Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Moench, O. Rose, eds.

A PARTICLE FILTERING FRAMEWORK FOR RANDOMIZED OPTIMIZATION ALGORITHMS

Enlu Zhou
Michael C. Fu
Steven I. Marcus

Institute for Systems Research
University of Maryland, College Park
College Park, MD 20742, U.S.A.

ABSTRACT

We propose a framework for optimization problems based on particle filtering (also called Sequential Monte Carlo method). This framework unifies and provides new insight into randomized optimization algorithms. The framework also sheds light on developing new optimization algorithms through the freedom in the framework and the various improving techniques for particle filtering.

1 INTRODUCTION

Many randomized optimization methods, such as the estimation of distribution algorithms (EDAs) (Muhlenbein and Paaß 1996, Lozano et al. 2006), the cross-entropy (CE) method (Rubinstein 1999, Rubinstein and Kroese 2004), and model reference adaptive search (MRAS) (Hu et al. 2007), fall into the category of model-based methods as classified in Zlochin et al. (2004). They share the similarities of iteratively repeating two steps: (1) generate candidate solutions from a distribution over the solution space, and (2) update the distribution using the candidate solutions. This distribution is often called a probabilistic model, as it often imposes a model on the relationship between the components that are needed to represent a solution. The choice and updating of the probabilistic model play a key role in determining the efficiency and accuracy of the algorithm. EDAs, CE and MRAS are very different in the updating procedure of the probabilistic model.

EDAs were first proposed by Muhlenbein and Paaß (1996) in the field of evolutionary computation, with the goal of eliminating the mutation and cross-over operations in genetic algorithms (GAs) in order to avoid partial solutions. EDAs generate offspring by sampling from a distribution over the solution space that is estimated from the candidate solutions of the previous iteration. The estimation of this distribution is often based on a probabilistic model that explicitly expresses the relationship between the underlying variables (Larranaga et al. 1999).

The cross-entropy (CE) method was originally introduced for estimating probabilities of rare events in complex stochastic networks (Rubinstein 1997), and later was modified slightly to be used for solving combinatorial and continuous optimization problems (Rubinstein 1999). A key idea of the CE method is to minimize the Kullback-Leibler (KL) divergence between a desired density (the optimal importance sampling density) and a family of parameterized densities, in particular an exponential family, since an analytical solution can be calculated in this case.

Hu et al. introduced the MRAS method in (Hu et al. 2007), which incorporates the key ideas of EDAs and the CE method. MRAS implicitly constructs a sequence of reference distributions and uses this sequence to facilitate and guide the parameter updating associated with a family of parameterized distributions. At each iteration, candidate solutions are sampled from the distribution (in the prescribed family) that has the minimum KL divergence with respect to the reference distribution of the previous iteration.

The aforementioned various ways of updating the probabilistic model motivate us to look for a unifying and systematic approach to the model-based methods for optimization. In this paper, we introduce a unifying framework based on particle filtering (also called sampling importance resampling (Arulampalam et al. 2002), and sequential Monte Carlo method (Doucet et al. 2001)). Particle filtering is a class of Monte Carlo simulation-based methods for recursively estimating the conditional density of the current state based on the observation history in a dynamic system. It was first introduced in Gordon et al. (1993), and soon gained popularity and has had a significant impact in many areas such as signal processing, computer vision, estimation and control (Doucet et al. 2001, Cappé et al. 2007). However, it has never been applied to the field of optimization, as far as we know.

We consider the global optimization problem:

$$x^* = \arg\max_{x \in \mathscr{X}} H(x),$$

which satisfies certain technical conditions and has a unique global optimal solution. Our main idea is to formulate the optimization problem as a filtering problem, in which the optimal solution is a static state to be estimated, and the conditional density of the state approaches a delta function concentrated on the optimal solution as the system evolves. The task of searching for the optimal solutions is carried out through the procedure of estimating the conditional density sequentially, and hence it is natural to apply particle filtering. We propose a plain particle filtering framework and a general particle filtering framework for optimization, where the former framework is a special case of the latter one and more intuitive, while the latter framework is a generalization and hence provides more opportunities for developing new algorithms.

The particle filtering framework provides new insights into the randomized optimization methods from another viewpoint, and sheds light on developing new optimization algorithms. For example, the CE method fits in the plain particle filtering framework, and in particular, the CE method corresponds to the projection particle filtering in Zhou et al. (2008). The possibilities of new algorithms come from the freedom in the particle filtering framework, as well as the rich literature of particle filtering, which includes many improving techniques that can be adapted to optimization.

The rest of the paper is organized as follows. In section 2, we give an overview of the particle filtering method. In section 3, we develop a particle filtering framework for randomized optimization algorithms. In section 4, we use the particle filtering framework to interpret CE. We discuss the directions for developing new algorithms and outline some future research in section 5.

2 OVERVIEW OF PARTICLE FILTERING

In this paper, we abuse notations a little: we use lower case letters to denote both random variables and realizations of the random variables.

Consider the discrete-time state-space model

$$\begin{aligned} x_k &= f(x_{k-1},u_k), k=1,2,\ldots, \\ y_k &= h(x_k,v_k), k=0,1,\ldots, \end{aligned} \tag{1}$$

where for all k, $x_k \in R^{n_x}$ is the state , $y_k \in R^{n_y}$ is the observation, $u_k \in R^{n_x}$ is the system noise, $v_k \in R^{n_y}$ is the observation noise, and n_x and n_y are the dimensions of x_k and y_k, respectively. We assume $\{u_k\}$ and $\{v_k\}$ are independent and identically distributed (i.i.d.) sequences, independent of each other, and also independent of the initial state x_0, which has the probability density function (p.d.f.) p_0. Let $p(x_k|x_{k-1})$ denote the transition density and $p(y_k|x_{k-1})$ the likelihood function.

The filtering problem consists of estimating the conditional densities

$$b_k(x_k) \triangleq p(x_k|y_{0:k}), k=0,1,\ldots, \tag{2}$$

where $y_{0:k}$ denotes the observations from time 0 to k. The conditional density $b_k(x_k)$ can be derived recursively via the Chapman-Kolmogorov equation and Bayes' rule as follows:

$$\begin{aligned} b_k(x_k) &= \frac{p(y_k|x_k)p(x_k|y_{0:k-1})}{p(y_k|y_{0:k-1})} \\ &= \frac{p(y_k|x_k)\int p(x_k|x_{k-1})b_{k-1}(x_{k-1})dx_{k-1}}{\int p(y_k|x_k)p(x_k|y_{0:k-1})dx_k}, \end{aligned} \tag{3}$$

where the denominator $\int p(y_k|x_k)p_k(x_k|y_{0:k-1})dx_k$ is just a normalizing constant.

It seems plausible that given $b_0(x_0)$, we can calculate $b_k(x_k)$ for any k by carrying out (3) recursively k times. However, the integrals in (3) are usually intractable. Moreover, the conditional density $b_k(x_k)$ may be any (infinite-dimensional) probability density, even if the prior $b_{k-1}(x_{k-1})$ is some nice distribution such as Gaussian. Therefore, Monte Carlo simulation is a very useful technique here, and in particular, importance sampling is crucial because of the difficulty in directly sampling from an arbitrary distribution.

Particle filtering is a class of filters that utilize Monte Carlo simulation and importance sampling techniques to estimate the conditional densities. It approximates the conditional density using a finite number of particles/samples and mimics the evolution of the conditional density through the propagation of particles. More specifically, particle filtering approximates $b_k(x_k)$ by a probability mass function

$$\hat{b}_k(x_k) = \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i), \tag{4}$$

where δ denotes the Kronecker delta function, $\{x_k^i\}_{i=1}^N$ are the random support points, and $\{w_k^i\}_{i=1}^N$ are the associated probabilities/weights.

We now show how to draw samples $\{x_k^i\}_{i=1}^N$ and calculate their corresponding weights $\{w_k^i\}_{i=1}^N$. From (3), we observe

$$b_k(x_k) \propto \int p(y_k|x_k)p(x_k|x_{k-1})b_{k-1}(x_{k-1})dx_{k-1}. \tag{5}$$

To carry out the integration on the right hand side, we need samples of (x_{k-1},x_k) drawn from $p(y_k|x_k)p(x_k|x_{k-1})b_{k-1}(x_{k-1})$. Instead of sampling directly from it, we introduce *importance densities* $q_k(x_k|x_{k-1},y_k)$ and $g_{k-1}(x_{k-1}|y_{0:k-1})$, from which we draw i.i.d. samples

$(x_{k-1},x_k)^i \triangleq (x_{k-1}^i, x_k^i)$ as follows:

$$x_{k-1}^i \sim g_{k-1}(x_{k-1}|y_{0:k-1}),\ i=1,\ldots,N, \quad (6)$$
$$x_k^i \sim q_k(x_k|x_{k-1}^i, y_k),\ i=1,\ldots,N. \quad (7)$$

We call $g_{k-1}(x_{k-1}|y_{0:k-1})$ the *resampling importance density*, and the reason will be seen shortly. Since $\{(x_{k-1},x_k)^i\}_{k=1}^N$ are i.i.d. samples drawn from $q_k(x_k|x_{k-1},y_k)g_{k-1}(x_{k-1}|y_{0:k-1})$, shorthanded as $q_k g_{k-1}$, the integral in (5) can be approximated as

$$\int \frac{p(y_k|x_k)p(x_k|x_{k-1})b_{k-1}(x_{k-1})}{q_k g_{k-1}} q_k g_{k-1} dx_{k-1}$$
$$\approx \sum_{i=1}^N \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)b_{k-1}(x_{k-1}^i)}{q_k(x_k^i|x_{k-1}^i,y_k)g_{k-1}(x_{k-1}^i|y_{0:k-1})}\delta(x_k - x_k^i).$$

Let the normalized weights be

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)b_{k-1}(x_{k-1}^i)}{q_k(x_k^i|x_{k-1}^i,y_k)g_{k-1}(x_{k-1}^i|y_{0:k-1})}, \quad (8)$$

then $b_k(x_{k-1}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$.

In summary, the algorithm of a *general particle filter* is as follows:

Algorithm 1 *General Particle Filter*

1. Initialization. *Sample* $\{x_0^i\}_{i=1}^N$ *i.i.d. from an initial p.d.f./p.m.f.* p_0. *Set* $k=1$.
2. Importance Sampling. *Sample* x_k^i *from* $q_k(x_k|x_{k-1}^i,y_k)$, $i=1,\ldots,N$.
3. Bayes' Updating. *Receive new observation* y_k. *The conditional density is approximated by* $\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$, *where weights* $\{w_k^i\}_{i=1}^N$ *are calculated and normalized according to (8).*
4. Importance Resampling. *Sample* $\{x_k^i\}_{i=1}^N$ *i.i.d. from* $g_k(x_k|y_{0:k})$.
5. $k \leftarrow k+1$ *and go to step 2.*

There are a few remarks on the general particle filter.

Remark 1 Resamping is mainly to counter the problem of *sample degeneracy*. Without resampling, after a few iterations, many particles/samples will have weights near zero and hence can be neglected in practice. A brute-force method to counter sample degeneracy is to use a very large number of samples, but a more efficient method is to resample so that new samples all have reasonably large weights. The optimal resampling should give all the new samples equal weights, and hence the optimal resampling importance density is

$$g_k^{opt}(x_k|y_{0:k}) = b_k(x_k). \quad (9)$$

Although $b_k(x_k)$ is not available, we have an approximate discrete distribution of it, i.e., $\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$. Therefore, in practice, the resampling importance density is often chosen to be

$$g_k(x_k|y_{0:k}) = \hat{b}_k(x_k). \quad (10)$$

The drawback of this choice is a problem often called *sample impoverishment* or *loss of diversity*, which means that samples of large weights are likely to have many copies while those with small weights are likely not to survive after resampling (with replacement). A remedy is to sample from a continuous approximation of $b_k(x_k)$, denoted as $\tilde{b}_k(x_k)$, instead of the discrete approximation $\hat{b}_k(x_k)$.

Remark 2 The choice of the importance density $q_k(x_k|x_{k-1}^i,y_k)$ is also important in overcoming the sample degeneracy problem. Given the optimal resampling importance density (9) and x_{k-1}^i, the optimal importance density has been shown in (Arulampalam et al. 2002) to be

$$q_k^{opt}(x_k|x_{k-1}^i,y_k) = p(x_k|x_{k-1}^i,y_k), \quad (11)$$

However, $p(y_k|x_{k-1}^i)$ is often very hard to evaluate, and hence many suboptimal importance densities are used in practice. A common choice is the transition density, i.e.,

$$q_k(x_k|x_{k-1}^i,y_k) = p(x_k|x_{k-1}^i). \quad (12)$$

This choice is intuitive and easy to implement, and also leads to simple evaluation of the weights. Substituting (12) into (8), and assuming that

$$\hat{b}_{k-1}(x_{k-1}^i) = b_{k-1}(x_{k-1}^i). \quad (13)$$

then the normalized weight becomes

$$w_k^i \propto p(y_k|x_k^i), \quad (14)$$

which is just the likelihood function and hence is easy to evaluate.

With the common choices of importance densities (12) and (10), the importance sampling step is equivalent to propagating the particles through the system dynamics, and the weight evaluation reduces to the simple evaluation of the likelihood function. We will refer to this type of particle filtering as the "plain particle filter" throughout this paper (it is actually the bootstrap filter in Gordon et al. (1993)), as a special case of the general particle filter. In summary, the algorithm of a plain particle filter is as follows:

Algorithm 2 *Plain Particle Filter*

1. Initialization. *Sample* $\{x_0^i\}_{i=1}^N$ *i.i.d. from initial p.d.f./p.m.f.* p_0. *Set* $k=1$.

2. Importance Sampling. *Sample* x_k^i *from* $p(x_k|x_{k-1}^i)$, $i = 1,\ldots,N$.
3. Bayes' Updating. *Receive new observation* y_k. *The conditional density is approximated by* $\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$, *where normalized weights are calculated as*

$$w_k^i \propto p(y_k|x_k^i), i = 1,2,\ldots,N.$$

4. Resampling. *Sample* $\{x_k^i\}_{i=1}^N$ *i.i.d. from* $\hat{b}_k(x_k)$.
5. $k \leftarrow k+1$ *and go to step 2.*

3 PARTICLE FILTERING FRAMEWORK FOR OPTIMIZATION

We consider the global optimization problem:

$$x^* = \arg\max_{x\in\mathscr{X}} H(x), \tag{15}$$

where the solution space $\mathscr{X}$ is a nonempty set in R^n, and $H(\cdot) : \mathscr{X} \to \mathscr{Y}$ is a real-valued function that is bounded, i.e., $\exists M_1 > -\infty,\ M_2 < \infty$ s.t. $M_1 \le H(x) \le M_2,\ \forall x \in \mathscr{X}$. We assume that (15) has a unique global optimal solution, i.e., $\exists x^* \in \mathscr{X}$ s.t. $H(x) < H(x^*),\ \forall x \ne x^*,\ x \in \mathscr{X}$.

The optimization problem (15) can be formulated as a filtering problem by constructing an appropriate state-space model. Let the state-space model be

$$\begin{aligned} x_k &= x_{k-1}, \quad k = 1,2,\ldots, \\ y_k &= H(x_k) - v_k, \quad k = 0,1,\ldots, \end{aligned} \tag{16}$$

where $x_k \in R^n$ is the unobserved state, $y_k \in R$ is the observation, $v_k \in R$ is the observation noise that is an i.i.d. sequence, and the unobserved initial state is $x_0 = x^*$. We assume that v_k has a p.d.f. $\varphi(\cdot)$. We further assume that we only observe realizations of y_k that are possible function values, i.e., $\exists x \in \mathscr{X}$ s.t. $y_k = H(x)$, even though some realizations of y_k are not possible function values. This is a justifiable assumption, since we will see later that the observations are actually chosen out of the sampled function values.

For the above state-space model, the transition density is

$$p(x_k|x_{k-1}) = \delta(x_k - x_{k-1}), \tag{17}$$

where δ denotes the Dirac delta function. The likelihood function is

$$\begin{aligned} p(y_k|x_k) &= \varphi(H(x_k) - y_k) \\ &= \varphi(H(x_{k-1}) - y_k). \end{aligned} \tag{18}$$

Substituting (17) and (18) into the recursive equation of conditional density (3), we obtain

$$b_k(x_k) = \frac{\varphi(H(x_k) - y_k) b_{k-1}(x_k)}{\int \varphi(H(x_k) - y_k) b_{k-1}(x_k) dx_k}. \tag{19}$$

The intuition of model (16) is that the optimal solution x^* is an unobserved static state, while we can only observe the optimal function values $y^* = H(x^*)$ with some noise. Moreover, we can only observe function values $y_k \le y^*$, since $y_k = H(x),\ x \in \mathscr{X}$. Equation (19) implies that at each iteration the conditional density (i.e., b_{k-1}) is tuned by the performance of solutions to yield a new conditional density (i.e., b_k) for drawing candidate solutions at next iteration. It should be expected that if y_k increases with k, the conditional density b_k will get closer to the density of x_k, i.e., a Dirac delta function concentrated on x^*. From the viewpoint of filtering, b_k is the posterior density of x_k that approaches the density of x_k. From the optimization viewpoint, b_k is a density defined on the solution space that becomes more and more concentrated on the optimal solution as k increases.

Therefore, the idea to solve the maximization problem (15) is to recursively estimate b_k of the model (16) while constructing an increasing sequence of observations $\{y_k\}$. We present *the plain particle filter framework for optimization* (PPFO), and then *the general particle filter framework for optimization* (GPFO). The former framework is a special case of the latter and provides more intuition, while the latter framework is more general and allows more variations in the development of new algorithms.

Algorithm 3 *Plain Particle Filter Framework for Optimization (PPFO)*

1. Initialization. *Specify* $\rho \in (0,1]$, *and an initial p.d.f./p.m.f.* b_0 *that is defined on* $\mathscr{X}$. *Sample* $\{x_1^i\}_{i=1}^N$ *i.i.d. from* b_0. *Set* $k = 1$.
2. Observation Construction. *Let* y_k *be the sample* $(1-\rho)$*-quantile of* $\{H(x_k^i)\}_{i=1}^N$. *If* $k > 1$ *and* $y_k \le y_{k-1}$, *then set* $y_k = y_{k-1}$.
3. Bayes' Updating. $\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$, *where weights are calculated as*

$$w_k^i = \varphi(H(x_k^i) - y_k), i = 1,2,\ldots,N,$$

   and normalized.
4. Resampling. *Construct a continuous approximation* $\tilde{b}_k(x_k)$ *from* $\hat{b}_k(x_k)$. *Sample* $\{x_{k+1}^i\}_{i=1}^N$ *i.i.d. from* $\tilde{b}_k(x_k)$.
5. Stopping. *If a stopping criterion is satisfied, then stop; else,* $k \leftarrow k+1$ *and go to step 2.*

At initialization, the PPFO algorithm draws samples from an initial distribution b_0 that is defined on $\mathscr{X}$. A

parameter ρ is specified to determine the $(1-\rho)$-quantile samples that will be used to construct a nondecreasing the observation sequence $\{y_k\}$. Since the transition probability is 1, the importance sampling step is omitted with suitable change of the indices. The Bayes' updating step assigns weights to the samples according to their performance. Slightly different from the plain particle filter, the resampling step here constructs a continuous density $\tilde{b}_k$ first, since the discrete approximation $\hat{b}_k$ does not provide any new samples. The new samples drawn from $\tilde{b}_k$ are more concentrated in the promising areas than the old samples. Similarly, the general particle filtering framework for optimization is as follows:

Algorithm 4 *A General Particle Filtering Framework for Optimization (GPFO)*

1. Initialization. *Specify* $\rho \in (0,1]$*, and an initial p.d.f./p.m.f.* b_0 *that is defined on* $\mathscr{X}$*. Sample* $\{x_0^i\}_{i=1}^N$ *i.i.d. from* b_0*. Set* $k=1$.
2. Importance Sampling. *Sample* x_k^i *from* $q_k(x_k|x_{k-1}^i, y_k)$*,* $i=1,\ldots,N$.
3. Observation Construction. *Let* y_k *be the sample* $(1-\rho)$*-quantile of* $\{H(x_k^i)\}_{i=1}^N$*. If* $k>1$ *and* $y_k \le y_{k-1}$*, then set* $y_k = y_{k-1}$.
4. Bayes' Updating. $\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$*, where weights are calculated as*

$$w_k^i = \frac{\varphi(H(x_k^i) - y_k) b_{k-1}(x_{k-1}^i)}{q_k(x_k^i|x_{k-1}^i, y_k) g_{k-1}(x_{k-1}^i|y_{0:k-1})},$$

and normalized.
5. Importance Resampling. *Sample* $\{x_k^i\}_{i=1}^N$ *i.i.d. from* $g_k(x_k|y_{0:k})$.
6. Stopping. *If a stopping criterion is satisfied, then stop; else,* $k \leftarrow k+1$ *and go to step 2.*

4 INTERPRETATION OF THE CROSS-ENTROPY METHOD

In this section, we use the particle filtering framework to interpret the cross entropy (CE) method. The CE method can be viewed as projection particle filtering, which fits in the plain particle filtering framework with a specific way to construct a continuous approximation of the conditional density, namely the density projection approach. The main difficulty in the particle filtering framework is to estimate a distribution from the samples, and this difficulty is solved in the CE method by projecting the empirical distribution of the samples to obtain an approximate continuous density. However, the density projection introduces an error from a filtering viewpoint.

In CE, the most common sample selection scheme is the so-called *truncated selection* (Zhang and Muhlenbein 2004), which selects the elite samples whose performance is above a threshold. In the following, we will focus on the truncated selection scheme in our interpretation. We will show that the truncated selection scheme is equivalent to setting the observation noise as a uniform random variable and the observation as the threshold in the particle filtering framework. Other selection schemes can be achieved by setting the observations and observation noise in other manners.

Recall that in the optimization problem (15), the objective function $H(x)$ is bounded by $M_1 \le H(x) \le M_2$. In the state-space model (16), let the observation noise v_k follow a uniform distribution $U(0, M_2 - M_1)$. Hence, the likelihood function is

$$p(y_k|x_k) = \begin{cases} \frac{1}{M_2 - M_1}, & \text{if } 0 \le H(x_k) - y_k \le M_2 - M_1; \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Since $y_k = H(x), x \in \mathscr{X}$, the inequality $H(x_k) - y_k \le M_2 - M_1$ always holds. Hence, (20) can be written in a more compact way as

$$p(y_k|x_k) = \frac{1}{M_2 - M_1} I_{\{H(x_k) \ge y_k\}}, \tag{21}$$

where $I_{\{\cdot\}}$ denotes the indicator function.

Substituting (21) into the conditional density evolution equation (19), we obtain

$$b_k(x_k) = \frac{I_{\{H(x_k) \ge y_k\}} b_{k-1}(x_k)}{\int I_{\{H(x_k) \ge y_k\}} b_{k-1}(x_k) dx_k}. \tag{22}$$

With i.i.d. samples $\{x_k^i\}_{i=1}^N$ drawn from b_{k-1}, $b_k(x_k)$ can be approximated by

$$\hat{b}_k(x_k) = \frac{\sum_{i=1}^N I_{\{H(x_k^i) \ge y_k\}} \delta(x_k - x_k^i)}{\sum_{i=1}^N I_{\{H(x_k^i) \ge y_k\}}}. \tag{23}$$

It is obvious to see that (22) is equivalent to selecting the elite solutions to tune the sampling distribution at the previous iteration, and (23) shows how it is implemented using Monte Carlo simulation. These two equations are the cornerstone of the rest of this section.

The standard CE method (we use the word "standard" to distinguish it from the extended version of standard CE (DeBoer et al. 2005)) for the optimization problem (15) is as follows:

Algorithm 5 *Standard CE Algorithm for Optimization*

1. *Choose an initial p.d.f./p.m.f.* $f(\cdot, \theta_0)$*,* $\theta_0 \in \Theta$*. Specify the parameter* $\rho \in (0,1]$*, and set* $k=1$.

2. *Generate samples* $\{x_k^i\}_{i=1}^N$ *from the density* $f(\cdot,\theta_{k-1})$ *and compute the sample* $(1-\rho)$*-quantile* y_k *of the performances* $\{H(x_k^i)\}_{i=1}^N$.
3. *Compute the new parameter according to*

$$\theta_k = \arg\max_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} I_{\{H(x_k^i)\geq y_k\}} \log f(x_k^i,\theta). \quad (24)$$

4. *If a stopping criterion is satisfied, then terminate; else, set* $k = k+1$ *and go to step 2.*

Equation (24) comes from the density projection of the optimal importance sampling density onto a parameterized family of densities $\{f(\cdot,\theta),\theta\in\Theta\}$. Projection particle filtering (Zhou et al. 2008) also uses the density projection technique, but for a very different reason. It projects the discrete approximation $\hat{b}_k$ onto the parameterized family $\{f(\cdot,\theta),\theta\in\Theta\}$ in order to obtain a continuous approximation $\tilde{b}_k$ that is characterized by only a few parameters, which is very useful in reducing the complexity of dynamic programming in a decision making problem. Specifically, projection particle filtering chooses a value of the parameter θ such that the Kullback-Leibler (KL) divergence between $\hat{b}_k$ and $f(\cdot,\theta)$ is minimized. The KL divergence between $\hat{b}_k$ and $f(\cdot,\theta)$ is:

$$\begin{aligned} D_{KL}(\hat{b}_k \| f(\cdot,\theta)) &= \int \hat{b}_k \log \frac{\hat{b}_k}{f(\cdot,\theta)} \\ &= \int \hat{b}_k \log \hat{b}_k - \int \hat{b}_k \log f(\cdot,\theta). \end{aligned}$$

Since the first term does not depend on $f(\cdot,\theta_k)$, minimizing the above equation is equivalent to solving the maximization problem

$$\max_{\theta\in\Theta} E_{\hat{b}_k}[\log f(\cdot,\theta)].$$

Since $\hat{b}_k(x_k)$ satisfies (23), the above maximization problem can be approximated by

$$\max_{\theta\in\Theta} \frac{\sum_{i=1}^N I_{\{H(x_k^i)\geq y_k\}} \log f(x_k^i,\theta)}{\sum_{i=1}^N I_{\{H(x_k^i)\geq y_k\}}},$$

which is equivalent to

$$\max_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} I_{\{H(x_k^i)\geq y_k\}} \log f(x_k^i,\theta). \quad (25)$$

Therefore, the optimization algorithm adapted from projection particle filtering is as follows:

Algorithm 6 *An Instantiation of Plain Particle Filter Framework for Optimization*

1. Initialization. *Specify* $\rho\in(0,1]$*, and an initial p.d.f./p.m.f.* $f(\cdot,\theta_0)$ *that is defined on* $\mathscr{X}$*. Sample* $\{x_1^i\}_{i=1}^N$ *i.i.d. from* $f(\cdot,\theta_0)$*. Set* $k=1$.
2. Observation Construction. *Let* y_k *be the sample* $(1-\rho)$*-quantile of* $\{H(x_k^i)\}_{i=1}^N$*. If* $k>1$ *and* $y_k \leq y_{k-1}$*, then set* $y_k = y_{k-1}$.
3. Bayes' Updating. *The discrete approximation is*

$$\hat{b}_k(x_k) = \frac{\sum_{i=1}^N I_{\{H(x_k^i)\geq y_k\}} \delta(x-x_k^i)}{\sum_{i=1}^N I_{\{H(x_k^i)\geq y_k\}}}.$$

4. Resampling. *Construct a continuous approximation* $\tilde{b}_k(x_k) = f(x_k,\theta_k)$ *by density projection, where*

$$\theta_k = \arg\max_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} I_{\{H(x_k^i)\geq y_k\}} \log f(x_k^i,\theta). \quad (26)$$

Sample $\{x_{k+1}^i\}_{i=1}^N$ *i.i.d. from* $\tilde{b}_k(x_k)$.
5. Stopping. *If a stopping criterion is satisfied, then stop; else,* $k \leftarrow k+1$ *and go to step 2.*

It is easy to see that this algorithm is essentially the same as the standard CE algorithm. The CE method avoids complicated estimation of the density b_k through the use of density projection. However, from a filtering viewpoint, the projection particle filtering introduces a projection error that is accumulated over iterations. The reason can be seen by scrutinizing the one-step evolution of the approximate density. Since samples $\{x_k^i\}_{i=1}^N$ are sampled from $\tilde{b}_{k-1} = f(\cdot,\theta_{k-1})$, the density that the algorithm actually tries to approximate at iteration k is

$$b_k(x_k) = \frac{I_{\{H(x_k)\geq y_k\}} f(x_k,\theta_{k-1})}{\int I_{\{H(x_k)\geq y_k\}} f(x_k,\theta_{k-1}))dx_k}.$$

Compared with the original equation (22) for b_k, b_{k-1} is replaced by its approximation $f(\cdot,\theta_{k-1})$, which introduces a projection error that is accumulated to the next iteration. This projection error can be corrected by taking $f(\cdot,\theta_{k-1})$ as an importance density and hence taken care of by the weights of the samples. This is a direction to improve the CE method that we will investigate further.

5 CONCLUSION AND FUTURE RESEARCH

We have introduced a particle filtering framework for randomized optimization algorithms. The CE method can be viewed as an instantiation of this framework, and interpreted from a filtering viewpoint.

The framework holds the promise for developing new optimization algorithms through the choice of observation noise, sampling and resampling importance densities as well as the various improving techniques for particle filtering. For

example, the choice of the distribution of the observation noise determines how the samples are weighted. Proper sampling and resampling importance densities can be chosen to adjust the trade-off between exploitation and exploration. Construction of the resampling importance density using the kernel method for density estimation (Musso et al. 2001), or approximation with Gaussian mixture (Kotecha and Djuric 2003) is very easy to implement and the obtained continuous distributions are easy to sample from. They add more exploration on the solution space, compared to a single Gaussian density that is often used in the CE method. Markov chain Monte Carlo (MCMC) step can be added after resampling (Gilks and Berzuini 2001) to further adjust the trade-off between exploitation and exploration, or add some local search.

There are three important lines of future research that we will pursue. First, we will continue to study the implication of the particle filtering framework on other randomized optimization algorithms, such as EDAs and MRAS, and the relationship between the various randomized optimization algorithms. Secondly, we will further study how to develop new algorithms using the particle filtering framework and the performance of these new algorithms. Finally, we want to investigate the convergence property of the particle filtering framework for optimization. Although convergence has been proved for EDAs (Zhang and Muhlenbein 2004), the CE method (Rubinstein 1999), and MRAS (Hu et al. 2007) individually, we are interested in a unifying convergence results under the particle filtering framework.

REFERENCES

Arulampalam, S., S. Maskell, N. J. Gordon, and T. Clapp. 2002. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50 (2): 174–188.

Cappé, O., S. J. Godsill, and E. Moulines. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE* 95 (5): 899–924.

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

Doucet, A., J. F. G. deFreitas, and N. J. Gordon. (Eds.) 2001. *Sequential Monte Carlo methods in practice*. New York: Springer.

Gilks, W., and C. Berzuini. 2001. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society* 63 (1): 127–146.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, Volume 140, 107–113.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. A model reference adaptive search method for global optimization. *Operations Research* 55:549–568.

Kotecha, J. H., and P. M. Djuric. 2003. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing* 51 (10): 2602–2612.

Larranaga, P., R. Etxeberria, J. A. Lozano, and J. M. Pena. 1999. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country.

Lozano, J. A., P. Larranaga, I. Inza, and E. Bengoetxea. (Eds.) 2006. *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*. New York: Springer Verlag.

Muhlenbein, H., and G. Paaß. 1996. From recombination of genes to the estimation of distributions: I. binary parameters. In *Parallel Problem Solving from Nature-PPSN IV*, ed. H. M. Voigt, W. Ebeling, I. Rechenberg, and H. P. Schwefel, 178–187. Berlin, Germany: Springer Verlag.

Musso, C., N. Oudjane, and F. LeGland. 2001. Improving regularized paricle filters. In *Sequential Monte Carlo Methods in Practice*, ed. A. Doucet, J. F. G. deFreitas, and N. J. Gordon. New York: Springer-Verlag.

Rubinstein, R. Y. 1997. Optimization of computer simulation models with rare events. *European Journal of Operational Research* 99:89–112.

Rubinstein, R. Y. 1999. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 2:127–190.

Rubinstein, R. Y., and D. P. Kroese. 2004. *The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning*. New York: Springer-Verlag.

Zhang, Q., and H. Muhlenbein. 2004. On the convergence of a class of estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* 8:127–136.

Zhou, E., M. C. Fu, and S. I. Marcus. 2008. Solving continuous-state POMDPs via density projection. Technical Report TR 2008-6, Institute for Systems Research, University of Maryland, College Park, MD.

Zlochin, M., M. Birattari, N. Meuleau, and M. Dorigo. 2004. Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research* 131:373–395.

AUTHOR BIOGRAPHIES

ENLU ZHOU is a Ph.D. student in the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland. She received the B.S. degree with highest honor in electrical engineering

from Zhejiang University in 2004. She was selected into the honor program of Mixed Class at Zhejiang University, and has been selected as a Future Faculty Fellow by the A. J. Clark School of engineering at University of Maryland. Her research interests lie in the areas of stochastic control and estimation, and simulation optimization, with applications towards financial engineering. Her email address is <enluzhou@umd.edu>.

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

STEVEN I. MARCUS is a Professor in the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland. He received the B.A. degree in electrical engineering and mathematics from Rice University in 1971, and the S.M. and Ph.D. degrees in electrical engineering from MIT in 1972 and 1975, respectively. From 1975 to 1991, he was with the Department of Electrical and Computer Engineering, University of Texas, Austin, where he was the L.B. (Preach) Meaders Professor in Engineering. At the University of Maryland, he has served as Director of the Institute for Systems Research from 1991-1996, and as Chair of the Electrical and Computer Engineering Department from 2000-2005. Currently, his research is focused on stochastic control and estimation, with applications in manufacturing and telecommunication networks. He is the Corresponding Editor of the *SIAM Journal on Control and Optimization* and a Fellow of IEEE. His email address is <marcus@umd.edu>.