Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

A NEW POPULATOIN-BASED SIMULATED ANNEALING ALGORITHM

Enlu Zhou
Xi Chen

Department of Industrial & Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, U.S.A

ABSTRACT

In this paper, we propose sequential Monte Carlo simulated annealing (SMC-SA), a population-based simulated annealing algorithm, for continuous global optimization. SMC-SA incorporates the sequential Monte Carlo method to track the converging sequence of Boltzmann distributions in simulated annealing, such that the empirical distribution will converge weakly to the uniform distribution on the set of global optima. Numerical results show that SMC-SA is a great improvement of the standard simulated annealing on all test problems and outperforms the popular cross-entropy method on badly-scaled objective functions.

1 INTRODUCTION

In this paper, we introduce a sequential Monte Carlo simulated annealing (SMC-SA) algorithm for continuous global optimization. It is well known that the Boltzmann distribution converges weakly to the uniform distribution concentrated on the set of global optima as the temperature decreases to zero (Romeijn and Smith 1994). At each iteration, simulated annealing essentially simulates an ergodic Markov chain whose stationary distribution is the Boltzmann distribution at current temperature, and the current state becomes the initial state for a new chain at the next iteration. Hence, the temperature has to decrease slowly enough such that the chain does not vary too much from iteration to iteration, which ensures the overall convergence for simulated annealing. Our idea is to track the sequence of Boltzmann distributions using a number of samples via sequential Monte Carlo method (SMC) (Doucet, de Freitas, and Gordon 2001), such that the empirical distribution yielded by our algorithm will also converge weakly to the uniform distribution concentrated on the set of global optima. One thing we should note is that our method does not have to be restricted to the Boltzmann distribution; it can be used to track any sequence of distributions as long as it converges to a degenerate distribution on one or more global optima and satisfies certain regularity conditions.

Simulated annealing (SA) is an attractive algorithm for optimization, due to its theoretical guarantee of convergence, good performance on many practical problems, and ease of implementation. It was first proposed by Kirkpatrick, Gelatt, and Vecchi (1983) through an analogy between optimization and the physical process of annealing. The early study of simulated annealing focused on combinatorial optimization, and some fundamental theoretical work include Geman and Geman (1984), Gidas (1985), Anily and Federgruen (1987), and Hajek (1988). Later, simulated annealing was extended to continuous global optimization, and rigorous convergence results were proved under various conditions, such as Dekkers and Aarts (1991), Belisle (1992), Romeijn and Smith (1994), Locatelli (1996), Locatelli (2000), and Yang (2000). Meanwhile, connections were exploited between simulated annealing and some other optimization algorithms, and many variations of simulated annealing were developed. The book of van Laarhoven and Aarts (1987) has a complete summary on simulated annealing for combinatorial optimization, and the survey paper of Henderson, Jacobson, and Johnson (2003) provides a good overview of the theoretical development of simulated annealing in both combinatorial and continuous optimization.

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

The standard SA generates one sample, or in other words candidate solution, at each iteration, and the sequence of candidate solutions converge asymptotically to the optima in probability. To speed up simulated annealing, many variations of simulated annealing were proposed to generate a population of samples or candidate solutions at each iteration. For example, Ruppeiner, Pedersen, and Salamon (1991) present an implementation of simulated annealing with an ensemble of random walkers searching the configuration space in parallel; Mahfoud and Goldberg (1995) propose an effective combination of simulated annealing and genetic algorithms, by incorporating the population approach and recombinative power of genetic algorithms into simulated annealing; Chu, Deng, and Reinitzy (1999) propose a parallel simulated annealing by monitoring and pooling performance statistics simultaneously on all processors and mixing states at intervals to ensure a Boltzmann distribution; Onbaşoğlu and Özdamar (2001) develop and compares five different parallel simulated annealing using different approaches for information exchange among processors; van Hentenryck and Vergados (2007) propose a population-based simulated annealing with intensification and diversification. Most recently, Molvalioglu, Zabinsky, and Kohn (2007) and Molvalioglu, Zabinsky, and Kohn (2009) introduce a multi-particle version of simulated annealing which consists of N-particle exploration and N-particle selection steps with a meta-control of the temperature. This method bears some similarity with SMC-SA in the sense that the exploration step can be viewed as a variation of the resampling step in SMC-SA and the selection step is essentially the SA move step in SMC-SA; however, SMC-SA has an importance updating step which plays an important role that will be explained later. The combination of the resampling and SA move steps in SMC-SA is also similar to that in the resample-move particle filter introduced in Gilks and Berzuini (2001), which is however developed for filtering (i.e. sequential state estimation).

Compared with the above algorithms, the main distinction of SMC-SA is in its motivation to closely track the sequence of Boltzmann distributions at each iteration through the use of sequential Monte Carlo method. Viewing each iteration of simulated annealing as trying to reach stationarity of a Markov chain whose stationary distribution is the current Boltzmann distribution, the importance updating step in SMC-SA brings a head start to each iteration in the sense that it updates the previous empirical distribution to a new one "close" to the current Boltzmann distribution, and the SA move step takes the new empirical distribution even "closer" to the stationary distribution. The resampling step in SMC-SA introduces additional approximation error, but it is necessary to keep the diversity of the samples. The goal of SMC-SA is to achieve faster convergence *in time* than the standard simulated annealing through the trade-off between the sample size (i.e., the number of samples or candidate solutions generated at each iteration) and the rate of temperature change: by generating more samples at each iteration we can reduce the time to reach the same accuracy of the solution. In fact, as the numerical results show, SMC-SA not only converges faster in time, but also achieves better solutions than the standard SA using the same total number of samples.

The rest of the paper is organized as follows: Section 2 revisits simulated annealing and motivates the development of SMC-SA; Section 3 introduces SMC-SA with explanations of the rationale behind it; Section 4 presents the numerical results of SMC-SA compared with the standard SA, multi-start SA, and the cross-entropy method; Section 5 concludes the paper.

2 REVISITING SIMULATED ANNEALING

We consider a maximization problem

$$\max_{x\in\mathscr{X}} H(x), \tag{1}$$

where the solution space $\mathscr{X}$ is a nonempty continuous set in R^n, and $H:\mathscr{X}\to R$ is a deterministic function that is bounded, i.e., $\exists H_l > -\infty$, $H_u < \infty$ s.t. $H_l \le H(x) \le H_u$, $\forall x \in \mathscr{X}$. We denote the optimal function value as H^*, i.e., there exists an maximum $x^* \in \mathscr{X}$ such that $H(x) \le H^* \triangleq H(x^*)$, $\forall x \in \mathscr{X}$.

Simulated annealing can be viewed as a Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953), which simulates the density proportional to $\exp\{H(x)/T\}$ by simulating an ergodic Markov chain whose stationary distribution is the Boltzmann distribution (Robert and Casella 2004). Hence, theoretically we have to simulate the chain for an infinite number of transitions before a sample is truly drawn from this Boltzmann distribution. Once the stationarity of the chain is achieved, we decrease the temperature, and then again have to simulate the new chain for an infinite number of transitions to achieve the stationary distribution which is the Boltzmann distribution at the new

temperature. This type of SA is conceptually simple and easier to analyze, but is clearly impractical. In practice, the most often used SA iteratively decreases the temperature and draws one sample. This is equivalent to simulating each Markov chain for only one transition, and hence, the chain almost never achieves stationarity before the temperature changes. Obviously, there could be some algorithms in between these two extremes, such as iteratively decreasing the temperature and drawing a few finite number of samples, which is equivalent to simulating each Markov chain for a few number of times at each temperature. The two extreme cases described above are summarized as follows:

- Infinite-Transition SA (ITSA): It can be viewed as a sequence of Markov chains. Each Markov chain is of infinite length, and converges to the Boltzmann distribution at the current temperature. The temperature is decreased in between subsequent Markov chains.
- Single-Transition SA (STSA): It can be viewed as a sequence of Markov chains. Each Markov chain has only one transition. The temperature is decreased in between subsequent Markov chains.

ITSA and STSA can be also viewed as "homogeneous SA" and "inhomogeneous SA" respectively (van Laarhoven and Aarts 1987), since ITSA can be viewed as a sequence of homogeneous Markov chains, and STSA as one single inhomogeneous Markov chain of infinite length, where the temperature is decreased in between subsequent transitions. "Homogeneous" means the transition probabilities/densities are fixed at all times, and "inhomogeneous" means that probabilities/densities are changing with time in a Markov chain. For the algorithm to converge to the global optima in probability, STSA requires the temperature to decrease slowly enough whereas there is no such requirement on ITSA (van Laarhoven and Aarts 1987). That can be intuitively explained as a result that the Markov chain corresponding to each temperature almost never achieves stationarity in STSA. If the temperature decreases slowly enough, then the subsequent Markov chains do not differ too much, such that when the current state becomes the initial state for the next Markov chain, it is not too far away from the stationary distribution.

In summary, the ITSA achieves stationarity of each subsequent chain perfectly, but it requires an infinite number of simulations of each chain; whereas the STSA only simulates each chain once but almost none of the chains achieve stationarity, and hence it requires a slow enough cooling speed. Motivated by this observation, we ask the question: *Can we follow the stationary distribution of each subsequent chain as close as possible in one step?* Our idea is to use sequential Monte Carlo (SMC) methods to follow the Boltzmann distribution at each temperature and use the Metropolis algorithm to rejuvenate new samples.

3 SEQUENTIAL MONTE CARLO SIMULATED ANNEALING

In this section, we propose the sequential Monte Carlo simulated annealing (SMC-SA) algorithm. The idea is to incorporate sequential Monte Carlo method to track the sequence of Boltzmann distributions in simulated annealing. It has three main steps: importance updating, resampling, and SA move. The importance updating step is based on the principle of importance sampling, and it essentially updates the empirical distribution from last iteration to a new one that is close to the target distribution of this iteration. More specifically, it takes the current Boltzmann distribution f_k as the target distribution, and the previous Boltzmann distribution f_{k-1} as the proposal distribution. Thus, given the previous samples are already distributed approximately according to f_{k-1} and the weights of these samples are updated in proportion to $f_k(\cdot)/f_{k-1}(\cdot)$, the new empirical distribution formed by these weighted samples will closely follow f_k. The resampling step redistributes the samples such that they all have equal weights. The SA move step performs one iteration of simulated annealing on each sample to generate a new sample or candidate solution. This step essentially takes the current empirical distribution as the initial distribution, and simulates one transition of the Markov chain whose stationary distribution is the current Boltzmann distribution f_k. Hence, the resultant empirical distribution will be brought even closer to f_k. The resampling step together with the SA move step prevents sample degeneracy, or in other words, keeps the sample diversity and thus the exploration of the solution space. We explain the main steps in more detail in the following.

3.1 Importance Updating

The importance updating step is based on importance sampling (Robert and Casella 2004), which essentially performs a change of measure. Thus, the expectation under one distribution can be estimated

using the samples drawn from another distribution with appropriate weighting. Specifically, let f and g denote two probability density functions. For any integrable function ϕ, its integration with respect to f equals to

$$I_\phi = \int \phi(x) f(x) dx = \int \phi(x) \frac{f(x)}{g(x)} g(x) dx. \tag{2}$$

If we draw independent and identically distributed (i.i.d.) samples $\{x^i\}_{i=1}^N$ from g and set their weights $\{w^i\}_{i=1}^N$ according to

$$W^i = \frac{f(x^i)}{g(x^i)}, \quad w^i = \frac{W^i}{\sum_{j=1}^N W^j},$$

then in view of (2), an estimate of I_ϕ is

$$\hat{I}_\phi = \frac{1}{N} \sum_{i=1}^N W^i \phi(x^i), \quad x^i \overset{\text{iid}}{\sim} g,$$

and an approximation of f is

$$\hat{f}(x) = \sum_{i=1}^N w^i \delta_{x^i}(x), \tag{3}$$

where δ denotes the Dirac delta function, which satisfies

$$\int \phi(x) \delta_y(x) dx = \phi(y).$$

In other words, $\{x^i, w^i\}_{i=1}^N$ is a weighted sample from f, and $\hat{f}$ defined in (3) is an empirical distribution of f.

In simulated annealing, suppose we already have i.i.d. samples $\{x_{k-1}^i\}_{i=1}^N$ from the last Boltzmann distribution, using the importance sampling principle described above we can obtain weighted samples $\{x_k^i, w_k^i\}_{i=1}^N$ that are distributed to the current Boltzmann distribution. More specifically, the Boltzmann density at time k is

$$f_k(x) = \frac{1}{Z_k} \exp\left\{\frac{H(x)}{T_k}\right\},$$

where $Z_k = \int \exp\{H(x)/T_k\} dx$ is the normalization constant, $H(x)$ is the objective function in (1), and T_k is often referred to as the temperature at time k. Noticing that

$$\frac{f_k(x)}{f_{k-1}(x)} = \frac{\exp\left\{H(x)\left(\frac{1}{T_k} - \frac{1}{T_{k-1}}\right)\right\}}{Z_k / Z_{k-1}}, \quad k = 2, \ldots,$$

an approximation of f_k is

$$\hat{f}_k(x) = \sum_{i=1}^N w_k^i \delta_{x_{k-1}^i}(x), \quad x_{k-1}^i \overset{\text{iid}}{\sim} f_{k-1},$$

where

$$w_k^i \propto \exp\left\{H(x_{k-1}^i)\left(\frac{1}{T_k} - \frac{1}{T_{k-1}}\right)\right\}, \quad \sum_{i=1}^N w_k^i = 1, \quad k = 2, \ldots.$$

Assuming that we do not have any prior knowledge about the optima, we draw the initial samples from a uniform distribution over the solution space, i.e.,

$$f_0(x) \propto 1, \quad \forall x \in \mathscr{X}.$$

Since

$$\frac{f_1(x)}{f_0(x)} \propto \exp\left\{\frac{H(x)}{T_1}\right\},$$

the weights at time 1 should satisfy

$$w_1^i \propto \exp\left\{\frac{H(x_0^i)}{T_1}\right\}, \quad \sum_{i=1}^{N} w_1^i = 1.$$

In the following, we refer to N as the sample size, i.e., the number of samples or candidate solutions generated at each iteration. We assume N is constant for every iteration.

3.2 Resampling

The importance updating step gives us $\hat{f}_k = \sum_{i=1}^{N} w_k^i \delta_{x_{k-1}^i}$, an approximation of f_k. However, the weighted samples $\{x_{k-1}^i, w_k^i\}_{i=1}^N$ will suffer from the problem of degeneracy that will be explained shortly. Therefore, the resampling step is needed to sample from the weighted samples $\{x_{k-1}^i, w_k^i\}_{i=1}^N$ in order to generate N i.i.d. new samples $\{\tilde{x}_k^i\}_{i=1}^N$, which are still approximately distributed according to f_k. In SMC-SA, we use sampling with replacement scheme for the resampling step. There are several other resampling schemes mainly for the purpose of variance reduction, such as stratified resampling, residual resampling (Liu and Chen 1998), and multinomial resampling (Gordon, Salmond, and Smith 1993), and their effects on the algorithm performance will be studied in the future.

The purpose of resampling can be explained from different perspectives. From sampling perspective, the resampling step together with the SA move step help to overcome sample degeneracy. Without resampling, after a few iterations, only few samples would have dominating weights and most others have weights close to 0. These negligible samples waste future computation effort, since they do not contribute much to the updating of the empirical distribution. In contrast, with resampling, samples with large weights would have multiple copies, and these identical copies lead to different samples because of the SA move step next. Hence, resampling keeps the diversity of samples and ensure that every sample is useful. From the optimization perspective, resampling brings more exploration to the neighborhood of good solutions. It is similar to the selection step in genetic algorithms, where the elite parents would have more offsprings.

3.3 SA Move

At iteration k, the SA move is one step of the Metropolis algorithm with the target distribution being the Boltzmann distribution with density f_k. As $\{\tilde{x}_k^i\}_{i=1}^N$ are the initial states of the Markov chain and are distributed "closely" according to f_k, new samples $\{x_k^i\}$ generated from $\{\tilde{x}_k^i\}$ by the SA move step are even "closer" to the stationary distribution f_k. The SA move step is essentially the same as the SA algorithm, and is described below for clarity.

Algorithm 1. *SA Move at iteration k in SMC-SA*

- *Choose a symmetric proposal distribution $g_k(\cdot|x)$, such as uniform or normal distributions with mean x.*
- *Generate $y_k^i \sim g_k(y|\tilde{x}_k^i), i = 1,\ldots,N$.*
- *Calculate acceptance probability*

$$\rho_k^i = \min\{\exp\left(\frac{H(y_k^i) - H(\tilde{x}_k^i)}{T_k}\right), 1\}.$$

- *Accept/Reject*

$$x_k^i = \begin{cases} y_k^i, & w.p.\ \rho_k^i; \\ \tilde{x}_k^i, & w.p.\ 1-\rho_k^i. \end{cases}$$

In summary, for the maximization problem (1), our proposed algorithm is as follows.

Algorithm 2. *Sequential Monte Carlo Simulated Annealing (SMC-SA)*

- *Input: sample size N, cooling schedule for* $\{T_k\}$.
- *Initialization: generate* $x_0^i \overset{\text{iid}}{\sim} \text{Unif}(\mathscr{X})$*, and compute normalized weights* $w_1^i \propto \exp\left\{\frac{H(x_0^i)}{T_1}\right\}$, $i = 1,2,\ldots,N$*. Generate i.i.d. samples* $\{x_1^i\}_{i=1}^N$ *from* $\{x_0^i, w_1^i\}_{i=1}^N$*. Set* $k = 2$.
- *At iteration k:*
  - *Importance Updating: compute normalized weights* $w_k^i \propto \exp\left\{H(x_{k-1}^i)\left(\frac{1}{T_k} - \frac{1}{T_{k-1}}\right)\right\}$.
  - *Resampling: draw i.i.d. samples* $\{\tilde{x}_k^i\}_{i=1}^N$ *from* $\{x_{k-1}^i, w_k^i\}_{i=1}^N$.
  - *SA Move: generate* x_k^i *from* $\tilde{x}_k^i$ *for each i,* $i = 1,\ldots,N$*, according to Algorithm 1.*
  - *Stopping: if a stopping criterion is satisfied, return* $\max_i H(x_k^i)$*; otherwise,* $k := k+1$ *and continue.*

4 NUMERICAL EXPERIMENTS

In this section, we present the numerical results of the proposed SMC-SA algorithm compared with the standard SA, multi-start SA, and the cross-entropy (CE) method (Rubinstein 1999) on several well-known unconstrained and continuous benchmark optimization problems.

4.1 Selected Benchmark Problems

The six selected benchmark problems (Corana, Marchesi, Martini, and Ridella 1987, Yao and Liu 1996, Kroese, Porotsky, and Rubinstein 2006, Pintér 1996) are listed below. The original problems are all minimization problems. The SMC-SA method is presented in maximization form, so we take the negative value of the objective function, and convert them to maximization problems. The plots of these objective functions in two dimensions are shown in Figure. 1.

(a) Dejong's 5th function (n=2)

$$H_a(x) = -\left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ji})^6}\right]^{-1},$$

where $a_{j1} = (-32,-16,0,16,32,-32,-16,0,16,32,-32,-16,0,16,32,$
$-32,-16,0,16,32,-32,-16,0,16,32)$
and
$a_{j2} = (-32,-32,-32,-32,-32,-16,-16,-16,-16,-16,0,0,0,0,0,$
$16,16,16,16,16,32,32,32,32,32)$. The global maximum is at $x^* = (-32,-32)^T$, and $H_a^* \approx -0.998$.

(b) Powel singular function (n=20)

$$H_b(x) = -\sum_{i=2}^{n-2}\left[(x_{i-1}+10x_i)^2 + 5(x_{i+1}-x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1}-x_{i+2})^4\right],$$

where $x^* = (0,\cdots,0)^T$, $H_b^* = 0$.

(c) Rosenbrock function (n=20)

$$H_c(x) = -\sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right],$$

where $x^* = (1, \cdots, 1)^T$, $H_c^* = 0$.

(d) Griewank function (n=20)

$$H_d(x) = -\left[\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1\right],$$

where $x^* = (0, \cdots, 0)^T$, $H_d^* = 0$.

(e) Trigonometric function (n=10)

$$H_e(x) = -1 - \sum_{i=1}^{n} \left[8\sin^2(7(x_i - 0.9)^2) + 6\sin^2(14(x_i - 0.9)^2) + (x_i - 0.9)^2\right],$$

where $x^* = (0.9, \cdots, 0.9)^T$, $H_e^* = -1$.

(f) Pintér's function (n=10)

$$\begin{aligned} H_f(x) \quad = \quad & -\left[\sum_{i=1}^{n} i x_i^2 + \sum_{i=1}^{n} 20i \sin^2(x_{i-1}\sin x_i - x_i + \sin x_{i+1})\right. \\ & \left. + \sum_{i=1}^{n} i \log_{10}(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2)\right], \end{aligned}$$

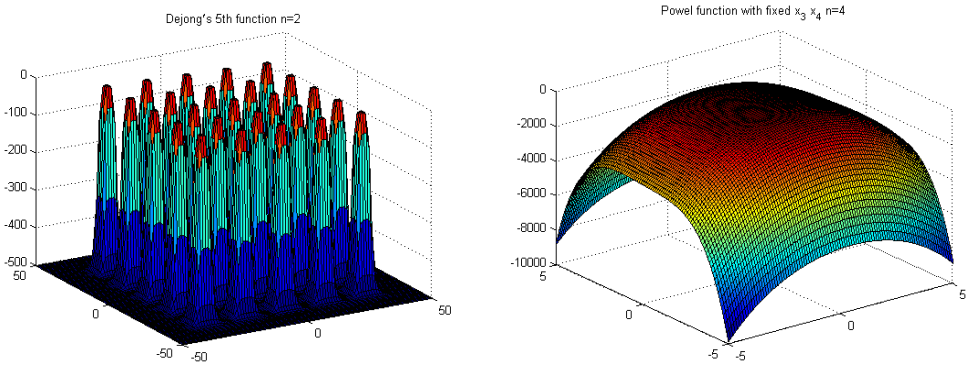
where $x^* = (0, \cdots, 0)^T$, $H_f^* = 0$.

4.2 Comparison of SMC-SA, standard SA, multi-start SA and CE

As a comparison of the proposed SMC-SA method, we also solved the above benchmark problems using the standard SA algorithm, multi-start SA, and CE method, and compared the average performance based on 100 independent runs.

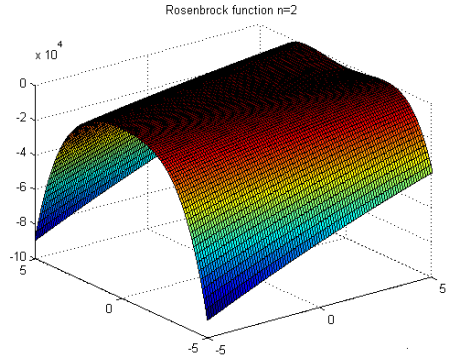
For SMC-SA, standard SA, and multi-start SA, we use the logarithm temperature reduction form, i.e., the temperature at the k^{th} iteration is $T_k = |H^*(x_{k-1})|/\log(k+1)$, where $H^*(x_{k-1})$ is the optimal sample function value at the $(k-1)^{th}$ iteration. The reason for using $|H^*(x_{k-1})|$ is because the weights w_k^i are calculated in proportion to the exponential function $\exp\left\{H(x_{k-1}^i)\left(\frac{1}{T_k} - \frac{1}{T_{k-1}}\right)\right\}$, which may get exploded if the argument of the exponential function is large, and may become identical values if the argument is in the flat tail of the exponential function. By using $|H^*(x_{k-1})|$ in the temperature, the weights will not depend too much on the value of $H(x_{k-1}^i)$.

In these four methods, the initial candidate solutions are all chosen randomly according to the uniform distribution on $[-50, 50]^n$. For SMC-SA, the proposed distribution in the SA move step is the normal distribution with standard deviation $\alpha\beta^k$ at iteration k, where $\alpha = 10$, $\beta = 0.995$ for objective functions H_a and H_b, and $\beta = 0.998$ for H_c, H_d, H_e, and H_f; the sample size is $N = 200$ for H_a, H_b, H_d and H_f, and $N = 1000$ for the high-dimensional problems H_c and H_e. For the standard SA and the multi-start SA, the parameter settings are the same as in SMC-SA for each problem, i.e, the same temperature function, proposal distributions, and the same sample size N in multi-start SA. For the CE method, we use the normal distributions as the parameterized family; the initial mean μ_0 is chosen randomly according to the uniform distribution on $[-50, 50]^n$, and the initial standard deviation is set to be $\Sigma_0 = 500 I_{n\times n}$; the quantile parameter ρ is set be 0.01; the sample size N is 400 for H_a, 500 for H_b, and 5000 for $H_c - H_f$. The standard CE method suffers from the "freezing" problem due to fast convergence of the parameterized family p.d.f. $f(\cdot, \theta_k)$ to a degenerated distribution, and the

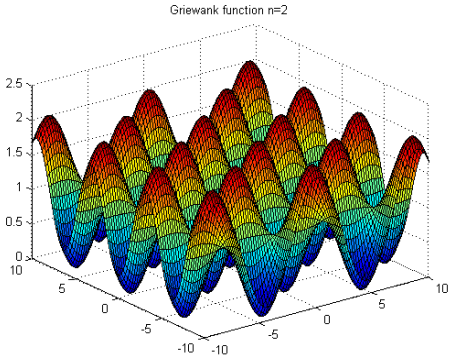


(a) Dejong's 5th function

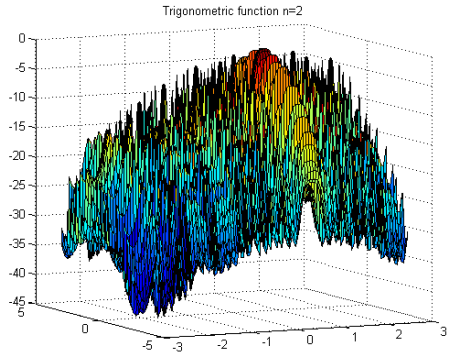
(b) Powel function



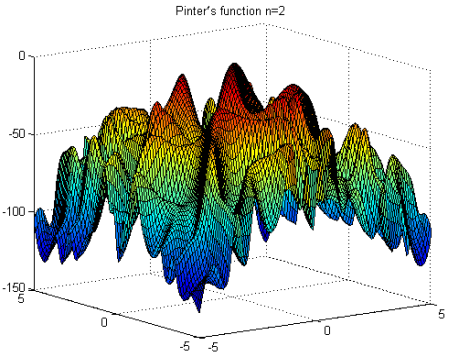
(c) Rosenbrock function



(d) Griewank function



(e) Trigonometric function



(f) Pinter's function

Figure 1: Test problems in two dimensions

solution will converge to a local optimum quickly. In the algorithm, we apply the smoothing parameter updating procedure to solve the "freezing" problem (DeBoer, Kroese, Mannor, and Rubinstein 2005). At iteration k, the parameter is updated smoothly by a weighted average of $\hat{\theta}_{k-1}$ and θ_k, i.e.,

$$\hat{\theta}_k = \nu\theta_k + (1-\nu)\hat{\theta}_{k-1}, \quad \nu \in (0,1) \tag{4}$$

The smoothing parameter is set to be $\nu = 0.2$, which is found to work best by trial and error in our experiments.

Table 1 shows the average performance based on 100 independent runs, where H^* is the true optimal value of $H(\cdot)$, $\bar{H}^*$ is the average optimal value computed by each of the methods, *std_err* is

the standard error of the optimal values, and M_ε is the number of ε-optimal solutions out of 100 runs. In this numerical experiment, we consider $\varepsilon = 10^{-5}$ for problem H_a, H_d, H_e, and H_f, and $\varepsilon = 0.01$ for problems H_b and H_c. Fig. 2 shows the average value of $H(\cdot)$ v.s. the total number of samples for these four methods.

		SMC-SA		multi-start SA		standard SA		CE ($\nu = 0.2$)	
	H^*	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε
H_a	-0.998	-0.998(1.34E-7)	100	-1.0024(0.0014)	19	-3.999(0.2117)	4	-1.544(0.0695)	51
H_b	0	-0.0064(4.95E-4)	81	-20.45(4.26)	0	-89.62(1.277)	0	-113.3(66.39)	69
H_c	0	-4.673(0.249)	5	-5.623(0.313)	4	-377.8(5.478)	0	-17.35(0.0113)	0
H_d	0	-1.80E-7(2.81E-9)	100	-2.44E-7(4.25E-9)	100	-0.274(0.0029)	0	-7.44E-12(3.09E-13)	100
H_e	-1	-1.225(0.0275)	56	-1.407(0.035)	2	-41.61(3.25)	0	-1(0.0E00)	100
H_f	0	-6.16E-16(1.86E-17)	100	-6.13E-6(4.87E-6)	98	-1.00E+3(85.91)	0	-0.1777(0.0037)	0

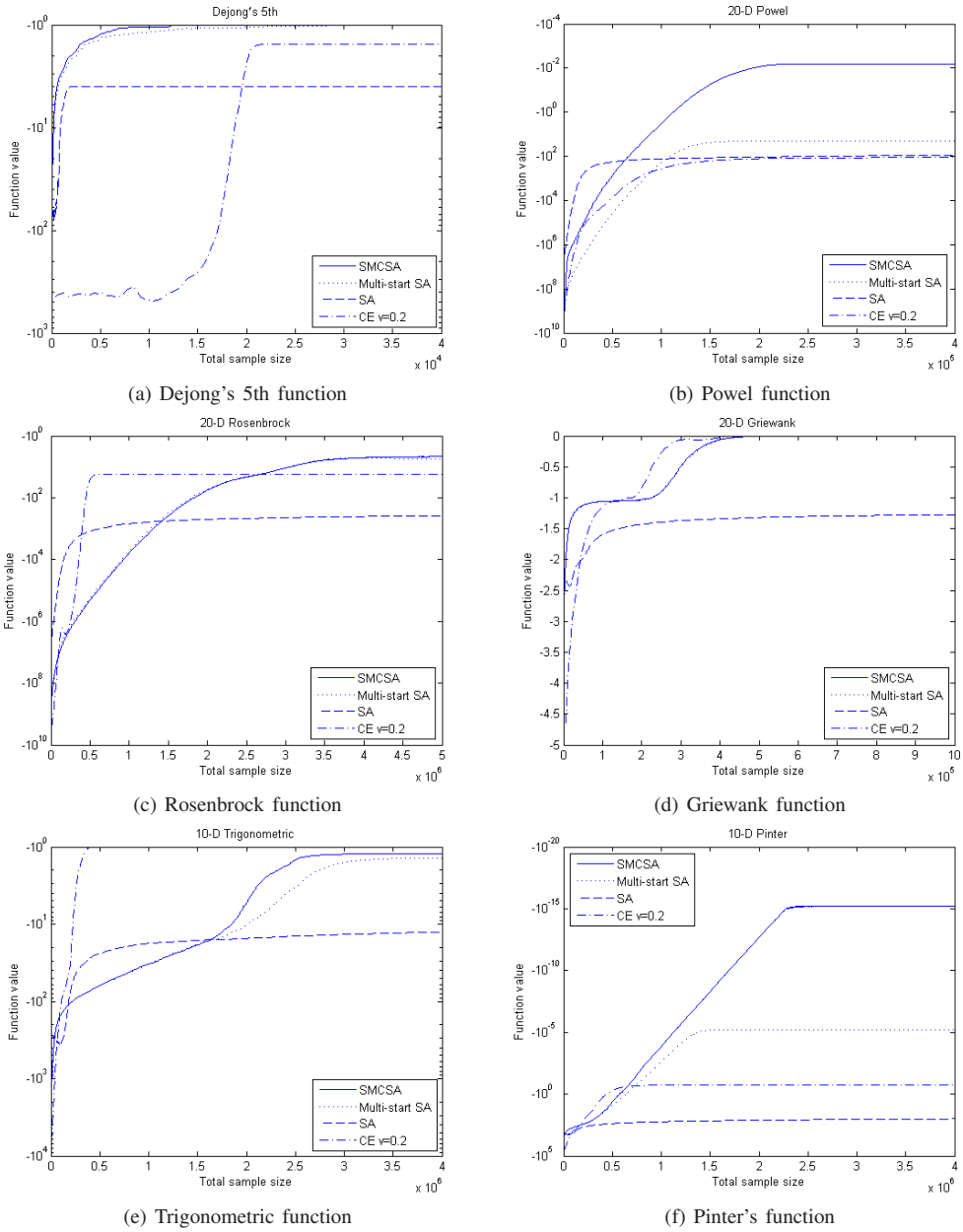
Table 1: Average Performance of SMC-SA, multi-start SA, standard SA and CE on Benchmark Problems

From the results, we may see that for all of these six benchmark problems, SMC-SA outperforms the standard SA. SMC-SA provides much more accurate solutions with smaller standard error, and it also converges faster than standard SA on problems H_a, $H_d - H_f$. SMC-SA performs better than multi-start SA on problems H_a, H_b, H_e and H_f in both accuracy and convergence rate, and performs slightly better than multi-start SA on problems H_c and H_d. In all the problems except H_d and H_e, SMC-SA performs better than CE in accuracy. SMC-SA converges faster than CE on the first two problems; on the last four problems, it converges faster than CE at the very beginning, and then slower.

Dejong's 5th function H_a has low dimension ($n = 2$) and limited number of local optima, and the optima are scattered as shown in Fig. 1(a). SMC-SA outperforms the other three methods in both accuracy and convergence rate. All the solutions by SMC-SA of the 100 replications are 10^{-5}-optimal solutions, whereas there are only 19 by multi-start SA and near half of the solutions by CE. SMC-SA is a population-based algorithm, and it explores the candidate solutions around several promising samples in each iteration. This avoids the candidate solution being trapped in one local optimum, and also concentrates the searching area around several promising candidate solutions. The better the performance of the solution is, the more samples it will generate around it. Moreover, in the low dimensional problem with limited number of local optima, using large enough sample size N, the candidate solutions are more likely to be guided to the global optimum. Multi-start SA takes the best solution among N independent standard SA solutions. The N candidate solutions at each iteration do not interact each other, and hence the new solutions will not concentrate around the elite samples. Standard SA stops producing better solution very early, since it is difficult to escape from a local optimum. CE performs better than standard SA, since it is also a population-based algorithm, and may not be easily trapped in the local optima, especially when using small smoothing parameter ν. But with small smoothing parameter, the convergence is slow and it needs more function evaluations to get good enough solutions.

Powel singular H_b and Rosenbrock H_c functions are high-dimensional ($n = 20$) badly-scaled functions. For Powel singular function H_b, SMC-SA performs much better than multi-start SA, standard SA, and CE. For Rosenbrock function H_c, none of four methods provide good enough solutions. SMC-SA converges slowly, but gives the best average optimal value -4.673; multi-start SA performs slightly worse than SMC-SA; standard SA provides the worst solution. For these two benchmark problems, there are a small number of local optima, but the functions are badly-scaled. SMC-SA improves the solution slightly during the iterations by exploring more around the promising samples. The objective function is smooth and has a small number of local optima, and thus, even with high dimensions, SMC-SA has a low chance to mislead all the sample to concentrate on certain local optimum. SMC-SA outperforms multi-start SA on the Powel singular function, but performs similarly as multi-start SA on the Rosenbrock function.

Griewank H_d ($n = 20$), Trigonometric H_e ($n = 10$), and Pinter H_f ($n = 10$) are high-dimensional problems with a large number of local optima, and the number of local optima increases exponentially with the problem dimension. Pinter H_f is also a badly-scaled function. In these three examples, the performance of SMC-SA, multi-start SA and CE is much better than standard SA, since population-based methods may not easily get trapped in local optima. SMC-SA converges very fast to a good solution at the very beginning, and then slowly improves the solution. For Griewank H_d, CE provides slightly more accurate solution than SMC-SA and multi-start SA; all the best solutions by SMC-SA, multi-start SA and CE are 10^{-5}-optimal solutions. For Trigonometric function H_e, CE performs better than SMC-SA and multi-start SA in both accuracy and convergence rate; all the 100 best solutions by



(a) Dejong's 5th function

(b) Powel function

(c) Rosenbrock function

(d) Griewank function

(e) Trigonometric function

(f) Pinter's function

Figure 2: Average Performance of SMC-SA, multi-start SA, standard SA and CE

CE are 10^{-5}-optimal solutions, and $M_{10^{-5}}$ is only 56 by SMC-SA and 2 by multi-start SA; SMC-SA provides highly accurate solutions but with lower probability. For Pinter's function H_f, SMC-SA and multi-start SA provide more accurate solutions than CE, and all the solutions are within 10^{-5} difference between the true optimal value 0, whereas CE has 0 solutions that are 0.01-optimal. For high-dimensional well-scaled problems with a large number of local optima, such as H_d and H_e, CE outperforms SMC-SA, and SMC-SA performs slightly better than multi-start SA. That is because with large number of local optima, SMC-SA may mislead to explore around local optima. To improve the performance of SMC-SA, we use a large exploration reduction rate $\beta = 0.998$, which allows more exploration without trapped in certain local optimum. For the badly-scaled and with large local optima problem H_f, SMC-SA performs better than multi-start SA and CE. Even when the number

of optima is large, we sacrifice the convergence rate to explore a larger area in order to improve the accuracy.

In summary, SMC-SA is a great improvement of the standard SA on all the test problems; SMC-SA works better than multi-start SA and CE on badly-scaled problems and problems with a small number of local optima; the CE method works better on well-scaled problems with a large number of local optima.

5 CONCLUSION

In this paper, we have proposed the sequential Monte Carlo simulated annealing (SMC-SA) algorithm for continuous global optimization. The main idea is to track the converging sequence of Boltzmann distribution using a population of samples via sequential Monte Carlo method. The numerical results show that SMC-SA is a great improvement of the standard SA on all the test problems and that SMC-SA works better than multi-start SA and CE on badly-scaled problems and problems with a small number of local optima.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant ECCS-0901543.

REFERENCES

Anily, S., and A. Federgruen. 1987. Simulated annealing methods with general acceptance probabilities. *Journal of Applied Probability* 24 (3): 657–667.

Belisle, C. J. P. 1992. Convergence theorems for a class of simulated annealing algorithms on R^d. *Journal of Applied Probability* 29:885–895.

Chu, K. W., Y. Deng, and J. Reinitzy. 1999. Parallel simulated annealing by mixing of states. *Journal of Computational Physics* 148:646–662.

Corana, A., M. Marchesi, C. Martini, and S. Ridella. 1987. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Transactions on Mathematical Software* 13:262–208.

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.

Dekkers, A., and E. Aarts. 1991. Global optimization and simulated annealing. *Mathematical Programming* 50 (3): 367–393.

Doucet, A., J. F. G. de Freitas, and N. J. Gordon. (Eds.) 2001. *Sequential* M*onte* C*arlo methods in practice*. New York: Springer.

Geman, S., and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE *Transactions of Pattern Analysis and Machine Intelligence* 6 (6): 721–741.

Gidas, B. 1985. Nonstationary Markov chains and convergence of the annealing algorithm. *Journal of Statistical Physics* 39:73–131.

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. *IEE Proceedings F (Radar and Signal Processing)* 140 (2): 107–113.

Hajek, B. 1988. Cooling schedules for optimal annealing. *Mathematics of Operations Research* 13:311–329.

Henderson, D., S. H. Jacobson, and A. W. Johnson. 2003. *Handbook of metaheuristics*, Volume 57 of *International Series in Operations Research & Management Science*, Chapter The Theory and Practice of Simulated Annealing, 287–319. Springer.

Kirkpatrick, S., C. D. Gelatt, and J. M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Kroese, D. P., S. Porotsky, and R. Y. Rubinstein. 2006. The cross-entropy method for continuous multiextremal optimization. *Methodology and Computing in Applied Probability* 8:383–407.

Liu, J., and R. Chen. 1998. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* 93:1032–1044.

Locatelli, M. 1996. Convergence properties of simulated annealing for continuous global optimization. *Journal of Applied Probability* 33:1127–1140.

Locatelli, M. 2000. Simulated annealing algorithms for continuous global optimization: Convergence conditions. *Journal of Optimization Theory and Applications* 104 (1): 121–133.

Mahfoud, S. W., and D. E. Goldberg. 1995. Parallel recombinative simulated annealing: a genetic algorithm. *Parallel Computing* 21:1–28.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21 (6): 1087–1092.

Molvalioglu, O., Z. B. Zabinsky, and W. Kohn. 2007. *Models and algorithms for global optimization*, Volume 4 of *Springer Optimization and Its Applications*, Chapter Multi-particle Simulated Annealing, 215–222. Springer.

Molvalioglu, O., Z. B. Zabinsky, and W. Kohn. 2009. The interacting-particle algorithm with dynamic heating and cooling. *Journal of Global Optimization* 43:329–356.

Onbaşoğlu, E., and L. Özdamar. 2001. Parallel simulated annealing algorithms in global optimization. *Journal of Global Optimization* 19:27–50.

Pintér, J. D. 1996. *Global optimization in action*. The Neitherlands: Kluwer Academic Publishers, Dordrecht.

Robert, C. P., and G. Casella. 2004. M*onte* C*arlo statistical methods*. Springer Texts in Statistics. New York: Springer.

Romeijn, H. E., and R. L. Smith. 1994. Simulated annealing for constrained global optimization. *Journal of Global Optimization* 5 (2): 101–126.

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

Ruppeiner, G., J. M. Pedersen, and P. Salamon. 1991. Ensemble approach to simulated annealing. *Journal de Physics I* 1:455–470.

van Hentenryck, P., and Y. Vergados. 2007. Population-based simulated annealing for traveling tournaments. In *Proceedings of the 22nd national conference on artificial intelligence*, Volume 1, 267–272.

van Laarhoven, P. J. M., and E. H. L. Aarts. 1987. *Simulated annealing: Theory and applications*. Springer.

Yang, R. L. 2000. Convergence of the simulated annealing algorithm for continuous global optimization. *Journal of Optimization Theory and Applications* 104:691–716.

Yao, X., and Y. Liu. 1996. Fast evolutionary programming. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, 451–460. Cambridge, MA: MIT Press.

AUTHOR BIOGRAPHIES

ENLU ZHOU is an Assistant Professor in the Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign. She received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009. Her research interests include stochastic control, nonlinear filtering, and simulation optimization. Her email address is <enluzhou@illinois.edu>.

XI CHEN is a Ph.D student in the Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign. Her research interests include simulation optimization and decision analysis. Her email address is <xchen37@illinois.com>.