Lecture Notes 16
Making Decisions Via Simulation

Prof. Marvin Nakayama

Outline

- Introduction
- Continuous Stochastic Optimization
- Small Number of Alternatives: Ranking and Selection
- Large Number of Alternatives: Discrete Stochastic Optimization

Reading Assignment: Chapter 10 of Law.

Introduction

- **Problem:** Given a number of alternative systems, identify the "best" one.
- **Example:** Given 10 alternative designs for a manufacturing system, which has the largest throughput?
- Performance of each alternative is estimated via simulation.
- Methods presented are chosen because of simplicity
- See Law textbook for
  - more details on exactly when they are applicable
  - whether improved versions of the algorithms are available.
- Current understanding of behavior of these algorithms incomplete
  - Apply with due caution.

Introduction

- Let θ be a parameter value denoting one alternative to consider.
- Let Θ denote the set of all possible parameters to compare.
- Methods categorized mainly based on the size of set Θ.
  1. Θ is an (uncountably infinite) interval of values
     - e.g., θ is an arrival rate
  2. Θ has a small number (≤ 20) of values
     - e.g., θ is the size of a single buffer
  3. Θ has a large (or countably infinite) number of values
     - e.g., θ is vector denoting the sizes of 10 buffers

Continuous Stochastic Optimization

- **Problem:** Given a set Θ of possible parameter values, find $\theta \in \Theta$ to

$$\min_{\theta \in \Theta} \mu(\theta)$$

- Let θ^* denote the (unknown) global minimizer of μ.
- For a given value of θ, we are not able to evaluate $\mu(\theta)$ analytically or numerically.
- Must obtain (noisy) estimates of $\mu(\theta)$ using simulation.
- First assume the set Θ of possible solutions is an (uncountably infinite) interval of real values:

$$\Theta = \left[\underline{\theta}, \overline{\theta}\right]$$

Robbins-Munro Algorithm

- One approach to $\min_{\theta \in \Theta} \mu(\theta)$
  1. Estimate the derivative $\mu'(\theta)$ using simulation.
  2. Use an iterative method to solve the equation $\mu'(\theta) = 0$.
- The most basic such method is the **Robbins-Monro Algorithm**.
  - Can be viewed as a stochastic version of Newton's method for finding a root of a nonlinear function.
  - Iterative algorithm identifies a sequence of parameter values $\theta_0, \theta_1, \theta_2, \ldots$ such that $\theta_n \to \theta^*$ as $n \to \infty$.

Robbins-Munro Algorithm

- After n iterations, the $(n+1)$th iteration selects θ_{n+1} as

$$\theta_{n+1} \;=\; \pi\left(\theta_n - \left(\frac{a}{n}\right) Z_n\right)$$

where

- $a > 0$ is a fixed parameter of the algorithm (the quantity a/n is sometimes called the "gain")
- Z_n is an unbiased estimator of derivative $\mu'(\theta_n)$
- Function $\pi(\,\cdot\,)$ projects the current parameter value onto the feasible set $\Theta = [\underline{\theta}, \overline{\theta}]$ via

$$\pi(\theta) = \begin{cases} \underline{\theta} & \text{if } \theta < \underline{\theta}; \\ \theta & \text{if } \underline{\theta} \le \theta \le \overline{\theta}; \\ \overline{\theta} & \text{if } \theta > \overline{\theta}. \end{cases}$$

Robbins-Munro Algorithm

- Algorithms that iterate as described above are called **stochastic approximation algorithms**.
- If the only local minimizer of μ is the global minimizer θ^*, then under very general conditions

$$\theta_n \to \theta^*$$

as $n \to \infty$ with probability 1.

- Otherwise, the algorithm can converge to some other local minimizer.

Robbins-Munro Algorithm

- For large n, can show θ_n has approximately normal distribution.
- Suppose procedure is run for
  - m independent replications (where $m = 5$ to 10)
  - each replication has n iterations
- Resulting m IID replicates are $\theta_{n,1}, \theta_{n,2}, \ldots, \theta_{n,m}$.
- Then
  - a point estimator for θ^* is

$$\bar{\theta}_m \;=\; \frac{1}{m} \sum_{j=1}^{m} \theta_{n,j}$$

  - confidence intervals can be formed in the usual way, based on the Student t distribution with $m - 1$ degrees of freedom.

Remarks on Robbins-Munro Algorithm

- Variants of previous procedure are available for multi-parameter problems.
- Main drawback of R-M algorithm:
  convergence can be slow and sensitive to choice of parameter a.
  - Current research focuses on finding methods that converge faster.
  - One easy, practical way to stabilize the algorithm a bit is to keep track and return the best parameter value seen so far.

Kiefer-Wolfowitz Algorithm

- A variant of R-M procedure is due to Kiefer and Wolfowitz.
- Replace the gradient estimate Z_n in R-M by a finite difference that is obtained by running the simulation at various values of parameter θ.
  - Finite-difference estimate of derivative at θ is
    $$\frac{\mu(\theta+h)-\mu(\theta)}{h} \quad \text{or} \quad \frac{\mu(\theta+h)-\mu(\theta-h)}{2h}$$
  - Choosing appropriate h is tricky.
  - K-W procedure breaks down as the dimension d of the parameter space increases.

Simultaneous perturbation stochastic approximation

- "Simultaneous perturbation stochastic approximation" (SPSA) tries to avoid the dimensionality problem of K-W as follows.
- At the kth iteration of a d-dimensional problem, a finite-difference is obtained by running the simulation at the two parameter values
  $$\theta \pm c\,\Delta_k,$$
  where
  - θ is the current value of the parameter vector (of length d)
  - c is a nonnegative real number that is a parameter of the algorithm
  - Δ_k is a d-vector of independent Bernoulli random variables, each equal to 1 or -1 with equal probability.

Back to Robbins-Munro Algorithm

- How do we obtain the derivative estimates needed for R-M algorithm?
- One popular method is via **likelihood ratios**.
- First suppose that we wish to estimate
  $$\mu \;=\; \mathcal{E}_p[\,h(X_1,\ldots,X_n)\,]$$
  - $\mathcal{E}_p$ denotes expectation when $X_1,\ldots,X_n \sim p$ IID
  - p is PMF with state space S
- Standard simulation approach to estimate μ is as follows:
  - For $k = 1, 2, \ldots, K$, generate n IID replicates of X, say $X_{1,k},\ldots,X_{n,k}$.
  - Set $\mu(k) = h(X_{1,k},\ldots,X_{n,k})$.
  - Estimate of μ is sample average of $\mu(1), \mu(2), \ldots, \mu(K)$.

Likelihood Ratios

- Suppose
  - q is another PMF with same state space S as p.
  - $q(s) > 0$ whenever $p(s) > 0$.
- Then μ can be estimated by sampling using q rather than p
- Let $X_1, \ldots, X_n \sim q$ IID, and set

$$L \;=\; \frac{\Pi_{i=1}^n p(X_i)}{\Pi_{i=1}^n q(X_i)}.$$

  - L is called a **likelihood ratio**
  - L represents likelihood of seeing observations $X_1, \ldots, X_n$ under PMF p relative to likelihood of seeing observations under PMF q.

Likelihood Ratios

Claim: $\mathcal{E}_q[\, h(X_1, \ldots, X_n)\, L\,] \;=\; \mathcal{E}_p[\, h(X_1, \ldots, X_n)\,]$,
where $\mathcal{E}_q$ denotes expectation under PMF q

Proof:

$$\begin{aligned}
&\mathcal{E}_q[\, h(X_1, \ldots, X_n)\, L\,] \\
&= \sum_{x_1 \in S} \cdots \sum_{x_n \in S} h(x_1, \ldots, x_n) \left(\frac{\Pi_{i=1}^n p(x_i)}{\Pi_{i=1}^n q(x_i)} \right) \\
&\quad \times \mathcal{P}_q(X_1 = x_1, \ldots, X_n = x_n) \\
&= \sum_{x_1 \in S} \cdots \sum_{x_n \in S} h(x_1, \ldots, x_n) \left(\frac{\Pi_{i=1}^n p(x_i)}{\Pi_{i=1}^n q(x_i)} \right) \prod_{i=1}^n q(x_i) \\
&= \sum_{x_1 \in S} \cdots \sum_{x_n \in S} h(x_1, \ldots, x_n) \prod_{i=1}^n p(x_i) \\
&= \mathcal{E}_p[\, h(X_1, \ldots, X_n)\,]
\end{aligned}$$

Likelihood Ratios

- Recall

$$\mathcal{E}_q[\, h(X_1, \ldots, X_n)\, L\,] \;=\; \mathcal{E}_p[\, h(X_1, \ldots, X_n)\,] \qquad (1)$$

- Thus, we can generate X's using q instead of p
  - Multiply performance $h(X_1, \ldots, X_n)$ by "correction factor" LR
- This idea can be applied in great generality.
  - For example, suppose X is RV with density function f.
  - If g is another density with $g(x) > 0$ whenever $f(x) > 0$, then (1) holds with

$$L \;=\; \frac{\Pi_{i=1}^n f(X_i)}{\Pi_{i=1}^n g(X_i)} \;=\; \prod_{i=1}^n \frac{f(X_i)}{g(X_i)}.$$

  - Similar results hold for very general stochastic processes.
  - Can replace n with a random time N.
- L can easily be computed in a recursive manner.

Likelihood Ratio Derivative Estimation

- Likelihood ratio techniques can be used to yield an estimate of $\mu'(\theta)$ based on simulation at the single parameter value θ.
- Suppose performance measure $\mu(\theta)$ satisfies

$$\mu(\theta) \;=\; \mathcal{E}_\theta[\, c(X, \theta)\,]$$

  - $c(X, \theta)$ is a "cost function" on realization X and parameter θ.
  - We write $\mathcal{E}_\theta$ to indicate that the distribution of the random variable X depends on θ.

Example: Likelihood Ratio Derivative Estimation

- Consider M/M/1 queue
  - arrival rate λ
  - service rate $\theta \in \Theta = [\underline{\theta}, \overline{\theta}]$.
- Define performance measure $\mu(\theta) = \mathcal{E}_\theta[\, c(X, \theta)\,]$
  - $X =$ average waiting time for first k customers
  - $c(x, \theta) = a\theta + bx$.
  - Cost function reflects tradeoff between
    - operating costs due to increase in service rate and
    - costs due to increased waiting times for customers.
- Ultimate goal is to determine $\theta \in \Theta$ to minimize $\mu(\theta)$.
  - Need derivative $\mu'(\theta)$ to apply Robbins-Munro algorithm.

Example: Likelihood Ratio Derivative Estimation

- Assume that $\lambda < \underline{\theta}$
  - So queue is stable for all possible choices of $\theta \in [\underline{\theta}, \overline{\theta}]$.
- As discussed before, we have

$$\begin{aligned} \mu(\theta + h) &= \mathcal{E}_{\theta+h}\,[\, c(X, \theta + h)\,] \\ &= \mathcal{E}_\theta\,[\, c(X, \theta + h)\, L(\theta + h)\,] \end{aligned}$$

  - Likelihood ratio

$$L(\theta + h) = \frac{d\mathcal{P}(\theta + h)}{d\mathcal{P}(\theta)}$$

  - $d\mathcal{P}(\theta + h)$ is likelihood of observed path under parameter $\theta + h$
  - $d\mathcal{P}(\theta)$ is likelihood of observed path under parameter θ

Example: Likelihood Ratio Derivative Estimation

- Recall

$$\begin{aligned} \mu(\theta) &= \mathcal{E}_\theta[\, c(X, \theta)\,], \\ L(\theta + h) &= d\mathcal{P}(\theta + h)/d\mathcal{P}(\theta), \\ \mu(\theta + h) &= \mathcal{E}_{\theta+h}\,[c(X, \theta + h)] = \mathcal{E}_\theta\,[c(X, \theta + h)\, L(\theta + h)] \end{aligned}$$

- Note that $L(\theta) = 1$, and under appropriate conditions,

$$\begin{aligned} \mu'(\theta) &= \lim_{h \to 0} \frac{\mu(\theta + h) - \mu(\theta)}{h} \\ &= \lim_{h \to 0} \mathcal{E}_\theta \left[\frac{c(X, \theta + h)\, L(\theta + h) - c(X, \theta)\, L(\theta)}{h} \right] \\ &= \mathcal{E}_\theta \left[\lim_{h \to 0} \frac{c(X, \theta + h)\, L(\theta + h) - c(X, \theta)\, L(\theta)}{h} \right] \\ &= \mathcal{E}_\theta \left[\frac{d}{d\theta} \Big\{ c(X, \theta)\, L(\theta) \Big\} \right] \\ &= \mathcal{E}_\theta \left[c'(X, \theta) + c(X, \theta)\, L'(\theta) \right] \end{aligned}$$

Example: Likelihood Ratio Derivative Estimation

- To estimate $\mu'(\theta)$, generate IID replicates $X_1, \ldots, X_n$, and compute the estimator

$$\widehat{\mu}'_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[c'(X_i, \theta) + c(X_i, \theta)\, L'_i(\theta) \right]$$

  - $L'_i(\theta) =$ value of $L'(\theta)$ on ith simulated replication.
- Confidence intervals are computed in the usual way.
- For Robbins-Monro algorithm
  - Often take n to be small, even perhaps taking $n = 1$.
  - In practice, often more efficient to execute many R-M iterations with relatively noisy derivative estimates than to execute fewer iterations with more precise derivative estimates.
  - Confidence intervals for $\mu'(\theta)$ not needed.

Example: M/M/1 Likelihood Ratio Derivative Estimation

- Recall: performance measure $\mu(\theta) = \mathcal{E}_\theta[\, c(X, \theta)\,]$
  - X = average waiting time of first k customers.
  - cost function $c(x, \theta) = a\theta + bx$.
- In the course of generating the first k waiting times, let
  - $A_1, \ldots, A_k$ be first k interarrival times $\sim$ Expo(λ)
  - $V_1, \ldots, V_k$ be first k service times $\sim$ Expo(θ).
- Likelihoods of $(A_1, \ldots, A_k, V_1, \ldots, V_k)$ under parameters θ and $\theta + h$ are

$$d\mathcal{P}(\theta) = \prod_{i=1}^{k} \left[\lambda e^{-\lambda A_i}\right] \left[\theta e^{-\theta V_i}\right],$$

$$d\mathcal{P}(\theta + h) = \prod_{i=1}^{k} \left[\lambda e^{-\lambda A_i}\right] \left[(\theta + h) e^{-(\theta+h) V_i}\right]$$

Example: M/M/1 Likelihood Ratio Derivative Estimation

- Likelihood ratio

$$L(\theta + h) = \frac{\prod_{i=1}^{k} (\theta + h) e^{-(\theta+h)V_i}}{\prod_{i=1}^{k} \theta e^{-\theta V_i}} = \prod_{i=1}^{k} \left(\frac{\theta + h}{\theta}\right) e^{-hV_i}$$

- Thus, since $c(x, \theta) = a\theta + bx$, can show that

$$L'(\theta) = \lim_{h \to 0} \frac{L(\theta + h) - L(\theta)}{h} = \sum_{i=1}^{k} \left(\frac{1}{\theta} - V_i\right),$$

$$\widehat{\mu}'_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\, c'(X_i, \theta) + c(X_i, \theta)\, L'_i(\theta)\,\right]$$

$$= \frac{1}{n} \sum_{j=1}^{n} \left[a + \left(a\theta + bX_j\right) \left(\sum_{i=1}^{k} \left(\frac{1}{\theta} - V_{ij}\right)\right)\right],$$

where X_j is the average of the k waiting times in the jth simulation, and V_{ij} is the ith service time during the jth simulation.

Remarks on Likelihood Ratio Derivative Estimation

- Derivative info useful even outside context of stochastic optimization.
  - Explore sensitivity of system performance to specified parameters.
- Other methods can be used to estimate derivatives
  - Infinitesimal perturbation analysis (IPA)
  - Weak derivatives
- Main drawback of likelihood-ratio approach is that variance of likelihood ratio increases with length of simulation.
  - Thus, the estimates can be very noisy for long simulation runs.
  - Requires many simulation replications to get reasonable estimates.

Small Number of Alternatives: Ranking and Selection

- Now let's look at the case of a finite number of alternatives.
- Unless stated otherwise, we will now assume that "bigger is better"
  - Our goal is to maximize some performance measure.
  - Can convert any minimization problem to maximization problem by maximizing the negative of original performance measure.
- Suppose that we are comparing k systems
  - k is on the order of 3 to 20.
- μ_i = value of performance measure for ith alternative.
  - E.g., μ_i might be expected revenue from system i over one year.
- Want to determine system i with largest μ_i.

Small Number of Alternatives: Ranking and Selection

- Sometimes we write

$$\mu_{(1)} \ < \ \mu_{(2)} \ < \ \cdots \ < \ \mu_{(k)}$$

so that $\mu_{(k)}$ is the optimal value of the performance measure.

- **Goal:** select best system with (high) probability $1 - \alpha$ whenever performance measure for best system is $> \delta$ better than all others
  - i.e., whenever $\mu_{(k)} - \mu_{(j)} > \delta$ for all $j \neq k$.
  - Parameter δ is called the **indifference zone**.
  - If there exist systems within the indifference zone (i.e., when $\mu_{(k)} - \mu_{(j)} \leq \delta$ for one or more values of j),
    - algorithm below selects a good solution (i.e., a system lying within the indifference zone) with probability $\geq 1 - \alpha$.

Small Number of Alternatives: Ranking and Selection

Inputs to the procedure are observations Y_{ij}

- $Y_{ij} = j$th output from alternative i.
- For a fixed alternative i, the observations $Y_{i1}, Y_{i2}, Y_{i3}, \ldots$ are assumed to be IID (i.e., based on independent simulation runs).
- Each Y_{ij} is required to have (at least approximately) a normal distribution.
  - In a terminating simulation, Y_{ij} could be an average over the jth replication of system i.
  - In a steady-state simulation, Y_{ij} could be the jth batch mean of system i.

Small Number of Alternatives: Ranking and Selection

- The following algorithm (due to Rinott, with modifications by Nelson and Matejcik) will
  - select the "best" system as defined above, with probability $1 - \alpha$
  - provide simultaneous $100(1 - \alpha)\%$ confidence intervals J_i on the quantities

$$\gamma_i \ = \ \mu_i - \max_{j \neq i} \mu_j$$

for $1 \leq i \leq k$; i.e.,

$$\mathcal{P}(\gamma_i \in J_i \text{ for all } 1 \leq i \leq k) \ \geq \ 1 - \alpha.$$

Small Number of Alternatives: Ranking and Selection

Recall: our algorithm will guarantee

$$\mathcal{P}(\gamma_i \in J_i \text{ for all } 1 \leq i \leq k) \ \geq \ 1 - \alpha.$$

- This is a stronger assertion than

$$\mathcal{P}(\gamma_i \in J_i) \ \geq \ 1 - \alpha \quad \text{for each} \quad 1 \leq i \leq k,$$

as with individual confidence intervals.
- This "multiple comparisons with the best" (MCB) approach indicates how close each system is to best of rest.
- This information is useful in identifying other good solutions.
- One of them may be needed if the "best" solution is ruled out because of secondary considerations
  - ease of implementation
  - maintenance costs

Ranking and Selection Algorithm

1. Specify
   - confidence level $1-\alpha$
   - indifference zone δ
   - initial sample size n_0 ($n_0 \geq 20$).
2. For $i = 1$ to k, take n_0 IID observations $Y_{i1}, Y_{i2}, \ldots, Y_{in_0}$
3. For $i = 1$ to k, compute

$$\bar{Y}_i = \frac{1}{n_0} \sum_{j=1}^{n_0} Y_{ij},$$

$$S_i^2 = \frac{1}{n_0 - 1} \sum_{j=1}^{n_0} \left(Y_{ij} - \bar{Y}_i\right)^2$$

4. Look up value of constant h from Table 8.3 (given at end of lecture notes).

5. For $i = 1$ to k, take $N_i - n_0$ additional observations Y_{ij}, independent of the previous observations, where

$$N_i = \max\left(n_0, \left\lceil \left(\frac{hS_i}{\delta}\right)^2 \right\rceil\right)$$

6. For $i = 1$ to k, compute

$$\widehat{\mu}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} Y_{ij}$$

7. Select system with largest value of $\widehat{\mu}_i$ as best.
8. Compute simultaneous $100(1-\alpha)\%$ confidence intervals $[a_i, b_i]$ for $\gamma_i = \mu_i - \max_{j \neq i} \mu_j$, where

$$a_i = \min(\,0,\, \widehat{\mu}_i - \max_{j \neq i} \widehat{\mu}_j - \delta\,),$$

$$b_i = \max(\,0,\, \widehat{\mu}_i - \max_{j \neq i} \widehat{\mu}_j + \delta\,).$$

Ranking and Selection Algorithm

- Justification of this algorithm is similar to (but more complicated than) the discussion in Section 10A in Law.
- See Rinott's 1978 paper in *Commun. Statist. A: Theory and Methods* for details.
- Damerdji and Nakayama (1999), *ACM TOMACS*, provides justification for the algorithm applied in steady-state simulations.
- Algorithm presented here is simpler to execute than the one discussed in Law.
- Improved versions of this procedure can be obtained through
  - common random numbers
  - fully sequential with variance updating
  - screening

Large Number of Alternatives: Discrete Stochastic Optimization

- **Goal:** $\max_{\theta \in \Theta} \mu(\theta)$
  - Value of $\mu(\theta)$ is estimated using simulation
  - Θ is a very large set of discrete design alternatives.
- **Example:**
  - θ is the number of servers at a service center
  - $\mu(\theta)$ is the expected value of the total revenue over a year of operation.
- Much ongoing research on random-search algorithms.

Large Number of Alternatives: Discrete Stochastic Optimization

- Most current algorithms generate a sequence $(\theta_n : n \geq 0)$
  - $\theta_{n+1} \in N(\theta_n) \cup \{\theta_n\}$
  - $N(\theta)$ is a "neighborhood" of θ
  - Maintain best solution found so far.
- Various algorithms differ in
  - Structure of the neighborhood
  - Rule used to determine θ_{n+1} from θ_n
  - Rule used to compute θ_n^*, the estimate of θ^* on the nth iteration
  - Rule used to stop the algorithm.

Large Number of Alternatives: Discrete Stochastic Optimization

- We present a typical algorithm of this type (due to Andradóttir).
- "Simulated annealing"-type rule to determine θ_{n+1} from θ_n
  - Usually accept a better solution
  - But with a small probability, sometimes accept a worse solution to avoid getting stuck at a local maximum.

Discrete Stochastic Optimization Algorithm

1. • Fix "temperature" $T > 0$, computing budget $N > 0$
   • Select a starting point θ_0
   • Generate an estimate of $\widehat{\mu}(\theta_0)$ of $\mu(\theta_0)$ via simulation. Set

$$\begin{aligned} A(\theta_0) &= \widehat{\mu}(\theta_0) && // \text{ accumulator for each alternative} \\ C(\theta_0) &= 1 && // \text{ count for each alternative} \\ n &= 0 && // \text{ iteration index} \\ \theta_0^* &= \theta_0 && // \text{ current estimate of best} \end{aligned}$$

2. • Select a candidate solution θ_n' randomly and uniformly from $\Theta - \{\theta_n\}$.
   • Generate an estimate $\widehat{\mu}(\theta_n')$ of $\mu(\theta_n')$ via simulation.

3. • If $\widehat{\mu}(\theta_n') \geq \widehat{\mu}(\theta_n)$, then set $\theta_{n+1} = \theta_n'$.
   • Otherwise, generate a random variable $U \sim$ unif[0, 1].
   - If

$$U \leq \exp\left[\frac{\widehat{\mu}(\theta_n') - \widehat{\mu}(\theta_n)}{T}\right]$$

   then set $\theta_{n+1} = \theta_n'$
   - Otherwise, set $\theta_{n+1} = \theta_n$.

4. Update accumulators and counts:

$$A(\theta_{n+1}) = A(\theta_{n+1}) + \widehat{\mu}(\theta_{n+1}),$$

$$C(\theta_{n+1}) = C(\theta_{n+1}) + 1.$$

5. • If
$$\frac{A(\theta_{n+1})}{C(\theta_{n+1})} > \frac{A(\theta_n^*)}{C(\theta_n^*)},$$
then set $\theta_{n+1}^* = \theta_{n+1}$.
   • Otherwise, set $\theta_{n+1}^* = \theta_n^*$.

6. • If $n \geq N$, return the final answer θ_{n+1}^* with estimated objective function value $A(\theta_{n+1}^*)/C(\theta_{n+1}^*)$.
   • Otherwise, set $n = n + 1$, and go to Step 2.

In the above algorithm, the neighborhood was defined as $N(\theta) = \Theta - \{\theta\}$. If the problem has more structure, the neighborhood may be defined in a problem-specific manner.

TABLE 8.3 Values of h Required by the Procedure Rinott + MCB

		k								
α	n_0	2	3	4	5	6	7	8	9	10
0.10	5	2.291	3.058	3.511	3.837	4.093	4.305	4.486	4.644	4.786
	6	2.177	2.871	3.270	3.552	3.771	3.951	4.103	4.235	4.352
	7	2.107	2.758	3.126	3.384	3.582	3.744	3.881	3.999	4.103
	8	2.059	2.682	3.031	3.273	3.459	3.609	3.736	3.845	3.941
	9	2.025	2.628	2.963	3.195	3.372	3.515	3.635	3.738	3.829
	10	1.999	2.587	2.913	3.137	3.307	3.445	3.560	3.659	3.746
	11	1.978	2.556	2.874	3.092	3.258	3.391	3.503	3.598	3.682
	12	1.962	2.531	2.843	3.056	3.218	3.349	3.457	3.551	3.632
	13	1.948	2.510	2.817	3.027	3.186	3.314	3.420	3.512	3.592
	14	1.937	2.493	2.796	3.003	3.160	3.285	3.390	3.480	3.558
	15	1.928	2.479	2.779	2.983	3.138	3.261	3.364	3.453	3.530
	16	1.919	2.467	2.764	2.966	3.119	3.241	3.343	3.430	3.506
	17	1.912	2.456	2.751	2.951	3.102	3.223	3.324	3.410	3.485
	18	1.906	2.447	2.739	2.938	3.088	3.208	3.308	3.393	3.467
	19	1.901	2.438	2.729	2.926	3.075	3.194	3.293	3.378	3.451
	20	1.896	2.431	2.720	2.916	3.064	3.182	3.280	3.364	3.437
	30	1.866	2.387	2.666	2.855	2.997	3.110	3.204	3.284	3.354
	40	1.852	2.366	2.641	2.827	2.966	3.077	3.169	3.247	3.315
	50	1.844	2.354	2.627	2.810	2.948	3.057	3.148	3.225	3.292
0.05	5	3.107	3.905	4.390	4.744	5.025	5.259	5.461	5.638	5.797
	6	2.910	3.602	4.010	4.303	4.533	4.722	4.884	5.025	5.150
	7	2.791	3.424	3.791	4.051	4.253	4.419	4.559	4.681	4.789
	8	2.712	3.308	3.649	3.889	4.074	4.225	4.353	4.463	4.561
	9	2.656	3.226	3.550	3.776	3.950	4.091	4.210	4.313	4.404
	10	2.614	3.166	3.476	3.693	3.859	3.993	4.106	4.204	4.290
	11	2.582	3.119	3.420	3.629	3.789	3.918	4.027	4.121	4.203
	12	2.556	3.082	3.376	3.579	3.734	3.860	3.965	4.055	4.135
	13	2.534	3.052	3.340	3.539	3.690	3.812	3.915	4.003	4.080
	14	2.517	3.027	3.310	3.505	3.654	3.773	3.874	3.960	4.035
	15	2.502	3.006	3.285	3.477	3.623	3.741	3.839	3.924	3.998
	16	2.489	2.988	3.264	3.453	3.597	3.713	3.810	3.893	3.966
	17	2.478	2.973	3.246	3.433	3.575	3.689	3.785	3.867	3.938
	18	2.468	2.959	3.230	3.415	3.556	3.669	3.763	3.844	3.914
	19	2.460	2.948	3.216	3.399	3.539	3.650	3.744	3.824	3.894
	20	2.452	2.937	3.203	3.385	3.523	3.634	3.727	3.806	3.875
	30	2.407	2.874	3.129	3.303	3.434	3.539	3.626	3.701	3.766
	40	2.386	2.845	3.094	3.264	3.392	3.495	3.580	3.652	3.716
	50	2.373	2.828	3.074	3.242	3.368	3.469	3.553	3.624	3.687