Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

STOCHASTIC OPTIMIZATION USING MODEL REFERENCE ADAPTIVE SEARCH

Jiaqiao Hu
Michael C. Fu
Steven I. Marcus

Institute for Systems Research
University of Maryland
College Park, MD 20854, U.S.A.

ABSTRACT

We apply the recently introduced approach of model reference adaptive search to the setting of stochastic optimization and report on simulation results.

1 INTRODUCTION

We consider an optimization problem of the form:

$$x^* \in \arg\max_{x \in \mathcal{X}} E_\psi[H(x, \psi)], \quad x \in \mathcal{X} \subseteq \Re^n, \qquad (1)$$

where $\mathcal{X}$ is the solution space, which can be either continuous or discrete, $H(\cdot, \cdot)$ is a deterministic, real-valued function, and ψ is a random variable (possibly depending on x) representing the stochastic effects of the system. We let $h(x) := E_\psi[H(x, \psi)]$, and assume $h(x)$ cannot be obtained easily, but instead, only the random variable $H(x, \psi)$ can be observed, e.g., via simulation, which makes (1) much more difficult to solve than its deterministic counterpart (cf. Fu 2002). We assume (1) has a unique global optimal solution.

In general, depending on the structure of the underlying solution space, the techniques for stochastic optimization problems could be quite different. When the solution space is continuous, there is a well-known class of methods called stochastic approximation (SA) for solving such problems. These methods rely on the estimation of the gradient of the objective function with respect to the decision variables; thus they generally find local optimal solutions. In terms of the different gradient estimation techniques used, the SA methods can be further characterized into different categories. Detailed reviews can be found in e.g., Fu (1994, 2005).

For problems with discrete decision variables, various randomized search methods have been suggested, including the stochastic ruler method (Yan and Mukai 1992; Alrefaei and Andradòttir 2001), random search methods (Andradòttir 1995, 1996; Hong and Nelson 2005), simulated annealing (Alrefaei and Andradòttir 1999), and nested partitions (Shi and Òlafsson 2000). Andradòttir (2005) and Òlafsson (2005) contain recent reviews of these techniques.

This paper presents a new unified approach called Stochastic Model Reference Adaptive Search (SMARS) for solving simulation-based optimization problems with either continuous or discrete solution spaces. SMARS is the generalization of the Model Reference Adaptive Search (MRAS) method for deterministic optimization introduced in Hu, Fu, and Marcus (2005a) with some appropriated modifications and extensions required for the stochastic setting. The method works with a parameterized probability distribution on the solution space and generates at each iteration a group of candidate solutions. These candidate solutions are then used to update the parameters associated with the distribution so that the future search will be biased toward the region containing high quality solutions. For complete technical developments of the approach and convergence proofs, the interested readers are referred to Hu, Fu, and Marcus (2005b).

The rest of the paper is structured as follows. In Section 2, we describe the proposed SMRAS method. In Section 3, we present the global convergence property of SMRAS. Illustrative numerical examples on both continuous and discrete domains are given in Section 4. Finally Section 5 contains some concluding remarks.

2 ALGORITHM

The algorithm is summarized in Figure 1. For a specified parameterized family of distributions $\{f(\cdot, \theta), \theta \in \Theta\}$, where Θ is the parameter space, the main body of the algorithm consists of the following steps:

(1) generate candidate solutions according the current distribution, say $f(\cdot, \theta_k)$;

Stochastic Model Reference Adaptive Search (SMRAS)

- **Initialization:** Specify $\rho_0 \in (0, 1]$, initial sample size $N_0 > 1$, $\alpha > 1$, $\varepsilon > 0$, a simulation allocation rule $\{M_k\}$, a function $S(\cdot) : \Re \to \Re^+$, a sequence of mixing coefficients $\{\lambda_k . k = 0, 1, \ldots\}$ satisfying $\lambda_k \geq \lambda_{k+1}$ and $\lambda_k \in (0, 1)$ $\forall k$, and an initial p.d.f. $f(x, \theta_0) > 0$ $\forall x \in \mathcal{X}$. Set $k \leftarrow 0$.
- **Repeat until a specified stopping rule is satisfied:**

1. Generate N_k samples $X_1^k, \ldots, X_{N_k}^k$ according to $\widetilde{f}(\cdot, \theta_k) := (1 - \lambda_k) f(\cdot, \theta_k) + \lambda_k f(\cdot, \theta_0)$.
2. Compute the sample $(1 - \rho_k)$-quantile $\widetilde{\gamma}_{k+1}(\rho_k, N_k) := \bar{H}_{k,(\lceil (1-\rho_k)N_k \rceil)}$, where $\lceil a \rceil$ is the smallest integer greater than a, and $\bar{H}_{k,(i)}$ is the ith order statistic of the sequence $\left\{ \bar{H}_k(X_i^k),\ i = 1, \ldots, N_k \right\}$.
3. **If** $k = 0$ **or** $\widetilde{\gamma}_{k+1}(\rho_k, N_k) \geq \bar{\gamma}_k + \varepsilon$, **then** do step 3a.

   3a. Set $\bar{\gamma}_{k+1} \leftarrow \widetilde{\gamma}_{k+1}(\rho_k, N_k)$, $\rho_{k+1} \leftarrow \rho_k$, $N_{k+1} \leftarrow N_k$, $X_{k+1}^\dagger \leftarrow X_{1-\rho_k}$, where $X_{1-\rho_k}$ represents the sample that achieves the sample $(1 - \rho_k)$-quantile of the sequence $\left\{ \bar{H}_k(X_i^k) \right\}$.

   else, find the largest $\bar{\rho} \in (0, \rho_k)$ such that $\widetilde{\gamma}_{k+1}(\bar{\rho}, N_k) \geq \bar{\gamma}_k + \varepsilon$.

   3b. **If** such a $\bar{\rho}$ exists, **then** Set $\bar{\gamma}_{k+1} \leftarrow \widetilde{\gamma}_{k+1}(\bar{\rho}, N_k)$, $\rho_{k+1} \leftarrow \bar{\rho}$, $N_{k+1} \leftarrow N_k$, $X_{k+1}^\dagger \leftarrow X_{1-\bar{\rho}}$.

   3c. **else** if no such $\bar{\rho}$ exists, set $\bar{\gamma}_{k+1} \leftarrow \bar{H}_k(X_k^\dagger)$, $\rho_{k+1} \leftarrow \rho_k$, $N_{k+1} \leftarrow \lceil \alpha N_k \rceil$, $X_{k+1}^\dagger \leftarrow X_k^\dagger$.

   endif
4. Compute θ_{k+1} as

$$\theta_{k+1} = \arg\max_{\theta \in \Theta} \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{[S(\bar{H}_k(X_i^k))]^k}{\widetilde{f}(X_i^k, \theta_k)} \widetilde{I}[\bar{H}_k(X_i^k), \bar{\gamma}_{k+1}] \ln f(X_i^k, \theta), \tag{2}$$

$$\text{where } \widetilde{I}(x, \gamma) := \begin{cases} 1 & \text{if } x \geq \gamma, \\ (x - \gamma + \varepsilon)/\varepsilon & \text{if } \gamma - \varepsilon < x < \gamma, \\ 0 & \text{if } x \leq \gamma - \varepsilon. \end{cases}$$

5. Set $k \leftarrow k + 1$.

Figure 1: Stochastic Model Reference Adaptive Search (SMRAS)

(2) compute a new parameter θ_{k+1} according to a specified rule by using the samples generated in the previous step in order to concentrate the future search toward more promising regions.

There are two simulation allocation rules in SMRAS: the sampling allocation rule $\{N_k, k = 0, 1 \ldots\}$, which determines the number of candidate solutions to be generated from the current sampling distribution $f(\cdot, \theta_k)$, and the observation allocation rule $\{M_k, k = 0, 1, \ldots\}$, which allocates M_k simulation observations to each of the candidate solutions generated at the kth iteration. We require both N_k and M_k to increase as the number of iteration grows for convergence. We use a parameter $\alpha > 1$ to control the rate of increase in $\{N_k\}$, and leave the sequence $\{M_k\}$ as user-specified. When M_k observations are allocated to a solution x at iteration k, we use $H_j(x)$ to denote the jth random observation of $H(x, \psi)$, and use $\bar{H}_k(x) = \frac{1}{M_k} \sum_{j=1}^{M_k} H_j(x)$ to denote the sample average of all M_k observations made at x.

The performance of the SMRAS algorithm depends on another sequence of quantities $\{\rho_k, k = 0, 1 \ldots\}$. The motivation behind the sequence is to concentrate the computational effort on the set of elite samples rather than the entire set of samples. At successive iterations of the algorithm, a sequence of thresholds $\{\bar{\gamma}_k, k = 1, 2, \ldots\}$ are generated according to the sequence of sample $(1 - \rho_k)$-quantiles, and only those samples that have performances better than these thresholds will be used in parameter updating. Thus, each ρ_k determines the proportion of N_k samples that will be used to update the probabilistic model at iteration k.

During the initialization step of SMRAS, a continuous and strictly increasing function $S(\cdot) : \Re \to \Re^+$ is specified. The function $S(\cdot)$ is used to account for the cases where the sample average approximations $\bar{H}_k(x)$ are negative for some x.

At each iteration k, random samples are drawn from the density/mass function $\widetilde{f}(\cdot, \theta_k)$, which is a mixture of the initial density $f(\cdot, \theta_0)$ and the density calculated from the previous iteration $f(\cdot, \theta_k)$. Intuitively, mixing in the initial density enables the algorithm to explore the entire solution space and thus maintain a global perspective during the search process.

At step 2, the sample $(1 - \rho_k)$-quantile $\widetilde{\gamma}_{k+1}$ with respect to $\widetilde{f}(\cdot, \theta_k)$ is calculated by first ordering the sample performances $\bar{H}_k(X_i^k)$, $i = 1, \ldots, N_k$ from smallest to largest, $\bar{H}_{k,(1)} \leq \bar{H}_{k,(2)} \leq \cdots \leq \bar{H}_{k,(N_k)}$, and then taking the $\lceil (1 - \rho_k)N_k \rceil$th order statistic. We use the function $\widetilde{\gamma}_{k+1}(\rho_k, N_k)$ to emphasize the dependencies of $\widetilde{\gamma}_{k+1}$ on both ρ_k and N_k, so that different sample quantile values can be distinguished by their arguments.

Step 3 of the algorithm is used to construct a sequence of thresholds $\{\bar{\gamma}_k, k = 1, 2, \ldots\}$ from the sequence of sample quantiles $\{\widetilde{\gamma}_k\}$, and to determine the appropri-

ate values of the ρ_{k+1} and N_{k+1} to be used in subsequent iterations. This is carried out by checking whether the condition $\widetilde{\gamma}_{k+1}(\rho_k, N_k) \geq \bar{\gamma}_k + \varepsilon$ is satisfied. If the inequality holds, then both the current ρ_k value and the new sample size N_k are satisfactory, and $\widetilde{\gamma}_{k+1}(\rho_k, N_k)$ is used as the current threshold value. Otherwise, we fix the sample size N_k and try to find if there exists a smaller $\bar{\rho} < \rho_k$ such that the above inequality can be satisfied with the new sample $(1-\bar{\rho})$-quantile. If such a $\bar{\rho}$ does exist, then the current sample size N_k is still deemed acceptable, and the new threshold value is updated by the sample $(1-\bar{\rho})$-quantile. On the other hand, if no such $\bar{\rho}$ can be found, then the sample size N_k is increased by a factor α, and the new threshold $\bar{\gamma}_{k+1}$ is calculated by using an additional variable $X_k^\dagger$ to remember the particular sample that achieves the previous threshold value $\bar{\gamma}_k$, and then simply allocating M_k observations to $X_k^\dagger$. It is important to note that in step 4, the set $\left\{x : \bar{H}_k(x) > \bar{\gamma}_k - \varepsilon, x \in \left\{X_1^k, \ldots, X_{N_k}^k\right\}\right\}$ could be empty, since it could happen that all the random samples generated at the current iteration are much worse than those generated at the previous iteration. If this is the case, the right hand side of equation (2) will be equal to zero, so any $\theta \in \Theta$ is a maximizer; we define $\theta_{k+1} := \theta_k$ in this case. Also note that a "soft" threshold function $\widetilde{I}(\cdot,\cdot)$, as opposed to the indicator function, is used in parameter updating.

In practice, different stopping rules can be used. The simplest method is to stop the algorithm when the total computational budget is exhausted or when the prescribed maximum number of iterations is reached; see Hu, Fu, and Marcus (2005b) for further details.

3 THEORETICAL CONVERGENCE

Hu, Fu, and Marcus (2005b) establishes convergence properties of the SMRAS algorithm for a particular family of distributions called the natural exponential family (NEF), which includes many common distributions, including the normal, Poisson, binomial, geometric, and certain multivariate forms of them.

Under appropriate assumptions on the objective function, the input parameters, the random variable $H_j(x)$, and the observation allocation rule $\{M_k, k = 0, 1, \ldots\}$, the following result is established in Fu, Hu, and Marcus (2005b).

Theorem 1 *Let φ be a positive constant satisfying the condition that the set $\left\{x : S(h(x)) \geq \frac{1}{\varphi}\right\}$ has a strictly positive Lebesgue measure. If there exist $\delta \in (0,1)$ and $\mathcal{T}_\delta < \infty$ such that $\alpha \geq [\varphi S^*]^2/[\lambda_k^{2/k}\delta]$ $\forall\, k \geq \mathcal{T}_\delta$, then*

$$\lim_{k\to\infty} E_{\theta_k}[\Gamma(X)] = \Gamma(x^*) \quad w.p.1, \tag{3}$$

where $E_{\theta_k}[\cdot]$ represents the expectation taken w.r.t. the distribution $f(\cdot, \theta_k)$.

In particular, for the multivariate normal case with mean vector μ_k and variance-covariance matrix Σ_k,

$$\lim_{k\to\infty} \mu_k = x^*, \quad \lim_{k\to\infty} \Sigma_k = 0_{n\times n} \quad \text{w.p.1},$$

where $0_{n\times n}$ represents a n-by-n zero matrix.

4 SIMULATION RESULTS

In this section, we test the performance of SMARS on both continuous and combinatorial stochastic optimization problems. In the former case, we first illustrate the global convergence of SMRAS by testing the algorithm on two multi-extremal functions; then we apply the algorithm to an inventory control problem. In the latter case, we consider the problem of optimizing the buffer allocations in a tandem queue with unreliable servers, which has been previously studied in e.g., Vouros and Papadopoulos (1998), and Allon et al. (2005).

In actual implementation of the algorithm, a smoothed parameter updating procedure (cf. De Boer et al. 2004) is used, i.e., first a smoothed parameter vector $\widehat{\theta}_{k+1}$ is computed at iteration k according to

$$\widehat{\theta}_{k+1} := \upsilon\, \theta_{k+1} + (1-\upsilon)\widehat{\theta}_k, \quad \forall\, k = 0, 1, \ldots, \text{ and } \widehat{\theta}_0 := \theta_0,$$

where θ_{k+1} is the parameter vector derived at step 3 of SMRAS, and $\upsilon \in (0, 1]$ is the smoothing parameter. Then $f(x, \widehat{\theta}_{k+1})$ instead of $f(x, \theta_{k+1})$ is used in step 1 to generate new samples. This modification does not affect the theoretical convergence.

4.1 Continuous Optimization

For continuous problems, we use multivariate normal p.d.f's as the parameterized probabilistic model.

4.1.1 Global Convergence

We consider the following two multi-extremal test functions

(1) Goldstein-Price function with additive noise

$$\begin{aligned} H_1(x,\psi) =& (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 \\ &+ 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2(18 - \\ &32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) + \psi, \end{aligned}$$

where $x = (x_1, x_2)^T$, and $\psi \sim N(0, 10^2)$. The function $h_1(x) = E_\psi[H_1(x,\psi)]$ has four local minima and a global minimum $h_1(0,-1) = 3$.

(2) Rosenbrock function with additive noise

$$H_2(x,\psi) = \sum_{i=1}^{4} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 + 1 + \psi,$$

where $x = (x_1, \ldots, x_5)^T$, and $\psi \sim N(0, 10^2)$. The function $h_2(x) = E_\psi[H_2(x, \psi)]$ has a global minimum $h_2(1, 1, 1, 1, 1) = 1$ and many local minima.

For both problems, the same set of parameters is used: $\beta = 1.02$, $\varepsilon = 0.1$, mixing coefficient $\lambda_k = \frac{1}{\sqrt{k+1}}\, \forall\, k$, initial sample size $N_0 = 100$, $\rho_0 = 0.9$, $\alpha = 1.03$, observation allocation rule $M_k = 1.1^k$, smoothing parameter $\upsilon = 0.2$. The initial mean vector is taken to be a vector of all 10s (which is very far from the optimal solution), and the initial variance-covariance matrix is a diagonal matrix with all diagonal elements equal to 100.

For each function, we performed 50 independent simulation runs of SMRAS. The averaged performance of the algorithm is shown in Table 1, where N_{avg} is the average total number of function evaluations needed to satisfy the stopping criteria, H_* and H^* are the worst and best function values obtained in 50 trials, and $\bar{H}$ is the averaged function values over the 50 replications. Figure 2 plots the progression of the average function values for (a) function H_1 through 45 iterations, and (b) function H_2 through 100 iterations.

Table 1: Performance of SMRAS on Two Test Functions (Standard Errors in Parentheses)

H_i	$N_{avg}(std\ err)$	H_*	H^*	$\bar{H}(std\ err)$
H_1	5.4e+04(3.9e+02)	3.05	3.00	3.01(1.64e-3)
H_2	1.0e+07(4.9e+05)	1.31	1.02	1.09(9.10e-3)

4.1.2 An Inventory Control Example

To further illustrate the algorithm, we consider an (s, S) inventory control problem with i.i.d exponentially distributed continuous demands, zero order lead times, full backlogging of orders, and linear ordering, holding and shortage costs. The inventory level is periodically reviewed, and an order is placed when the inventory position (on hand plus that on order) falls below the level s, and the amount of the order is the difference between S and the current inventory position. Let D_t denote the demand in period t, X_t the inventory position in period t, p the per period per unit demand lost penalty cost, h the per period per unit inventory holding cost, c the per unit ordering cost, and K the set-up cost per order. The inventory position $\{X_t\}$ evolves according to the following dynamics:

$$X_{t+1} = \begin{cases} S - D_{t+1} & X_t < s, \\ X_t - D_{t+1} & X_t \geq s. \end{cases}$$

The goal is to choose values of s and S to minimize the steady-state average cost per period:

$$(s^*, S^*) = \arg\min J(s, S) := \arg\min \lim_{t\to\infty} J_t(s, S),$$

where $J_t(s, S) := \frac{1}{t}\sum_{i=1}^{t}\left[I\{X_i < s\}(K + c(S - X_i)) + hX_i^+ + pX_i^-\right]$, $I\{\cdot\}$ is the indicator function, $x^+ = \max(0, x)$, and $x^- = \max(0, -x)$. Although the cost function is convex, this property is not exploited.

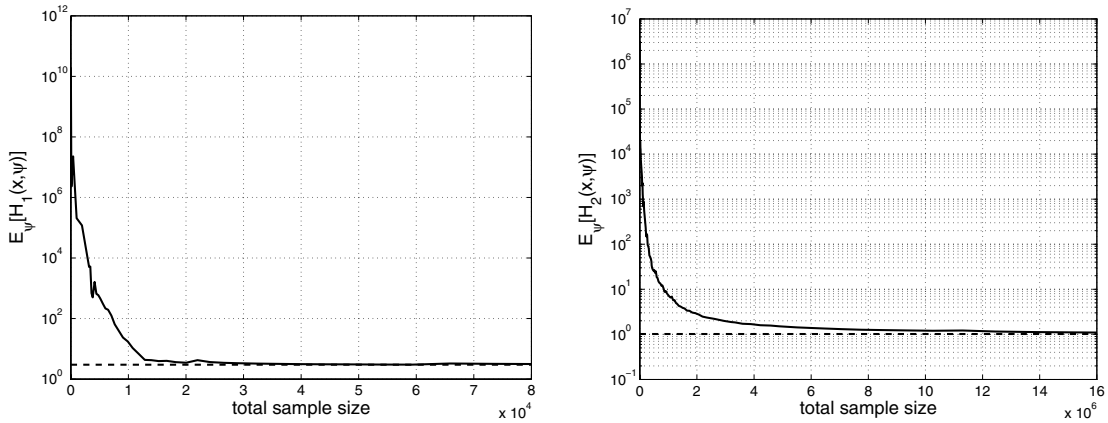
Eight cases taken from Fu and Healy (1997) were used to test the performance of the SMRAS algorithm. Shown in Table 2 are the cost coefficients and the optimal solutions, with $c = h = 1$ and exponentially distributed demands with mean $E[D]$.

Table 2: (s, S) Inventory Control Example Test Cases

Case	$E[D]$	p	K	J^*	(s^*, S^*)
1	200	10	100	740.9	(341,541)
2	200	10	10000	2200.0	(0,2000)
3	200	100	100	1184.4	(784,984)
4	200	100	10000	2643.4	(443,2443)
5	5000	10	100	17078	(11078,12078)
6	5000	10	10000	21496	(6496,16496)
7	5000	100	100	28164	(22164,23164)
8	5000	100	10000	32583	(17582,27582)

In our simulation experiments, the initial mean vector is taken to be $(2000, 4000)^T$ for all eight cases, and the covariance matrices are initialized as diagonal matrices with all diagonal elements equal to 10^5 for cases $1-4$ and 10^6 for cases $5-8$. The other parameters are: $\beta = 1.05$, $\varepsilon = 0.1$, $\lambda_k = \frac{1}{\sqrt{k+1}}\ \forall\ k$, $N_0 = 100$, $\rho_0 = 0.95$, $\alpha = 1.05$, $M_k = 1.2^k$, smoothing parameter $\upsilon = 0.3$. The steady-state cost per period is simulated by averaging the accumulated cost over 50 periods after a warm-up period with length 50.

Figure 3 shows the typical performance of SMRAS for the first four test cases when the total number of simulation periods is set to 10^6. The locations of the optimal solutions are marked by $\star$. We see that the algorithm converges rapidly to the neighborhood of the optimal solution in the first few iterations and then spends most of the computational effort in that small region. Numerical results for all eight test cases are given in Table 3. In the table, N_p indicates the total number of periods (including the warm-up periods) simulated, and the entries represent the averaged function values J of the final sample solutions obtained for different choices of N_p, each one based on 25 independent simulation replications.



(a) (b)

Figure 2: Average Performance of SMRAS on (a) Goldstein-price Function; (b) 5-D Rosenbrock Function

Table 3: Performance of SMRAS on (s, S) Inventory Example (Each Case Based on 25 Independent Simulation Runs; Standard Errors in Parentheses)

$Case$	$N_p = 10^5$	$N_p = 10^6$	$N_p = 5 \times 10^6$	$N_p = 10^7$	J^*
1	1169.7(43.5)	742.6(0.32)	741.6(0.14)	741.2(0.06)	740.9
2	2371.6(37.8)	2223.9(3.57)	2202.0(0.20)	2200.8(0.17)	2200.0
3	1413.1(28.0)	1213.8(5.90)	1188.8(0.78)	1185.8(0.28)	1184.4
4	2709.0(13.4)	2667.2(4.89)	2647.2(0.61)	2645.0(0.42)	2643.4
5	18694.6(195.5)	17390.4(48.5)	17245.5(32.81)	17119.3(9.25)	17078
6	24001.7(340.8)	21808.5(53.6)	21780.0(34.00)	21520.9(5.80)	21496
7	32909.1(579.5)	28778.5(82.2)	28598.8(50.25)	28290.1(33.45)	28164
8	36520.0(538.0)	32881.7(216.9)	32860.2(52.56)	32682.8(36.68)	32583

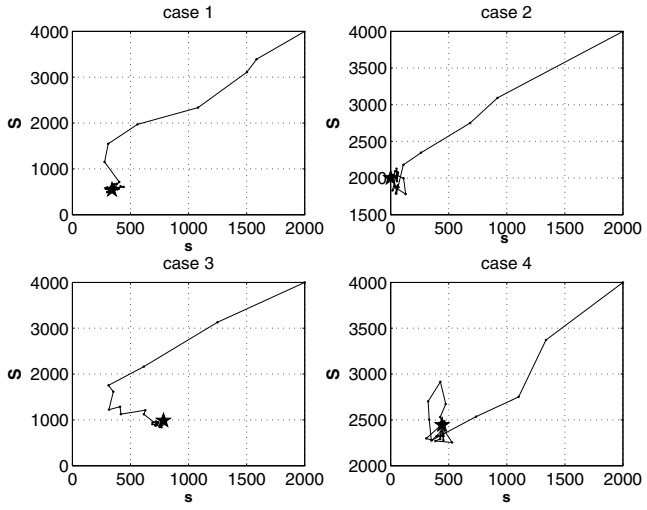


Figure 3: Typical Performance of SMRAS on the First Four Test Cases ($N_p = 10^6$)

4.2 Combinatorial Optimization

To illustrate the performance of SMRAS on discrete stochastic optimization problems, we consider the buffer allocation problem (BAP) in a service facility with unreliable servers. The system consists of m servers in series, which are separated by $m-1$ buffer locations. Each job enters the system from the first server, goes through all intermediate servers and buffer locations in a sequential order, and finally exits from the last server. The service times at each server are independent exponentially distributed with service rate $\mu_i,\ i = 1, \ldots, m$. The servers are assumed to be unreliable, and are subject to random failures. When a server fails, it has to be repaired. The time to failure and the time for repair are both i.i.d. exponentially distributed with respective rates f_i and $r_i,\ i = 1, \ldots, m$. A server is blocked when the buffer associated with the server coming next to it is full and is starved when no jobs are offered to it. Thus, the status of a server (busy/broken) will affect the status of all other servers in the system. We assume that the failure rate of each server remains the same, regardless of its current status. Given n limited buffer spaces, our goal is to find an optimal way of allocating these n spaces to the $m-1$ buffer locations such that the throughput (average production rate) is maximized.

When applying SMRAS, we have used the same technique as in Allon et al. (2005) to generate admissible buffer allocations, and the basic idea is to choose the probabilistic model as an $(n+1)$-by-$(m-1)$ matrix P, whose (i, j)th entry specifies the probability of allocating $i-1$ buffer spaces to the jth buffer location.

For the numerical experiments, we consider two cases: (i) $m = 3$, $n = 1, 2, \ldots, 10$, $\mu_1 = 1$, $\mu_2 = 1.2$ $\mu_3 = 1.4$, failure rates $f_i = 0.05$ and repair rates $r_i = 0.5$ for all $i = 1, 2, 3$; (ii) $m = 5$, $n = 1, \ldots, 10$, $\mu_1 = 1$, $\mu_2 = 1.1$, $\mu_3 = 1.2$, $\mu_4 = 1.3$, $\mu_5 = 1.5$, $f_i = 0.05$ and $r_i = 0.5$ for all $i = 1, \ldots, 5$.

Apart from their combinatorial nature, an additional difficulty in solving these problems is that different buffer allocation schemes (samples) have similar performances. Thus, when only noisy observations are available, it could be very difficult to discern the best allocation from a set of candidate allocation schemes. Because of this, in SMRAS we choose the performance function $S(\cdot)$ as an exponential function with a relatively larger base $\beta = 10$. The other parameters are as follows: $\varepsilon = 0.001$, $\lambda_k = 0.01$ $\forall\, k$, initial sample size $N_0 = 10$ for case (i) and $N_0 = 20$ for case (ii), $\rho = 0.9$, $\alpha = 1.2$, observation allocation rule $M_k = (1.5)^k$, smoothing parameter $\upsilon = 0.7$, and the initial P^0 is taken to be a uniform matrix with each column sum equal to one, i.e., $P^0_{i,j} = \frac{1}{n+1}$ $\forall\, i,\ j$. We start all simulation replications when the system is empty. The steady-state throughputs are simulated after 100 warm-up periods, and then averaged over the subsequent 900 periods.

Tables 4 and 5 give the performances of SMRAS for each of the respective cases (i) and (ii), where $\overline{P}_{MRAS}$ is the averaged total number of periods simulated over 16 independent trials, $Alloc$ is the optimal allocation scheme, $\overline{N}_{MRAS}$ is the averaged total number of allocations generated, N_{A^*} is the number of times the best allocation was found out of 16 runs, $\overline{\mathcal{T}}_{MRAS}$ is the averaged throughput value calculated by the algorithm, and $\mathcal{T}^*$ represents the exact optimal solution. In each table, we also included for comparison the results of Allon et al. (2004), which are based on 10 independent runs of the CE method, where $\overline{N}_{CE}$ is the averaged total number of allocations used by CE and $\overline{\mathcal{T}}_{CE}$ is the averaged throughput value computed by CE.

5 CONCLUSIONS

We have presented a unified approach for solving stochastic optimization problems for both continuous and discrete solution spaces. The approach is generic – requiring only a few mild regularity conditions on the underlying problem – and is provably convergent to the global optimal solution. Preliminary simulation studies comparing its performance with the cross-entropy method indicate the approach has great promise.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant DMI-0323220, and by the Air Force of Scientific Research under Grant FA95500410210.

REFERENCES

Allon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein. 2005. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research* 134(1): 137-152.

Alrefaei, M. H. and S. Andradóttir. 1995. A modification of the stochastic ruler method for discrete stochastic optimization. *European Journal of Operational Research* 133: 160-182.

Alrefaei, M. H. and S. Andradóttir. 1999. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science* 45(5): 748-764.

Andradóttir, S. 1995. A method for discrete stochastic optimization. *Management Science* 41: 1946-1961.

Andradóttir, S. 2005. An overview of simulation optimization via random search. Chapter 21 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

De Boer, P. T., D. P. Kroese, S. Mannor, and R. Y. Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of Operations Research* 134(1): 19-67.

Fu, M. C. and K. J. Healy. 1997. Techniques for simulation optimization: An experimental study on an (s, S) inventory system. *IIE Transactions* 29(3): 191-199.

Fu, M.C. 1994. Optimization via simulation: A review. *Annals of Operations Research* 53: 199-248.

Fu, M.C. 2002. Optimization for simulation: Theory vs. Practice (Feature Article). *INFORMS Journal on Computing* 14 (3): 192-215.

Fu, M.C. 2005. Stochastic Gradient Estimation. Chapter 19 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier, 2005.

Hong, L. J. and B.L. Nelson. 2005. Discrete Optimization via Simulation using COMPASS, *Operations Research*. to appear.

Hu, J., M. C. Fu, and S. I. Marcus. 2005a. A Model Reference Adaptive Search Algorithm for Global Optimization. submitted for publication. also available at `http://techreports.isr.umd.edu/ARCHIVE/dsp_reportList.php?year=2005&center=ISR`.

Hu, J., M. C. Fu, and S. I. Marcus. 2005b. A Model Reference Adaptive Search Algorithm for Stochastic Global Optimization, submitted for publication. also available at `http://techreports.isr.umd.edu/ARCHIVE/dsp_reportList.php?year=2005&center=ISR`.

Ólaffson, S. 2005. Metaheuristics. Chapter 22 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

Table 4: BAP case (i): Performance of SMRAS vs. CE

n	$\overline{P}_{MRAS}(std\ err)$	$Alloc\ (N_{A^*})$	$\overline{N}_{MRAS}$	$\overline{T}_{MRAS}(std\ err)$	$\overline{N}_{CE}$	$\overline{T}_{CE}$	$\mathcal{T}^*$
1	3.31e+4(4.87e+2)	[1,0] (16)	33	0.634(4.06e-4)	12	0.634	0.634
2	4.68e+4(3.15e+3)	[1,1] (16)	47	0.674(6.35e-4)	24	0.672	0.674
3	4.39e+4(1.51e+3)	[2,1] (16)	44	0.711(6.11e-4)	47	0.700	0.711
4	4.98e+4(3.45e+3)	[3,1] (14)	50	0.735(6.47e-4)	84	0.735	0.736
5	5.04e+4(3.68e+3)	[3,2] (13)	50	0.758(1.06e-3)	114	0.757	0.759
6	6.40e+4(6.29e+3)	[4,2] (12)	64	0.776(1.39e-3)	155	0.769	0.778
7	5.91e+4(4.27e+3)	[5,2] (14)	59	0.792(1.04e-3)	260	0.781	0.792
8	6.39e+4(4.79e+3)	[5,3] (10)	64	0.805(1.20e-3)	245	0.804	0.806
9	6.06e+4(3.46e+3)	[6,3] (10)	61	0.817(6.53e-4)	491	0.814	0.818
10	6.37e+4(5.69e+3)	[7,3] (12)	64	0.826(9.88e-4)	498	0.826	0.827

Table 5: BAP case (ii): Performance of SMRAS vs. CE

n	$\overline{P}_{MRAS}(std\ err)$	$Alloc\ (N_{A^*})$	$\overline{N}_{MRAS}$	$\overline{T}(std\ err)$	$\overline{N}_{CE}$	$\overline{T}_{CE}$	$\mathcal{T}^*$
1	1.02e+5(7.49e+3)	[0,1,0,0] (16)	102	0.523(6.79e-4)	26	0.521	0.521
2	1.29e+5(1.48e+4)	[1,1,0,0] (16)	129	0.555(3.86e-4)	92	0.548	0.551
3	1.75e+5(1.57e+4)	[1,1,1,0] (16)	175	0.587(4.57e-4)	108	0.582	0.582
4	2.51e+5(2.59e+4)	[1,2,1,0] (11)	251	0.606(1.20e-3)	256	0.602	0.603
5	3.37e+5(4.20e+4)	[2,2,1,0] (10)	337	0.626(6.57e-4)	450	0.620	0.621
6	4.69e+5(5.52e+4)	[2,2,1,1] (8)	469	0.644(1.10e-3)	342	0.642	0.642
7	4.56e+5(5.82e+4)	[2,2,2,1] (7)	456	0.659(1.10e-3)	539	0.657	0.659
8	4.45e+5(5.49e+4)	[3,2,2,1] (7)	445	0.674(1.10e-3)	576	0.673	0.674
9	5.91e+5(5.61e+4)	[3,3,2,1] (6)	591	0.689(1.39e-3)	828	0.689	0.689
10	5.29e+5(5.40e+4)	[3,3,3,1] (8)	529	0.701(1.10e-3)	1070	0.700	0.701

Shi, L. and S. Ólafsson. 2000. Nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability* 2(3): 271-291.

Vouros, G. A. and H. T. Papadopoulos. 1998. Buffer allocation in unreliable production lines using a knowledge based system. *Computer & Operations Research* 25(12): 1055-1067.

Yan, D. and H. Mukai. 1992. Stochastic discrete optimization. *SIAM Journal on Control and Optimization* 30: 594-612.

AUTHOR BIOGRAPHIES

JIAQIAO HU is a Ph.D. candidate in the Department of Electrical and Computer Engineering and the Institute for Systems Research at the University of Maryland. He received a B.S. in Automation from Shanghai Jiao Tong University and an M.S. in Applied Mathematics from the University of Maryland, Baltimore County. His research interests include Markov decision processes, applied probability, and simulation optimization. He received third place in the 2005 Genetic and Evolutionary Computation Conference (GECCO) physical traveling salesman problem competition. His e-mail address is <jqhu@glue.umd.edu>.

MICHAEL C. FU is a Professor in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research and an affiliate appointment in the Department of Electrical and Computer Engineering, all at the University of Maryland. He received degrees in mathematics and EE/CS from MIT, and an M.S. and Ph.D. in applied mathematics from Harvard University. His research interests include simulation and applied probability modeling, particularly with applications towards manufacturing systems, inventory control, and financial engineering. He currently serves as Simulation Area Editor of *Operations Research*. He is co-author (with J.Q. Hu) of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, which received the INFORMS College on Simulation Outstanding Publication Award in 1998. His e-mail address is <mfu@rhsmith.umd.edu>.

STEVEN I. MARCUS is a Professor in the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland. He received the B.A. degree in electrical engineering and mathematics from Rice University in 1971, and the S.M. and Ph.D. degrees in electrical engineering from MIT in 1972 and 1975, respectively. From 1975 to 1991, he was with the Department of Electrical and Computer Engineering, University of Texas, Austin, where he was the L.B. (Preach) Meaders Professor in Engineering. At the University of Maryland, he has served as Director of the Institute for Systems Research from 1991-1996, and as Chair of the Electrical and Com-

puter Engineering Department from 2000-2005. Currently, his research is focused on stochastic control and estimation, with applications in manufacturing and telecommunication networks. He is the Editor-in-Chief of the *SIAM Journal on Control and Optimization* and a Fellow of IEEE. His e-mail address is <marcus@eng.umd.edu>.