Chapter 10
A Survey of Some Model-Based Methods for Global Optimization

Jiaqiao Hu, Yongqiang Wang, Enlu Zhou, Michael C. Fu, and Steven I. Marcus

10.1 Introduction

Global optimization aims at characterizing and computing global optimal solutions to problems with nonconvex, multimodal, or badly scaled objective functions; it has applications in many areas of engineering and science. In general, due to the absence of structural information and the presence of many local extrema, global optimization problems are extremely difficult to solve exactly. There are many different types of methods in the literature on global optimization, which can be categorized based on different criteria. For instance, they can be classified either based on the properties of problems to be solved (combinatorial or continuous, nonlinear, linear, convex, etc.) or by the properties of algorithms that search for new candidate solutions such as *deterministic* or *random search* algorithms. Random search algorithms can further be classified as *instance-based* or *model-*

J. Hu
Department of Applied Mathematics and Statistics, State University at Stony Brook, Stony Brook, NY 11794, USA
e-mail: jqhu@ams.sunysb.edu

Y. Wang • S.I. Marcus (✉)
Department of Electrical and Computer Engineering & Institute for Systems Research, University of Maryland, College Park, MD 20742, USA
e-mail: yqwang@umd.edu; marcus@umd.edu

E. Zhou
Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, IL 61801, USA
e-mail: enluzhou@illinois.edu

M.C. Fu
The Robert H. Smith School of Business & Institute for Systems Research, University of Maryland, College Park, MD 20742, USA
e-mail: mfu@umd.edu

D. Hernández-Hernández and A. Minjárez-Sosa (eds.), *Optimization, Control, and Applications of Stochastic Systems*, Systems & Control: Foundations & Applications,
DOI 10.1007/978-0-8176-8337-5_10,

based algorithms according to the mechanism of generating new candidate solutions [46].

Instance-based algorithms maintain a single solution or population of candidate solutions, and the construction of new generate of candidate solutions depends explicitly on the previously generated solutions. Some well-known instance-based algorithms include simulated annealing [25], genetic algorithms [16,36], tabu search [15], nested partitions [35], generalized hill climbing [22, 23], and evolutionary programming [12]. Model-based search algorithms are a class of new solution techniques and were introduced only in recent years [18, 27, 32–34, 42]. In model-based algorithms, new solutions are generated via an intermediate probabilistic model that is updated or induced from the previously generated solutions. Thus, there is only an implicit/indirect dependency among the solutions generated at successive iterations of the algorithm. Specific model-based algorithms include annealing adaptive search (AAS) [31, 41], the cross-entropy (CE) method [32–34], and estimation of distribution algorithms (EDAs) [27, 42]. Instance-based algorithms have been extensively studied in past decades. After briefly reviewing some model-based algorithms, this chapter focuses on several model-based methods that have been developed recently.

10.2 Global Optimization and Previous Work

10.2.1 Problem Statement

In many engineering design and optimization applications, we are concerned with finding parameter values that achieve the optimum of an objective function. Such problems can be mathematically stated in the generic form:

$$x^* \in \arg\max_{x \in \mathbf{X}} H(x), \tag{10.1}$$

where x is a vector of n decision variables, the solution space $\mathbf{X}$ is a nonempty (often compact) subset of $\Re^n$, and the objective function $H : \mathbf{X} \to \Re$ is a bounded deterministic function.

Throughout this chapter, we assume that there exists a global optimal solution to (10.1), i.e., $\exists x^* \in \mathbf{X}$ such that $H(x) \leq H(x^*)\ \forall x \neq x^*,\ x \in \mathbf{X}$. In practice, this assumption can be justified under fairly general conditions. For example, for continuous optimization problems with compact solution spaces, the existence of an x^* is guaranteed by the well-known Weierstrass theorem, whereas in discrete optimization, the assumption holds trivially when $\mathbf{X}$ is a (nonempty) finite set. Note that no further structural assumptions, such as convexity or differentiability, are imposed on the objective function, and there may exist many locally optimal solutions. In other words, our focus is on general global optimization problems with little known structure. This setting arises in many complex systems of interest, e.g.,

when the explicit form of H is not readily available and the objective function values can only be assessed via "black-box" evaluations.

10.2.2 Previous Work on Random Search Methods

In this section, we review a class of global optimization algorithms collectively known as random search methods. A random search method usually refers to an algorithm that is iterative in nature, and uses some sort of randomized mechanism to generate a sequence of iterates, e.g., candidate solutions or probabilistic models, in order to successively approximate the optimal solution. What type of iterates an algorithm produces and how these iterates are generated are what differentiates approaches. A major advantage of stochastic search methods is that they are robust and easy to implement, because they typically only rely on the objective function values rather than structural information such as convexity and differentiability. This feature makes these algorithms especially prominent in optimization of complex systems with little structure.

From an algorithmic point of view, a random search algorithm can further be classified as being either *instance-based* or *model-based* [46]. In instanced-based algorithms, an iterate comprises a single or a set/population of candidate solution(s), and the construction of new candidate solutions depends explicitly on previously generated solutions. Such algorithms can be represented abstractly by the following framework:

1. Given a set/population of candidate solutions $Y^{(k)}$ (which might be a singleton set), generate a set of new candidate solutions $X^{(k)}$ according to a specified random mechanism.
2. Update the current population $Y^{(k+1)}$ based on population $Y^{(k)}$ and candidate solutions in $X^{(k)}$; increase the iteration counter k by 1 and reiterate from Step 1.

Thus the two major steps in an instance-based algorithm are the generation step that produces a set of candidate solutions, and the selection/update step that determines whether a newly generated solution in $X^{(k)}$ should be included in the next generation. Over the past few decades, a significant amount of research effort has been centered around instance-based methods, with numerous algorithms proposed in the literature and their behaviors relatively well studied and understood. Some well-known examples include simulated annealing [25], genetic algorithms [16,36], tabu search [15], nested partitions [35], generalized hill climbing [22, 23], and evolutionary programming [12].

We focus on model-based methods, which differ from instance-based approaches in that candidate solutions are generated at each iteration by sampling from an intermediate probability distribution model over the solution space. The idea is to iteratively modify the distribution model based on the sampled solutions to bias the future search toward regions containing high-quality solutions. In its most basic

from, a model-based algorithm typically consists of the following two steps: let g_k be a probability distribution on $\mathbf{X}$ at the kth iteration of an algorithm:

1. Randomly generate a set/population of candidate solutions $X^{(k)}$ from g_k.
2. Update g_k based on the sampled solutions in $X^{(k)}$ to obtain a new distribution g_{k+1}; increase k by 1 and reiterate from step 1.

The underlying idea is to construct a sequence of iterates (probability distributions) $\{g_k\}$ with the hope that $g_k \to g^*$ as $k \to \infty$, where g^* is a limiting distribution that assigns most of its probability mass to the set of optimal solutions. So it is the probability distribution (as opposed to candidate solutions as in instance-based algorithms) that is propagated from one iteration to the next.

Clearly, the two key questions one needs to address in a model-based algorithm are how to generate samples from a given distribution g_k and how to construct the distribution sequence $\{g_k\}$. In order to address these questions, we provide brief descriptions of three model-based algorithms: annealing adaptive search (AAS) [31, 41], the cross-entropy (CE) method [32–34], and estimation of distribution algorithms (EDAs) [27, 42].

The annealing adaptive search algorithm was originally introduced in Romeijn and Smith [18] as a means to understand the behavior of simulated annealing. The algorithm generates candidate solutions by sampling from a sequence of Boltzmann distributions parameterized by time-dependent temperatures. As the temperature decreases to zero, the sequence of Boltzmann distributions becomes more concentrated on the set of optimal solutions, so that a solution sampled at later iterations will be close to the global optimum with high probability. For the class of Lipschitz optimization problems, it is 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 [31, 41]. However, the idealized AAS is not intended to be a practically useful algorithm, because the problem of sampling exactly from a given Boltzmann distribution is known to be extremely difficult. This implementation issue has motivated a number of algorithms that approximate AAS, where a primary focus has been on the design and refinement of Markov chain-based sampling techniques embedded within the AAS framework [40, 41].

The CE method was motivated by an adaptive algorithm for estimating probabilities of rare events in complex stochastic networks [32], which involves variance minimization. It was later realized [33] that the method can be modified to solve combinatorial and continuous optimization problems. The CE method uses a family of parameterized probability distributions on the solution space and tries to find the parameter of the distribution that assigns maximum probability to the set of optimal solutions. Implicit in CE is an optimal importance sampling distribution concentrated only on the set of optimal solutions. The key idea is to use an iterative scheme to successively estimate the optimal parameter that minimizes the Kullback-Leibler (KL) divergence between the optimal distribution and the family of parameterized distributions. Although there have been extensive developments regarding implementation and successful practical applications of CE (see [34]), 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., [17] 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 [21] by relating the algorithm to recursions of stochastic approximation type (see Sect. 10.6).

EDAs were first introduced in the field of evolutionary computation. They inherit the spirit of the well-known 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 [27] 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 [42].

There are many other model-based algorithms proposed for global optimization. Some interesting examples include ant colony optimization (ACO) [9], probability collectives (PCs) [39], and particle swarm optimization (PSO) [24]. We do not provide a comprehensive description of all of them, but instead present some recently developed frameworks and approaches that allow us to view these algorithms in a unified setting. These approaches, including model reference adaptive search (MRAS) [18], the particle-filtering (PF) approach [43], the evolutionary games approach [38], and the stochastic approximation gradient approach [20, 21], will be discussed in detail in the following sections.

10.3 Model Reference Adaptive Search

As we have seen from Sect. 10.2, model-based algorithms differ from each other in the choices of the distribution sequence $\{g_k\}$. Examples of the $\{g_k\}$ sequence include (a) Boltzmann distributions, used in AAS; (b) optimal importance sampling measure, primarily used in the CE method; and (c) proportional selection schemes, used in EDAs, ACOs, and PCs.

However, in all the above cases, the construction of g_k often depends on the objective function H, whose explicit form may not be available. In addition, since g_k may not have any special structure, sampling exactly from the distribution is in general intractable. To address these computational challenges arising in model-based methods, we have formalized in [18] a general approach called model reference adaptive search (MRAS), where the basic idea is to use a convenient parametric distribution as a surrogate to approximate g_k and then sample candidate

solutions from the surrogate distribution. More specifically, the method starts by specifying a family of parameterized distributions $\{f_\theta,\ \theta \in \Theta\}$ (with Θ being the parameter space) and then projects g_k onto the family to obtain a sampling distribution f_{θ_k}, where the projection is implemented at each iteration by finding an optimal parameter θ_k that minimizes the Kullback-Leibler (KL) divergence between g_k and the parameterized family [34], i.e.,

$$\theta_k = \arg\min_{\theta\in\Theta} \mathscr{D}(g_k, f_\theta) := \arg\min_{\theta\in\Theta} \left(\int_{\mathbf{X}} \ln \frac{g_k(x)}{f_\theta(x)} g_k(\mathrm{d}x) \right). \tag{10.2}$$

The idea is that the parameterized family is specified with some structure (e.g., family of normal distributions parameterized by means and variances) so that once its parameter is specified, sampling from the corresponding distribution can be performed relatively easily and efficiently. Another advantage is that the task of constructing the entire surrogate distribution now simplifies to the task of finding its associated parameters. Roughly speaking, each sampling distribution f_{θ_k} obtained via (10.2) can be viewed as a compact approximation of g_k, and consequently the entire sequence $\{f_{\theta_k}\}$ may (hopefully) retain some nice properties of the distribution sequence $\{g_k\}$. Thus, to ensure the convergence of the MRAS method, it is intuitively clear that the sequence $\{g_k\}$ should be chosen in a way so that it can be shown to converge to a limiting distribution concentrated only on the set of optimal solutions. Since the distribution g_k is primarily used to guide the parameter updating process and to express the desired properties of the MRAS method, it is called the *reference* distribution.

We now provide a summary of the MRAS method:

0. Select a sequence of reference distributions $\{g_k\}$ with desired convergence properties and choose a parameterized family $\{f_\theta\}$.
1. Given θ_k, sample N candidate solutions $X_k^1, \ldots, X_k^N$ from f_{θ_k}.
2. Update the parameter θ_{k+1} by minimizing the KL divergence
$$\theta_{k+1} = \arg\min_{\theta} \mathscr{D}(g_{k+1}, f_\theta);$$
increase k by 1 and reiterate from step 1.

Note that the algorithm above assumes that the expectation/integral involved in the KL divergence (cf. (10.2)) can be evaluated exactly. In practice, it is often estimated by an empirical average based on samples obtained at step 1.

The MRAS framework accommodates many algorithms aforementioned in Sect. 10.2. For example, when Boltzmann distributions are used as reference models, the resulting algorithm becomes AAS with an additional projection step. The algorithm instantiation considered in [18] uses the following recursive procedure to construct the g_k sequence:

$$g_{k+1}(x) = \frac{H(x) g_k(x)}{\int_{\mathbf{X}} H(x) g_k(\mathrm{d}x)}, \tag{10.3}$$

where $g_0(x)$ is a given initial distribution on $\mathbf{X}$ and we have assumed for simplicity that $H(x) > 0$ for all $x \in \mathbf{X}$ to prevent negative probabilities. This form of reference distributions has also been used in a class of EDAs with proportional selection schemes. It weights the new distribution g_{k+1} by the value of the objective function $H(x)$, so that each iteration of (10.3) improves the expected performance in the sense that

$$E_{g_{k+1}}[H(X)] := \int_{\mathbf{X}} H(x) g_{k+1}(\mathrm{d}x) = \frac{\int_{\mathbf{X}} H^2(x) g_k(\mathrm{d}x)}{\int_{\mathbf{X}} H(x) g_k(\mathrm{d}x)} \geq E_{g_k}[H(X)],$$

so solutions with better performance are given more probability under g_{k+1}. This results in a $\{g_k\}$ sequence that converges to a degenerate distribution at the optimal solution. Furthermore, it is shown in [18] that the CE method can also be recovered by replacing g_k in the right-hand side of (10.3) with f_{θ_k}. In other words, there is a sequence of reference distributions implicit in CE that takes the form

$$g_{k+1}(x) = \frac{H(x) f_{\theta_k}(x)}{\int_{\mathbf{X}} H(x) f_{\theta_k}(\mathrm{d}x)}. \tag{10.4}$$

Since g_{k+1} in (10.4) is obtained by tilting the sampling distribution f_{θ_k} with the objective function H, it improves the expected performance of f_{θ_k}, i.e.,

$$E_{g_{k+1}}[H(X)] = \frac{\int_{\mathbf{X}} H^2(x) f_{\theta_k}(\mathrm{d}x)}{\int_{\mathbf{X}} H(x) f_{\theta_k}(\mathrm{d}x)} \geq \int_{\mathbf{X}} H(x) f_{\theta_k}(\mathrm{d}x) := E_{\theta_k}[H(X)].$$

Therefore, it is reasonable to expect that the projection of g_{k+1} on the parameterized family, $f_{\theta_{k+1}}$, also improves f_{θ_k}, i.e., $E_{\theta_{k+1}}[H(X)] \geq E_{\theta_k}[H(X)]$. This view of CE leads to an important monotonicity property of the method, generalizing that of [34], which is only proved for the one-dimensional case.

10.3.1 Convergence Result

For the family of natural exponential distributions (NEFs), the optimization problem involved at step 2 of the MRAS method can be solved analytically in closed form, which makes the approach very convenient to implement in practice. We recall the definition of NEFs.

Definition 10.3.1. A parameterized family $\{f_\theta, \theta \in \Theta \subseteq \Re^d\}$ is said to belong to the natural exponential family if there exist mappings $\Gamma : \Re^n \to \Re^d$ and $K : \Re^d \to \Re$ such that each f_θ in the family can be represented in the form $f_\theta(x) = \exp\left(\theta^T \Gamma(x) - K(\theta)\right)$, where $K(\theta)$ is a normalization constant given by $K(\theta) = \ln \int_{\mathbf{X}} \exp(\theta^T \Gamma(x)) \mathrm{d}x$.

The function $K(\theta)$ plays an important role in the theory of NEFs. It is strictly convex in the interior of Θ with gradient $\nabla_\theta K(\theta) = E_\theta[\Gamma(X)]$ and Hessian matrix $\mathrm{Cov}_\theta[\Gamma(X)]$. We define the mean vector function

$$m(\theta) := E_\theta[\Gamma(X)].$$

Since the Jacobian of $m(\theta)$ is strictly positive definite, we have from the inverse function theorem that $m(\theta)$ is a one-to-one invertible function of θ. Generally speaking, $m(\theta)$ can be viewed as a transformed version of the sufficient statistic $\Gamma(x)$, whose value contains all necessary information to estimate the parameter θ. For example, for the univariate normal distribution $\mathbf{N}(\mu, \sigma^2)$ with mean μ and variance σ^2, it can be seen that $\Gamma(x) = (x, x^2)^T$ and $\theta = (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2})^T$. Thus, $m(\theta) = E_\theta[\Gamma(X)]$ becomes $(\mu, \sigma^2 + \mu^2)^T$, which can be uniquely solved for μ and σ^2 given the value of $m(\theta)$.

When NEFs are used as the parameterized family, we have the following convergence theorem for the instantiation of MRAS considered in [18].

Theorem 10.3.1. *When $\{g_k\}$ in (10.3) are used as reference distributions in MRAS, let $\{\theta_k\}$ be the sequence of parameters generated by the algorithm based on the sampled candidate solutions. Under appropriate assumptions (see [18]),*

$$\lim_{k\to\infty} m(\theta_k) = \Gamma(x^*) \ w.p.1.$$

The interpretation of Theorem 10.3.1 relies on the parameterized family used in MRAS and, in particular, on the specific form of the sufficient statistic $\Gamma(x)$. We consider two special cases of Theorem 10.3.1. (a) In continuous optimization when multivariate normal distributions with mean vector μ and covariance matrix Σ are used as the parameterized family, then it is easy to show that Theorem 10.3.1 implies $\lim_{k\to\infty} \mu_k = x^*$ and $\lim_{k\to\infty} \Sigma_k = 0_{n\times n}$ $w.p.1$, where $0_{n\times n}$ represents an n-by-n zero matrix. In other words, the sequence of sampling distributions $\{f_{\theta_k}\}$ will converge to a delta distribution with all probability mass concentrated on x^*. (b) For a discrete optimization problem with feasible domain $\mathbf{X}$ that contains l distinct values denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_l$, the parameterized family can be specified in terms of an l-by-1 probability vector Q, whose ith entry q_i represents the probability that a (random) solution will take the ith value $\mathbf{x}_i$. A probability mass function on $\mathbf{X}$, when parameterized by Q, can thus be expressed as

$$f_\theta(x) = \prod_{i=1}^{l} q_i^{I\{x=\mathbf{x}_i\}} := \mathrm{e}^{\theta^T \Gamma(x)},$$

where $I\{\cdot\}$ is the indicator function, $\theta = [\ln q_1, \ldots, \ln q_l]^T$, and the sufficient statistic $\Gamma(x) = [I\{x = \mathbf{x}_1\}, \ldots, I\{x = \mathbf{x}_l\}]^T$. Therefore, a simple application of Theorem 10.3.1 yields

$$\lim_{k\to\infty} \sum_{x\in\mathbf{X}} \prod_{i=1}^{l} (q_i^k)^{I\{x=\mathbf{x}_i\}} I\{x = \mathbf{x}_j\} = I\{x^* = \mathbf{x}_j\} \ \forall j \ w.p.1,$$

where q_i^k is the ith entry of the probability vector Q_k obtained at the kth iteration of the algorithm. This in turn implies that $\lim_{k\to\infty} q_i^k = I\{x^* = \mathbf{x}_i\}$ $w.p.1.$, i.e., the sequence of Q_k will convergence to a degenerate probability vector assigning unit mass to x^*.

We remark that Theorem 10.3.1 does not address the convergence rate of the algorithm. Moreover, the proof techniques used in [18] cannot be directly carried over to analyze other algorithms such as CE, due to the dependency of g_k on the parameterized family (cf. (10.4)). In Sect. 10.6, we show that with some appropriate modifications of the MRAS method, we can arrive at a general framework linking model-based methods to recursive algorithms of stochastic approximation type, which makes the convergence and convergence rate analysis of these algorithms more tractable.

10.4 Particle-Filtering Approach

Filtering refers to the estimation of an unobserved state in a dynamical system based on noisy observations that arrive sequentially in time (c.f. [8] for an introduction). The idea behind the particle-filtering approach is to transform the optimization problem into a filtering problem. Using a novel interpretation, the distribution sequence $\{g_k\}$ in model-based optimization corresponds to the sequence of conditional distributions of the unobserved state given the observation history in filtering, and hence, $\{g_k\}$ is updated from a Bayesian perspective. A class of simulation-based filtering techniques called particle filtering can then be employed to sample from $\{g_k\}$, leading to a framework for model-based optimization algorithms.

More specifically, the optimization problem (10.1) can be transformed into a filtering problem by choosing an appropriate state-space model, such as the following:

$$\begin{aligned} X_k &= X_{k-1},\ k = 1,2,\ldots, \\ Y_k &= H(X_k) - V_k,\ k = 1,2,\ldots, \end{aligned} \tag{10.5}$$

where $X_k \in \Re^n$ is the unobserved state, $Y_k \in \Re$ is the observation, and $\{V_k, k = 1,2,\ldots\}$ is an i.i.d. sequence of nonnegative random variables that have a p.d.f. φ. A prior distribution on X_0 is denoted by g_0. The goal of filtering is to compute the conditional density g_k of the unobserved state X_k given the past observations $\{Y_1 = y_1, \ldots, Y_k = y_k\}$ for $k = 1,2,\ldots$. Let $\mathscr{F}$ denote the σ-field of Borel sets of $\Re^n$. Then the conditional density g_k satisfies

$$P(X_k \in A | Y_{1:k} = y_{1:k}) = \int_A g_k(x)\mathrm{d}x,\ \forall A \in \mathscr{F},$$

where $Y_{1:k} = \{Y_1, \ldots, Y_k\}$, and $y_{1:k} = \{y_1, \ldots, y_k\}$. Using Bayes rule, the evolution of $g_k(x)$ can be derived as follows:

$$\begin{aligned} g_k(x) &= p(x|y_{0:k-1}, y_k) \\ &= \frac{p(y_k|x)p(x|y_{0:k-1})}{p(y_k|y_{0:k-1})} \\ &= \frac{\varphi(H(x) - y_k)g_{k-1}(x)}{\int \varphi(H(x) - y_k)g_{k-1}(x)\mathrm{d}x}, \end{aligned} \tag{10.6}$$

where the last line uses the density functions induced by (10.5).

The intuition of (10.5) and (10.6) and their connection with optimization can be explained as follows: the unobserved state $\{X_k\}$ is constant with the underlying value being the optimum x^*, which needs to be estimated; the observations $\{y_k\}$ are noisy observations of the optimal function value $H(x^*)$ and come from the sample function values in an optimization algorithm; the conditional density g_k is a density estimate of the optimum x^* at iteration k based on the sample function values $\{y_1, \ldots, y_k\}$. Equation (10.6) implies that g_k is tuned the more promising area where $H(x)$ is greater than y_k since $\varphi(H(x) - y_k)$ is positive if $H(x) \geq y_k$ and is zero otherwise. Hence, randomization in the optimization algorithm is brought in by the randomness of V_k, and the choice of the p.d.f. of V_k, φ, results in different sample selection or weighting schemes in the algorithm. In order to ensure the resultant optimization algorithm monotonically approaches the optimum, the following general condition (C) on φ is imposed:

(C) The p.d.f. $\varphi(\cdot)$ is positive, strictly increasing, and continuous on its support $[0, \infty)$.

It is shown in [45] that if φ satisfies the condition, then for an arbitrary, fixed observation sequence $\{y_1, y_2, \ldots\}$, the estimate of the function value is monotonically increasing, i.e.,

$$E_{g_{k+1}}[H(X)] \geq E_{g_k}[H(X)].$$

Hence, it has the same monotonicity property as MRAS and CE. Furthermore, the estimate of the optimal function value asymptotically converges to the true optimal function value as stated in the following theorem that is also shown in [45].

Theorem 10.4.2. *Suppose the following conditions hold:*

(i) For all $H(x) < H(x^*)$, the set $\{z \in \mathbf{X} : H(z) \geq H(x)\}$ has strictly positive measure with respect to the initial sampling distribution, i.e., $\int_{\{z \in \mathbf{X}: H(z) \geq H(x)\}} g_0(x)\mathrm{d}x > 0$.

(ii) There is a unique optimum x^*, and $H(x)$ is continuous at x^*.

(iii) φ satisfies the condition (C).

Then for an arbitrary, fixed observation sequence $\{y_1, y_2, \ldots\}$,

$$\lim_{k \to \infty} E_{g_k}[H(X)] = H(x^*).$$

The conditions (i) and (ii) ensure that any neighborhood of the optimum always has a positive probability to be sampled. The result implies that the samples drawn from g_k in the limit will be concentrated on the optimum.

10.4.1 Algorithms

The distribution sequence $\{g_k\}$ in general does not have a closed-form solution. Various numerical filtering methods (cf. [5] for a recent survey) are available to numerically approximate $\{g_k\}$. However, the most akin to model-based optimization algorithms is the particle-filtering technique, which is a more recent class of approximate filtering methods based on Sequential Monte Carlo (SMC) simulation (cf. the tutorial [1] and the more recent tutorial [11] for a quick reference and the book [10] for a more comprehensive account). Despite its abundant successful applications in many areas, particle filtering has rarely been explored in optimization.

The basic particle filter is a sequential importance sampling resampling algorithm, each iteration of which is composed of an importance sampling step to propagate the particles (i.e., samples) from the previous iteration to the current, a Bayes updating step to update the weights of the particles, and a resampling step to generate new particles in order to prevent sample degeneracy. Applying it to the distribution sequence $\{g_k\}$ specified in (10.6) leads to the particle filtering for optimization (PFO) framework as follows:

0. *Initialization.* Specify g_0, and draw i.i.d. samples $\{X_1^i\}_{i=1}^{N_1}$ from g_0. Set $k = 1$.
1. *Bayes updating.* Take y_k to be a sample function value $H(X_k^i)$ according to a certain rule. Compute the weight w_k^i for sample X_k^i according to

$$w_k^i \propto \varphi(H(X_k^i) - y_k), i = 1, 2, \ldots, N_k,$$

   and normalize the weights such that they sum up to 1.
2. *Resampling.* Generate i.i.d. samples $\{X_{k+1}^i\}_{i=1}^{N_{k+1}}$ from the weighted samples $\{w_k^i, X_k^i\}_{i=1}^{N_k}$ using regularized method, density projection method, or resample-move method.
3. *Stopping.* If a stopping criterion is satisfied, then stop; else, increase k by 1 and reiterate from step 1.

Note that the simple method of sampling with replacement cannot be used in the resampling step since it does not generate new values for the samples and hence does not explore new candidate solutions for the purpose of optimization. Several other known resampling methods can be used to generate new candidate solutions and can also be easily implemented, including the regularized method [28], the density projection method [44], and the resample-move method [13]. The regularized method draws new i.i.d. samples from a continuous mixture distribution, where each continuous kernel of the mixture distribution is centered at each sample

X_k^i and the weight of that kernel is equal to the probability mass w_k^i of X_k^i. The density projection method resembles MRAS and CE in finding a parameterized density f_{θ_k} by minimizing the KL divergence between the discrete distribution $\{w_k^i, X_k^i\}$ and the parameterized family. The resample-move method applies a Markov chain Monte Carlo (MCMC) step to move the particles after they are generated by sampling with replacement. Depending on the resampling methods, the convergence properties of the different instantiations of PFO are also slightly different, but all readily follow from the existing convergence results of the corresponding particle filters in the literature [6, 14, 44] under suitable assumptions.

We end this section with a final remark that the PFO framework provides a new perspective on CE and MRAS. We will use the truncated selection scheme for sample selection as an illustration. Suppose that the objective function $H(x)$ is bounded by $H_1 \le H(x) \le H_2$. In the state-space model (10.5), let the observation noise V_k follow a uniform distribution $U(0, H_2 - H_1)$, and then φ, the p.d.f. of V_k, satisfies

$$\varphi(u) = \begin{cases} \frac{1}{H_2 - H_1}, & \text{if } 0 \le u \le H_2 - H_1; \\ 0, & \text{otherwise.} \end{cases} \tag{10.7}$$

Since y_k is a sample function value, the inequality $H(x) - y_k \le H_2 - H_1$ holds with probability 1, so substituting (10.7) into (10.6) yields

$$g_k(x) = \frac{I\{H(x) \ge y_k\} g_{k-1}(x)}{\int I\{H(x) \ge y_k\} g_{k-1}(x) \mathrm{d}x}.$$

The standard CE method can be viewed as PFO with the above choice of distribution sequence $\{g_k\}$ and the density projection method for resampling, so the samples $\{X_k^i\}$ are generated from $f_{\theta_{k-1}}$ and the weights of the samples are computed according to $w_k^i \propto I\{H(X_k^i) \ge y_k\}$. However, the approximation of g_{k-1} by $f_{\theta_{k-1}}$ introduces an approximation error, which is accumulated to the next iteration. This approximation error can be corrected by taking $f_{\theta_{k-1}}$ as an importance density and hence can be taken care of by the weights of the samples. That is, in the case of MRAS or CE in which the sequence $\{y_k\}$ is monotonically increasing, the weights are computed according to

$$w_k^i = \frac{g_k(X_k^i)}{f_{\theta_{k-1}}(X_k^i)} \propto \frac{I\{H(X_k^i) \ge y_k\}}{f_{\theta_{k-1}}(X_k^i)}.$$

This instantiation of PFO coincides with an instantiation of MRAS. More details on a unifying perspective on EDAs, CE, and MRAS are given in [45].

10.5 Evolutionary Games Approach

The main idea of the evolutionary games approach is to formulate the global optimization problem as an evolutionary game and to use dynamics from evolutionary game theory to study the evolution of the candidate solutions. Searching for the optimal solution is carried out through the dynamics of reaching equilibrium points in evolutionary games. Specifically, we establish a connection between evolutionary game theory and optimization by formulating the global optimization problem as an evolutionary game with continuous strategy spaces. We show that there is a strong connection between a particular equilibrium set of the replicator dynamics and the global optimal solutions. By using Lyapunov theory, we also show that the particular equilibrium set is asymptotically stable under mild conditions. Based on the connection between the equilibrium points and global optimal solutions, we develop a model-based evolutionary optimization (MEO) algorithm.

First, we set up an evolutionary game with a continuous strategy space. Let $\mathscr{B}$ be the Borel σ-field on $\mathbf{X}$, the strategy space of the game; for each t, let $\mathbb{P}_t$ be a probability measure defined on $(\mathbf{X}, \mathscr{B})$. Let Δ denote set of all the strategies (probability measures) on $\mathbf{X}$. Each point $x \in \mathbf{X}$ can be viewed as a pure strategy. Roughly speaking, the fraction of agents playing the pure strategy x at time t is $\mathbb{P}_t(\mathrm{d}x)$. An agent playing the pure strategy x obtains a fitness $\phi(H(x))$, where $\phi(\cdot) : \Re \to \Re^+$ is a strictly increasing function. An appropriate chosen $\phi(\cdot)$ can facilitate the expression of the model updating rule presented later. Let X be a random variable with probability distribution $\mathbb{P}_t$. The fractions of agents adopting different strategies in the continuous game is described by the probability measure $\mathbb{P}_t$ defined on the strategy space $\mathbf{X}$, so the average payoff of the whole population is given by

$$E_{\mathbb{P}_t}[\phi(H(X))] = \int_{\mathbf{X}} \phi(H(x))\mathbb{P}_t(\mathrm{d}x).$$

In evolutionary game theory [29], the evolution of this probability measure is governed by some dynamics such as the so-called replicator dynamics. Let $\mathscr{A}$ be a measurable set in $\mathbf{X}$. If the replicator dynamics with a continuous strategy space is adopted, we have

$$\dot{\mathbb{P}}_t(\mathscr{A}) = \int_{\mathscr{A}} (\phi(H(x)) - E_{\mathbb{P}_t}[\phi(H(X))])\mathbb{P}_t(\mathrm{d}x). \tag{10.8}$$

From (10.8), we can see that if $\phi(H(x))$ outperforms $E_{\mathbb{P}_t}[\phi(H(X))]$ at x, the probability measure around x will increase. If there exists a probability density function p_t, such that $\mathbb{P}_t(\mathrm{d}x) = p_t\mu(\mathrm{d}x)$, where $\mu(\cdot)$ is the Lebesgue measure defined on $(\mathbf{X}, \mathscr{B})$, then (10.8) becomes

$$\dot{p}_t(x) = (\phi(H(x)) - E_{\mathbb{P}_t}[\phi(H(X))])p_t(x), \tag{10.9}$$

which governs the evolution of the probability density function on the continuous strategy space. When $p_t(x)$ is used as our model to generate candidate solutions for the global optimization problem (10.1), the differential equation (10.9) can be used to update the model $p_t(x)$, with the final goal of making the probability density function $p_t(x)$ converge to a small set containing the global optimal solution. Then, the global optimization problem can be easily solved by sampling from the obtained probability density function.

10.5.1 Convergence Analysis

In this section, we study the properties of the equilibrium points of (10.8) and their connection with the global optimal solutions for the optimization problem, by employing the tools of equilibrium analysis in game theory and stability analysis in dynamic systems.

Assume that the optimization problem (10.1) has m global optimal solutions $\{x_i^\star, i = 1,\ldots,m\}$. It is easy to see that $\mathbb{P}^{(x)} = \delta(x - x_i^\star)$ for $i = 1,\ldots,m$ are equilibrium points of (10.8), and we might guess there is a strong connection between the equilibrium points of (10.8) and the optimal solutions of the global optimization problem (10.1). We enforce the following assumption on function ϕ.

Assumption 10.5.1 *$\phi(\cdot)$ is a continuous and strictly increasing function; there exist constants $\mathscr{L}$ and $\mathscr{M}$ such that $\mathscr{L} \le \phi(H(x)) \le \mathscr{M}$ for all $x \in \mathbf{X}$.*

The following theorem shows that the overall fitness of the strategy (probability measure) $\mathbb{P}_t$ is monotonically increasing over time.

Theorem 10.5.3. *Let $\mathbb{P}_t$ be a solution of the replicator dynamics (10.8). Under Assumption 10.5.1, the average payoff of the entire population $E_{\mathbb{P}_t}[\phi(H(X))]$ is monotonically increasing with time t. If $\mathbb{P}_t$ is not an equilibrium point of (10.8), then $E_{\mathbb{P}_t}[\phi(H(X))]$ is strictly increasing with time t.*

To further study the properties of the equilibrium points of the replicator dynamics (10.8), the Prokhorov metric is used to measure the distance between different strategies (probability measures):

$$\rho(\mathbb{P},\mathbb{Q}) := \inf\{\varepsilon > 0 : \mathbb{Q}(\mathscr{A}) \le \mathbb{P}(\mathscr{A}^\varepsilon) + \varepsilon \text{ and } \mathbb{P}(\mathscr{A}) \le \mathbb{Q}(\mathscr{A}^\varepsilon) + \varepsilon, \quad \forall \mathscr{A} \in \mathscr{B}\},$$

where $\mathscr{A}^\varepsilon := \{x : \exists \tilde{y} \in \mathscr{A}, d(\tilde{y},x) < \varepsilon\}$, in which d is a metric defined on $\mathbf{X}$. Then, the convergence of $\rho(\mathbb{Q}_n,\mathbb{Q}) \to 0$ is equivalent to the weak convergence of $\mathbb{Q}_n$ to $\mathbb{Q}$ [3].

Definition 10.5.2. Let $\mathscr{E}$ be a set in Δ. For a point $\mathbb{P} \in \Delta$, define the distance between $\mathbb{P}$ and $\mathscr{E}$ as $\rho(\mathbb{P},\mathscr{E}) = \inf\{\rho(\mathbb{P},\mathbb{Q}), \forall \mathbb{Q} \in \mathscr{E}\}$. $\mathscr{E}$ is called Lyapunov stable if for all $\varepsilon > 0$, there exists $\eta > 0$ such that $\rho(\mathbb{P}_0,\mathscr{E}) < \eta \Longrightarrow \rho(\mathbb{P}_t,\mathscr{E}) < \varepsilon$ for all $t > 0$.

Definition 10.5.3. Let $\mathscr{E}$ be a set in Δ. $\mathscr{E}$ is called asymptotically stable if $\mathscr{E}$ is Lyapunov stable and there exists $\eta > 0$ such that $\rho(\mathbb{P}_0, \mathscr{E}) < \eta \Longrightarrow \rho(\mathbb{P}_t, \mathscr{E}) \to 0$ as $t \to \infty$.

Definition 10.5.4. $\Delta_0 \subset \Delta$ is the set containing all $\mathbb{P}_0$ for which there exists a $x_k^\star$ such that $\mathbb{P}_0(\tilde{\mathscr{A}}) > 0$ for any set $\tilde{\mathscr{A}} \in \mathscr{B}$ that contains $x_k^\star$ and has a positive Lebesgue measure $\mu(\tilde{\mathscr{A}}) > 0$. Let $\mathscr{C} = \{\mathbb{P}^\star : \mathbb{P}^\star = \lim_{t\to\infty} \mathbb{P}_t$ starting from some $\mathbb{P}_0 \in \Delta_0\}$.

To present the main convergence result, we also need the following assumption.

Assumption 10.5.2 *There is a finite number of global optimal solutions $\{x_1^\star, \ldots, x_m^\star\}$ for the optimization problem (10.1), where m is a positive integer.*

Theorem 10.5.4. *If Assumptions 10.5.1 and 10.5.2 hold, then for any $\mathbb{P}^\star \in \mathscr{C}$, there exist $\alpha_i \geq 0$, for $i = 1, \ldots, m$ with $\sum_{i=1}^m \alpha_i = 1$ such that $\mathbb{P}^{(x)} = \sum_{i=1}^m \alpha_i \delta(x - x_i^\star)$; the set $\mathscr{C}$ can be represented as $\mathscr{C} = \{\mathbb{P}^\star : \mathbb{P}^{(x)} = \sum_{i=1}^m \alpha_i \delta(x - x_i^\star)$, for some $\sum_{i=1}^m \alpha_i = 1, \alpha_i \geq 0, \forall i = 1, \ldots, m\}$, and in addition, the set $\mathscr{C}$ is asymptotically stable.*

10.5.2 Model-Based Evolutionary Optimization

From the above analysis, we know that the global optimal solutions can be obtained by generating samples from equilibrium distributions of the replicator dynamics (10.8); these equilibrium distributions can be approached by following trajectories of (10.8) starting from $\mathbb{P}_0 \in \Delta_0$. Note that by Theorem 10.5.4, the equilibrium points obtained by starting from $\mathbb{P}_0 \in \Delta_0$ are of the form $\mathbb{P}^{(x)} = \sum_{i=1}^m \alpha_i \delta(x - x_i^\star)$, where $\sum_{i=1}^m \alpha_i = 1$ and $\alpha_i \geq 0$ for $i = 1, \ldots, m$, which suggests using a sum of Dirac functions to approximate p_t. Assume a group of candidate solutions $\{y_t^i\}_{i=1}^N$ is generated from p_t; then the probability density function p_t can be approximated by $\hat{p}_t(x) = \sum_{i=1}^N w_t^i \delta(x - x_t^i)$, where δ denotes the Dirac function and $\{w_t^i\}_{i=1}^N$ are weights satisfying $\sum_{i=1}^N w_t^i = 1$. If we use this approximation $\hat{p}_t$ as our probabilistic model and substitute it into (10.9), we have

$$\frac{\partial w_t^i}{\partial t} = \Big(\phi(H(x_t^i)) - \sum_{j=1}^N w_t^j \phi(H(x_t^j))\Big) w_t^i, \quad \forall i = 1, \ldots, N. \tag{10.10}$$

The corresponding discrete-time version of (10.10) is

$$w_{k+1}^i = \frac{\phi(H(x_k^i))}{\sum_{j=1}^N w_k^j \phi(H(x_k^j))} w_k^i, \quad \forall i = 1, \ldots, N. \tag{10.11}$$

We can let $\phi(\cdot)$ be an exponential function so that the denominator of the right-hand side of (10.11) is not equal to zero. Although an updated density approximation $\hat{p}_{k+1}(x) = \sum_{i=1}^N w_{k+1}^i \delta(x - x_k^i)$ is obtained, it cannot be used directly to generate new candidate solutions. We construct a new continuous density to approximate $\hat{p}_{k+1}$,

which is done by projecting $\hat{p}_{k+1}$ onto some parameterized family of distributions g_θ. The idea of projection onto a parameterized family has also been used in CE and MRAS, as discussed above. Specifically, we minimize the KL divergence between the parameterized distribution g_θ and $\hat{p}_{k+1}$:

$$\theta_{k+1} = \arg\min_{\theta\in\Theta} \mathscr{D}\big(\hat{p}_{k+1}, g_\theta\big), \tag{10.12}$$

where Θ is the domain of θ. After some algebraic operations, we can show that solving (10.12) is equivalent to: $\max_{\theta\in\Theta} \sum_{i=1}^{N} w_{k+1}^i \ln g_\theta(y_k^i)$.

All the above analysis is carried out when replicator dynamics, e.g., (10.8) and (10.9), are used. There are some other dynamics in evolutionary game theory such as imitation dynamics, logit dynamics, and Brown-von Neumann-Nash dynamics that can be used to update the weights $\{w_k^i\}$. To present the algorithm in a more general setting, the updating of weights is denoted as

$$w_k^i = D_d\Big(\phi(H(x_{k-1}^i))I_{\{H(x_{k-1}^i)\geq\gamma_{k-1}\}}, \sum_{j=1}^{N} w_{k-1}^j \phi(H(x_{k-1}^j))I_{\{H(x_{k-1}^j)\geq\gamma_{k-1}\}}, w_{k-1}^i\Big), \tag{10.13}$$

where γ_{k-1} is a constant that is used to select good candidate solutions; D_d is a function of three variables, which is used to represent the updating rule. For example, when D_d is derived from replicator dynamics, we have

$$w_k^i = \frac{\frac{1}{N}\phi(H(x_{k-1}^i))I_{\{H(x_{k-1}^i)\geq\gamma_{k-1}\}}}{\sum_{j=1}^{N}\frac{1}{N}\phi(H(x_{k-1}^j))I_{\{H(x_{k-1}^j)\geq\gamma_{k-1}\}}} w_{k-1}^i, \quad \forall i = 1,\ldots,N.$$

Based on the above analysis, a Monte Carlo simulation version of the MEO algorithm is given as follows.

Model-Based Evolutionary Optimization Algorithm (MEO)

0. Initialization. Specify N as the total number of candidate solutions generated at each iteration. Choose $\rho \in (0,1]$ and an initial g_{θ_0} defined on $\mathbf{X}$. Set $k = 0$, $w_0^i = 1/N$ for $i = 1,\ldots,N$, and $\gamma_0 = -\infty$.
1. Quantile calculation. Generate N candidate solutions $\{x_k^i\}_{i=1}^N$ from g_{θ_k}. Calculate the $1-\rho$ quantile γ_k of $\{x_k^i\}_{i=1}^N$. If $\gamma_k < \gamma_{k-1}$ and $k > 1$, set $\gamma_k = \gamma_{k-1}$ and $w_{k-1}^i = 1/N$ for $i = 1,\ldots,N$. Set $k = k+1$ and go to Step 2.
2. Updating the probabilistic model. The discrete approximation of the model is $\hat{p}_k(x) = \sum_{i=1}^{N} w_k^i \delta(x - x_{k-1}^i)$, where $\{w_k^i\}$ are updated according to (10.13).
3. Density projection. Construct g_θ by projecting the density $\hat{p}_k = \sum_{i=1}^{N} w_k^i \delta(x - x_{k-1}^i)$ onto g_θ: $\theta_k = \arg\max_{\theta\in\Theta} \sum_{i=1}^{N} w_k^i \ln g_\theta(x_{k-1}^i)$.
4. Stop if some stopping criterion is satisfied; otherwise go to Step 1.

Generally, it is not easy to solve the optimization problem (10.12), which depends on the choice of g_θ. However, for g_θ in an exponential family, analytical solutions can be obtained. A comprehensive exposition of the evolutionary games approach is given in [37, 38].

10.6 Stochastic Approximation Approach

In this section, we present a stochastic approximation framework to study model-based algorithms [21]. The framework is based on the MRAS method presented in Sect. 10.3 and is intended to combine the robust features of model-based algorithms encountered in practice with rigorous convergence guarantees. Specifically, by exploiting a natural connection between model-based algorithms and the well-known stochastic approximation (SA) method [2,4,26,30], we show that, regardless of the type of decision variables involved in (10.1), algorithms conforming to the framework can be equivalently formulated in the form of a generalized stochastic approximation procedure on a transformed continuous parameter space for solving a sequence of stochastic optimization problems with differentiable structures. This viewpoint, which is new to this type of random search algorithms, allows us to study the asymptotic convergence and rate properties of these algorithms by using existing theory and tools from SA.

The key idea that leads to the proposed framework is based on replacing the reference sequence $\{g_k\}$ in the original MRAS method by a more general distribution sequence in the recursive form:

$$\hat{g}_{k+1}(x) = \alpha_k g_{k+1}(x) + (1-\alpha_k) f_{\theta_k}(x),\ \alpha_k \in (0,1)\ \forall k, \tag{10.14}$$

which is a mixture of the reference distribution g_{k+1} and the sampling distribution f_{θ_k} obtained at the kth iteration. Such a mixture $\hat{g}_{k+1}$ retains the properties of g_{k+1} while, on the other hand, ensures that its difference from f_{θ_k} is only incremental. Thus, the intuition is that if one were to replace g_{k+1} with $\hat{g}_{k+1}$ in minimizing the KL divergence $\mathscr{D}(\hat{g}_{k+1}, f_\theta)$, then the new sampling distribution $f_{\theta_{k+1}}$ obtained would also stay close to the current sampling distribution f_{θ_k}.

When $\{\hat{g}_k\}$ instead of $\{g_k\}$ is used at step 2 of MRAS to minimize the KL divergence, the following lemma reveals a key link between the two successive mean vector functions of the projected probability distributions [21].

Lemma 10.6.1. *If f_θ belongs to NEFs and the new parameter θ_{k+1} obtained via minimizing $\mathscr{D}(\hat{g}_{k+1}, f_\theta)$ is an interior point of the parameter space Θ for all k, then*

$$m(\theta_{k+1}) - m(\theta_k) = -\alpha_k \nabla_\theta \mathscr{D}(g_{k+1}, f_\theta)|_{\theta=\theta_k}. \tag{10.15}$$

Basically, Lemma 10.6.1 states that regardless of the specific form of g_k, the mean vector function $m(\theta_k)$ (i.e., a one-to-one transformation of θ_k) is updated at each step

along the gradient descent direction of the *time-varying* objective function for the minimization problem $\min_\theta \mathscr{D}(g_{k+1}, f_\theta)$. In particular, in the case of the CE method, i.e., when g_{k+1} in (10.15) takes the form $g_{k+1}(x) = \frac{H(x) f_{\theta_k}(x)}{\int_{\mathbf{X}} H(x) f_{\theta_k}(\mathrm{d}x)}$ (cf. (10.4)), it can be seen that recursion (10.15) becomes

$$m(\theta_{k+1}) - m(\theta_k) = \alpha_k \nabla_\theta \ln E_\theta[H(X)]|_{\theta=\theta_k}. \tag{10.16}$$

Hence, $m(\theta_k)$ is updated along the gradient direction of the objective function for the maximization problem $\max_\theta \ln E_\theta[H(X)]$, the optimal solution to which is a sampling distribution f_{θ^*} that assigns maximum probability to the set of optimal solutions of (10.1). Note that the parameter sequence $\{\alpha_k\}$ turns out to be the gain sequence for the gradient iteration, so that the special case $\alpha_k \equiv 1$ corresponds to the original MRAS method. This suggests that all model-based algorithms that fall under the MRAS framework can be equivalently viewed as gradient-based recursions on the parameter space Θ for solving a sequence of optimization problems with differentiable structures. This new interpretation of model-based algorithms provides a key insight to understand how these algorithms address hard optimization problems with little structure.

In actual implementation, when integrals/expectations are replaced by sample averages based on Monte Carlo sampling, (10.15) and (10.16) become recursive algorithms of stochastic approximation type with direct gradient estimation. Thus, it is clear that the rich body of tools and results from stochastic approximation can be incorporated into the framework to analyze model-based algorithms.

10.6.1 Convergence of the CE Method

The convergence of the CE algorithm has recently been studied in [19,21] by casting a Monte Carlo version of recursion (10.16) in the form of a generalized Robbins-Monro algorithm in terms of the true gradient, bias, and an error term due to random sampling and then following the arguments of the ordinary differential equation (ODE) approach [2,4]. The main convergence results are summarized below, where for notational convenience, we define $\eta := m(\theta)$ and $\eta_k := m(\theta_k)$.

Theorem 10.6.5. *(Convergence of CE) Under some regularity conditions (see [21]), the sequence of iterates $\{\eta_k\}$ generated by the CE algorithm converges w.p.1 to a compact connected internally chain recurrent set of the ODE*

$$\frac{\mathrm{d}\eta(t)}{\mathrm{d}t} = L(\eta),\ t \geq 0, \tag{10.17}$$

where $L(\eta) := \nabla_\theta \ln E_\theta[H(X)]|_{\theta=m^{-1}(\eta)}$.

Theorem 10.6.5 indicates that the long-run behavior (e.g., local/global convergence) of CE is primarily governed by the asymptotic solution of an underlying ODE. This result formalizes our prior observation in [18], which provides counterexamples indicating that CE and its variants are in general local improvement methods. Under the more stringent assumption that the convergence of $\{\eta_k\}$ occurs to a unique limiting point η^*, the following asymptotic normality result was obtained in [21].

Theorem 10.6.6. *(Asymptotic normality of CE) Under some appropriate conditions (see Theorem* 4.1 *of [21]),*

$$k^{\frac{\tau}{2}}(\eta_k - \eta^*) \xrightarrow{dist} \mathbf{N}\big(0, \Sigma\big) \text{ as } k \to \infty,$$

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

10.6.2 Model-Based Annealing Random Search

To further illustrate the stochastic approximation approach, we present an algorithm instantiation of the framework called model-based annealing random search (MARS) [20]. MARS can essentially be viewed as an implementable version of the annealing adaptive search (AAS) algorithm, in that it provides an alternative approach to address the implementation difficulty of AAS (cf. Sect. 10.2). The basic idea is to use a sequence of NEF distributions to approximate the target Boltzmann distributions and then use the sequence as surrogate distributions to generate candidate points. Thus, by treating Boltzmann distributions as reference distributions, candidate solutions are drawn at each iteration of MARS *indirectly* from a Boltzmann distribution by sampling exactly from its approximation. This is in contrast to Markov chain-based techniques [41] that aim to *directly* sample from the Boltzmann distributions.

The MARS algorithm is conceptually very simple and is summarized below:

0. Choose a parameterized family $\{f_\theta\}$, an annealing schedule used in the Boltzmann distribution, and a gain sequence $\{\alpha_k\}$.
1. Given θ_k, sample N candidate solutions $X_k^1, \ldots, X_k^N$ from f_{θ_k}.
2. Update the parameter $\theta_{k+1} = \arg_\theta \min \mathscr{D}(\widetilde{g}_{k+1}, f_\theta)$; increase k by 1 and reiterate from step 1.

At Step 2 of MARS, the reference distribution is given by $\widetilde{g}_{k+1}(x) = \alpha_k \bar{g}_{k+1}(x) + (1-\alpha_k) f_{\theta_k}(x)$, where $\bar{g}_{k+1}$ is an empirical estimate of the true Boltzmann distribution $g_{k+1}(x) := \frac{\mathrm{e}^{H(x)/T_k}}{\int_{\mathbf{X}} \mathrm{e}^{H(x)/T_k}\mathrm{d}x}$ based on the sampled solutions $X_k^1, \ldots, X_k^N$, and $\{T_k\}$ is a sequence of decreasing temperatures that controls how fast the sequence of Boltzmann distributions will degenerate.

Under its equivalent gradient interpretation, Lemma 10.6.1 shows that the mean vector function $m(\theta_{k+1})$ of the new distribution $f_{\theta_{k+1}}$ obtained at step 2 of MARS can be viewed as an iterate generated by a gradient descent algorithm for solving the iteration-varying minimization problem $\min_\theta \mathscr{D}(\bar{g}_{k+1}, f_\theta)$ on the parameter space Θ, i.e.,

$$m(\theta_{k+1}) - m(\theta_k) = -\alpha_k \nabla_\theta \mathscr{D}(\bar{g}_{k+1}, f_\theta)|_{\theta=\theta_k}. \tag{10.18}$$

Note that since the reference distribution $\bar{g}_{k+1}$ may change shape with k, a primary difference between MARS and CE is that the gradient in (10.18) is time-varying vs. stationary in (10.16). Stationarity in general only guarantees local convergence, whereas the time-varying feature of MARS provides a viable way to ensure that the algorithm escapes from local optima, leading to global convergence. By the properties of NEFs, recursion (10.18) can be further written as

$$\begin{aligned} m(\theta_{k+1}) - m(\theta_k) &= -\alpha_k \big(m(\theta_k) - E_{g_{k+1}}[\Gamma(X)] + E_{g_{k+1}}[\Gamma(X)] - E_{\bar{g}_{k+1}}[\Gamma(X)]\big) \\ &= -\alpha_k \nabla_\theta \mathscr{D}(g_{k+1}, f_\theta)|_{\theta=\theta_k} - \alpha_k \big(E_{g_{k+1}}[\Gamma(X)] - E_{\bar{g}_{k+1}}[\Gamma(X)]\big). \end{aligned}$$

This becomes a Robbins-Monro-type stochastic approximation algorithm in terms of the true gradient and a noise term due to the approximation error between g_{k+1} and $\bar{g}_{k+1}$. Thus, in light of the existing theories from stochastic approximation, the convergence analysis of MARS essentially boils down to the issue of inspecting whether the Boltzmann distribution g_{k+1} can be closely approximated by its empirical estimate $\bar{g}_{k+1}$. The following results are obtained in [20].

Theorem 10.6.7. *(Global convergence of MARS) Under some appropriate conditions (see Theorem 3.1 of [20]),*

$$\lim_{k\to\infty} m(\theta_k) = \Gamma(x^*) \ w.p.1.$$

Theorem 10.6.8. *(Asymptotic normality of MARS) Let $\alpha_k = a/k^\alpha$ and the sample size be polynomially increasing $N_k = ck^\beta$ for constants $a > 0$, $c > 0$, $\alpha \in (\frac{1}{2}, 1)$, and $\beta > \alpha$. Under some additional conditions on $\{T_k\}$,*

$$k^{\frac{\alpha+\beta}{2}} \big(m(\theta_k) - \Gamma(x^*)\big) \xrightarrow{\text{dist}} \mathbf{N}(0, \Sigma) \ as \ k \to \infty,$$

where Σ is some positive definite covariance matrix.

Numerical results on high-dimensional multi-extremal benchmark problems reported in [20] show that MARS may yield high-quality solutions within a modest number of function evaluations and provide superior performance over some of the existing algorithms.

10.7 Conclusions

We reviewed several recent contributions to model-based methods for global optimization, including algorithms and convergence results for model reference adaptive search, the particle-filtering approach, the evolutionary games approach, and the stochastic approximation gradient approach. These approaches analyze model-based methods from different perspectives, providing useful tools to explore properties of the updating mechanism of probabilistic models and to facilitate proofs of convergence of model-based algorithms.

Acknowledgements This work was supported in part by the National Science Foundation (NSF) under Grants CNS-0926194, CMMI-0856256, CMMI-0900332, CMMI-1130273, CMMI-1130761, EECS-0901543, and by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-10-1-0340.

References

1. Arulampalam, S., Maskell, S., Gordon, N.J., Clapp, T.: A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing **50**(2), 174–188 (2002)
2. Benaim, M.: A dynamical system approach to stochastic approximations. SIAM Journal on Control and Optimization **34**, 437–472 (1996)
3. Billingsley, P.: Convergence of Probability Measures. John Wiley & Sons, Inc., New York (1999)
4. Borkar, V.: Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press; New Delhi: Hindustan Book Agency (2008)
5. Budhiraja, A., Chen, L., Lee, C.: A survey of numerical methods for nonlinear filtering problems. Physica D: Nonlinear Phenomena **230**, 27–36 (2007)
6. Chopin, N.: Central limit theorem for sequential Monte Carlo and its applications to Bayesian inference. The Annals of Statistics **32**(6), 2385–2411 (2004)
7. Costa, A., Jones, O., Kroese, D.: Convergence properties of the cross-entropy method for discrete optimization. Operations Research Letters (35), 573–580 (2007)
8. Davis, M.H.A., Marcus, S.I.: An introduction to nonlinear filtering. The Mathematics of Filtering and Identification and Applications. Amsterdam, The Netherlands, Reidel (1981)
9. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation **1**, 53–66 (1997)
10. Doucet, A., deFreitas, J.F.G., Gordon, N.J. (eds.): Sequential Monte Carlo Methods In Practice. Springer, New York (2001)
11. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of Nonlinear Filtering. Cambridge University Press, Cambridge (2009)
12. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Natural Computing Series, Springer (2003)
13. Gilks, W., Berzuini, C.: Following a moving target - Monte Carlo inference for dynamic Bayesian models. Journal of the Royal Statistical Society **63**(1), 127–146 (2001)
14. Gland, F.L., Oudjane, N.: Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filter. The Annals of Applied Probability **14**(1), 144–187 (2004)
15. Glover, F.W.: Tabu search: A tutorial. Interfaces **20**, 74–94 (1990)

16. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston, MA (1989)
17. Homem-De-Mello, T.: A study on the cross-entropy method for rare-event probability estimation. INFORMS Journal on Computing **19**, 381–394 (2007)
18. Hu, J., Fu, M.C., Marcus, S.I.: A model reference adaptive search method for global optimization. Operations Research **55**(3), 549–568 (2007)
19. Hu, J., Hu, P.: On the performance of the cross-entropy method. In: Proceedings of the 2009 Winter Simulation Conference, pp. 459–468 (2009)
20. Hu, J., Hu, P.: Annealing adaptive search, cross-entropy, and stochastic approximation in global optimization. Naval Research Logistics **58**, 457–477 (2011)
21. Hu, J., Hu, P., Chang, H.: A stochastic approximation framework for a class of randomized optimization algorithms. IEEE Transactions on Automatic Control (forthcoming) (2012)
22. Jacobson, S., Sullivan, K., Johnson, A.: Discrete manufacturing process design optimization using computer simulation and generalized hill climbing algorithms. Engineering Optimization **31**, 247–260 (1998)
23. Johnson, A., Jacobson, S.: A class of convergent generalized hill climbing algorithms. Applied Mathematics and Computation **125**, 359–373 (2001)
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks. IEEE Press, Piscataway, NJ, pp. 1942–1948 (1995)
25. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
26. Kushner, H.J., Clark, D.S.: Stochastic Approximation Methods for Constrained and Unconstrained Systems and Applications. Springer-Verlag, New York (1978)
27. Larrañaga, P., Lozano, J. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publisher, Boston, MA (2002)
28. Musso, C., Oudjane, N., Gland, F.L.: Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
29. Oechssler, J., Riedel, F.: On the dynamics foundation of evolutionary stability in continuous models. Journal of Economic Theory **107**, 223–252 (2002)
30. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics **22**, 400–407 (1951)
31. Romeijn, H., Smith, R.: Simulated annealing and adaptive search in global optimization. Probability in the Engineering and Informational Sciences **8**, 571–590 (1994)
32. Rubinstein, R.Y.: Optimization of computer simulation models with rare events. European Journal of Operations Research **99**, 89–112 (1997)
33. Rubinstein, R.Y.: The cross-entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability **2**, 127–190 (1999)
34. Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning. Springer, New York, NY (2004)
35. Shi, L., Olafsson, S.: Nested partitions method for global optimization. Operations Research **48**(3), 390–407 (2000)
36. Srinivas, M., Patnaik, L.M.: Genetic algorithms: A survey. IEEE Computer **27**, 17–26 (1994)
37. Wang Y., Fu, M.C., Marcus, S.I.: Model-based evolutionary optimization. In Proceedings of the 2010 Winter Simulation Conference. IEEE Press, Piscataway, NJ, pp. 1199–1210 (2010)
38. Wang, Y., Fu, M.C., Marcus, S.I.: An evolutionary game approach for model-based optimization. Working paper (2011)
39. Wolpert, D.H. Finding bounded rational equilibria part i: Iterative focusing. In Proceedings of the Eleventh International Symposium on Dynamic Games and Applications, T. Vincent (Editor), Tucson AZ, USA (2004)
40. Zabinsky, Z., Smith, R., McDonald, J., Romeijn, H., Kaufman, D.: Improving hit-and-run for global optimization. Journal of Global Optimization **3**, 171–192 (1993)
41. Zabinsky, Z.B.: Stochastic Adaptive Search for Global Optimization. Kluwer, The Netherlands (2003)

42. Zhang, Q., Mühlenbein, H.: On the convergence of a class of estimation of distribution algorithm. IEEE Trans. on Evolutionary Computation **8**, 127–136 (2004)
43. Zhou, E., Fu, M.C., Marcus, S.I.: A particle filtering framework for randomized optimization algorithms. In Proceedings of the 2008 Winter Simulation Conference. IEEE Press, Piscataway, NJ, pp. 647–654 (2008)
44. Zhou, E., Fu, M.C., Marcus, S.I.: Solving continuous-state POMDPs via density projection. IEEE Transactions on Automatic Control **55**(5), 1101–1116 (2010)
45. Zhou, E., Fu, M.C., Marcus, S.I.: Particle filtering framework for a class of randomized optimization algorithms. Under review (2012)
46. Zlochin, M., Birattari, M., Meuleau, N., Dorigo, M.: Model-based search for combinatorial optimization: A critical survey. Annals of Operations Research **131**, 373–395 (2004)