

Chapter 12

Model-Based Stochastic Search Methods

Jiaqiao Hu

Abstract Model-based algorithms are a class of stochastic search methods that have successfully addressed some hard deterministic optimization problems. However, their application to simulation optimization is relatively undeveloped. This chapter reviews the basic structure of model-based algorithms, describes some recently developed frameworks and approaches to the design and analysis of a class of model-based algorithms, and discusses their extensions to simulation optimization.

12.1 Introduction

In this chapter, we address the problem of finding the values of a set of design parameters that attain the optimum of an objective function, written in the following general form:

$$x^* \in \arg \max_{x \in \Theta} h(x), \tag{12.1}$$

where Θ is the feasible region, which is often a non-empty compact subset of \mathbb{R}^d , and $h : \Theta \rightarrow \mathbb{R}$ is a bounded, deterministic objective function. Stochastic optimization often refers to the case where the objective function h itself takes the form of an expectation

$$h(x) = E[H(x, \xi)],$$

where ξ is a random variable representing the stochastic uncertainty of the system, which for example could be a sample path, and only estimates of the “noisy” sample performance H are available. In this chapter, simulation optimization refers to the special case of stochastic optimization when the sample performance H is assessed in a path-wise manner through computer simulation. In contrast with

J. Hu (✉)
Stony Brook University, Stony Brook, NY, USA
e-mail: jqhu@ams.sunysb.edu

deterministic optimization, simulation optimization problems are characterized by random uncertainties in their performance measures, and the objective functions are often highly nonlinear with respect to the underlying decision variables.

When the objective function is differentiable, a well-known class of methods for solving simulation optimization problems is stochastic approximation [4, 5, 28, 29], the topic of Chaps. 6 and 7. These methods mimic the classical gradient algorithms in deterministic optimization and rely on the estimation of the gradient of the objective function. Because they are gradient-based, these methods generally find local optimal solutions. Sample average approximation [14, 27] is another approach that often exploits structural information such as differentiability, linearity or convexity. The main idea of this approach is to transform a simulation optimization problem into a deterministic one by expending a large amount of simulation effort on each visited solution to obtain a precise estimate of the objective function value. The resulting deterministic counterpart of the original stochastic problem is then solved by a deterministic optimization algorithm.

Due to the limited structural knowledge for general simulation optimization problems, it is natural to adapt random search methods from deterministic optimization to these types of problems. A random search method is usually recursive and approximates the optimal solution by a sequence of iterates (e.g., candidate solutions, promising subsets, probability models) generated according to a specified random mechanism. These methods differ primarily in the type of iterates an algorithm produces and in the choices of the random strategy used to generate the iterates. Because random search methods typically only rely on the objective function values rather than structural information such as convexity and differentiability, they are robust, easy to implement, and can be applied to a broad range of optimization problems with very different characteristics.

From an algorithmic point of view, a random search algorithm can be broadly classified as being either *instance-based* or *model-based* [44]. In instance-based algorithms, an iterate corresponds to a single or a subset of candidate solution(s), and the construction of new iterates depends explicitly on iterates generated in previous iterations. These include both population-based algorithms such as genetic algorithms [10], which produce a collection of candidate solutions at each iteration, and methods like nested partitions [36] that are based on repeatedly identifying a promising subset of the feasible region as the search proceeds. Currently, random search-based simulation optimization is primarily dominated by instance-based methods, with numerous algorithms proposed in the literature and their behaviors relatively well studied and understood. In addition to the two aforementioned methods, some typical examples include response surface methods [3], simulated annealing [26], tabu search [9], stochastic ruler methods [38], stochastic comparison [11], and the COMPASS algorithm [15]; see Chaps. 2, 10, and 11, as well as [2] for a review of this class of methods.

While the field of simulation optimization has significantly evolved in terms of instance-based algorithms, less attention has been devoted to the study of model-based methods. Unlike instance-based algorithms, the model-based methods are based on sampling candidate solutions from an intermediate (usually parameterized) probability distribution over the feasible region. The idea is to iteratively modify the

distribution model based on the sampled solutions to bias the search towards regions containing high quality solutions. Therefore, each iterate in a model-based algorithm corresponds to a distribution function, which can be abstractly viewed as a “model” characterizing the promising regions of the solution space. Examples of this type of method include ant colony optimization (ACO) [8], estimation of distribution algorithms (EDAs) [30], annealing adaptive search (AAS) [32, 39], probability collectives (PCs) [37], and the cross-entropy (CE) method [35]. These algorithms have successfully solved some hard nonlinear, non-differentiable problems and are becoming increasingly prominent in deterministic optimization.

Unfortunately, because model-based algorithms are generally heuristic in nature, very few of them have found their way into the simulation optimization literature. In this chapter, we aim to stimulate new research ideas in this area by presenting some recently developed frameworks and approaches for the design and analysis of a class of model-based algorithms. As a starting point, we introduce these developments in the context of deterministic optimization with the objective function h being viewed as a “black box” that returns the exact function value for a specified solution. We then proceed by providing specific examples and algorithms to illustrate the key modifications needed, as well as issues and challenges involved in extending model-based algorithms to general simulation optimization settings.

12.2 A Brief Review of Model-Based Methods

The basic idea of model-based algorithms is to use a sequence of probability distribution functions to successively characterize the promising regions of the solution space. So in a model-based algorithm, it is the probability distribution rather than candidate solutions (as in an instance-based algorithm) that is propagated from one iteration to another. Most algorithms that fall into this category are iterative methods involving the basic steps of the framework below.

Basic Model-Based Optimization Framework

- Step 1. Randomly generate a population of candidate solutions $X^{(k)}$ from g_k , where g_k is the probability distribution on Θ at the k th iteration.
 - Step 2. Evaluate/estimate the performance of generated candidate solutions.
 - Step 3. Update g_k based on the performance of the sampled solutions in $X^{(k)}$ to construct a new distribution g_{k+1} ; increase k by 1 and reiterate from Step 1.
-

Note that since a population of candidate solutions is generated at each step, such algorithms retain the primary strengths of population-based approaches such as genetic algorithms, while providing more flexibility and robustness in exploring the entire feasible region (i.e., via sampling from g_k). Clearly, a major algorithmic question in model-based algorithms is how the update in Step 3 is carried out. Important practical implementation issues are the efficient construction/representation of the

probability distribution g_k and efficient sampling from g_k over the feasible region Θ . How these issues are addressed is what differentiates particular approaches.

Pure random search [39] (see also Chap. 11) can be viewed as one of the simplest model-based methods, in the sense that g_k is taken to be a fixed uniform distribution over Θ (not updated at all). The algorithm proceeds by generating a sequence of uniformly distributed random points over the feasible region and using the best candidate solution obtained thus far as an estimate of the optimal solution. Although the idea behind the algorithm is simple, the complexity of the algorithm increases exponentially with the dimension of the solution space.

A nontrivial improvement of pure random search is the annealing adaptive search (AAS) algorithm [32, 39], described in detail in Chap. 11, which replaces the fixed uniform distribution in a pure random search method by a sequence of Boltzmann distributions parameterized by iteration-varying temperatures T_k . These Boltzmann distributions are constructed in such a way that as the temperature decreases to zero, the sequence of distributions will become more concentrated on the set of optimal solutions. So solutions sampled from Boltzmann distributions with small values of T_k will be close to the optimum with high probability. For the class of Lipschitz optimization problems, it has been shown that the expected number of iterations required by AAS to achieve a given level of precision increases at most linearly in the problem dimension [32, 39]. However, the idealized AAS is not readily implementable in practice for solving optimization problems, because the problem of sampling exactly from a given Boltzmann distribution is known to be very difficult. This implementation issue has been addressed in a number of papers (see e.g., [39, 40] and the references therein), and the basic idea is to use Markov chain Monte Carlo techniques to sample asymptotically from the Boltzmann distribution.

The cross-entropy (CE) method [35] was originally motivated by the problem of estimating probabilities of rare events in simulation [33], before it was discovered that it could be modified to solving deterministic optimization problems. The key idea of CE is to use a family of parameterized distributions to successively approximate an optimal (importance sampling) distribution concentrated only on the set of (near) optimal solutions, which is carried out by iteratively estimating the optimal parameter that minimizes the Kullback–Leibler (KL) divergence between the parameterized distribution and the target optimal distribution. Since its introduction, there have been extensive developments regarding implementation and practical applications of CE (see [35]). Those that are particularly relevant to our discussion include the adaptation of CE to handle stochastic network combinatorial optimization problems [34], the application of the method to solving buffer allocation problems in a simulation-based environment [1], and the work of [31], which uses CE as a direct policy search approach to solving stochastic dynamic programming problems. The literature analyzing the convergence properties of the CE method is relatively sparse, with most of the existing results limited to specific settings; see, e.g., [13] for a convergence proof of a variational version of CE in the context of estimation of rare event probabilities, and [7] for probability one convergence proofs of CE for discrete optimization problems. General convergence

and asymptotic rate results for CE were recently obtained in [20, 22] by relating the algorithm to recursions of stochastic approximation type (see Sect. 12.4).

Two other well-established model-based methods are the ant colony optimization (ACO) [8] and the estimation of distribution algorithms (EDAs) [30]. ACO was inspired by the behavior of a colony of biological ants, which are capable of solving shortest path problems by exchanging their local information indirectly through a certain chemical substance called pheromone. ACO is frequently applied to solving combinatorial problems, e.g., the traveling salesman problem. In such problems, the generation of candidate solutions (tours) is based on the series of random moves performed by a collection of artificial ants called agents, which are controlled by an empirical distribution constructed based on each agent's local experience. ACO has been formally extended to stochastic settings for solving stochastic combinatorial optimization problems. One such extension is called the stochastic ant colony optimization (S-ACO) [12], which uses Monte-Carlo sampling to estimate the expectation involved in evaluating the objective function. The probability one convergence of S-ACO to the global optimal solution has been established in [12].

EDAs inherit the spirit of genetic algorithms (GAs), but eliminate the crossover and mutation operators to avoid the disruption of partial solutions. In EDAs, a new population of candidate solutions are generated according to the probability distribution induced or estimated from the promising solutions selected from the previous generation. Unlike CE, EDAs often take into account the interrelations between the underlying decision variables needed to represent the individual candidate solutions. At each iteration of the algorithm, a high-dimensional probabilistic model that better represents the interdependencies between the decision variables is induced; this step constitutes the most crucial and difficult part of the method. We refer the reader to [30] for a review of the way in which different probabilistic models are used as EDA instantiations. A proof of convergence of a class of EDAs, under the idealized infinite population assumption, can be found in [41].

There are many other model-based algorithms proposed for optimization. Some notable examples include probability collectives (PCs) [37], particle swarm optimization [25], the particle filtering approach [42], and gradient-guided stochastic search [43]. A complete description of all of them is outside the scope of this chapter. Instead, we will focus our discussions on some recently developed approaches that allow us to arrive at a systematic framework to study a class of model-based algorithms in a uniform manner. We review specific algorithms under the framework, present their asymptotic convergence properties, and discuss their adaptations to simulation optimization.

12.3 A Model Reference Optimization Framework

We begin with a description of the model reference adaptive search (MRAS) method introduced by Hu et al. [18], which will serve as a basis for subsequent discussions.

Model-based methods construct a sequence of distributions $\{g_k\}$ with some desired convergence properties (ideally including $g_k \rightarrow g^*$ as $k \rightarrow \infty$, with g^*

being a limiting distribution assigning all of its probability mass to the set of optimal solutions). Some common approaches for constructing such a sequence include: (a) various proportional selection schemes—used in ACOs, EDAs, and PCs; (b) Boltzmann selection schemes—used in AAS and the continuous EDA algorithm in [6]; and (c) optimal importance sampling measure—primarily used in the CE method.

However, in all cases, the obvious difficulties are that the sequence $\{g_k\}$ often depends on h , which may not be available in any explicit form, and that the problem of sampling exactly from even a known (but arbitrary) distribution g_k is in general intractable. In [18], a general approach called model reference adaptive search (MRAS) is proposed, where these difficulties are circumvented by sampling from a *surrogate distribution* that approximates g_k . The idea of MRAS is to specify a family of parameterized distributions $\{f_\varphi(\cdot), \varphi \in \Phi\}$ (with Φ being the parameter space) and then project g_k onto the family to obtain a sequence of sampling distributions $\{f_{\varphi_k}\}$ with desired convergence properties. The projection is carried out by finding an optimal parameter φ_k that minimizes the Kullback–Leibler (KL) divergence between g_k and the parameterized family $\{f_\varphi(\cdot), \varphi \in \Phi\}$ (cf. also [35]), i.e.,

$$\varphi_k = \arg \min_{\varphi \in \Phi} \mathcal{D}(g_k, f_\varphi),$$

where

$$\mathcal{D}(g_k, f_\varphi) := \int_{\Theta} \ln \frac{g_k(x)}{f_\varphi(x)} g_k(x) \nu(dx) = E_{g_k} \left[\ln \frac{g_k(X)}{f_\varphi(X)} \right], \quad (12.2)$$

ν is the Lebesgue/discrete measure on Θ , and $E_g[\cdot]$ is the expectation taken with respect to the density/mass functions g . The hope is that the parameterized family is specified with some structure so that once the parameter is determined, sampling from each of these distributions should be a relatively easy task. Moreover, the task of updating the entire distribution can be simplified to the task of updating its associated parameters, and the sequence $\{g_k\}$, henceforth referred to as *reference distributions*, is only used implicitly to guide the parameter updating procedure.

Note that there are other ways of constructing surrogate distributions. For example, in AAS, Markov chain Monte Carlo (MCMC) techniques are frequently used to approximate the target Boltzmann distribution at each iteration [32], and in traditional EDAs, empirical distribution models are directly constructed to generate new candidate solutions [30, 41]. However, these algorithms can also be accommodated within the above model reference framework by projecting the target distributions onto a family of parameterized distributions. Specifically, under such a framework, the three example sequences of reference distributions $\{g_k\}$ can be expressed in the following recursive forms:

(a) proportional selection scheme: $g_k(x) = \frac{S(h(x))g_{k-1}(x)}{E_{g_{k-1}}[S(h(X))]};$

(b) Boltzmann selection:

$$g_k(x) = \frac{e^{h(x)/T_k}}{\int_{x \in \Theta} e^{h(x)/T_k} v(dx)} = \frac{e^{h(x)(\frac{1}{T_k} - \frac{1}{T_{k-1}})} g_{k-1}(x)}{\mathbb{E}_{g_{k-1}} [e^{h(X)(\frac{1}{T_k} - \frac{1}{T_{k-1}})}]} := \frac{S(h(x))g_{k-1}(x)}{\mathbb{E}_{g_{k-1}} [S(h(X))]},$$

with $\{T_k\}$ being a sequence of parameters determined by an annealing schedule;

(c) importance sampling measure: $g_k(x) = \frac{S(h(x))f_{\phi_k}(x)}{\mathbb{E}_{\phi_k} [S(h(X))]},$

where $S(\cdot)$ is a non-negative increasing (possibly iteration-varying) function, and $\mathbb{E}_{\phi}[\cdot]$ is the expectation taken with respect to f_{ϕ} . The algorithm instantiation considered in [18] uses the recursive procedure corresponding to case (a) to construct the g_k sequence. This form of reference distributions weights g_k by the value of the performance measure by giving more mass to solutions with good performance. The resulting g_{k+1} has the property that it improves the expected performance of g_k , since

$$\mathbb{E}_{g_{k+1}} [S(h(X))] = \frac{\mathbb{E}_{g_k} [S^2(h(X))]}{\mathbb{E}_{g_k} [S(h(X))]} \geq \mathbb{E}_{g_k} [S(h(X))].$$

This property ensure the convergence of the sequence $\{g_k\}$ to a degenerate distribution concentrated on the set of optimal solutions. The general structure of the MRAS method is outlined below.

Basic Model Reference Adaptive Search (MRAS) Optimization Framework

Step 0. Select a parameterized family $\{f_{\phi}\}$ and the $\{g_k\}$ sequence with desired convergence properties.

Step 1. Given ϕ_k , generate N candidate solutions X_k^1, \dots, X_k^N by sampling from f_{ϕ_k} .

Step 2. Update the parameter ϕ_{k+1} based on the sampled solutions by minimizing the KL-divergence

$$\phi_{k+1} = \arg \min_{\phi \in \Phi} \mathcal{D}(g_{k+1}, f_{\phi});$$

set $k \leftarrow k + 1$ and reiterate from Step 1 until a stopping criterion is satisfied.

In some model-based algorithms such as CE and EDAs, there is often an additional selection step embedded in the above procedure. The idea is to concentrate the computational effort on the set of promising solutions by using only a portion of the samples—the set of “elite” samples—to update the probability model at Step 2. Also note that in general, the expectation involved in the KL-divergence

(cf. (12.2)) cannot be evaluated exactly. So in practice, the objective function in the minimization problem at Step 2 is often replaced by its sample average approximation. This leads to an estimator of φ_{k+1} that is biased for any finite sample size N . This bias issue will be discussed later in Sect. 12.5.

12.3.1 Convergence Result

For distributions in the natural exponential family (NEF), the minimization in Step 2 of the basic MRAS framework can be carried out in analytical form, which makes the method easy to implement efficiently.

Definition 12.1. A parameterized family $\{f_\varphi(\cdot), \varphi \in \Phi \subseteq \mathbb{R}^m\}$ on Θ is called a natural exponential family if there exist mappings $\Gamma: \mathbb{R}^d \rightarrow \mathbb{R}^m$ and $K: \mathbb{R}^m \rightarrow \mathbb{R}$ such that $f_\varphi(x) = \exp(\varphi^T \Gamma(x) - K(\varphi))$, where $K(\varphi) = \ln \int_\Theta \exp(\varphi^T \Gamma(x)) \nu(dx)$ is a normalization constant.

The NEF has many interesting properties. For the purpose of our discussion, we recall that $K(\varphi)$ is strictly convex in the interior of Φ and the mean parameter function defined by

$$m(\varphi) := \nabla_\varphi K(\varphi) = E_\varphi[\Gamma(X)] \quad (12.3)$$

is a one-to-one invertible mapping of φ . Intuitively, $m(\varphi)$ is essentially a transformed version of the sufficient statistic $\Gamma(x)$, whose value contains all information necessary in estimating the parameter φ . For example, in the univariate normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2 , it can be seen that $\Gamma(x) = (x, x^2)^T$ and $\varphi = (\mu/\sigma^2, -1/(2\sigma^2))^T$. Thus, given the value of $m(\varphi)$, the equation $m(\varphi) = E_\varphi[\Gamma(X)] = (\mu, \sigma^2 + \mu^2)^T$ can be uniquely solved for μ and σ^2 .

When NEFs are used in the framework with the sample size N adaptively increasing, the convergence result for the instantiation of MRAS considered in [18] takes the form

$$\lim_{k \rightarrow \infty} m(\varphi_k) = \Gamma(x^*) \text{ w.p.1. as } k \rightarrow \infty.$$

Since $m(\varphi)$ is one-to-one, this shows that the sequence of sampling distributions $\{f_{\varphi_k}\}$ will converge to a limiting distribution f_{φ^*} for g_k sequences of proportional selection scheme type. In addition, it has been argued in [18] that in many special cases of interest, the limiting distribution turns out to be a degenerate distribution on the set of optimal solutions. For example, when multivariate normal distributions with mean vector μ_k and covariance matrix Σ_k are used as parameterized distributions, the convergence result translates to $\lim_{k \rightarrow \infty} \mu_k = x^*$ and $\lim_{k \rightarrow \infty} \Sigma_k = \mathbf{0}_{n \times n}$ w.p.1, where $\mathbf{0}_{n \times n}$ represents an n -by- n zero matrix.

12.3.2 Simulation Optimization

In [19], the MRAS method has been generalized to stochastic settings where the objective function $h(x)$ in (12.1) can only be estimated, e.g., via a simulation model or real-time observations. We begin by providing a high-level description of the method.

Stochastic MRAS Framework for Simulation Optimization

- Step 0. Select a parameterized family $\{f_\varphi\}$ and an idealized $\{g_k\}$ sequence with desired convergence properties.
- Step 1. Given φ_k , generate N candidate solutions $\Lambda_k := \{X_k^1, \dots, X_k^N\}$ by sampling from f_{φ_k} .
- Step 2. Take M_k simulation observations for each $x \in \Lambda_k$ and estimate $h(x)$ by the sample average $\bar{H}(x) := \frac{1}{M_k} \sum_{j=1}^{M_k} H(x, \xi_j)$, where ξ_j is the simulation noise in the j th replication run.
- Step 3. Update the parameter φ_{k+1} based on the sampled solutions by minimizing the KL-divergence

$$\varphi_{k+1} = \arg \min_{\varphi \in \Phi} \mathcal{D}(\tilde{g}_{k+1}, f_\varphi);$$

set $k \leftarrow k + 1$ and reiterate from Step 1 until a stopping criterion is satisfied.

The basic structure of the stochastic MRAS method (SMRAS) is similar to that of MRAS for deterministic optimization, the main addition being the requirement of an additional performance estimation step (i.e., Step 2 above). So in addition to the sample size N used in MRAS, we also need to specify the number of simulation replications to be allocated to each sampled candidate solution. At each iteration, the sample mean based on M_k observations is used to estimate the true performance $h(x)$. Another modification from the original MRAS method occurs at Step 3, where a distribution \tilde{g}_k is used as an approximation of the idealized g_k distribution in minimizing the KL-divergence. The sequence $\{\tilde{g}_k\}$ is obtained by replacing the true objective function $h(x)$ in the construction of g_k with its sample average approximation $\bar{H}(x)$.

There are two general types of approaches for handling the simulation noise involved in evaluating the objective function. One type of approaches (e.g., sample average approximation) relies on highly precise estimates of the objective function values by allocating a significant amount of simulation replications to each visited solution. The other type of approaches (e.g., stochastic approximation) does not require precise performance estimates, but generally involves some forms of averaging, so that the estimation error due to simulation noise will automatically cancel out over the course of a large number of iterations. The SMRAS method falls in between these two types of approaches and requires increasingly precise

estimates of the objective function values as the search progresses. In particular, for the proportional selection scheme type $\{g_k\}$ sequence, conditions on $\{M_k\}$ that ensure the convergence of the method are provided in [19]. It has been shown that when the simulation noises (not necessarily i.i.d.) satisfy the large deviation principle, M_k is required to grow linearly with the number of iterations k .

As compared with the MRAS method for deterministic optimization, the updating formula for φ_{k+1} (or equivalently $m(\varphi_{k+1})$, due to the invertibility of the mapping m) in SMRAS has an additional simulation error term. For general reference distribution sequences, a large deviation approach similar to that of [19] can be applied to determine the conditions on $\{M_k\}$ under which the error term converges to zero with probability one. Again, when NEFs are used in the SMRAS framework and $\{g_k\}$ is constructed using the proportional selection scheme, essentially the same convergence result as stated in Sect. 12.3.1, i.e.,

$$\lim_{k \rightarrow \infty} m(\varphi_k) = \Gamma(x^*) \text{ w.p.1. as } k \rightarrow \infty,$$

has been established under some appropriate conditions on the algorithm input parameters, the sample size N , and the number of simulation replications M_k ; see [19].

12.4 A Stochastic Approximation Framework

In this section, we present a stochastic approximation (cf. Chap.6) framework to analyze a general class of model-based algorithms. We show that a slight modification of the MRAS method introduced in Sect. 12.3 will lead to an interesting connection between model-based algorithms and the well-known stochastic approximation method. This connection implies that for a general non-differentiable (deterministic) optimization problem, a model-based algorithm implicitly transforms the underlying problem into an equivalent stochastic optimization problem with smooth differentiable structures, and the algorithm itself can be viewed as a gradient-based recursion over a transformed parameter space for solving the equivalent smoothed problem. This interpretation of model-based algorithms not only explains why these algorithms work well for hard optimization problems with little structure, but also allows us to investigate their theoretical properties such as convergence and rate of convergence by using theory and tools from stochastic approximation.

The main idea of the stochastic approximation framework is to replace the reference distribution sequence $\{g_k\}$ in the original MRAS method by a sequence $\{\hat{g}_k\}$ of the form:

$$\hat{g}_{k+1}(x) = \alpha_k g_{k+1}(x) + (1 - \alpha_k) f_{\varphi_k}(x), \quad \alpha_k \in [0, 1] \forall k, \quad (12.4)$$

where f_{φ_k} is the sampling distribution obtained at the k th iteration of an algorithm and α_k is a mixture coefficient. Note that by taking \hat{g}_{k+1} as the weighted average of g_{k+1} and f_{φ_k} , it is “forced” to stay close to both distributions. Thus, if \hat{g}_{k+1} is used in place of g_{k+1} in minimizing the KL-divergence $\mathcal{D}(\hat{g}_{k+1}, f_{\varphi})$ at Step 2 of the MRAS method, the new sampling distribution $f_{\varphi_{k+1}}$ obtained will not deviate significantly from the current sampling distribution f_{φ_k} . Since an NEF distribution f_{φ} is uniquely characterized by its associated parameter φ , the above intuition can be formally stated in terms of the difference between the two successive mean parameter functions (cf. (12.3)) of the projected probability distributions [20, 22]:

$$m(\varphi_{k+1}) - m(\varphi_k) = -\alpha_k \nabla_{\varphi} \mathcal{D}(g_{k+1}, f_{\varphi})|_{\varphi=\varphi_k}. \quad (12.5)$$

Equation (12.5) explicitly brings out the updating direction of the parameter functions at each step, which is in the direction of the negative gradient of the iteration-varying objective function for the minimization problem $\min_{\varphi \in \Phi} \mathcal{D}(g_{k+1}, f_{\varphi}) \forall k$. This suggests that regardless of the type of decision variables involved in the original problem (12.1), algorithms conforming to the framework are essentially gradient-based recursions for solving a sequence of optimization problems on the parameter space Φ with smooth differentiable structures. In the special case of the CE method, i.e., when g_{k+1} in the right-hand-side of recursion (12.4) is replaced with $S(h(x))f_{\varphi_k}(x)/E_{\varphi_k}[S(h(X))]$, it can be seen that (12.5) becomes

$$m(\varphi_{k+1}) - m(\varphi_k) = \alpha_k \frac{E_{\varphi_k}[S(h(X))(\Gamma(X) - m(\varphi_k))]}{E_{\varphi_k}[S(h(X))]} = \alpha_k \nabla_{\varphi} \ln E_{\varphi}[S(h(X))]|_{\varphi=\varphi_k}. \quad (12.6)$$

So the updating direction is in the gradient of the objective function for the maximization problem $\max_{\varphi \in \Phi} \ln E_{\varphi}[S(h(X))]$. The optimal solution of this optimization problem is a parameter φ^* whose associated sampling distribution f_{φ^*} assigns maximum probability to the set of optimal solutions of (12.1).

Letting $\eta_k := m(\varphi_k)$ and using the invertibility of the mapping m , we can write (12.5) in the abstract form

$$\eta_{k+1} = \eta_k - \alpha_k L(\eta_k), \quad (12.7)$$

where $L(\eta_k)$ represents the gradient of the underlying (possibly iteration-varying) objective function at η_k , which, for the three example $\{g_k\}$ sequences discussed in Sect. 12.3, takes the general form

$$L(\eta_k) = \frac{E_{\varphi_k}[S(h(X))G(X, \eta_k)]}{E_{\varphi_k}[S(h(X))]} \quad (12.8)$$

for some appropriate function $G(x, \eta_k)$. In actual implementation, expectations are replaced by sample averages based on Monte Carlo sampling, (12.7) becomes stochastic approximation with direct gradient estimation:

$$\tilde{\eta}_{k+1} = \tilde{\eta}_k - \alpha_k \tilde{L}(\tilde{\eta}_k), \quad (12.9)$$

where \tilde{L} is an estimator for L based on sampled candidate solutions. The most straightforward estimator is

$$\tilde{L}(\tilde{\eta}_k) = \frac{N^{-1} \sum_{i=1}^N S(h(X_k^i)) G(X_k^i, \tilde{\eta}_k)}{N^{-1} \sum_{i=1}^N S(h(X_k^i))}.$$

Thus, it is clear that the rich body of tools and results from stochastic approximation [28, 29] can be incorporated into the framework to analyze model-based algorithms.

12.4.1 Convergence Results

Convergence of the CE Method

The convergence of the CE algorithm has been studied in [20, 22] by writing (12.9) in the form of a generalized Robbins–Monro algorithm in terms of the true gradient, a bias term, and a noise term caused by Monte Carlo random sampling, and then following a standard ordinary differential equation (ODE) argument (cf. e.g., [4, 5, 28]). Basically, it has been shown in [22] that the asymptotic behavior of CE is governed by the properties of a limit set of an underlying ODE. In addition, if the limit set consists purely of isolated equilibrium points of the ODE, then the sequence of $\{\tilde{\eta}_k\}$ generated by CE will converge to a unique limiting point η^* w.p.1. Under such a condition, the following asymptotic convergence rate result has also been established in [22]:

$$k^{\frac{\tau}{2}}(\tilde{\eta}_k - \eta^*) \xrightarrow{d} \mathcal{N}(0, \Sigma) \text{ as } k \rightarrow \infty,$$

where $\tau \in (0, 1)$ is an appropriate constant and Σ is a positive definite covariance matrix.

Model-Based Annealing Random Search (MARS)

The stochastic approximation framework also allows for a lot of flexibility in developing provably convergent algorithms that perform well in practice. For example, [21] has investigated the use of Boltzmann distributions as reference models in the framework to address the implementation difficulty of annealing adaptive search, leading to a new globally convergent algorithm called model-based annealing random search (MARS). The algorithm complements existing research based on MCMC sampling techniques in the sense that it samples candidate solutions from a sequence of NEF distributions that approximates the target Boltzmann distributions, whereas MCMC techniques are sequential sampling procedures that sample directly from the Boltzmann distributions. The major steps of the MARS algorithm are given next.

Basic MARS Algorithm

Step 0. Select an NEF distribution family $\{f_\varphi, \varphi \in \Phi\}$, a sequence of temperature parameters $\{T_k\}$, and a gain sequence $\{\alpha_k\}$.

Step 1. Given φ_k , generate N candidate solutions X_k^1, \dots, X_k^N by sampling from f_{φ_k} .

Step 2. Update the parameter

$$\varphi_{k+1} = \arg \min_{\varphi \in \Phi} \mathcal{D}(\hat{g}_{k+1}, f_\varphi);$$

set $k \leftarrow k + 1$ and reiterate from Step 1.

At Step 2 of MARS, the reference distribution \hat{g}_{k+1} is given by $\hat{g}_{k+1}(x) = \alpha_k \bar{g}_{k+1}(x) + (1 - \alpha_k) f_{\varphi_k}(x)$, which is the mixture of the current sampling distribution f_{φ_k} with \bar{g}_{k+1} , an empirical estimate of the true Boltzmann distribution $g_{k+1}(x) := \frac{e^{h(x)/T_k}}{\int_{\mathcal{X}} e^{h(x)/T_k} dx}$ based on the sampled solutions X_k^1, \dots, X_k^N .

In light of Eq. (12.5), the mean parameter function $m(\varphi_{k+1})$ corresponding to the new parameter φ_{k+1} obtained at Step 2 of MARS can be viewed as an iterate generated by the gradient recursion

$$m(\varphi_{k+1}) = m(\varphi_k) - \alpha_k \nabla_{\varphi} \mathcal{D}(\bar{g}_{k+1}, f_{\varphi})|_{\varphi=\varphi_k}. \quad (12.10)$$

By the properties of NEFs, the gradient in (12.10) can be expressed in terms of a true gradient term involving the Boltzmann distribution g_{k+1} and an error term due to random sampling, leading to a Robbins–Monro type stochastic approximation algorithm

$$\begin{aligned} m(\varphi_{k+1}) &= m(\varphi_k) - \alpha_k \left(m(\varphi_k) - \mathbb{E}_{g_{k+1}}[\Gamma(X)] + \mathbb{E}_{g_{k+1}}[\Gamma(X)] - \mathbb{E}_{\bar{g}_{k+1}}[\Gamma(X)] \right) \\ &= m(\varphi_k) - \alpha_k \nabla_{\varphi} \mathcal{D}(g_{k+1}, f_{\varphi})|_{\varphi=\varphi_k} - \alpha_k \left(\mathbb{E}_{g_{k+1}}[\Gamma(X)] - \mathbb{E}_{\bar{g}_{k+1}}[\Gamma(X)] \right). \end{aligned} \quad (12.11)$$

Note that (12.11) generalizes a typical stochastic approximation recursion in that the function $\mathcal{D}(g_{k+1}, f_{\varphi})$ may change shape with k . This time-varying feature of MARS actually turns out to be a desirable property, because the idealized sequence of Boltzmann distributions $\{g_k\}$ converges to a limiting distribution g^* as k goes to infinity. This will in turn imply the convergence of the sequence of the optimal solutions $\{\varphi_k\}$ to a global optimizer φ^* . Under some appropriate conditions on the algorithm input parameters, the following convergence result has been obtained in [21]:

$$\lim_{k \rightarrow \infty} m(\varphi_k) = \Gamma(x^*) \text{ w.p.1.}$$

In addition, for a polynomially increasing sample size $N = ak^\beta$ and a gain sequence of the form $\alpha_k = c/k^\alpha$ for constants $a > 0$, $c > 0$, $\alpha \in (\frac{1}{2}, 1)$, and $\beta > \alpha$, an asymptotic normality result of the following form is also given in [21]:

$$k^{\frac{\alpha+\beta}{2}} (m(\varphi_k) - \Gamma(x^*)) \xrightarrow{d} \mathcal{N}(0, \Sigma) \text{ as } k \rightarrow \infty,$$

where Σ is a positive definite covariance matrix.

12.4.2 Simulation Optimization

The simulation optimization setting, where $H(x, \xi)$ is obtained in a simulation replication, requires an additional simulation allocation rule $\{M_k\}$, which allocates M_k simulation observations to each of the N candidate solutions generated at the k th iteration. Thus, in constructing gradient estimators, if the true performance at a sampled solution X_k^i is replaced by the sample average

$$\bar{H}_k(X_k^i) = \frac{1}{M_k} \sum_{j=1}^{M_k} H(X_k^i, \xi_j),$$

then an estimator of the true gradient $L(\tilde{\eta}_k)$ in (12.8) will take the form

$$\bar{L}_N(\tilde{\eta}_k) = \frac{N^{-1} \sum_{i=1}^N S(\bar{H}_k(X_k^i)) G(X_k^i, \tilde{\eta}_k)}{N^{-1} \sum_{i=1}^N S(\bar{H}_k(X_k^i))}.$$

Consequently, the gradient iteration (12.9) can be carried out at each step by replacing the true performance at a sampled solution X_k^i by its sample average approximation $\bar{H}_k(X_k^i)$, leading to a recursion of the form

$$\tilde{\eta}_{k+1} = \tilde{\eta}_k - \alpha_k \tilde{L}_N(\tilde{\eta}_k) + \alpha_k (\tilde{L}_N(\tilde{\eta}_k) - \bar{L}_N(\tilde{\eta}_k)). \quad (12.12)$$

Under some appropriate conditions on the allocation rule $\{M_k\}$, the expectation (conditional on the current sampled solutions) of the error term $\tilde{L}_N(\tilde{\eta}_k) - \bar{L}_N(\tilde{\eta}_k)$ goes to 0 as $k \rightarrow \infty$. Thus by treating the simulation noise $\tilde{L}_N(\tilde{\eta}_k) - \bar{L}_N(\tilde{\eta}_k)$ as a vanishing bias term, the (possibly local) convergence and convergence rate analysis of recursion (12.12) can be studied along the same line as in the deterministic optimization case.

Application of MARS to Finite-Horizon Markov Decision Processes

To illustrate the adaptation of the stochastic approximation framework to simulation setting, we modify and extend the MARS algorithm to a stochastic setting and present a simulation-based algorithm called approximate stochastic annealing (ASA) for solving finite-horizon Markov decision processes (MDPs) [16]. The idea is to interpret an MDP as a stochastic optimization problem on the (randomized) policy space (where candidate solutions are policies) and then use ASA as a specific optimization strategy to directly search the policy space to find good policies.

Consider a discrete-time finite \mathcal{H} -horizon stochastic system $x_{t+1} = f(x_t, a_t, w_t)$ for $t = 0, 1, \dots, \mathcal{H} - 1$, where x_t represents the system state at time t taking values from a finite state space \mathcal{X} , a_t is the control applied at time t chosen from a finite action set \mathcal{A} , $\{w_t\}$ is a sequence of random vectors representing the stochastic uncertainty of the system, and f is the next-state transition function. Let $R_t(x_t, a_t, w_t)$ be the one-stage reward for action a_t taken in state x_t at time t . Define Π as the set of non-stationary deterministic Markovian policies $\pi = \{\pi_t, t = 0, \dots, \mathcal{H} - 1\}$, where each $\pi_t : \mathcal{X} \rightarrow \mathcal{A}$ is a function that specifies the action to be applied at time t for each $x \in \mathcal{X}$. For an initial state $x_0 = x$, the expected total reward (value function) associated with a policy π is given by

$$V^\pi(x) := \mathbb{E} \left[\sum_{t=0}^{\mathcal{H}-1} R_t(x_t, \pi_t(x_t), w_t) \mid x_0 = x \right]. \quad (12.13)$$

The objective is to find an optimal policy $\pi^* \in \Pi$ that maximizes the expected total reward for a given state x , i.e.,

$$V^{\pi^*}(x) = \max_{\pi \in \Pi} V^\pi(x). \quad (12.14)$$

At each iteration, ASA searches for improved policies by sampling from a probability distribution function $\phi(\pi, q)$ over the policy space Π , where q is a parameter vector taking values from some parameter space. The distribution is then modified using a Boltzmann selection scheme based on the simulated/estimated value functions of the sampled policies. One simple way to specify the parameterized distribution $\phi(\pi, q)$ is to use an $|\mathcal{X}|$ -by- $|\mathcal{A}|$ -by- \mathcal{H} stochastic matrix q_k , whose (i, j, t) th entry $q_k(i, j, t)$ specifies the probability that the action a_j is applied to state x_i at time t . Such a stochastic matrix q_k gives rise to a probability mass function over Π :

$$\phi(\pi, q_k) := \prod_{t=0}^{\mathcal{H}-1} \prod_{i=1}^{|\mathcal{X}|} \prod_{j=1}^{|\mathcal{A}|} [q_k(i, j, t)]^{I\{\pi \in \Pi_{i,j}(t)\}} \quad \forall \pi \in \Pi, \quad (12.15)$$

where $I\{\cdot\}$ is the indicator function and $\Pi_{i,j}(t) := \{\pi : \pi_t(x_i) = a_j\}$ denotes the set of deterministic policies that assign action a_j to state x_i at time t . Thus, as in

the MARS algorithm, the goal is to iteratively update the entries $q_k(i, j, t)$ so that $\phi(\boldsymbol{\pi}, q_k)$ will be a close approximation to the desired Boltzmann distribution

$$g_{k+1}(\boldsymbol{\pi}) = \frac{e^{V^\pi/T_k}}{\sum_{\boldsymbol{\pi} \in \Pi} e^{V^\pi/T_k}}.$$

This results in the following adaptive policy search procedure.

Approximate Stochastic Annealing (ASA) Algorithm for Solving MDPs

- Step 0. Select a sequence of temperature parameters $\{T_k\}$ and a gain sequence $\{\alpha_k\}$.
- Step 1. Given q_k , sample N policies $\Lambda_k := \{\pi_k^1, \dots, \pi_k^N\}$ from $\phi(\boldsymbol{\pi}, q_k)$.
- Step 2. For each $\boldsymbol{\pi} \in \Lambda_k$, perform M_k independent simulation replication runs and let $\bar{V}_k^\pi = \frac{1}{M_k} \sum_{l=1}^{M_k} V_l^\pi$, where V_l^π is an estimate of the value function V^π obtained in the l th replication run.
- Step 3. Update the q matrix

$$q_{k+1} = \arg \min_q \mathcal{D}(\hat{g}_{k+1}, \phi(\cdot, q)).$$

Set $k \leftarrow k + 1$ and reiterate from Step 1.

Note that at Step 3, the KL-divergence is with respect to $\hat{g}_{k+1}(\boldsymbol{\pi}) = \alpha_k \tilde{g}_{k+1}(\boldsymbol{\pi}) + (1 - \alpha_k) \phi(\boldsymbol{\pi}, q_k)$, where \tilde{g}_{k+1} is an empirical estimate of the true Boltzmann distribution g_{k+1} based on the sampled policies in Λ_k and value function estimates V_l^π .

Since the parameterized distribution $\phi(\boldsymbol{\pi}, q_k)$ given in (12.15) belongs to NEFs, it is not difficult to show that the entries of the q_k matrix updated at Step 3 satisfy the following recursion:

$$\begin{aligned} q_{k+1}(i, j, t) - q_k(i, j, t) &= \alpha_k \left[\mathbb{E}_{\tilde{g}_{k+1}} [I\{\boldsymbol{\pi} \in \Pi_{i,j}(t)\}] - q_k(i, j, t) \right] \\ &= \alpha_k \left[\mathbb{E}_{g_{k+1}} [I\{\boldsymbol{\pi} \in \Pi_{i,j}(t)\}] - q_k(i, j, t) \right] \\ &\quad - \alpha_k \left[\mathbb{E}_{g_{k+1}} [I\{\boldsymbol{\pi} \in \Pi_{i,j}(t)\}] - \mathbb{E}_{\tilde{g}_{k+1}} [I\{\boldsymbol{\pi} \in \Pi_{i,j}(t)\}] \right]. \end{aligned} \quad (12.16)$$

Since g_{k+1} assigns more weight to policies with better performance, the first term on the right-hand-side of the second equality implies that the entries of q_k are updated in a direction that “pursues” the optimal policy $\boldsymbol{\pi}^*$, whereas the second term can be viewed as a noise term caused by the approximation error between \tilde{g}_{k+1} and g_{k+1} . Thus the convergence analysis of ASA essentially boils down to the issue of inspecting whether the Boltzmann distribution g_{k+1} can be closely approximated by its empirical estimate \tilde{g}_{k+1} . In particular, under some mild conditions on the algorithm input parameters, the following result is obtained in [16]:

$$q_k(i, j, t) \rightarrow I\{\boldsymbol{\pi}^* \in \Pi_{i,j}(t)\} \quad \forall i, j, t \text{ as } k \rightarrow \infty \text{ w.p.1,}$$

which indicates that the sequence of stochastic matrices q_k generated by the algorithm will converge to a limiting matrix assigning unit probability mass to the optimal policy π^* .

12.5 A Stochastic Averaging Approach

As we have seen in previous sections, convergence analysis of model-based algorithms typically requires a sample size N that increases polynomially with the number of algorithm iterations. In practice, this translates to using a per-iteration computational effort that grows without bound as the number of iterations increases, which may have a negative impact on the algorithm's practical performance, especially in the setting where simulation/function evaluations are expensive.

This efficiency issue has been tackled in [17], where the basic idea is to maintain a population of probability distribution models (rather than just a single model as in a typical model-based algorithm) at each iteration and then adaptively allocate a given computing budget among different models in order to maximize the expected performance of the algorithm. In this section, we present an approach that aims to improve the sampling efficiency of model-based algorithms from a different perspective, focusing primarily on reducing the number of candidate solutions generated per iteration. This is carried out through embedding a stochastic averaging procedure within model-based algorithms to make more efficient use of the past sampling information. The material in this section is based on [23, 24].

For simplicity, we consider the general reference distribution introduced in Sect. 12.3:

$$g_k(x) = \frac{S(h(x))g_{k-1}(x)}{\mathbb{E}_{g_{k-1}}[S(h(X))]}.$$

By expanding the above recursion, we can write g_k in terms of the initial distribution g_0 as

$$g_k(x) = \frac{S_k(h(x))g_0(x)}{\mathbb{E}_{g_0}[S_k(h(X))]}, \quad (12.17)$$

where S_k is some appropriate iteration-varying function that depends on S . Substituting (12.17) into the minimization problem $\min_{\varphi} \mathcal{D}(g_{k+1}, f_{\varphi})$ and dropping terms that are constants with respect to φ , it can be seen that the parameter φ_{k+1} obtained at Step 2 of the MRAS method of Sect. 12.3 can be equivalently obtained by solving the following optimization problem

$$\varphi_{k+1} = \arg \max_{\varphi \in \Theta} \left(Q_{k+1}(\varphi) := \int_{x \in \Theta} S_{k+1}(h(x)) \ln f_{\varphi}(x) dx \right). \quad (12.18)$$

As discussed previously, in model-based algorithms, the integral involved in the Q -function $Q_{k+1}(\varphi)$ is estimated by generating N i.i.d. candidate solutions X_k^1, \dots, X_k^N from f_{φ_k} , and then replacing $Q_{k+1}(\varphi)$ by its sample average approximation

$$\bar{Q}_{k+1}(\varphi) := \frac{1}{N} \sum_{i=1}^N \frac{S_{k+1}(h(X_k^i))}{f_{\varphi_k}(X_k^i)} \ln f_{\varphi}(X_k^i).$$

Although $\bar{Q}_{k+1}(\varphi)$ is an unbiased estimator of $Q_{k+1}(\varphi)$, the corresponding optimization step will lead to an estimator of φ_{k+1} that is biased for any finite sample size N , because the optimal solution to (12.18) involves a ratio of integrals/expectations. Consequently, common implementations of these algorithms either require hundreds or even thousands of candidate solutions to be generated per iteration [6, 35], or require the use of a sample size N that increases at least polynomially with k in order to reduce the ratio bias effect [13, 18, 21, 22].

This bias issue has been addressed in [23, 24] by replacing the sample average approximation $\bar{Q}_k(\varphi)$ with the stochastic averaging procedure

$$\hat{Q}_{k+1}(\varphi) = (1 - \beta_k) \hat{Q}_k(\varphi) + \beta_k \frac{1}{N} \sum_{i=1}^N \frac{S_{k+1}(h(X_k^i))}{f_{\varphi_k}(X_k^i)} \ln f_{\varphi}(X_k^i), \quad (12.19)$$

with $\hat{Q}_1(\varphi) := \frac{1}{N} \sum_{i=1}^N (S_1(h(X_0^i))/f_{\varphi_0}(X_0^i)) \ln f_{\varphi}(X_0^i)$, where β_k is a step size constant satisfying $\beta_k \in (0, 1] \forall k$. Note that this procedure incrementally updates the current estimate of the Q -function as new sampling information becomes available at each iteration. In addition, due to the recursive nature of (12.19), all candidate solutions generated in the previous iterations contribute to the estimation of the Q -function $Q_{k+1}(\varphi)$. Consequently, it is reasonable to expect that the number of samples per iteration N can be significantly reduced or even held at a small constant value.

It is interesting to note that when NEF is used, $\hat{Q}_{k+1}(\varphi)$ can be expressed as a linear combination of the parameter vector φ and the function $K(\varphi)$:

$$\hat{Q}_{k+1}(\varphi) = \varphi^T \mathcal{S}_{k+1} - K(\varphi) \mathcal{R}_{k+1},$$

where the quantities \mathcal{S}_k and \mathcal{R}_k can be computed via the respective recursions

$$\begin{aligned} \mathcal{S}_{k+1} &= \mathcal{S}_k + \beta_k \left(\frac{1}{N} \sum_{i=1}^N \frac{S_{k+1}(h(X_k^i))}{f_{\varphi_k}(X_k^i)} \Gamma(X_k^i) - \mathcal{S}_k \right), \\ \mathcal{R}_{k+1} &= \mathcal{R}_k + \beta_k \left(\frac{1}{N} \sum_{i=1}^N \frac{S_{k+1}(h(X_k^i))}{f_{\varphi_k}(X_k^i)} - \mathcal{R}_k \right), \end{aligned}$$

with $\mathcal{S}_1 := \frac{1}{N} \sum_{i=1}^N S_1(h(X_0^i))/f_{\varphi_0}(X_0^i) \Gamma(X_0^i)$ and $\mathcal{R}_1 := \frac{1}{N} \sum_{i=1}^N S_1(h(X_0^i))/f_{\varphi_0}(X_0^i)$. Thus, by substituting $\hat{Q}_{k+1}(\varphi)$ for $Q_{k+1}(\varphi)$ in (12.18), we have the following optimization problem:

$$\varphi_{k+1} = \arg \max_{\varphi \in \Theta} \left(\varphi^T \mathcal{S}_{k+1} - K(\varphi) \mathcal{R}_{k+1} \right),$$

whose unique closed-form solution is given by

$$m(\varphi_{k+1}) = \frac{\mathcal{S}_{k+1}}{\mathcal{R}_{k+1}} \text{ or equivalently } \varphi_{k+1} = m^{-1} \left(\frac{\mathcal{S}_{k+1}}{\mathcal{R}_{k+1}} \right).$$

The above stochastic averaging idea has been combined with the MARS algorithm of Sect. 12.4, leading to a two-time-scale stochastic approximation type of algorithm called MARS with stochastic averaging (MARS-SA).

MARS with Stochastic Averaging (MARS-SA)

Step 0. Select temperature parameters $\{T_k\}$ and gain sequences $\{\alpha_k\}$ and $\{\beta_k\}$.

Step 1. Generate N i.i.d candidate solutions $\Lambda_k := \{X_k^1, \dots, X_k^N\}$ from $f_{\varphi_k}(x)$.

Step 2. Update \mathcal{S}_{k+1} and \mathcal{R}_{k+1} according to the recursions:

$$\begin{aligned} \mathcal{S}_{k+1} &= \mathcal{S}_k + \beta_k \left(\frac{1}{N} \sum_{x \in \Lambda_k} \frac{e^{h(x)/T_{k+1}}}{f_{\varphi_k}(x)} \Gamma(x) - \mathcal{S}_k \right), \\ \mathcal{R}_{k+1} &= \mathcal{R}_k + \beta_k \left(\frac{1}{N} \sum_{x \in \Lambda_k} \frac{e^{h(x)/T_{k+1}}}{f_{\varphi_k}(x)} - \mathcal{R}_k \right). \end{aligned}$$

Step 3. Compute a new parameter φ_{k+1} as

$$m(\varphi_{k+1}) = \alpha_k \frac{\mathcal{S}_{k+1}}{\mathcal{R}_{k+1}} + (1 - \alpha_k) m(\varphi_k).$$

Set $k \leftarrow k + 1$ and reiterate from Step 1.

By exploiting the connection of MARS-SA to stochastic approximation method, it is shown in [23, 24] that the algorithm converges globally even when the per iteration sample size N is held at a small constant value. In addition, preliminary empirical results reported in [24] indicate that the new algorithm can be more efficient (in terms of the number of performance evaluations) than the original MARS algorithm.

12.6 Conclusions and Open Research Questions

In this chapter, we have provided an overview of three model-based optimization methods and discussed their extensions to simulation optimization. In particular, the MRAS method discussed in Sect. 12.3 offers a general framework to design and implement model-based algorithms, whereas the stochastic approximation method

presented in Sect. 12.4 provides a systematic approach to analyze the convergence properties of these algorithms. In addition, we have also outlined in Sect. 12.5 a stochastic averaging procedure that addresses the estimator bias issue in model-based algorithms and aims to make these algorithms computationally more efficient. These methods are illustrated through a number of exemplary algorithms that are both provably convergent and exhibit promising empirical performance.

There are many challenging research issues that remain to be addressed. For example, the theoretical convergence and empirical performance of model-based algorithms are greatly influenced by the choices of reference distributions. So a natural research question is how the reference distributions should be chosen for specific problems. Moreover, since existing convergence results are all asymptotic in nature, a theoretical issue is to study whether finite-time performance bounds (e.g., similar to those in stochastic adaptive search [39]) can also be developed for these algorithms.

Model-based algorithms generally do not make use of problem structure, whereas in a continuous-variable simulation optimization setting, there may be additional information available (e.g., stochastic gradient estimates, Lipschitz continuity) from problem knowledge. It is well-known that effective use of structure may dramatically improve the solution efficiency. Therefore, another interesting research direction is to investigate how to incorporate problem structure information into model-based algorithms, as well as to identify classes of problems for which this can be done in a systematic manner.

The extension of model-based algorithms to simulation optimization is carried out in a relatively straightforward manner by introducing an additional simulation allocation sequence $\{M_k\}$ to obtain increasingly precise performance estimates as the search proceeds. This motivates the design of new model-based algorithms that do not rely on expending additional simulation effort on performance estimation ($M_k = 1$ for all k), e.g., in a way that the simulation noise will act like martingale difference noise in stochastic approximation and automatically average out over a large number of iterations. This is yet another avenue of research that merits further investigation.

Acknowledgements This work was supported in part by the National Science Foundation (NSF) under Grant CMMI-1130761 and by the Air Force Office of Scientific Research (AFOSR) under Grant FA95501010340.

References

1. G. Allon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134:137–151, 2005.
2. S. Andradóttir. An overview of simulation optimization with random search. In S. G. Henderson and B. L. Nelson, editors, *Handbooks in Operations Research and Management Science: Simulation*. Elsevier, 2006.

3. R. R. Barton and M. Meckesheimer. Metamodel-based simulation optimization. In S. G. Henderson and B. L. Nelson, editors, *Handbooks in Operations Research and Management Science: Simulation*. Elsevier, 2006.
4. M. Benaim. A dynamical system approach to stochastic approximations. *SIAM Journal on Control and Optimization*, 34:437–472, 1996.
5. V. S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press; New Delhi: Hindustan Book Agency, 2008.
6. Y. Cai, X. Sun, and P. Jia. Probabilistic modeling for continuous eda with boltzmann selection and kullback-leibler divergence. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 389–396, 2006.
7. A. Costa, O. D. Jones, and D. Kroese. Convergence properties of the cross-entropy method for discrete optimization. *Operations Research Letters*, 35(5):573–580, 2007.
8. M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.
9. F. W. Glover. Tabu search: A tutorial. *Interfaces*, 20:74–94, 1990.
10. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Kluwer Academic Publishers, Boston, MA, 1989.
11. W. B. Gong, Y. C. Ho, and W. Zhai. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*, 10:384–404, 1999.
12. W. J. Gutjahr. A converging aco algorithm for stochastic combinatorial optimization. In *Proceedings of SAGA 2003 Stochastic Algorithms: Foundations and Applications*, pages 10–25, 2003.
13. T. Homem-de-Mello. A study on the cross-entropy method for rare-event probability estimation. *INFORMS Journal on Computing*, 19:381–394, 2007.
14. T. Homem-de-Mello. On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling. *SIAM Journal on Optimization*, 19:524–551, 2008.
15. L. J. Hong and B. L. Nelson. Discrete optimization via simulation using compass. *Operations Research*, 54:115–129, 2006.
16. J. Hu and H. S. Chang. An approximate stochastic annealing algorithm for finite horizon markov decision processes. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 5338–5343, 2010.
17. J. Hu, H. S. Chang, M. C. Fu, and S. I. Marcus. Dynamic sample budget allocation in model-based optimization. *Journal of Global Optimization*, 50:575–596, 2011.
18. J. Hu, M. C. Fu, and S. I. Marcus. A model reference adaptive search method for global optimization. *Operations Research*, 55:549–568, 2007.
19. J. Hu, M. C. Fu, and S. I. Marcus. A model reference adaptive search method for stochastic global optimization. *Communications in Information and Systems*, 8:245–276, 2008.
20. J. Hu and P. Hu. On the performance of the cross-entropy method. In *Proceedings of the 2009 Winter Simulation Conference*, pages 459–468. IEEE, Piscataway, NJ, 2009.
21. J. Hu and P. Hu. Annealing adaptive search, cross-entropy, and stochastic approximation in global optimization. *Naval Research Logistics*, 58:457–477, 2011.
22. J. Hu, P. Hu, and H. S. Chang. A stochastic approximation framework for a class of randomized optimization algorithms. *IEEE Transactions on Automatic Control*, 57:165–178, 2012.
23. J. Hu and C. Wang. Discrete optimization via approximate annealing adaptive search with stochastic averaging. In *Proceedings of the 2011 Winter Simulation Conference*, pages 4206–4216. IEEE, Piscataway, NJ, 2011.
24. J. Hu, E. Zhou, and Q. Fan. Model-based annealing random search with stochastic averaging. *ACM Transactions on Modeling and Computer Simulation*, forthcoming, 2014.
25. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
26. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

27. A. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, 2001.
28. H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems and Applications*. Springer-Verlag, New York, 1978.
29. H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
30. P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, Boston, MA, 2002.
31. S. Mannor, R. Y. Rubinstein, and Y. Gat. The cross-entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning*, pages 512–519, 2003.
32. H. E. Romeijn and R. L. Smith. Simulated annealing and adaptive search in global optimization. *Probability in the Engineering and Informational Sciences*, 8:571–590, 1994.
33. R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99:89–112, 1997.
34. R. Y. Rubinstein. Combinatorial optimization, ants and rare events. In S. Uryasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, pages 304–358, 2001.
35. R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer, New York, 2004.
36. L. Shi and S. Ólafsson. Nested partitions method for global optimization. *Operations Research*, 48:390–400, 2000.
37. D. H. Wolpert. Finding bounded rational equilibria part i: Iterative focusing. In T. Vincent, editor, *Proceedings of the Eleventh International Symposium on Dynamic Games and Applications*, 2004.
38. D. Yan and H. Mukai. Stochastic discrete optimization. *SIAM Journal on Control and Optimization*, 30:594–612, 1992.
39. Z. B. Zabinsky. *Stochastic Adaptive Search for Global Optimization*. Kluwer Academic Publishers, 2003.
40. Z. B. Zabinsky, R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman. Improving hit-and-run for global optimization. *Journal of Global Optimization*, 3:171–192, 1993.
41. Q. Zhang and H. Mühlenbein. On the convergence of a class of estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 8:127–136, 2004.
42. E. Zhou, M. C. Fu, and S. I. Marcus. A particle filtering framework for randomized optimization algorithms. In *Proceedings of the 2008 Winter Simulation Conference*, pages 647–654. IEEE, Piscataway, NJ, 2008.
43. E. Zhou and J. Hu. Gradient guided adaptive stochastic search. *IEEE Transactions on Automatic Control*, 59:1818–1832, 2014.
44. M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research*, 131:373–395, 2004.