S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13020-0

Chapter 20

An Overview of Simulation Optimization via Random Search

Sigrún Andradóttir

School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
E-mail: sa@gatech.edu

Abstract

This chapter provides an overview of the use of random search methods for optimizing system performance via simulation. More specifically, we consider a broad class of optimization algorithms that sample a set of feasible system designs in each iteration, conduct simulations at the sampled designs in an effort to estimate the performance of these designs, and then use the simulation results to decide on what designs should be sampled in the next iteration and on the current estimate of the optimal system design. Consequently, the class of optimization algorithms under consideration is broad enough to include simulated annealing, tabu search, and genetic algorithms as special cases. We provide a discussion of when random search methods are guaranteed to converge almost surely to a globally optimal system design, as well as a description of desirable features such methods should have in order to yield attractive empirical performance.

1 Introduction

This chapter is concerned with the use of random search to solve simulation optimization problems of the form

$$\min_{\theta\in\Theta} f(\theta), \tag{1}$$

where Θ is the feasible region and $f:\Theta \to \mathbb{R}$ is the objective function. In practice, the feasible region Θ consists of all possible designs of a system under consideration and $f(\theta)$ denotes the expected performance of that system under the design $\theta \in \Theta$. We assume that the system under consideration is complex enough that the expected performance $f(\theta)$ of each system design $\theta \in \Theta$ cannot be determined exactly, but is instead estimated through simulation. The feasible region Θ is often embedded in a real space (i.e., $\Theta \subset \mathbb{R}^d$, where d is a positive integer), but this is not a requirement for applying random search methods to solve the optimization problem (1). We will outline the basic form of random search methods and discuss the convergence of such methods and other desirable features the methods should have. Other approaches for solving simulation optimization problems of the form (1) are described in Chapters 18, 19 and 21 of this volume.

Random search methods involve sampling points from the feasible region Θ of the underlying optimization problem (1) based on some sampling strategy, evaluating the performance of the objective function f at the chosen points, and then updating the sampling strategy based on the observed history (i.e., the feasible points that have been sampled so far and the associated objective function values), before proceeding to the next iteration. Hence, random search methods constitute a broad class of optimization techniques that can be applied to solve both deterministic and stochastic optimization problems with either discrete or continuous decision parameters (or both). Simulation optimization is a special case of stochastic optimization where the required objective function values $f(\theta)$ are estimated via computer simulation, and hence involve some noise.

Most existing random search methods were originally developed for solving deterministic optimization problems where there is no noise in the estimated objective function values. There are two primary ways in which simulation optimization techniques address the problem of having noise in the estimated objective values (it is also possible to combine ideas from the two approaches). One approach involves expending a significant amount of computer effort at each point visited by the algorithm to obtain a relatively precise estimate of the objective function values at these points (especially as the search for an optimal solution progresses), and then using a deterministic optimization approach to solve the underlying optimization problem. Kleywegt et al. (2001) study the application of this sample path approach to solve discrete optimization problems; see also Healy and Schruben (1991), Rubinstein and Shapiro (1993), Robinson (1996), Shapiro and Wardi (1996), Chen and Schmeiser (2001) and Homem-de-Mello (2003), among others, for related work.

The other approach does not involve obtaining highly precise estimates of the objective function values each time the algorithm visits a feasible point, so that the algorithm must at each step decide on how to proceed based only on limited information. This means that techniques that were originally intended for deterministic optimization generally need to be modified to yield good performance in the presence of noise. The basic issue is that the fact that a particular point *appears* at first to be good (bad), in that it has a small (large) estimated objective function value, see Equation (1), does not necessarily imply that this point is in fact good (bad). Consequently, the optimization technique must proceed cautiously based on the information that is available at the time, so that the method can recover quickly from errors made due to misleading information. For example, it is possible for the method to obtain misleading estimated objective function values suggesting that a bad subset of the feasible region Θ is preferable to a good subset of Θ. When this happens, the method should not completely lose progress made previously with respect to identifying the location of a global optimal solution. Stochastic approximation methods fall within this category of simulation optimization techniques, see, for example, Chapter 19 in this volume and Robbins and Monro (1951), Kiefer and Wolfowitz (1952), Kushner and Clark (1978), Benveniste et al. (1990), Pflug (1996), Kushner and Yin (1997) and Spall (2003). Currently existing random search methods for simulation optimization fall in both categories in that they may require highly precise estimates of the objective function values as the search progresses or not.

Throughout this chapter, we focus on the use of random search methods to optimize the expected performance of stochastic systems using simulation when the number of potential system configurations (i.e., the number of elements of the feasible region Θ) is either finite or countably infinite. This is consistent with the current use of random search methods for solving simulation optimization problems. The objective function values $f(\theta)$ can be estimated using either transient or steady-state simulation, depending on whether we are interested in optimizing the performance of the underlying system over a finite time horizon, or in the long run, respectively.

The outline of this chapter is as follows: In Section 2 we describe the basic structure of random search methods and briefly review the literature on such methods designed for solving simulation optimization problems. In Section 3 we discuss conditions under which random search methods can be shown to be almost surely convergent to the set of global optimal solutions of the optimization problem (1). In Section 4 we discuss properties that random search methods should have in addition to guaranteed convergence to perform well in practice. Finally, Section 5 contains a brief summary of this chapter.

2 A brief review of random search methods

In this section, we provide the general structure and a brief review of random search methods. We start by providing the general structure of random search methods applied to solve the simulation optimization problem (1).

Generic random search algorithm for simulation optimization:

Step 0 (initialize). Choose the initial sampling strategy $\mathcal{S}_1$ and let $n = 1$.

Step 1 (sample). Select $\theta_n^{(1)}, \ldots, \theta_n^{(M_n)} \in \Theta$ according to the sampling strategy $\mathcal{S}_n$.

Step 2 (simulate). Estimate $f(\theta_n^{(i)})$, for $i = 1, \ldots, M_n$, using simulation.

Step 3 (update). Use the simulation results obtained so far in Step 2 to compute an estimate of the optimal solution θ_n^* and to choose an updated sampling strategy $\mathcal{S}_{n+1}$. Let $n = n + 1$ and go to Step 1.

Note that the number M_n of feasible points sampled in iteration n of the algorithm above is a parameter of the sampling strategy $\mathcal{S}_n$ used in iteration n, and consequently does not need to be chosen in advance of executing the algorithm. Also, note that the statement of the algorithm above does not include a stopping criterion. This is typical of the random search literature, and is consistent with the fact that convergence results for random search methods are typically asymptotic in nature; i.e., the sequence $\{\theta_n^*\}$ will be shown to converge in some sense as n approaches infinity. In practice, it is of course necessary to augment this algorithm with a suitable stopping criterion (because otherwise the algorithm involves executing an infinite loop).

The generic random search algorithm given above clearly outlines a broad class of techniques for solving the simulation optimization problem (1). These techniques differ primarily in the choice of the sampling strategy $\{\mathcal{S}_n\}$. For example, the sampling strategy can be "point-based", leading to methods such as simulated annealing and tabu search that involve sampling points in the neighborhood of the current point that the algorithm is visiting, where the neighborhood of a feasible point $\theta \in \Theta$ is a set $N(\theta) \subset \Theta \setminus \{\theta\}$, and the neighborhood structure $N = \{N(\theta)\colon\ \theta \in \Theta\}$ is connected in the sense that it has the feature that for all $\theta, \theta' \in \Theta$, $\theta \neq \theta'$, there exist $\theta_1, \ldots, \theta_k \in \Theta$ such that $\theta_{i+1} \in N(\theta_i)$ for $i = 0, \ldots, k$, where $\theta_0 = \theta$ and $\theta_{k+1} = \theta'$. The sampling strategy can also be "set-based", leading to methods such as (stochastic) branch-and-bound and nested partitions (see Chapter 21) that sample points in a particular subset of the feasible space Θ. Finally, the sampling strategy can be "population-based", leading to methods such as genetic algorithms that generate a new collection of points using properties of the current collection of points. Random search methods also differ in the choice of the sequence of estimates $\{\theta_n^*\}$ of the optimal solution. This is discussed in Section 3.

A detailed discussion of specific random search methods is outside the scope of this chapter. Instead, we focus on desirable properties of random search methods (i.e., guaranteed convergence and attractive empirical performance) and how these properties may be achieved. Nevertheless, to assist readers who are interested in using random search methods to solve simulation optimization problems, we now briefly describe some existing random search methods. Tabu search, nested partitions, and genetic algorithms are reviewed in Chapter 21 of this volume, and hence will not be discussed here.

Stochastic ruler methods constitute one class of random search methods, see Yan and Mukai (1992) and Alrefaei and Andradóttir (2001, 2005). These are point-based methods in which the quality of different feasible points and future movements of the algorithm are decided by comparing estimated objective function values to observations of the stochastic ruler, which is a uniform random variable covering the range of values the objective function f can take. Stochastic comparison and descent algorithms form another class of random search methods. This class would include the point-based methods of Andradóttir (1995, 1996, 1999), Gong et al. (1999) and Prudius and Andradóttir (2004, 2006a), and also the set-based COMPASS method of Hong and Nelson

(2005). These methods involve comparing estimated objective function values at different points with each other, rather than with a stochastic ruler. Algorithm movements take place when a point is found that appears to be better (i.e., has a smaller objective function value, see Equation (1)) than previously visited points. Simulated annealing methods constitute a third class of random search methods, see Gelfand and Mitter (1989), Gutjahr and Pflug (1996), Fox and Heine (1996), Alrefaei and Andradóttir (1999) and Prudius and Andradóttir (2005, 2006b). These point-based methods are designed to escape from locally optimal solutions (and hence be globally convergent), and consequently will sometimes move from the current point to an apparently inferior candidate point in the hope that this will allow the algorithm to eventually find even better points elsewhere in the feasible region Θ. Other related work includes the branch-and-bound method of Norkin et al. (1998a, 1998b), and also the results of Homem-de-Mello (2003) on the convergence of random search.

3 Convergence

One desirable feature of random search methods for discrete simulation optimization is that such methods can be shown to converge almost surely to the set of global optimal solutions of the underlying (discrete) simulation optimization problem under very general conditions. We now review such convergence results. The material in this section is based on Andradóttir (1999, 2006); see also Andradóttir (2000).

In the process of using the generic random search algorithm described in Section 2 to solve a simulation optimization problem of the form (1), simulations will be conducted at any point θ in the feasible region Θ each time the algorithm visits that point (i.e., each time when $\theta \in \{\theta_n^{(1)}, \ldots, \theta_n^{(M_n)}\}$). These simulation results are then aggregated to obtain estimates of the objective function value $f(\theta)$, for each $\theta \in \Theta$, that improve with each visit the algorithm makes to θ. We will use the notation $C_n(\theta)$ to refer to the number of times the feasible point θ has been visited in the first n iterations, and $f_n(\theta)$ to refer to the aggregated estimate of $f(\theta)$ available after n iterations have been completed.

We first consider the case when the feasible region Θ is finite, see Andradóttir (1999). In this case, the estimated optimal solution θ_n^* after n iterations have been completed will be chosen from the subset of the feasible points visited so far by the method whose elements have the best (smallest, see Equation (1)) estimated objective function values. Hence,

$$\theta_n^* \in \arg\min_{\theta \in \widetilde{\Theta}_n} f_n(\theta) \tag{2}$$

for all $n \geqslant 1$, where

$$\widetilde{\Theta}_n = \bigl\{\theta \in \Theta \colon\ C_n(\theta) \geqslant 1\bigr\}.$$

If the estimates $f_n(\theta)$ of the objective function are strongly consistent, for all $\theta \in \Theta$, then almost sure convergence to the set of global optimal solutions

$$\Theta^* = \{\theta \in \Theta\colon f(\theta) \leqslant f(\theta'), \forall \theta' \in \Theta\}$$

of the simulation optimization problem (1) is guaranteed. Assuring the strong consistency of the objective function estimates $f_n(\theta)$, for all $\theta \in \Theta$, would typically require that the underlying random search method be guaranteed to visit each feasible point $\theta \in \Theta$ infinitely often with probability one as the number of iterations grows.

For all $\theta \in \Theta$ and $t \geqslant 0$, let $X_\theta(t)$ denote the state of the simulated system at time t when the value of the decision parameter that we want to optimize is given by θ. In transient simulation optimization, the objective function values take the form $f(\theta) = \mathrm{E}\{Y_\theta\}$, where the random variable Y_θ only depends on the system behavior $X_\theta(t)$ until some finite time τ_θ, where τ_θ is a stopping time with respect to the stochastic process $\{X_\theta(t)\}$ (so that $f(\theta)$ can be estimated using transient simulation). For example, if for all $\theta \in \Theta$, $X_\theta(t)$ denotes the number of customers in a $G/G/s$ queue at time t, for all $t \geqslant 0$, $\tau_\theta = T \in (0, \infty)$, and $Y_\theta = \int_0^T X_\theta(t)\,\mathrm{d}t/T$, then $f(\theta)$ represents the average number of customers in the system over the period $[0, T]$. The strong consistency of the estimated objective function values $f_n(\theta)$ in transient simulation optimization is usually achieved by generating one or more observations of Y_θ at each feasible point $\theta \in \{\theta_n^{(1)}, \ldots, \theta_n^{(M_n)}\}$ visited in each iteration n of the algorithm, averaging these observations to obtain an estimate of $f(\theta)$, and then letting $f_n(\theta)$ be the average of the $C_n(\theta)$ estimates of $f(\theta)$ that have been obtained so far by the algorithm. The strong law of large numbers would then be used to prove the strong consistency of the estimated objective function values $f_n(\theta)$, assuming that each feasible point $\theta \in \Theta$ is visited infinitely often by the underlying algorithm with probability one.

In steady-state simulation optimization, we usually have that

$$f(\theta) = \lim_{t\to\infty} \int_0^t h_\theta\bigl(X_\theta(u)\bigr)\,\mathrm{d}u \Big/ t$$

(assuming that the limit exists and equals a constant almost surely), where h_θ is a deterministic function. For example, in the $G/G/s$ example given in the previous paragraph, if for all $\theta \in \Theta$, $h_\theta(x) = x$ for all $x \in \mathbb{R}$, then $f(\theta)$ represents the long-run average number of customers in the system. The strong consistency of the estimators $f_n(\theta)$ in steady-state simulation optimization is usually proved using renewal or regenerative process theory, and can be achieved in various ways, see Andradóttir (2000). One approach involves saving enough information about the simulation at each point $\theta \in \widetilde{\Theta}_n$ visited so far by the algorithm, so that the simulation can be restarted when the algorithm revisits the point θ. Then if the process $\{X_\theta(t)\}$ is simulated for T time units each time when $\theta \in \{\theta_n^{(1)}, \ldots, \theta_n^{(M_n)}\}$, the estimate $f_n(\theta)$ will be an observation of

$$\frac{1}{TC_n(\theta)} \int_0^{TC_n(\theta)} h_\theta\bigl(X_\theta(u)\bigr)\,\mathrm{d}u$$

(we could also truncate some observations at the beginning of the run). Another way involves starting simulations from scratch each time the algorithm visits a point $\theta \in \Theta$, letting the length T_k of the simulation run conducted the kth time the algorithm visits a feasible point grow to infinity with k and averaging all the resulting $C_n(\theta)$ estimates, so that

$$f_n(\theta) = \frac{1}{C_n(\theta)} \sum_{k=1}^{C_n(\theta)} \frac{1}{T_k} \int_0^{T_k} h_\theta\bigl(X_{\theta,k}(u)\bigr)\,\mathrm{d}u$$

for all $n \geqslant 1$ and $\theta \in \widetilde{\Theta}_n$, where $\{X_{\theta,k}(t)\}$ denotes the sample path of $\{X_\theta(t)\}$ generated the kth time the algorithm visits $\theta \in \Theta$ for all $k \geqslant 1$ (as before, we could truncate some observations at the beginning of each replication).

In both cases (transient and steady-state simulation optimization), one can ensure that each point $\theta \in \Theta$ is visited infinitely often by including a pure search component in the random search method (e.g., in each iteration n, generate a point at random from the feasible region with a small probability). Moreover, many existing random search methods have the property that $M_n = 1$ for all n, and the sequence of points $\{\theta_n^{(1)}\}$ visited by the algorithm is a Markov chain. If this Markov chain is time-homogeneous, irreducible and positive recurrent, then each feasible point $\theta \in \Theta$ will be visited infinitely often with probability one.

We now consider the case when the feasible region Θ is countably infinite, see (Andradóttir, 2006). In this case, we constrain the estimated optimal solution θ_n^* after n iterations of the random search method under consideration have been completed to be chosen from the subset of the feasible points visited *sufficiently often* so far by the algorithm whose elements have the best (smallest) estimated objective function values. (This approach for estimating the optimal solution may also perform better than the approach (2) even when Θ is finite, see Prudius and Andradóttir, 2006a.) Moreover, it is often useful to also constrain θ_n^* to lie in a deterministic set Θ_n, where $\Theta_n \subset \Theta_{n+1}$ for all n and $\bigcup_{n=1}^\infty \Theta_n = \Theta$. Consequently, we let

$$\theta_n^* \in \arg\min_{\theta \in \widetilde{\Theta}'_n} f_n(\theta), \tag{3}$$

where

$$\widetilde{\Theta}'_n = \bigl\{\theta \in \Theta_n\colon\, C_n(\theta) \geqslant K_n\bigr\}$$

and $\{K_n\}$ is a nondecreasing sequence of positive integers that converges to infinity at a sublinear rate. For example, the sequence $\{\Theta_n\}$ would typically be chosen so that the number of elements $|\Theta_n|$ in Θ_n increases at a polynomial rate with n (e.g., if $\Theta \subset \mathbb{R}^d$, then we could choose $\Theta_n = \Theta \cap \prod_{i=1}^{d}[\tilde{\theta}_i - P_i(n), \tilde{\theta}_i + P_i(n)]$ for all $n \geqslant 1$, where $(\tilde{\theta}_1, \ldots, \tilde{\theta}_d)$ is the user's best initial guess of an optimal solution $\theta^* \in \Theta^*$ and $P_1(n), \ldots, P_d(n)$ are polynomial functions that increase with n), and one could choose $K_n = [an^\alpha]$ for all $n \geqslant 1$, where $a > 0$, $0 < \alpha < 1$, and $[x]$ is the integer closest to x for all $x \in \mathbb{R}$. (The sequence $\{K_n\}$ is constrained to grow at a sublinear rate in n because it is typically possible to show that the number of visits $C_n(\theta)$ to the feasible point θ in the first n iterations of the algorithm grows at a linear rate with n for all θ in some subset of interest of the feasible region. Such feasible points θ would then be included in the set $\widetilde{\Theta}'_n$ with probability one for large n. If the number M_n of points sampled in iteration n grows superlinearly with n, then it is possible that $C_n(\theta)$ will also grow superlinearly with n for some interesting set of feasible points θ, in which case the sequence K_n could be chosen to grow at a linear or superlinear rate with n.)

Suppose that Θ is countably infinite, Equation (3) is used to estimate the optimal solution, the underlying random search method is guaranteed to visit each feasible point sufficiently often as the number of iterations grows (so that the intersection between $\widetilde{\Theta}'_n$ and Θ^* is nonempty for sufficiently large n), and the estimates of the objective function value $f(\theta)$ obtained after the feasible point θ has been visited k times by the algorithm, denoted $\hat{f}_k(\theta)$, are "sufficiently close to" the true objective function value $f(\theta)$ for large k and all $\theta \in \Theta$. Then almost sure convergence to the set Θ^* of global optimal solutions to the simulation optimization problem (1) is guaranteed, see Theorem 3.1 and Section 4 of Andradóttir (2006) for the details. Assuring that each point $\theta \in \Theta$ is visited sufficiently often can typically be achieved in the same manner as when Θ is finite (i.e., by incorporating a pure search component in the method or ensuring that the sequence of points visited by the algorithm constitutes a time-homogeneous, irreducible, and positive recurrent Markov chain). Ensuring that the estimated objective function values $\hat{f}_k(\theta)$ are sufficiently close to the true objective function values $f(\theta)$ for large k and all $\theta \in \Theta$ can usually be ascertained using large deviations theory, see Andradóttir (2006) for more details and Dembo and Zeitouni (1993) for an introduction to large deviations theory.

4 Efficiency

In Section 3 we reviewed under what conditions a random search method is guaranteed to converge almost surely to the set of global optimal solutions Θ^*. These conditions are quite general, and consequently leave a lot of flexibility to develop simulation optimization algorithms that are both provably globally convergent and also highly efficient. Consequently, we believe that there is no reason to consider simulation optimization algorithms in practice that are not guaranteed to converge almost surely to the set of global optimal solutions under weak conditions. Fortunately, many existing random search methods that do not have this property at present can be easily modified in a way that assures almost sure and global convergence (e.g., by adding a pure search component to the method).

However, knowing that a simulation optimization algorithm is globally convergent with probability one does not necessarily imply that this algorithm will perform well in practice. In fact, it is easy to see that certain algorithms that are obviously undesirable are nevertheless guaranteed to converge almost surely to the set of global optimal solutions Θ^* with probability one. For example, when Θ is finite, then repeated enumeration (to repeatedly visit all the points in the state space in a particular order) will converge almost surely to Θ^* as long as the point with the best estimated objective function value is used to estimate the optimal solution (see Equation (2)) and the estimated objective function values obtained after k visits to each feasible point are strongly consistent as k grows.

Moreover, no-free-lunch theorems for deterministic optimization (see Wolpert and Macready, 1997) show that without any knowledge about the structure of the underlying optimization problem, all optimization algorithms will exhibit the same average performance (with the average taken over all possible objective functions). More specifically, suppose that we want to optimize a function $f:\mathcal{X} \rightarrow \mathcal{Y}$, where both $\mathcal{X}$ and $\mathcal{Y}$ are finite (e.g., because of finite precision). Suppose furthermore that $d_m = \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\}$ is a time ordered set of m distinct points visited by an optimization algorithm (not counting times when feasible points are revisited), with $d_m^x = \{d_m^x(1), \ldots, d_m^x(m)\}$ being the successive (distinct) feasible points and $d_m^y = \{d_m^y(1), \ldots, d_m^y(m)\}$ being the corresponding objective function values. Then Theorem 1 of Wolpert and Macready (1997) states that if a_1 and a_2 are two optimization algorithms, then

$$\sum_f \mathrm{P}\{d_m^y | f, m, a_1\} = \sum_f \mathrm{P}\{d_m^y | f, m, a_2\}.$$

In other words, the average probability of observing a particular value of d_m^y (averaged over all possible objective functions f) does not depend on the choice of optimization algorithm. This result immediately implies that for any measure of algorithm performance that is a function of d_m^y, the average performance of all optimization algorithms (averaged over all possible f) is the same. Consequently, for an optimization algorithm to do better than repeated enumeration, the underlying optimization problem needs to have some known structure that the algorithm can exploit. Ideas about how such "landscape" structure can be measured can be found in the literature on evolutionary computing, see for example Vassilev et al. (2000) and Reidys and Stadler (2002).

In contrast with the literature on deterministic optimization, which is dominated by methods designed to solve highly structured optimization problems (e.g., linear programs), simulation optimization problems frequently have very little structure that the optimization algorithm can exploit, and even if the underlying problem has a lot of structure, the user may not be aware of this fact. Moreover, since most simulation practitioners use simulation languages to implement their models, for a simulation optimization technique to find widespread use it is desirable that the method be suitable for incorporation in simulation languages. This requires the method to perform well on a broad class of problems having very different characteristics. Consequently, most random search methods are designed to exploit the structure that good points (i.e., points with low objective function values, see (1)) tend to be clustered together. Of course, if more structure is available (e.g., convexity), then this can be exploited to make the search more efficient by narrowing down the location of the global optimal solution(s) as the search progresses.

When a method that is designed to exploit the structure that good points tend to be clustered together has identified a relatively good feasible point (or collection of feasible points), it is sensible to focus the search near that point. Point-based methods do this by focusing the search in a neighborhood of the current point (see Section 3), set-based methods do it by focusing the search in a set containing the current point, and population-based methods focus the search on points that can be generated by manipulating the current collection of points. Additionally, for good performance, these methods must have a mechanism for finding attractive points quickly. This generally translates into ensuring that the method can move rapidly around within the feasible region. A mechanism that accomplishes this needs to be incorporated in the sampling strategy used by the algorithm, regardless of whether it is point-based, set-based, or population-based.

Achieving the goals described in the previous paragraph depends on the set of feasible points under consideration by the sampling strategy $\mathcal{S}_n$ in each iteration n (including the size of this set), and also on how this set is searched. For point-based methods, this translates into the choice of the neighborhoods $N(\theta)$ and the decision about how these neighborhoods are searched. Based on the discussion in the previous paragraph, the neighborhood $N(\theta)$ of a feasible point $\theta \in \Theta$ should include points close to θ (to allow the method to focus the search near θ if desired) and also allow for rapid movements within the feasible region Θ (this would generally involve including some points that are far from θ in $N(\theta)$). Since the choice of the neighborhood structure $N = \{N(\theta)\colon\ \theta \in \Theta\}$ can have a substantial effect on the performance of the random search method under consideration, it is important to try to capture the essential features of the underlying optimization problem (1) in this choice, especially with respect to deciding what points should be considered to be close to each other.

With respect to the size of the neighborhoods, at one extreme one can use very large neighborhoods (e.g., $N(\theta) = \Theta \setminus \{\theta\}$ for all $\theta \in \Theta$), while at the other extreme the neighborhoods can be very localized around the current point (e.g., $N(\theta) = \Theta \cap \{\theta' \in \mathbb{R}^d\colon\ \|\theta' - \theta\| \leqslant 1\}$ when $\Theta \subset \mathbb{R}^d$). Clearly, large neighborhoods allow rapid movements within the feasible region and facilitate achieving global convergence, but they can be less effective than small neighborhoods in focusing the search around desirable points. On the other hand, small neighborhoods can lead to good performance if the algorithm is started in a good point, but can perform poorly if the feasible region Θ is large and many steps are required to move from the starting point(s) $\theta_1^{(1)}, \ldots, \theta_1^{(M_1)}$ to the global optimal solutions (e.g., if Θ is the set of integers, $N(\theta) = \{\theta - 1, \theta + 1\}$ for all $\theta \in \Theta$, $M_1 = 1$, and $\inf_{\theta^* \in \Theta^*} |\theta_n^{(1)} - \theta^*|$ is large).

When it comes to deciding how the neighborhoods $N(\theta)$ should be searched, at one extreme one could sample one point from the neighborhood of the current point and compare it with the current point, while at the other extreme one could search for the best point in the neighborhood and compare that point with the current point. Procedures that search very large neighborhoods constitute an active area of research in the combinatorial optimization community, see Ahuja et al. (2002) for a recent review. However, doing this effectively requires the exploitation of some underlying structure that the optimization problem has. Since such structure is often either not present or unknown in simulation optimization problems, this approach does not appear to be promising for developing simulation optimization algorithms that are suitable for widespread use, for example through being incorporated in a simulation language. Hence, the use of large neighborhood structures for simulation optimization would typically be accompanied by a sampling mechanism over the neighborhoods, rather than an optimization approach.

On the other hand, when the neighborhoods are relatively small in size, then it is possible to either sample from them or optimize over them. In the latter case, there is an opportunity to combine random search methods with ranking-and-selection methods, see Chapter 17 of this volume for a review. However, care needs to be taken to avoid expending too much computer time on optimizing over neighborhoods early in the search when the neighborhoods may not include any desirable points, or on choosing between several points exhibiting very similar performance (except perhaps near the conclusion of the search). Other opportunities for combining random search methods and ranking-and-selection techniques in the simulation optimization setting include using ranking and selection to ensure that the method will only change the estimate of the optimal solution or focus the search in a different subset of the feasible region Θ if there is strong evidence that the change is in fact an improvement over the status quo, and for "clean up" at the end of the search (i.e., to ensure that the final estimate of the optimal solution is in fact the best point visited by the algorithm with high probability). These issues are discussed by Pichitlamken and Nelson (2003) and Boesel et al. (2003), respectively.

The discussion above suggests that it is desirable for a random search method to incorporate both global and local search components, and to maintain the right balance between the two as the search progresses. The global search component is useful for quickly identifying desirable subsets of the feasible region Θ (with small objective function values). Once such subsets have been found, then it is beneficial to search locally within these areas to identify improved solutions. More details about how both global and local search can be included in random search methods, and how appropriate balance can be maintained between the two, are discussed by Prudius and Andradóttir (2004, 2006a). A related issue involves deciding on the appropriate balance between the number of points sampled by the algorithm and how much effort is spent on estimating the objective function value at each sampled point, see for example Yakowitz et al. (2000).

Note that if a random search method is guaranteed to be globally convergent with probability one, then this method does include a mechanism for escaping from feasible points that are locally, but not globally optimal (in other words, such a mechanism should be included in the method). Another desirable feature a random search algorithm may have (in addition to almost sure and global convergence, the ability to focus the search in desirable regions, and the ability to move quickly within Θ) is the ability to focus the search in areas of the feasible region that either appear to be desirable (i.e., have relatively low objective function values) or have not been sampled much previously. This is accomplished in tabu search by excluding certain points from consideration (e.g., points that have been visited recently by the algorithm), and in COMPASS by using a neighborhood structure that focuses the search in the region whose elements are closer to the best point seen so far than they are to other points that the algorithm has already visited.

Furthermore, it is desirable for random search methods to be highly adaptive in order to take as much advantage as possible of the information that it gathers throughout the search to guide the search. In our generic random search algorithm for simulation optimization given in Section 2, we incorporate this by allowing the sampling strategies $\{\mathcal{S}_n\}$ to adapt to all information collected by the algorithm. For example, the neighborhoods used by point-based methods can be chosen adaptively, and the same is true of how these neighborhoods are searched. The only restrictions on this are that adapting to the information collected by the algorithm should in fact improve performance (e.g., the algorithm should not be misled by the noise in the estimated objective function values), this should be done in a way that maintains the guarantee of almost sure convergence, and it is preferable for the algorithm not to require the intervention of the user as it adapts to the available information (e.g., for the method to be suitable for inclusion in a simulation language).

One way of adapting the search to the information collected over time by the random search method is to use the aggregated function estimates $f_n(\theta)$ to guide the search, rather than simply for computing the sequence $\{\theta_n^*\}$ of estimates of the optimal solution, see Equations (2) and (3). The most commonly used alternative to this approach is to use only the simulation results obtained in the current iteration of the algorithm to decide on the next action taken by the algorithm (this is often done to ensure that the progression of the algorithm can be modeled as a Markov chain, which is useful for proving results about the behavior of the algorithm). Since for all $\theta \in \Theta$, the aggregated estimate $f_n(\theta)$ of the objective function value $f(\theta)$ is more precise than an estimate of $f(\theta)$ obtained in a single replication (as long as the feasible point θ has been visited more than once by the random search method), and since the presence of noise in the estimated objective function values makes it more difficult to solve the simulation optimization problem (1), it is reasonable to expect that using the aggregated estimates $f_n(\theta)$ to guide the search will lead to better empirical behavior than using only the most recently obtained estimates. However, this is not always the case, because there are situations where the random search algorithm being used can benefit from the added simulation noise (in addition to any randomness inherent in the algorithm itself). For example, if the random search method being used is a descent method (so that it does not have a mechanism for escaping from solutions that are locally, but not globally optimal) and if the underlying optimization problem has locally optimal solutions that are not globally optimal, then there may be instances where the algorithm will converge to a locally optimal solution when the aggregated estimates $f_n(\theta)$ are used, but it will converge to a globally optimal solution when the most recently available estimates of $f(\theta)$ are used (because the noise in the most recently available estimates allows the algorithm to escape from locally optimal solutions). For more discussion on the use of aggregated objective function estimates in simulation optimization, the reader is referred to Prudius and Andradóttir (2005, 2006b).

The discussion in this section is concerned with features a random search method should have to exhibit attractive empirical performance. More details about the design of random search methods are given by Prudius and Andradóttir (2006a). Finally, note that it would of course be desirable to be able to prove results about how fast a random search method is likely to converge in practice. This would include both asymptotic rate-of-convergence and attraction results, see Andradóttir (1999). Similarly, results about the finite-horizon behavior of random search methods, such as determining how many iterations are required to reach a certain level of performance with a specified probability, would also be valuable. However, such rate-of-convergence results are only of practical value if the conditions under which they hold do not unduly limit the flexibility in the algorithm design (so that an algorithm that satisfies these conditions is not likely to perform worse in practice than the algorithms with the best empirical performance on the class of optimization problems under consideration).

5 Summary

In this chapter, we have outlined the features a random search method should have to both be provably convergent and also exhibit attractive empirical performance. We showed that almost sure and global convergence is not difficult to achieve, and as a result it seems reasonable to focus on random search methods that have this property. We have also discussed other desirable characteristics of random search methods, including the ability to move rapidly within the feasible region to identify desirable areas worthy of further investigation, to focus the search in desirable subsets of the feasible region that have not been extensively explored before, and to adapt to information collected by the method about the optimization problem at hand as it becomes available. The application of these principles towards the design of effective random search methods for simulation optimization is presently an active and worthy research area.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. DMI-0000135, Grant No. DMI-0217860 and Grant No. DMI-0400260. The author thanks the editors of this volume, Shane G. Henderson and Barry L. Nelson, for guidance about the scope of this chapter and other helpful comments. The author also appreciates the comments provided by an anonymous referee.

References

Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P. (2002). A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics* 23, 75–102.

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

Alrefaei, M.H., Andradóttir, S. (2001). A modification of the stochastic ruler method for discrete stochastic optimization. *European Journal of Operational Research* 133, 160–182.

Alrefaei, M.H., Andradóttir, S. (2005). Discrete stochastic optimization using variants of the stochastic ruler method. *Naval Research Logistics* 52, 344–360.

Andradóttir, S. (1995). A method for discrete stochastic optimization. *Management Science* 41, 1946–1961.

Andradóttir, S. (1996). A global search method for discrete stochastic optimization. *SIAM Journal on Optimization* 6, 513–530.

Andradóttir, S. (1999). Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 9, 349–380.

Andradóttir, S. (2000). Rate of convergence of random search methods for discrete optimization using steady-state simulation. Preprint.

Andradóttir, S. (2006). Simulation optimization with countably infinite feasible regions: Efficiency and convergence. *ACM Transactions on Modelling and Computer Simulation*, in press.

Benveniste, A., Métivier, M., Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, Berlin.

Boesel, J., Nelson, B.L., Kim, S.-H. (2003). Using ranking and selection to "clean up" after simulation optimization. *Operations Research* 51, 814–825.

Chen, H., Schmeiser, B.W. (2001). Stochastic root finding via retrospective approximation. *IIE Transactions* 33, 259–275.

Dembo, A., Zeitouni, O. (1993). *Large Deviations Techniques and Applications*. Jones and Bartlett, Boston, MA.

Fox, B.L., Heine, G.W. (1996). Probabilistic search with overrides. *The Annals of Applied Probability* 6, 1087–1094.

Gelfand, S.B., Mitter, S.K. (1989). Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications* 62, 49–62.

Gong, W.-B., Ho, Y.-C., Zhai, W. (1999). Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization* 10, 384–404.

Gutjahr, W.J., Pflug, G.Ch. (1996). Simulated annealing for noisy cost functions. *Journal of Global Optimization* 8, 1–13.

Healy, K.J., Schruben, L.W. (1991). Retrospective simulation response optimization. In: Nelson, B.L., Kelton, W.D., Clark, G.M. (Eds.), *Proceedings of the 1991 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 901–906.

Homem-de-Mello, T. (2003). Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 13, 108–133.

Hong, L.J., Nelson, B.L. (2005). Discrete optimization via simulation using COMPASS. *Operations Research* 54, 115–129.

Kiefer, J., Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* 23, 462–466.

Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T. (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12, 479–502.

Kushner, H.J., Clark, D.S. (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, NY.

Kushner, H.J., Yin, G.G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, NY.

Norkin, V.I., Ermoliev, Y.M., Ruszczyński, A. (1998a). On optimal allocation of indivisibles under uncertainty. *Operations Research* 46, 381–395.

Norkin, V.I., Pflug, G.Ch., Ruszczyński, A. (1998b). A branch and bound method for stochastic global optimization. *Mathematical Programming A* 83, 425–450.

Pflug, G.Ch. (1996). *Optimization of Stochastic Models: The Interface between Simulation and Optimization*. Kluwer, Boston, MA.

Pichitlamken, J., Nelson, B.L. (2003). A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 13, 155–179.

Prudius, A.A., Andradóttir, S. (2004). Simulation optimization using balanced explorative and exploitative search. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 545–549.

Prudius, A.A., Andradóttir, S. (2005). Two simulated annealing algorithms for noisy objective functions. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 797–802.

Prudius, A.A., Andradóttir, S. (2006a). Balanced explorative and exploitative search with estimation. Working paper.

Prudius, A.A., Andradóttir, S. (2006b). An averaging framework for simulation optimization with applications to simulated annealing. Working paper.

Reidys, C.M., Stadler, P.F. (2002). Combinatorial landscapes. *SIAM Review* 44, 3–54.

Robbins, H., Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400–407.

Robinson, S.M. (1996). Analysis of sample-path optimization. *Mathematics of Operations Research* 21, 513–528.

Rubinstein, R.Y., Shapiro, A. (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, Chichester, England.

Shapiro, A., Wardi, Y. (1996). Convergence analysis of gradient descent stochastic algorithms. *Journal of Optimization Theory and Applications* 91, 439–454.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Hoboken, NJ.

Vassilev, V.K., Fogarty, T.C., Miller, J.F. (2000). Information characteristics and the structure of landscapes. *Evolutionary Computation* 8, 31–60.

Wolpert, D.H., Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 67–82.

Yakowitz, S., L'Ecuyer, P., Vázquez-Abad, F. (2000). Global stochastic optimization with low-dispersion point sets. *Operations Research* 48, 939–950.

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