© Journal of Applied Mathematics & Decision Sciences, 4(1), 17–38 (2000)
Reprints Available directly from the Editor. Printed in New Zealand.

STRATIFIED FILTERED SAMPLING IN STOCHASTIC OPTIMIZATION

ROBERT RUSH
Investment Policy and Research Group
John Hancock Mutual Life Insurance Company
Boston, Massachusetts

JOHN M. MULVEY
Princeton University Princeton, NJ

JOHN E. MITCHELL AND THOMAS R. WILLEMAIN
Rensselaer Polytechnic Institute
Troy, New York

Abstract. We develop a methodology for evaluating a decision strategy generated by a stochastic optimization model. The methodology is based on a pilot study in which we estimate the distribution of performance associated with the strategy, and define an appropriate stratified sampling plan. An algorithm we call filtered search allows us to implement this plan efficiently. We demonstrate the approach's advantages with a problem in asset / liability management for an insurance company.

Keywords: variance reduction, stratified sampling, stochastic optimization, performance evaluation.

1. Introduction and Background

Many significant problems dictate the development of strategies for handling sequential decision-making under uncertainty, e.g., natural resource planning [25], financial planning [3, 6], and telecommunications network expansion planning [27]. In such situations, there exists a planning horizon that consists of T stages. The beginning of stage 1 represents the current time. During each stage the decision-maker must select a course of action that affects the actions he/she can take in subsequent stages. See figure 1. The end of the planning horizon usually represents the point at which some critical action must be taken or some critical goal achieved. The goal of multi-stage stochastic optimization (MSO) is to develop a sequence of decisions that maximizes the extent to which this goal is achieved. Estimating the expected performance of a decision strategy returned by the MSO process is crucial to increasing the technology's effectiveness. Proper evaluation allows decision-makers to compare MSO-generated strategies with other alternatives available to them in a statistically valid manner.

In this paper, we present a variance reduction technique called stratified filtered sampling (SFS) that greatly improves the computational efficiency of this evalua-

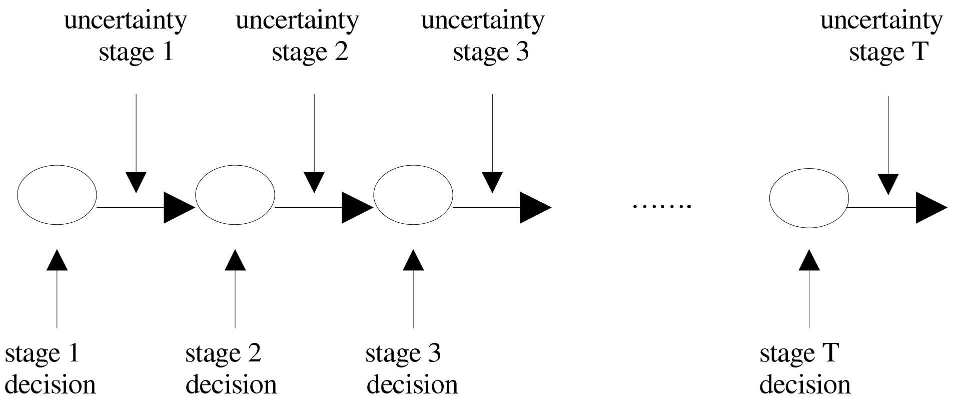


Figure 1. Depiction of sequential decision making under uncertainty.
(Heavy arcs represent the random variables for the uncertainty in each stage.)

tion process. Variance reduction methods attempt to reduce the standard error of the estimate of expected performance without introducing bias into the estimation process. The technique presented herein is based on research associated with the Ph.D. dissertation of the first author [26].

1.1. The Multi-Stage Stochastic Optimization Process

We first introduce some notation. We let X_t, for $t = 1, ..., T$, be the set of feasible decisions available at the start of period t, with $X = (X_1, ..., X_T)$. We let x_t, for $t = 1, ..., T$, be the decision(s) actually made at the start of period t, with $x = (x_1, ..., x_T)$. Also, we let Ω_t, for $t = 1, ..., T$, be the vector of random variables representing the uncertain components associated with period t, with $\Omega = (\Omega_1, ..., \Omega_T)$. Ω has associated with it a probability function Θ. Let ω_t be a distinct realization of Ω_t, with $\omega = (\omega_1, ..., \omega_T)$. We call ω a scenario. Finally, the function $\mathtt{z}(x, \omega)$ measures the performance associated with x under scenario ω; we call $\mathtt{z}$ the performance function. (The vehicle employed to measure performance varies from application to application. The measures we employ in our work incorporate the decision-maker's attitude towards risk, as Section 5 will demonstrate.)

An asset management example will clarify this notation further. Our goal in this example is to maximize expected wealth at the end of a 10-year planning horizon. We allow adjustments to our portfolio (buying and selling of assets) at the start of each year. The set of possible buying and selling decisions at the start of each year corresponds to X_t; the actual buying and selling we implement corresponds to x_t. Furthermore, Ω_t corresponds to the collection of possible returns on the assets in our portfolio in year t, ω_t to the returns that actually manifest themselves. Therefore $\mathtt{z}(x, \omega)$ represents the wealth we accumulate over the 10-year planning horizon via

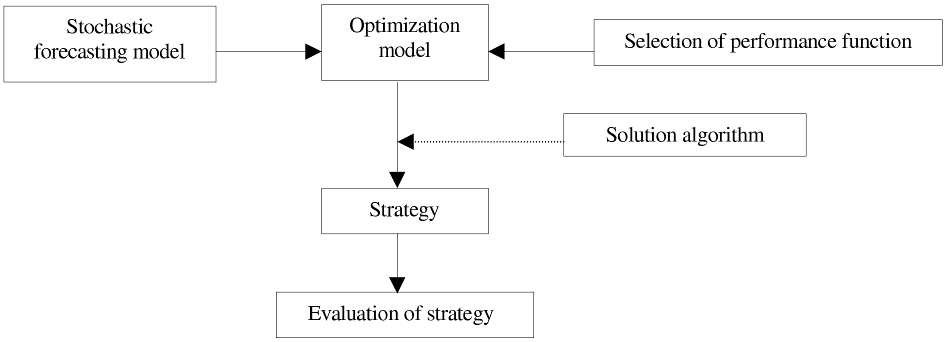


Figure 2. The MSO process

a progression of distinct buying and selling decisions (x) for a given sequence of actual asset returns (ω).

The MSO process has four principal components: selection of the performance function, optimization, stochastic forecasting, and evaluation. See Figure 2. As stated earlier, in our work the first component revolves around selection of an appropriate vehicle for addressing the decision-maker's attitude towards risk, e.g., the von Neumann-Morganstern expected utility model [28] or the nonlinear penalty-based method [7].

The optimization component recommends the best actions for each stage of the planning horizon. A common form (CF) of the associated model is as follows:

$$\begin{array}{lllr} \text{Model CF:} & \mathtt{max}_{\mathtt{x}} & \int_{\Omega} \mathtt{z}(x,\omega)\Theta(\mathtt{d}\omega) & \text{(CF-obj)} \\ & \text{s.t.} & A_{t,\omega} x_t = b_{t,\omega} & \text{(CF-a)} \\ & & A_{t,\omega} = \mathtt{u}_t(\omega, x_{t-1}) & \text{(CF-b)} \\ & & b_{t,\omega} = \mathtt{v}_t(\omega, x_{t-1}) & \text{(CF-c)} \end{array}$$

Constraints (CF-a) define the set of feasible decisions in stage t under scenario ω. (CF-b) and (CF-c) show, through the vector-valued functions $\mathtt{u}_t$ and $\mathtt{v}_t$, that (CF-a) depend on both the manifestation of uncertainty and the decisions made in the previous period. We can readily adapt (CF) to handle the case in which periods prior to the previous one have an impact.

Unfortunately, there usually exists no closed form expression for the integration in (CF-obj); this effectively precludes direct solution efforts. To address this, we employ the third component of the MSO process: a stochastic forecasting model. We generate with this model a set of scenarios to serve as a proxy for the uncertainty space Ω; these scenarios comprise the solution generation set. Our goal is to optimize over the solution generation set rather than Ω. (Examples of stochastic forecasting systems for asset / liability management include Russell's vector autoregressive system [6], the Towers' Perrin CAP:Link system [20], and Wilkie's investment model [29].)

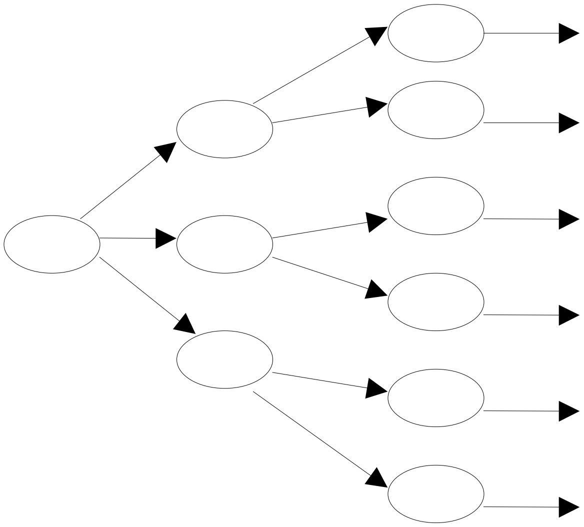


Figure 3. A 3-stage, 6-scenario tree
The nodes again represent decisions, but the arcs are now distinct realizations of uncertainty.

By construction, any decision strategy developed by optimizing over a distinct solution generation set performs well when applied to scenarios in the set. The likelihood of the uncertainty manifesting itself as one of these scenarios, however, is exceedingly small. Consequently, proper evaluation of the strategy (the final phase of the process) is essential. Its goal is to assess the performance of a decision strategy in scenarios outside the relevant solution generation set, or more specifically, estimate to an acceptable accuracy the expected performance of the generated strategy with respect to the universe of all possible scenarios.

The approach employed to generate the solution generation set and solve the resulting optimization problem determines to a large extent our ability to conduct this testing. We present two basic approaches. The first generates the scenarios as a *tree*. See Figure 3. Each path through the tree defines a scenario, with each arc representing a specific manifestation of uncertainty for a distinct stage of the planning horizon. Just as a given arc appears in multiple paths, so too does the associated manifestation of uncertainty appear in multiple scenarios.

To define the relevant optimization problem (that we call the tree form), we require some additional notation. Let TREE denote the solution generation set, Θ^{ω} the probability of scenario ω occurring, and $x_{\omega,t}$ the decision made in scenario ω at stage t, with $x^{\omega} = (x_{\omega,t})$, for $t = 1, ..., T$. Finally, let $N_{\omega}(t)$ be the set of all scenarios that are identical to scenario ω through stage t (recall the possible "overlap" among scenarios described above). We have the following:

$$\begin{array}{llcl} \text{Model TF:} & \mathtt{max}_{\mathtt{x}}: & \sum_{\omega\in\mathtt{TREE}}\{\Theta^{\omega}z(x^{\omega},\omega)\} & \text{(TF-obj)} \\ & \text{s.t.:} & A_{t,\omega}x_t = b_{t,\omega} & \text{(TF-a)} \\ & & A_{t,\omega} = u_t(\omega, x_{t-1}) & \text{(TF-b)} \\ & & b_{t,\omega} = v_t(\omega, x_{t-1}) & \text{(TF-c)} \\ & & x_{\tilde{\omega},t} = x_{\dot{\omega},t} \text{ for all } \tilde{\omega}, \dot{\omega} \text{ in } N_{\omega}(t) & \text{(TF-d)} \end{array}$$

(TF) differs from (CF) in its objective function and in the presence of constraints (TF-d). As mentioned earlier, (TF)'s objective function averages performance over the members of the solution generation set rather than integrating over the space Ω. Constraints (TF-d) are called *nonanticipativity* constraints; they require that all decisions made with the same information be identical.

(TF) is a deterministic program that maximizes a concave objective function (usually) over a convex region (always). All the advantages of convex minimization are therefore relevant. Efforts to find effective solution methodologies have given rise to the field of multi-stage stochastic programming with recourse. Kall and Wallace [15] and Birge [4] provide excellent introductions and guidance for further reading.

Evaluating the solution generated by a (TF) model is difficult because it does not generate a strategy that can address arbitrary realizations of uncertainty. Only the decision returned for the first stage can handle any realization of uncertainty. The remaining decisions - those for stages 2 or beyond - are defined (meaningful) only for the specific scenarios on which they are based. This causes difficulty when applying Monte Carlo simulation to estimate the expected performance [12].

In the second approach, the scenarios in the solution generation set possess a string structure. See Figure 4. Because independent sampling from the forecasting model creates each scenario, there is no overlap among different scenarios, as in a scenario tree. Our goal in this approach is to find the best member of a particular family of *decision rules*. A decision rule is a function r that maps Ω_t into X_t ; it thereby dictates the actions to be taken at any stage in the planning period. (Formally, we have $r : \Omega_t \rightarrow X_t$) A family of decision rules is a set of functions $R = \{r_1, r_2, \ldots\}$, with each member of R representing a distinct instance of the given family. The values assumed by a vector of parameters $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ dictate the member of R that operates in a particular decision-making environment.

We refer to the optimization problem associated with this approach as the string form (SF). Let r be an arbitrary member of decision rule family R, $x^r_{\omega,t}$ the decision made in scenario ω at stage t as dictated by rule r, and $x^r_{\omega} = (x^r_{\omega,t})$, for $t = 1, \ldots, T$. Denoting the relevant solution generation set as STRINGS, we get the following:

$$\begin{array}{llcl} \text{Model SF:} & \mathtt{max}_{\alpha} & \sum_{\omega\in\mathtt{STRINGS}}\{\Theta^{\omega}z(x^r_{\omega},\omega)\} & \text{(SF-obj)} \\ & \text{s.t.:} & x^r_{\omega,t} = r(\omega_t) & \text{(SF-a)} \\ & & r = g(\alpha) & \text{(SF-b)} \\ & & \alpha \in A & \text{(SF-c)} \end{array}$$

Here, constraints (SF-a) describe the dependence of the actions taken on the form assumed by the decision rule. Constraints (SF-b) describe the dependence of the actual rule to be implemented on α. Constraints (SF-c) establish limitations on the

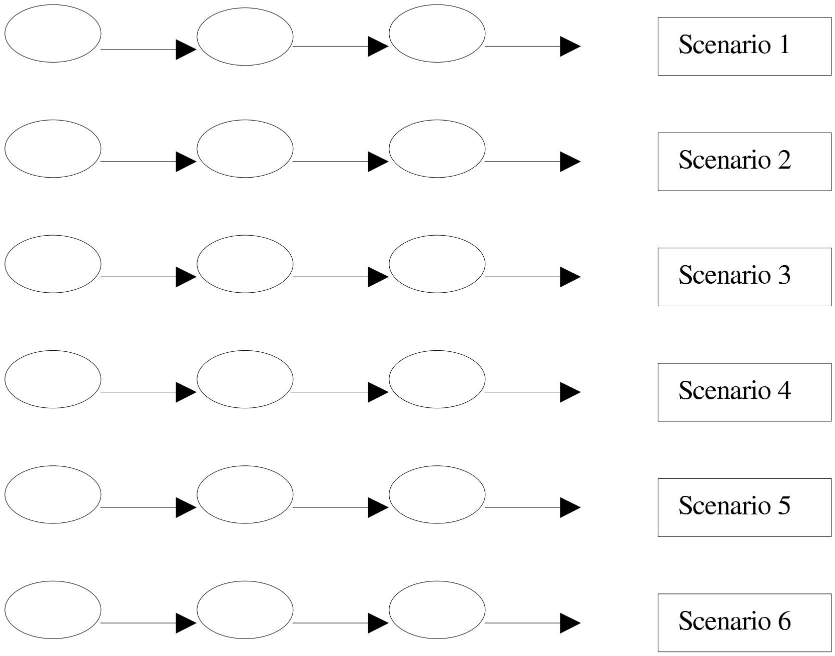


Figure 4. A solution generation set containing six 3-stage scenarios in string form

set of possible rules. Solving model (SF) is difficult because its objective function is often non-concave (recall it is a maximization problem).

Offsetting this disadvantage is the model's capacity to support Monte Carlo-based evaluation. An (SF) model returns a decision rule; this rule is by definition a strategy that can address any realization of uncertainty throughout the planning horizon. (Recall the mapping on which the rule is based.) Consequently, testing on scenarios outside the solution generation set is much easier than with strategies returned by (TF) models. In effect, model (SF) purchases increased testability at the expense of the solution difficulties caused by its non-concavity. For a more detailed examination of the relative merits of the tree- and string-based approaches, see Rush [26].

1.2. An Overview of Stratified Filtered Sampling

We specifically developed stratified filtered sampling to address the evaluation of rules generated by string-form MSO models. As its name implies, the methodology is an extension of the well-known sampling technique called stratified sampling (see Cochran [9], hereafter referred to as Cochran). Previous applications of stratified sampling to Monte Carlo simulation have relied principally on stratification of the n-dimensional hypercube of uniform random numbers that drive the simulation (see, for example, Niederreiter [24]). SFS relies on direct stratification of the distribution of performance associated with the MSO-generated strategy (hereafter called the performance distribution); it selectively evaluates those scenarios that contribute most to the variability of performance. The estimate it generates we call the *stratified performance estimate*, or SPE. We note that other techniques for variance reduction exist: correlation induction (which includes antithetic sampling [8] and Latin hypercube sampling [18]), control variates [16], conditional expectation [17], importance sampling [10], and the relatively new "quasi Monte-Carlo" methods [14]. Although developed for use within an MSO context, stratified filtered sampling, like these other techniques, applies to other simulation environments as well.

The SPE-generation process consists of two steps. In the first, we conduct a pilot study to estimate the performance distribution, and design the stratified sampling scheme. In the second step, we employ the filtered search algorithm to efficiently implement the proposed scheme.

The rest of this paper is organized as follows. Section 2 introduces the "classical" version of stratified sampling, motivating the advantages of our approach. Section 3 describes the components of the SPE-generation procedure. Section 4 shows that SPE is unbiased. Section 5 discusses the application of SFS to asset / liability management for an insurance company, and demonstrates the computational savings afforded by filtered search. Section 6 presents concluding remarks.

2. Introduction to Stratified Sampling

Let P be a population with mean μ_P and standard deviation σ_P . Assume that P has been divided into J mutually exclusive and collectively exhaustive subpopulations, or strata: $P_1, \ldots, P_J$. Let λ_j be the probability that a random draw from P will yield an element of stratum P_j, for $j = 1, \ldots, J$. Let σ_j for $j = 1, \ldots, J$ be the standard deviation of stratum j. A stratified sampling plan defines, for a given total sample size N_{SS}, the size of the random sample to be drawn from each stratum j , N_j. Let $\bar{y}_j$ be the sample mean associated with the sample drawn from stratum j. The stratified sampling based estimate of μ_p is $\bar{y}_{SS} = \sum_j (\lambda_j \bar{y}_j)$. We have the following results for "classical" stratified sampling:

THEOREM 1 *(Cochran, p. 91): Assume that λ_j for $j = 1, \ldots, J$ is known with certainty. Then $\bar{y}_{SS}$ is an unbiased estimate of μ_P.*

THEOREM 2 *(Neyman [23]): The variance of $\bar{y}_{SS}$ is minimized for a total sample of size N_{SS} if*

$$N_j = N_{SS} \left(\frac{\sigma_j \lambda_j}{\sum_j (\sigma_j \lambda_j)} \right), \textit{ for } j = 1, \ldots, J.$$

THEOREM 3 *Theorem 3 (Cochran, p. 115): LetN_j^* be the optimal size of the random sample drawn from stratum j (as defined in Theorem 2). Assume that this optimal allocation plan is not implemented, and that instead we draw a sample of size $\underline{N}_j$ from each stratum j, such that $\sum_j (\underline{N}_j) = N_{SS}$. Then the proportional increase in the variance of $\bar{y}_{SS}$resulting from these deviations from the optimal stratified sampling plan is*

$$\frac{1}{N_{SS}} \sum_j \left(\frac{(\underline{N}_j - N_j^*)^2}{N_j^*} \right).$$

Two critical issues are the definition of the strata boundaries and the determination of the appropriate number of strata. Cochran provides some basic theory, as well as references for further reading. For the present discussion, we note that the principal goal underlying both questions is the maximization of both intra-stratum homogeneity and inter-stratum heterogeneity (Mulvey [21]). In other words, a good stratification identifies those regions of the population that contribute most to variability and isolates them as strata. Sampling thus is focused on those regions for which uncertainty is greatest. Figure 5 presents a form the performance distribution might assume. Clearly, the right tail contributes most to overall variability. The pilot study will yield enough information about the distribution's shape to allow us to isolate this tail as a distinct stratum.

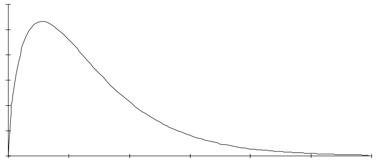


Figure 5. Possible form of f^r

3. Generating the Stratified Performance Estimate

We first review and/or introduce the following notation:

- $r :=$ the decision rule being evaluated.
- $u :=$ an appropriately-sized vector of uniform $(0, 1)$ random numbers, with U the space of all such vectors.
- $\omega :=$ a scenario.
- $f^r :=$ the density function of the performance distribution for rule r, with mean μ^r.
- $z(x^r_\omega, \omega) :=$ the value of the performance function associated with rule r under scenario ω. When ω is generated randomly, $z(x^r_\omega, \omega)$ is a tantamount to a random draw from f^r. For the sake of notational convenience, we abbreviate $z(x^r_\omega, \omega)$ as $z(x^r, \omega)$.
- $G :=$ the (continuous) function which maps U into Ω . G is the scenario generator function.

In terms of this notation, we generate SPE via stratification of the set of performance values for which f^r is the density. The description of stratified sampling in section 2 reveals three problems we must address in order to accomplish this. Problem 1 is that we usually have no knowledge of the structure of f^r. This information is vital for identifying the collections of scenarios for which performance is extreme - a key to the effectiveness of stratified sampling. Problem 2 also is due to lack of knowledge of f^r - the λ's are not known with certainty. Thus, we cannot directly invoke Theorem 1 to prove that SPE is unbiased. (Section 4 shows that SPE is in fact an unbiased estimator of μ^r). The third problem is that we have no ready means of separating the scenario space into a collection of regions that we can map in a well-defined fashion into the strata on which our stratified sampling

Preliminaries
- Define rule r
- Define acceptable standard error of estimation

Phase A: Devise Stratified Sampling Plan via Pilot Study
A.1 Estimate f ^{r}.
A.2 Define strata for f ^{r}.
A.3 Insure precision of relevant strata parameters
A.4 Define size of sample required from each stratum

Phase B: Implement Stratified Sampling Scheme
B.1 Naïve implementation
B.2 Filtered search implementation

Figure 6. The SPE Generation Process

plan is based. Consequently, the implementation of the stratified sampling plan is computationally very difficult.

We address problems 1 and 3 by employing a two-phase procedure to generate SPE. In the first phase (phase A), we randomly sample from f^r to obtain adequate knowledge of its structure. This provides the blueprint for the stratified sampling plan. The second phase (phase B) implements the stratified sampling plan. We have identified two possible implementation schemes. The first, the "naïve" implementation, is guaranteed to work, but may require great computational effort. The second is a much less computationally intensive scheme we call "filtered search". Figure 6 displays the SPE generation process.

The "Preliminaries" - definition of the relevant strategy and selection of the acceptable standard error - require no explanation. We provide the details of Phases A and B below.

Phase A: Devise Stratified Sampling Plan via Pilot Study

A.1: Estimate f^r. Randomly sample from f^r to construct an adequate representation of its structure, to wit:

- Draw a sample of size N from U.
- Define the corresponding N scenarios: $\omega = G(u)$, for each u generated.
- Calculate $z(x^r, \omega)$ for each ω generated.

A.2: Define strata for f^r.

Separate f^r into J mutually exclusive and collectively exhaustive regions. (Recall that it is not our primary purpose in this paper to describe how to construct these strata. For our tests, however, 10-15 strata have worked very well.) Let $\mathtt{SAMP}_j$, for $j = 1, \ldots, J$ be the set of random draws from f^r contained in each stratum j. Let $n_j = |\mathtt{SAMP}_j|$, the cardinality of $\mathtt{SAMP}_j$. Let $\hat{\sigma}_j$ be the sample standard deviation of $\mathtt{SAMP}_j$.

A.3: Insure precision of estimates of relevant strata parameters.

The relevant parameters here are λ_j and σ_j. Recall that λ_j is the probability that a random draw from f^r will fall in stratum j, σ_j the true standard deviation of stratum j, $\hat{\lambda}_j = \frac{n_j}{N}$ the sample estimate of λ_j, and $\hat{\sigma}_j$ the sample estimate of σ_j . Theorem 3 clearly reveals the importance of estimating the λ_j's and σ_j's well. To address this issue, we implement the following procedure:

- Calculate $\mathtt{cv}_j = \frac{\sqrt{\hat{\lambda}(1-\hat{\lambda})/N}}{\hat{\lambda}_j}$, for $j = 1, \ldots, J$.
- Let $\mathtt{max} = \mathtt{max}_j(\mathtt{cv}_j)$.
- Adequate Precision Check: Is $\mathtt{max} \leq \delta$? (We suggest $\delta = .20$.)

  if YES: Stop. (Precision of estimates is acceptable.)
  if NO: Go to CORRECT_N.

CORRECT_N

- Determine, based on the current values of the λ_j's, the number of additional draws from f^r needed to satisfy the Adequate Precision Check; call this number B.
- Draw from f^r B times.
- Place each draw from f^r into the appropriate stratum as defined in A.2.
- Update $\hat{\lambda}_j, \hat{\sigma_j}, n_j$, and $\mathtt{cv}_j$ for $j = 1, \ldots, J$.
- Recompute max.
- Return to Adequate Precision Check.

Note that $\mathtt{cv}_j$ is the estimated coefficient of variation for $\hat{\lambda}_j$. The Adequate Precision Check therefore insures that the error in estimating λ_j is small relative to the magnitude of the estimate itself. Furthermore, the procedure indirectly improves the precision with which $\hat{\sigma}_j$ estimates by σ_j increasing the cardinality of the $\mathtt{SAMP}_j$'s.

A.4: Define size of sample required from each stratum.

Let $N_{\mathtt{SPE}}$ be the size of the entire sample needed for the SPE-generation process; $N_{\mathtt{SPE}}$ will be a function of the desired standard error. Let $N_{\mathtt{SPE}}(j)$ be the required size of the random sample from stratum j. As in section 2, let $\bar{y}_j$ be the sample mean associated with the sample from stratum j. Then the stratified performance estimator of μ^r is

$$\mathtt{SPE} = \sum_j \hat{\lambda}_j \bar{y}_j$$

$\mathtt{SPE}$ is identical to $\bar{y}_{\mathtt{SS}}$ from section 2 except that we have replaced λ_j with its estimate, $\hat{\lambda}_j$. The Adequate Precision Check certifies the quality with which we estimate the λ_j's. We estimate $\mathtt{SPE}$'s standard error as:

$$\sqrt{\sum_j \left\{ \hat{\lambda}^2 \left(\hat{\sigma}^2 / N_{\mathtt{SPE}}(j) \right) \right\}}$$

We compute $N_{\mathtt{SPE}}(j)$ using the optimal allocation scheme laid out in Theorem 2:

$$N_{\mathtt{SPE}}(j) = N_{\mathtt{SPE}} \left(\frac{\hat{\lambda}_j \hat{\sigma}_j}{\sum_j \left(\hat{\lambda}_j \hat{\sigma}_j \right)} \right)$$

Calculating the appropriate value for $N_{\mathtt{SPE}}$ is straightforward; we simply increase it until the desired value of standard error is indicated. Once this step is complete, the stratified sampling plan is complete.

Phase B: Implement Stratified Sampling Scheme

The specification of the stratified sampling scheme in A.4 defines the size of the random sample, $N_{\mathtt{SPE}}(j)$, needed from each stratum. Recall, however, that upon leaving A.3 we already have a sample of size n_j from each stratum j. We therefore require an additional sample of size $M_j = \mathtt{max}[N_{\mathtt{SPE}}(j) - n_j, 0]$ from each stratum j in order to satisfy the requirements of the stratified scheme.

B.1: The Naïve Implementation.

This implementation scheme draws randomly from f^r until the required number of additional draws from each stratum have been obtained (figure 7). Because we cannot control the value of these draws, the procedure will almost certainly draw more than necessary from some of the strata.

This implementation scheme can be inefficient. To explain the reason for this, we require the following definitions:

Definition 1 *The* difficulty ratio *for each stratum $j = \mathtt{DR}_j = \frac{M_j}{\lambda_j}$.*

Definition 2 *The* critical stratum *is that stratum for which the difficulty ratio is largest.*

We thus can state:

LEMMA 1 *The expected number of draws from f^r required to satisfy the sample size requirement for each stratum j is equal to $\mathtt{DR}_j$.*

Proof: Immediate.

Lemma 1 shows that on average more simulation runs will be necessary to satisfy the sample size requirement for the critical stratum (hereafter referred to as

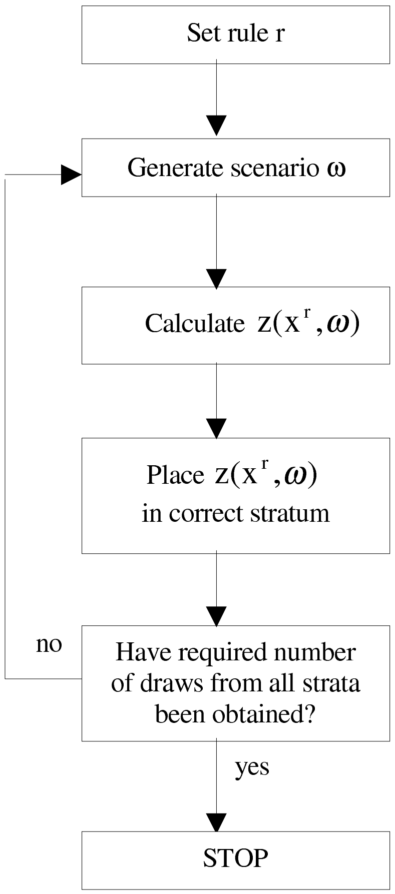

Set rule r
Generate scenario ω
Calculate z(x^r,ω)
Place z(x^r,ω) in correct stratum
Have required number of draws from all strata been obtained?
no
yes
STOP

Figure 7. The naïve implementation

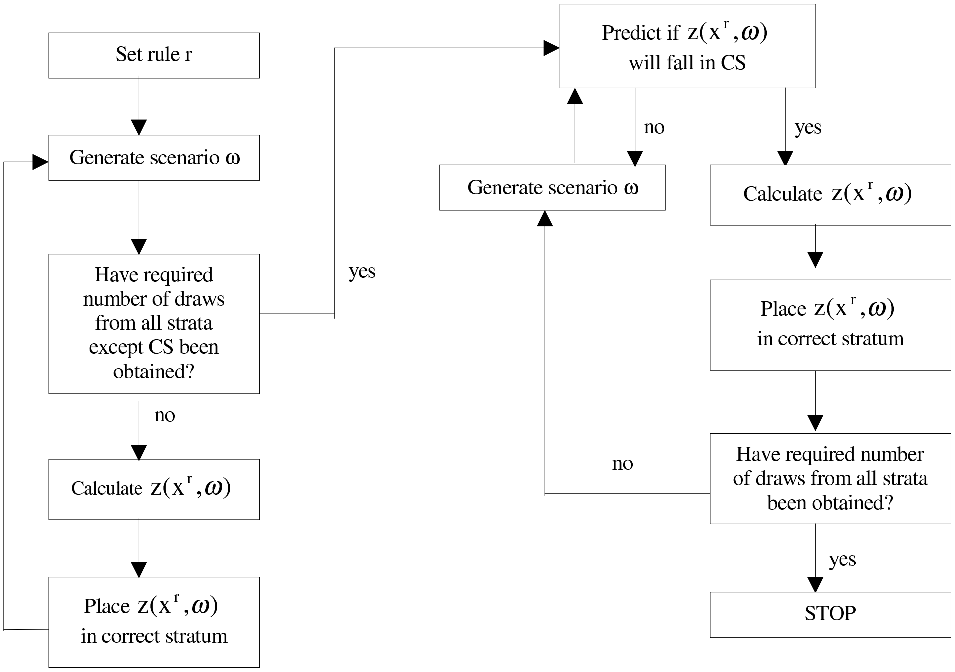

Set rule r
Generate scenario ω
Have required number of draws from all strata except CS been obtained?
yes
no
Calculate $z(x^r, \omega)$
Place $z(x^r, \omega)$ in correct stratum
Predict if $z(x^r, \omega)$ will fall in CS
no
yes
Generate scenario ω
Calculate $z(x^r, \omega)$
Place $z(x^r, \omega)$ in correct stratum
Have required number of draws from all strata been obtained?
no
yes
STOP

Figure 8. Filtered search implementation

CS) than for any other stratum. It therefore constitutes the chief computational obstacle associated with the naïve implementation. This computational difficulty can potentially negate the original benefit of the stratified sampling approach.

B.2: “Filtered Search” implementation.

We address the problems caused by the CS. The basis for the computational benefits of the filtered search approach is that the effort required to generate ω is usually very small relative to that required to calculate $z(x^r, \omega)$, i.e., generation is cheaper than evaluation. Consequently, the majority of the effort required to generate SPE usually lies not in the generation of the scenarios, but rather in the subsequent evaluation of the performance function for all scenarios generated.

Figure 8 presents the modified implementation. The central idea is that once we have obtained the requisite draws from all strata except the CS, we no longer blindly draw from f^r. Instead, we predict for each subsequently generated scenario ω whether or not $z(x^r, \omega)$ will fall within the CS or not. (We construct the predictor function with the information obtained during the pilot study, as explained below.) Only if we predict that $z(x^r, \omega)$ will fall within the CS do we actually conduct the relevant simulation run. Thus, we incur the computational burden of calculating $z(x^r, \omega)$ only if it seems likely that doing so will serve our purpose.

The key to the success of the Filtered Search scheme is the quality of the predictor mechanism. As in all 0-1 classification problems, two errors are possible (See Table 1). The quality of the predictor mechanism is primarily dependent on the minimization of Prob (Type I error). There are two reasons for this. The first concerns computational efficiency. Because elements of the CS occur by definition with extremely low frequency, we must minimize the possibility that we miss one. The second reason concerns the avoidance of bias in the estimation of the mean of the CS. (As section 4 will show, this is essential if SPE is to be unbiased.) To see how bias can occur, refer again to Figure 5. Assume that the critical stratum there is defined as $[T, \infty)$. Ambiguity over how to classify a given scenario ω will occur most frequently when $z(x^r, \omega)$ is "close" to T. Therefore, a predictor mechanism with high Type I error rate will tend to correctly classify scenarios for which $z(x^r, \omega)$ is "far away" from T, and misclassify scenarios for which $z(x^r, \omega)$ is "close" to T. The result can be a biased sample mean.

		REALITY	
		CS	*Not CS*
DECISION	*Predict CS*	Correct	Type II error
	Predict not CS	Type I error	Correct

Table 1. Possible errors for predicting membership in critical stratum

Many techniques are available for 0-1, or binary, classification, such as logistic regression [13], discriminant analysis [19], CART [5], neural networks [11], and bilinear separation via mathematical programming [2], all of which can be adapted to emphasize the minimization of the Type I error rate. Because the example application in section 5 employs logistic regression, we close this section with a brief introduction to the methodology.

Binary logistic regression is a form of regression in which entities of interest can belong to one of two possible groups, say 0 and 1; our goal is to predict membership. (In our situation, the entities of interest are distinct scenarios ω ; we want to predict whether will $z(x^r, \omega)$ fall in the critical stratum or not.) Let p = probability that the entity of interest is a member of group 1. The technique predicts membership by employing as its dependent variable the log-likelihood ratio that the entity of interest is a member of group 1, or the logit of p:

$$L = \ln(\frac{p}{1-p})$$

and fitting the following model:

$$L = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \mathit{error}$$

Maximum-likelihood, not least squares, is the basis of the fitting process. We estimate p as follows:

$$\hat{p} = \frac{\texttt{exp}(\hat{L})}{1 + \texttt{exp}(\hat{L})}$$

where

$$\hat{L} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_n X_n$$

As one would expect, the default for assigning the value 1 to the entity of interest is a value for $\hat{p}$ greater than .5. Changing this threshold allows us to control Type I and II error rates.

4. Showing that SPE is Unbiased

Recall that $\texttt{SPE} = \sum_j \hat{\lambda}_j \bar{y}_j$. We need to show that $\texttt{E}(\texttt{SPE}) = \mu^r$. We have the following:

LEMMA 2 *Let P be an arbitrary population, with (true) mean μ^P . Let $P_1, \ldots, P_J$ be a set of mutually exclusive and collectively exhaustive strata for P. For each stratum $j, j = 1, \ldots, J$, define the following: μ_j = (true) mean of stratum j and λ_j = (true) probability that an arbitrary member of P is an element of stratum j. Then $\mu^P = \sum_j \lambda_j \mu_j$*

Proof: See Cochran, p. 91.

LEMMA 3 *For any j, let $\hat{\lambda}_j$ and $\bar{y}_j$ be, respectively, the estimated probability that a random draw from f^r will come from stratum j and the sample mean of the $N_{\texttt{SPE}}(j)$ draws from stratum j required by the stratified sampling scheme. Assume that our control of the Type I error rate is sufficient for $\texttt{E}(\bar{y}_j) = \mu_j$. Then $\texttt{E}(\hat{\lambda}_j \bar{y}_j) = \lambda_j \mu_j$.*

Proof: We condition on the random variable $\hat{\lambda}_j$, to wit:

$$\begin{aligned}
\texttt{E}(\hat{\lambda}_j \bar{y}_j) &= \texttt{E}\left\{\texttt{E}\left\{\hat{\lambda}_j \bar{y}_j | \hat{\lambda}_j\right\}\right\} \\
&= \sum_l \left(\texttt{E}\left[\hat{\lambda}_j \bar{y}_j | \hat{\lambda}_j = l\right] \cdot \texttt{Pr}(\hat{\lambda}_j = l)\right) \\
&= \sum_l \left(\texttt{E}\left[l \bar{y}_j\right] \cdot \texttt{Pr}(\hat{\lambda}_j = l)\right) \\
&= \sum_l \left(l\texttt{E}\left[\bar{y}_j\right] \cdot \texttt{Pr}(\hat{\lambda}_j = l)\right)
\end{aligned}$$

$$\begin{aligned}
&= \sum_l \left(l\mu_j \cdot \mathtt{Pr}(\hat{\lambda}_j = l) \right), \text{ because we assumed our control of the Type I error rate allows } \bar{y}_j \text{ to be unbiased} \\
&= \mu_j \sum_l \left(l \cdot \mathtt{Pr}(\hat{\lambda}_j = l) \right) \\
&= \mu_j \mathtt{E}(\hat{\lambda}_j) \\
&= \mu_j \lambda_j
\end{aligned}$$

THEOREM 4 *Under the conditions of Lemma 3,* SPE *is an unbiased estimator of* μ^r*.*

Proof:

$$\begin{aligned}
\mathtt{E}(\mathtt{SPE}) &= \mathtt{E}\left(\sum_j \hat{\lambda}_j \bar{y}_j \right) \\
&= \sum_j \left\{ \mathtt{E}\left(\hat{\lambda}_j \bar{y}_j \right) \right\} \\
&= \sum_j \lambda_j \mu_j \text{ , by Lemma 3.} \\
&= \mu^r, \text{ by Lemma 2.}
\end{aligned}$$

Although we have now proven the desired result, we also state the following for the sake of completeness:

LEMMA 4 *Under the conditions of Lemma 3,* $\mathtt{Cov}(\hat{\lambda}_j, \bar{y}_j) = 0$.

Proof:

$$\mathtt{Cov}(\hat{\lambda}_j, \bar{y}_j) = \mathtt{E}(\hat{\lambda}_j \bar{y}_j) - \mathtt{E}(\hat{\lambda}_j)\mathtt{E}(\bar{y}_j) = \lambda_j\mu_j - \lambda_j\mu_j = 0.$$

5. Example Application

We describe the manner in which multi-stage stochastic optimization allowed us to address issues in multi-year asset / liability management faced by a major reinsurance firm, focusing particularly on the role played by stratified filtered sampling and the effectiveness of filtered search. The firm had a strategy for deciding both the extent of its underwriting and the amount of its capital it would make available for investment. Its problem was to decide how to allocate this available capital among five different investment instruments (A, B, C, D, and E) over a five year planning horizon.

Our first task was to select a performance function. In consultation with the firm, we decided to employ the utility of the firm's net asset position (NAP) at the end of the planning horizon as our measure of performance. (Recall the importance we assign to consideration of the decision-maker's attitude towards risk.) The firm calculated its net asset position at the end of the planning horizon via: $\texttt{NAP} = \frac{wp_5 - wp_0}{wp_0}$, where wp_5 represents wealth position at the end of year 5, and wp_0 initial wealth. (Thus, even though the firm chose a five-year (period) planning horizon, i.e., T = 5, we needed to track the firm's wealth position at six different times.) The utility model we adopted is due to Bell [1]: $\texttt{util}(y) = y - B\exp(-C \cdot y)$.

There were 30 different sources of uncertainty, i.e., Ω in this instance is a 30-vector: the annual returns for each of the investment instruments ($\texttt{ret}_{i,t}$, for $i = A, \ldots, E$ and $t = 1, \ldots, 5$), and the dollar value of the claims paid out at the end of each year (c_t, for $t = 1, \ldots, 5$). We employed two different stochastic forecasting systems to deal with this uncertainty. The CAP:Link system developed by the first author [20] addressed investment return uncertainty. (In other words, CAP:Link serves as the scenario generation function for investment returns. As such, its core engine is a numerical approximation of the probability function Θ - recall section 1.1 - associated with the multivariate return distribution.) A proprietary system developed by the firm addressed claim uncertainty.

Optimization to find an acceptable asset allocation strategy was next. Discussion with the firm resulted in reducing the set of eligible instruments to A, B, and E. Our recommendation (to which the firm agreed) was to find a suitable instance of the "fixed-mix" family of decision rules; we thus opted for the string approach to MSO described in the Introduction. A fixed-mix rule specifies the relative proportion that each investment instrument should occupy in an asset portfolio at the beginning of each stage of the relevant planning period. Only when the returns on all available instruments are identical will these relative proportions continue to satisfy the "fixed-mix". The rule therefore specifies the specific buying and selling necessary to rebalance the portfolio at the start of each year (stage) of the planning horizon. Thus, (x^r, ω) in this situation corresponds to the sequence of buying and selling decisions dictated by fixed-mix rule r under scenario ω. With a fixed-mix rule, wealth position changes over the course of the planning horizon in the following manner:

$$wp_{t+1} = \sum_i \{(1 + \texttt{ret}_{i,t})\,(\alpha_i wp_t)\} - c_t,$$

where α_i is the proportion of wealth to be placed in instrument i per the buying and selling dictated by the relevant "fixed-mix" rule. With $\texttt{NAP}[r, \omega]$ representing the final net asset position achieved by fixed-mix rule r under scenario ω, we formally defined the performance function as:

$$z(x^r, \omega) = \texttt{NAP}[r, \omega] - B \cdot \exp(-C \cdot \texttt{NAP}[r, \omega]),$$

with $B = 10$ and $C = 4$. Creating a solution generation set with 1000 scenarios and letting $\alpha = (\alpha_A, \alpha_B, \alpha_E)$, we constructed the following string model:

$$\max_{\alpha} \frac{1}{1000} \sum_{\omega} z(x^r, \omega)$$

$$\text{s.t.: } \alpha_A + \alpha_B + \alpha_E = 1$$

$$\alpha \geq 0$$

Solving this model yielded a strategy we labeled r1: 40% of available capital in investment A, 40% in investment B, and 20% in investment E.

The SFS process - particularly the filtered search component - proved very successful in evaluating this strategy. Implementation of steps A.1 through A.3 of the SPE-generation procedure required 10,000 draws from f^{r1}. We defined 13 strata (see Table 2).

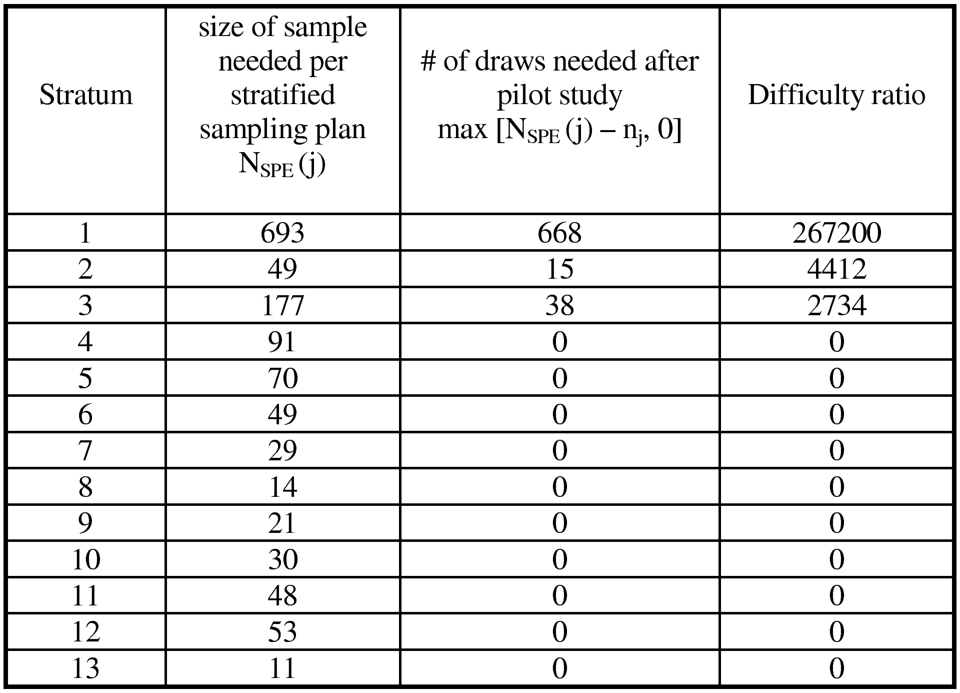
Stratum	Stratum boundaries	Cardinality after pilot sudy n_j	Estimated st. dev. after pilot study $\hat{\sigma}_j$	Estimated probability of occurrence $\hat{\lambda}_j$	Coefficient of variation
1	$-\infty < u \leq -85$	25	189.23	0.0025	.19975
2	$-85 < u \leq -50$	34	9.89	0.0034	.17125
3	$-50 < u \leq -20$	139	8.67	0.0139	.08423
4	$-20 < u \leq -10$	224	2.77	0.0224	.06606
5	$-10 < u \leq -5$	330	1.44	0.033	.05413
6	$-5 < u \leq -2.5$	474	.71	0.0474	.04483
7	$-2.5 < u \leq -1.25$	552	.36	0.0552	.04137
8	$-1.25 < u \leq -.6$	512	.19	0.0512	.04305
9	$-.6 < u \leq 0$	821	.17	0.0821	.03344
10	$0 < u \leq .5$	1392	.15	0.1392	.02487
11	$.5 < u \leq -1.0$	2289	.14	0.2289	.01835
12	$1.0 < u \leq 1.5$	2596	.14	0.2596	.01689
13	$1.5 < u < \infty$	612	.13	0.0612	.03917

Table 2. Core information obtained through pilot study

In consultation with the company's management, we set the desired standard error = .035. Application of (2) and (3) of SPE-generation step A.4 dictated that that $N_{\mathtt{SPE}}$ be 1335. (Note that the sample standard deviation of the 10,000 draws from f^{r1} was 14.772. We thus would require 178,000 simple random samples from f^{r1} to achieve what the stratified approach achieves in 1335.) Table 3 presents the stratified sampling scheme for $N_{\mathtt{SPE}} = 1335$, as well as the difficulty ratio for each stratum. Clearly, stratum 1 is the critical stratum. This makes sense, given that its estimated standard deviation is at least two orders of magnitude larger than that of any other stratum.

Next, we constructed the logistic regression predictor function required for filtered search. The original 10,000 draws from f^{r1} served as the model building data set. There were initially twenty independent variables available for the model: the returns for instruments A, B, and E, and the claim payouts. We reduced this to eight by aggregating the five annual returns for each instrument into a joint measure of return for the whole planning horizon, in the following manner:

$$\text{aggregate return for instrument } i = \mathtt{agret}_i = \prod_{t=1}^{5} (1 + \mathtt{ret}_{i,t}) - 1.$$



Stratum	size of sample needed per stratified sampling plan $N_{SPE}(j)$	# of draws needed after pilot study max $[N_{SPE}(j) - n_j, 0]$	Difficulty ratio
1	693	668	267200
2	49	15	4412
3	177	38	2734
4	91	0	0
5	70	0	0
6	49	0	0
7	29	0	0
8	14	0	0
9	21	0	0
10	30	0	0
11	48	0	0
12	53	0	0
13	11	0	0

Stratum 1 is the critical stratum due to its difficulty ratio.

Table 3. Summary of stratified sampling plan

The model fitting process included six of these into the final model:

$$\hat{L} = -10.236 - 8.650c_1 - 7.595c_2 - 6.476c_3 - 5.857c_4 - 1.863c_5 - 1.016\mathtt{agret}_E.$$

Testing of the resulting predictor function yielded the following error rates: Prob (Type II error) = .007581, Prob (Type I error) $\approx$ 0. Clearly, the magnitude of the claims is the most significant factor in predicting membership in the critical stratum. To understand this, recall from Table 2 that the threshold for the critical stratum is util(NAP) = -85, and note that util(NAP) ≤ -85 occurs only when NAP $\leq\sim$ .56, i.e., when at least 56% of the initial asset position is lost during the planning period.

Table 2 shows that the only strata for which $M_j > 0$ are strata 1 ($M_1 = 668$), 2 ($M_2 = 15$), and 3 ($M_3 = 38$), with stratum 1 the critical stratum. We applied filtered search to "find" these final 668 + 15 + 38 = 721 draws. We obtained the required number of draws from strata 2 and 3 (plus 7 additional draws from the critical stratum) after 2768 simulation runs. The filtering process began at this point. An additional 213,880 scenarios were generated with 3324 actually evaluated to find the remaining 661 draws from the critical stratum. Compare this with the corresponding results of the naïve implementation in finding the remaining 721:

216,648 scenarios generated, with 216,648 scenarios evaluated. Filtered search required 98% fewer evaluations than naïve search. Also, the total number of scenarios generated by filtered search was 2768 + 213,880 = 216,648, the same number generated by the naïve implementation. The equality of these numbers shows that the logistic regression predictor function did not commit a single Type I error.

We close this discussion by noting the success of our efforts; the firm eventually adopted an asset allocation strategy very similar to r1.

6. Conclusions

We have motivated the need for rigorous evaluation of strategies generated by multi-stage stochastic optimization models, and presented a methodology called stratified filtered sampling that both fulfills this need and has value as a general tool for variance reduction. The methodology is based on stratification of the distribution of performance associated with the generated strategy. The principal obstacle to implementing such an approach is the inability to sample efficiently from a given region of this distribution. We presented a robust methodology called filtered search that overcomes this obstacle, insuring the computational tractability of the methodology.

We are currently investigating techniques to adapt algorithms for solving multi-stage stochastic optimization models to account for the "sampled" nature of the scenarios on which they are based. As we have discussed herein, many current algorithms treat the scenarios as the collectively exhaustive set of ways in which the relevant uncertainty might manifest itself. Of course, the scenarios are samples from the infinite set of possible manifestations. One particular solution strategy we find promising calls for the direction of search in the optimization process to be defined by statistical significance.

Acknowledgements: The authors wish to thank the anonymous reviewer for a review process that greatly improved the quality of this paper.

References

1. Bell, D. E., "Risk, Return, and Utility", Management Science, v41, 1995, 23-30.
2. Bennett, K. P. and Mangasarian, O. L., "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software, v1, 1992, 23-34.
3. Berger, A. J. and Mulvey, J. M., "Asset and Liability Management for Individual Investors", in World Wide Asset and Liability Modeling (W. T. Ziemba and J. M. Mulvey, eds.), Cambridge University Press, 1997.
4. Birge, J. R.; "Stochastic Programming Computation and Applications", INFORMS Journal on Computing, v9, 1997, 111 - 133.
5. Breiman, L., Friedman, J., Olshen, R., and Stone, C., Classification and Regression Trees, Wadsworth, Belmont (CA), 1984.

6. Carino, D. R., Kent, T., Myers, D. H., Stacy, C., Sylvanus, M., Turner, A., Watanabe, K., and Ziemba, W. T., "The Russell-Yasuda Kasai Model: An Asset Liability Model for a Japanese Insurance Company using Multi-Stage Stochastic Programming", Interfaces, v24, 1994, 29-49.
7. Carino, D. R. and Turner, A. L., "Multistage Planning for Asset al.location", in World Wide Asset and Liability Modeling (W. T. Ziemba and J. M. Mulvey, eds.), Cambridge University Press, 1997.
8. Cheng, R. C. H., "The Use of Antithetic Variates in Computer Simulations", Journal of the Operational Research Society, v33, 1982, 229-237.
9. Cochran, W. G., Sampling Techniques, John Wiley & Sons, New York, 1977.
10. Glynn, P. W. and Iglehart, D. L., "Importance Sampling for Stochastic Simulation", Management Science, v35, 1989, 1367-1392.
11. Hassoun, M. H., Fundamentals of Artificial Neural Networks, MIT Press, Cambridge, 1995.
12. Holmer, M. R., McKendall, R., Vassiadou-Zeniou, C., and Zenios, S., "Dynamic Models for Fixed Income Portfolio Management under Uncertainty", to appear in Journal of Economic Dynamics and Control.
13. Hosmer, D. W. and Lemeshow, S., Applied Logistic Regression, John Wiley & Sons, New York, 1989.
14. Joy, C., Boyle, P. P., and Tan, K. S., "Quasi-Monte Carlo Methods in Numerical Finance", Management Science, v42, 1996, 926-938.
15. Kall, P. and Wallace, S. W., Stochastic Programming, Wiley, New York (NY), 1994.
16. Lavenberg, S. S., Moeller, T. L., and Welch, P. D., "Statistical Results on Control Variates with Application to Queuing Network Simulation", Operations Research, v30, 1982, 182-202.
17. Lavenberg, S. S. and Welch, P. D., "Using Conditional Expectation to Reduce Variance in Discrete Event Simulation", Proc. 1979 Winter Simulation Conference, San Diego, 1979, 291-294.
18. McKay M. D., Beckman, R. J., and Conover, W. J., " A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from A Computer Code", Technometrics, v21, 1979, 239-245.
19. Morrison, D. F., Multivariate Statistical Methods, McGraw-Hill, New York, 1990.
20. Mulvey, J. M, "Generating Scenarios for the Towers Perrin Investment System", Interfaces, v26, 1996, 1-15.
21. Mulvey, J. M., "Multivariate Stratified Sampling by Optimization", Management Science, v29, 1983, 715-724.
22. Mulvey, J. M., Correnti, S., and Lummis, J, "Total Integrative Risk Management: Insurance Elements", Princeton University Report, SOR-97-2.
23. Neyman, J., "On the Two Different Aspects of the Representative Method: the Method of Stratified Sampling and the Method of Purposive Selection", Journal of the Royal Statistical Society, v97, 1934, 558-606.
24. Niederreiter, H., Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992.
25. Pickens, J. B., Hof, J. G., and Kent, B. M., "Use of Chance-Constrained Programming to Account for Stochastic Variation in the A-matrix of Large-Scale Linear Programs: A Forestry Example", in Stochastic Progamming, Part II (J. R. Birge and R. J.-B. Wets, eds.), Basel, Switzerland, 1991, 511-526.
26. Rush, R., "Decision-Constrained Stochastic Programming for Asset Liability Management", unpublished Ph.D. dissertation, Rensselaer Polytechnic Institute, May, 1998.
27. Sen, S., Doverspike, R. D., and Cosares, S., "Network Planning with Random Demand", Telecommunications Systems, v3, 1994, 11-30.
28. von-Neumann, J. and Morgenstern, O., Theory of Games and Economic Behavior, Princeton University Press, Princeton (NJ), 1947.
29. Wilkie, A. D., "Stochastic Investment Models: Theory and Application", Insurance: Mathematics and Economics, v6, 1987, 65 - 83.