, there has been significant work on studying the behavior of solutions to $(SP_n)$ for large sample sizes (see Section 5). There are analogous convergence results for algorithms based on internal sampling. Such results take a number of forms but perhaps the most fundamental is to show that limit points of the sequence of solutions are, say, almost surely optimal to $(SP)$. Next, it is desirable to have a statement regarding the rate of convergence and an associated asymptotic distribution. These consistency and limiting distribution results are aimed at justifying sampling-based methods and may be viewed as establishing solution quality. However, the approach discussed in this section centers on the question: Given a candidate solution $\hat{x} \in X$, what can be said regarding its quality? Because candidate solutions may be obtained by internal or external sampling schemes or via another, heuristic, method,
procedures that can directly test the quality of $\hat{x}$, regardless of its origin, are very attractive.

One natural way of defining solution quality is by the optimality gap, $Ef(\hat{x}, \xi) - z^*$. An optimal solution has an optimality gap of zero, but in our setting we hope to make probabilistic statements such as

$$P\{Ef(\hat{x}, \xi) - z^* \leq \epsilon\} \geq \alpha, \quad (2)$$

where $\epsilon$ is a random confidence interval width and $\alpha$ is a confidence level, e.g., $\alpha = 0.95$. Unfortunately, exact confidence intervals such as (2) can be difficult to obtain even in relatively simple statistical settings so we attempt to construct approximate confidence intervals

$$P\{Ef(\hat{x}, \xi) - z^* \leq \epsilon\} \approx \alpha. \quad (3)$$

To form a confidence interval (3) for $Ef(\hat{x}, \xi) - z^*$ we estimate the mean of a gap random variable $G_n = U_n - L_n$ that is expressed as the difference between upper and lower bound estimators and satisfies $EG_n \geq Ef(\hat{x}, \xi) - z^*$.

In many problems it is relatively straightforward to estimate the performance of a suboptimal decision $\hat{x}$ via simulation. For example, the standard sample mean estimator, $U_n = \frac{1}{n} \sum_{i=1}^{n} f(\hat{x}, \xi^i)$, provides an unbiased estimate of the expected cost of using decision $\hat{x}$, i.e., $Ef(\hat{x}, \xi)$.

To construct a confidence interval for the optimality gap we also want an estimate of $z^*$. However, unbiased estimates of $z^*$ are difficult to obtain so an estimator $L_n$ that satisfies $EL_n \leq z^*$ is used. Mak et al. [50] show that if the objective in (SP$_n$) is an unbiased estimate of $Ef(x, \xi)$ then $EZ_n^* \leq z^*$, i.e., $z_n^*$ is one possible lower bound estimator $L_n$. Higle and Sen [32] perform a Lagrangian relaxation of a reformulation of (SP$_n$) which uses explicit “nonanticipativity” constraints. The resulting lower bound is weaker in expectation than $z_n^*$ but has the computational advantage that the optimization problem separates by scenario.

Once observations of $G_n$ can be formed, we can appeal to the batch means method and use the central limit theorem [50], or a nonparametric approach [30, 32], to construct approximate confidence intervals (3). Another approach to examining solution quality is to test the null hypothesis that the (generalized) Karush-Kuhn-Tucker (KKT) optimality conditions are satisfied; see Shapiro and Homem-de-Mello [62]. Higle and Sen [30] also consider the KKT conditions but use them to derive bounds on the optimality gap.

4. Variance Reduction Techniques

When applying the “crude” Monte Carlo method to estimate $Ef(x, \xi)$ for fixed $x$, we use the standard sample mean estimator based on i.i.d. terms, $\frac{1}{n} \sum_{i=1}^{n} f(x, \xi^i)$. The error associated with this estimate is proportional to

$$\left[ \frac{\text{var} f(x, \xi)}{n} \right]^{1/2}. \quad (4)$$

This error can be decreased by increasing the sample size. However, obtaining an additional digit of accuracy requires increasing the sample size by a factor of 100. If $f$ is defined as the optimal value of a mathematical program or as the performance measure of a simulation model, increasing the number of evaluations of $f$ in this fashion can be prohibitively expensive. Variance reduction techniques (VRTs) effectively decrease the numerator in (4) instead of increasing the denominator. Many problems for which crude Monte Carlo would yield useless results are instead made computationally tractable via VRTs. As described in Section 2, sampling is also used to estimate $\nabla Ef(x, \xi)$, but for simplicity we primarily restrict our attention to VRTs for estimating $Ef(x, \xi)$.

Some VRTs, including control variates (CVs) and importance sampling (IS), exploit special structures of $f(x, \xi)$. Suppose that we have $\Gamma_x(\xi)$, with known mean $\mu_T$, which is believed to approximate (be positively correlated with) $f(x, \xi)$. In CVs we attempt to “subtract out” variation by generating observations of
Another group of VRTs attempts to more regularly spread the sampled observations over the support of $\xi$. Such techniques include stratified sampling and Latin hypercube sampling as well as quasi-Monte Carlo techniques in which the sequence of observations is deterministic. Empirical results in Higle [29] for two-stage stochastic linear programming compare the variance reduction obtained by stratified sampling, antithetic variates, IS, and CVs and suggest that a CV procedure performs relatively well, particularly on high-variance problems.

5. Theoretical Justification for Sampling

In Section 2 we formed an approximating problem for external sampling procedures by using the sample mean estimator of $Ef(x, \xi)$. Here we redefine $(SP_n)$ as

$$z^*_n = \min_{x \in X} E_n f(x, \xi), \quad (SP_n)$$

with $x^*_n$ again denoting an optimal solution. In $(SP)$ the expected value operator $E$ is with respect to the “true” probability measure $P$ while in $(SP_n)$, $E_n$ is with respect to a measure $P_n$ that is a statistical estimate of $P$. If Monte Carlo methods are used to generate i.i.d. replicates from $P$ then $P_n$ is the associated (random) empirical measure.

Since $z^*_n$ is an estimator of $z^*$ and $x^*_n$ an estimator of an optimal solution to $(SP)$, it is natural to study the behavior of these estimators for large sample sizes. For example, under what conditions do we obtain consistency and what can be said concerning rates of convergence? Positive answers to such questions provide theoretical justification for employing external Monte Carlo sampling techniques to solve $(SP)$.

In general, $(SP_n)$ and $(SP)$ may have multiple optimal solutions and so we cannot expect $\{x^*_n\}$ to converge. Instead, establishing consistency of $x^*_n$ amounts to showing that the accumulation points of the sequence are almost surely optimal to $(SP)$. If, for example, the samples are i.i.d. then by the strong law of large numbers we have $E_n f(x, \xi) \to Ef(x, \xi)$, a.s., for all $x$. Unfortunately, this does not ensure that
\( \{x_n^*\} \) has accumulation points that are optimal to \((SP)\) and that \( z_n^* \rightarrow z^*, \ a.s. \ [4]\).

The notion of epi-convergence plays a fundamental role in establishing consistency results for \( x_n^* \) and \( z_n^*; \) see Attouch and Wets [4]. A sequence of functions \( \{\phi_n\} \) is said to epi-converge to \( \phi \) (written \( \phi_n^\text{epi} \rightarrow \phi \)) if the epi-graphs of \( \phi_n \), \( \{(x, \beta) : \beta \geq \phi_n(x)\} \), converge to that of \( \phi \). Epi-convergence is weaker than classical uniform convergence. Kall [40] provides an excellent review of various types of convergence, their relations, and their implications for approximations of optimization models. Epi-convergence is a value property because of the following result:

Suppose \( \phi_n^\text{epi} \rightarrow \phi \). If \( \hat{x} \) is an accumulation point of \( \{x_n^*\} \), where \( x_n^* \in \text{argmin} \phi_n(x) \), then \( \hat{x} \in \text{argmin} \phi(x) \).

Constrained optimization is captured in this result because \( \phi_n \) and \( \phi \) are defined to be extended-real-valued functions that take value \(+\infty\) at infeasible points. While it is possible that the sequence of optimizers \( \{x_n^*\} \) has no accumulation points, this potential difficulty is avoided if the feasible region \( X \) is compact (i.e., closed and bounded).

Because of the implications of epi-convergence, there is considerable interest in determining sufficient conditions on \( f \), \( P_n \), and \( P \) under which \( E_n f(x, \xi) \xrightarrow{\text{epi}} Ef(x, \xi) \), a.s. Note that because \( \{P_n\} \) are random measures, the epi-convergence of the approximating functions is with probability one (also called epi-consistency). Under this hypothesis the accumulation points of \( \{x_n^*\} \) are almost surely optimal to \((SP)\); see Dupačová and Wets [18].

Sufficient conditions for achieving \( E_n f(x, \xi) \xrightarrow{\text{epi}} Ef(x, \xi) \), a.s. are examined in Dupačová and Wets [18], Kall [41], Robinson [54], and Robinson and Wets [55]. Roughly speaking, we will obtain epi-consistency if \( f \) is sufficiently smooth, \( P_n \) converges weakly to \( P \) with probability one, and the tails of the distributions are well-behaved relative to \( f \). See [2, 59] for results when \( f \) is discontinuous.

For two-stage stochastic programming in which the recourse matrix \( W \) in (1) is deterministic and \( P_n \) is the empirical measure, King and Wets [45] obtain consistency results under modest assumptions. We note that is possible to develop consistency results using other (stronger) types of convergence of \( E_n f(x, \xi) \) to \( Ef(x, \xi) \); see, for example, Pflug et al. [51].

There is a large literature on consistency, stability, and rates of convergence for solutions of \((SP_n)\). Much of this work may be viewed as generalizing earlier results on constrained maximum likelihood estimation by Aitchison and Silvey [1] and Huber [35]. Under restrictive assumptions, asymptotic normality for \( \sqrt{n} (z_n^* - z^*) \) and \( \sqrt{n} (x_n^* - x^*) \) may be obtained, e.g., [18]. However, when inequality constraints in \( X \) play a nontrivial role we cannot, in general, expect to obtain limiting distributions that are normal [43, 61]. King and Rockafellar [43] obtain a limiting distribution for \( \sqrt{n} (x_n^* - x^*) \) that is the solution of a (random) quadratic program.

References


David P. Morton and Elmira Popova
Operations Research and Industrial Engineering
The University of Texas at Austin
Austin, Texas
USA
E-mail address: morton@mail.utexas.edu
elmira@mail.utexas.edu

AMS 1991 Subject Classification: 90 C 15, 65 C 05, 65 K 05, 90 C 31, 62 F 12.
Key words and phrases: stochastic programming, simulation-based optimization, Monte Carlo methods.