ECE490 Final Project

Paper Review Report on SIMULATED ANNEALING: A REVIEW AND A NEW SCHEME

Yizhen Lu (yizhenl3)

Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

May 16, 2022

Abstract

In this report, I review the SIMULATED ANNEALING: A REVIEW AND A NEW SCHEME[1] paper by Thomas Guilmeau, Emilie Chouzenoux, and Víctor Elvira. I will first talk about the concepts of the paper, discussing what the paper is aiming to address, and how it achieved the expected goal. Then, I will re-derive some theorems and conclusions that the paper deducts as well as re-implement the algorithms the paper mentions to try my best on re-producing the numerical results the paper provides on the two test problems. Eventually, I will compare results from my own implementation with the results in the paper, and in case there are discrepancies, I will analyze potential causes for the disagreements.

1 Introduction

Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. It is often used when the search space is discrete and when the objective function is non-convex. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to exact algorithms such as gradient descent or branch and bound.

The paper looks into SA-based stochastic optimization method for finding the global minimum of a non-convex optimization problem. As this is a commonly-known hard task widely applied in fields like machine learning and signal processing, the paper goes through the common approaches of solving such problems using SA as well as its improved variants, Fast Simulated Annealing(FSA) and Sequential Monte Carlo Simulated Annealing(SMC-SA), in a unifying manner. Besides, based on the current algorithms, the paper attempts to develop a new algorithm, Curious Simulated Annealing (CSA), that combines the two improved implementations of the SA method. The paper states that this new algorithm inherits the original convergence guarantees of FSA and SMC-SA while obviously improving the performance.

2 Problem Formulation

The paper is mainly solving the optimization problem formulated as:

$$\text{minimize}_{x\in\chi} f(x), \tag{1}$$

where χ is a compact non-empty subset of $\mathbb{R}^d$. The objective function f is supposed to be defined on χ, continuous and thus bounded from both above and below on χ. Besides the general assumptions, there are some specific requirements that the function needs to satisfy that may break the generality.

- $f(x) \geq 0, \forall x \in \chi$
- $S_* := \{x \in \chi, f(x) = 0\}$ is non-empty with null Lebesgue measure.

2.1 Common Notations

Notation	Meaning
$\Delta f := \sup_{x\in\chi} f(x)$	the gap between the infimum and supremum of f
$\|\cdot\|_\infty$	the supremum norm
$\|\cdot\|_2$	the euclidean norm
$\mathcal{B}(\chi)$	Borel algebra of χ
$\mathcal{M}(\chi)$	the set of probability measure on $(\chi, \mathcal{B}(\chi))$
$\|\cdot\|_{TV}$	the total variation norm on $\mathcal{M}(\chi)$
ι_A	the binary indicator function of a set A, take value 1 for $x \in A$ and 0 elsewhere
$\delta_x \in \mathcal{M}(\chi)$	the Dirac measure, for $x \in \chi, A \in \mathcal{B}(\chi)$, it is such that $\delta_x(A) = 1$ if and only if $x \in A$
$(\cdot)_+$	for every $x \in \mathbb{R}$, $(x)_+ = \max(0, x)$
Markov kernel M : $\chi \times \mathcal{B}(\chi) \to \mathbb{R}^+$	$m \in \mathcal{M}(\chi), mM \in \mathcal{M}(\chi)$ $mM(dy) = \int_\chi m(dx)M(x, dy)$

Table 1: Common measure theory concepts and notations

3 Contributions

There are two main contributions for the paper. The first part is reviewing SA-based global optimization strategies in a unifying manner. Then, the paper proposes a new algorithm, Curious Simulated Annealing, which is developed upon combining features of two improved SA-implementations, fast simulated annealing (FSA) and sequential Monte-Carlo simulated annealing(SMC-SA).

The paper focuses on SA-based stochastic optimization method for finding the global minimum of a non-convex optimization problem. As this is a commonly-known hard task widely applied in fields like machine learning and signal processing, the paper goes through the common approaches of solving such problems using SA as well as its improved variants, Fast Simulated Annealing(FSA) and Sequential Monte Carlo Simulated Annealing(SMC-SA), in a unifying manner. The paper reviews these algorithms and provides theorems that retain the algorithms to be still highly generic methods as well as proves that certify these algorithms are mathematically guaranteed to be converging. Besides, based on the current algorithms, the paper attempts to develop a new algorithm, Curious Simulated Annealing (CSA), that combines the two improved implementations of the SA method. The paper states that this new algorithm inherits the original convergence guarantees of FSA and SMC-SA while obviously improving the performance by comparing the output of these algorithms over the same optimization problem as well as the cooling schedules of these algorithms.

4 Convergence Analysis

Since the convergence may require details of the algorithms, they are placed in the appendix for reference. In this section I only re-derive the convergence analysis for two algorithms: Original simulated annealing and sequential monte-carlo simulate annealing as the proof of the other two approaches largely resemble these two methods.

4.1 Convergence of SA

Theorem 1 Under suitable ergodicity hypothesis on G, if there exist $\xi \in (0,1)$ such that

$$T_k = \frac{(1+\xi)\Delta f}{\log(k+2)}, \forall k \in \mathbb{N}, \tag{2}$$

then $\|\mu_k - \pi_k\|_{TV} \to 0$ and $\lim_{k\to\infty} \mathbb{P}(x_k \in S_\epsilon) = 1, \forall \epsilon > 0$.

Prove We would start with the **Theorem 5.1, Lemma 4.1(b), and Theorem 3.3** of [2], we conclude that $\|\mu_k - \pi_k\|_{TV} \to 0$ as long as the following conditions are satisfied:

$$\lim_{k\to\infty} \sum_{i=r_k}^{k} \exp(-\sum_{j=n_{i+1}}^{n_{i+1}} \Delta f \frac{1}{T_j}) = \infty \tag{3}$$

$$|\frac{1}{T_{n_{k+2}}} - \frac{1}{T_{n_{r_k}}}| \text{ is bounded} \tag{4}$$

To prove Equation 3, we can simply choose $n_k = k$ and $r_k = |k - k^{(1-\frac{\epsilon}{2})}|$, and denote $a = \frac{1}{1+\epsilon}$:

$$\begin{aligned}
\lim_{k\to\infty} \sum_{i=r_k}^{k} \exp(-\sum_{j=n_{i+1}}^{n_{i+1}} \Delta f \frac{1}{T_j}) &\geq \sum_{i=r_{k+1}}^{k+1} \mathbf{i}^{\frac{-1}{i+\epsilon}} \\
&\geq \int_{r_{k+1}}^{k+1} x^{-a} dx \\
&= \frac{1}{1-a}[(k+1)^{1-a} - [(k+1) - (k+1)^{(1-\frac{\epsilon}{2})}]^{1-a}] \\
&\geq \frac{1}{a\epsilon}(k+1)^{a\epsilon}[1 - (1 - a\epsilon(k+1)^{-\frac{\epsilon}{2}})] \\
&\geq (k+1)^{\epsilon(a-\frac{1}{2})} \xrightarrow{k\to\infty} \infty
\end{aligned}$$

To prove Equation 4, we denote $1 - \frac{\epsilon}{2} = b$. Define $y_0 = k+2, y_{j+1} = g(y_j)$, for $j \geq 0$, where $g(y) = [y - y^b]$. We can get clearly for fixed n we have $y_{n-1} \geq [k - k^b] \geq y_n$. An easy exercise shows that $y - g(y) \geq g(y) - g(g(y))$. We deduce

$$y_n \leq y_0 - n(y_{n-1} - y_n) \tag{5}$$

$$\begin{aligned} y_{n-1} - y_n &\geq [k - k^b] - g([k - k^b]) \\ &= k - k^b - k + k^b + [k - k^b]^b = [k - k^b]^b \\ &\geq \frac{k^b}{2} \end{aligned}$$

From these two relationships we can get:

$$n \leq \frac{y_0 - y_n}{y_{n-1} - y_n} \leq \frac{k + 2 - g([k - k^b])}{\frac{k^b}{2}} \leq C, \tag{6}$$

where C is a finite constant. Then we can plug in the definition for T_k and get:

$$|\frac{1}{T_{n_{k+2}}} - \frac{1}{T_{n_{r_k}}}| \leq (T_{n_{k+2}})^{-2} \sum_{j=0}^{n} |T_{y_i} - T_{y_{i+1}}| \leq \frac{C \log^2((k+2)+2)}{\log^2(y_n)} \leq C' \tag{7}$$

Therefore, we prove that Equation 3 indeed holds and Equation 4 is bounded. We can derive that **Theorem 1** has been proven true.

4.2 Convergence of SMC-SA

Theorem 2 Consider $\mu_k(dx) = \frac{1}{N_k} \sum_{n=1}^{N_k} \delta_{x_k^{(n)}}(dx)$ and $\mathcal{F}_k$ the history of all past samples until iteration k of SMC-SA algorithm. Then, if the cooling schedule is logarithmic and the sequence $\{N_k\}_{k \in \mathbb{N}}$ increases fast enough, under ergodicity hypothesis on G, there exists a sequence $\{c_k\}_{k \in \mathbb{N}} \searrow 0$, such that for any bounded function ϕ,

$$\mathbb{E}[|\mu_k(\phi) - \pi_k(\phi)||\mathcal{F}_{k-1}] \leq c_k \|\phi\|_\infty. \tag{8}$$

Prove We would start with the initialization step:

$$\langle \mu_0, \phi \rangle = \frac{\sum_{i=0}^{N_0} w(x_i)\phi(x_i)}{\sum_{i=0}^{N_0} w(x_i)}, x_i \overset{\text{i.i.d.}}{\sim} v, \tag{9}$$

where $w(x_i) = \pi_0^d(x_i) v^d(x_i)$. Using Taylor expansion, we get:

$$\mathbb{E}[\langle \mu_0, \phi \rangle] = \mathbb{E}_{\pi_0}[\phi] + \frac{\mathbb{E}_{\pi_0}[\phi]\text{Var}_v(w) - \text{Cov}_v(w, w\phi)}{N_0} + O(N_0^{-2}) \tag{10}$$

$$\begin{aligned} \mathbb{E}[\langle \mu_0 - \pi_0, \phi \rangle] &= \frac{|\langle \pi_0, \phi \rangle \langle v, w^2 \rangle - \langle v, w^2 \phi \rangle|}{N_0} \\ &\leq \frac{\langle v, w^2(\mathbb{E}_{\pi_0}[\phi] - \phi) \rangle}{N_0} \\ &\leq \frac{\|w\|^2 \|\phi\|}{N} \triangleq c_0 \|\phi\|, \end{aligned}$$

$$\begin{aligned}\mathbb{E}[|\left\langle \tilde{\mu_k}^{N_k} - \pi_k, \phi \right\rangle ||\mathcal{F}_{k-1}] &\leq \mathbb{E}[|\left\langle \tilde{\mu_k}^{N_k} - \tilde{\mu_k}, \phi \right\rangle ||\mathcal{F}_{k-1}] + \mathbb{E}[|\left\langle \tilde{\mu_k} - \pi_k, \phi \right\rangle ||\mathcal{F}_{k-1}] \\ &= \mathbb{E}[|\left\langle \tilde{\mu_k}^{N_k} - \tilde{\mu_k}, \phi \right\rangle ||\mathcal{F}_{k-1}] + \mathbb{E}[|\frac{\langle \mu_{k-1}, \Psi_k\phi\rangle}{\langle \mu_{k-1}, \Psi_k\rangle} - \frac{\langle \pi_{k-1}, \Psi_k\phi\rangle}{\langle \pi_{k-1}, \Psi_k\rangle}||\mathcal{F}_{k-1}] \\ &\leq (\frac{1}{\sqrt{N_k}} + c_{k-1}\|\Psi_k\|)\|\phi\|\end{aligned}$$

$$\begin{aligned}\mathbb{E}[|\left\langle \mu_k(\phi) - \pi_k(\phi) \right\rangle ||\mathcal{F}_{k-1}] &= \mathbb{E}[|\left\langle \tilde{\mu_k}^{N_k} P_k - \pi_k, \phi \right\rangle ||\mathcal{F}_{k-1}] \\ &\leq (1-\epsilon_k)(\frac{1}{\sqrt{N_k}} + c_{k-1}\|\Psi_k\|)\|\phi\| \\ &\leq (1-\epsilon_k)(\frac{1}{\sqrt{N_k}} + c_{k-1}\text{exp}\{H^*|\frac{1}{T_k} - \frac{1}{T_{k-1}}|\})\|\phi\| \\ &\triangleq c_k\|\phi\|.\end{aligned}$$

Thus, **Theorem 2** has been proven as well.

5 Numerical Experiments

5.1 Rosenbrock function

The objective function (P1) is the Rosenbrock function in $\mathbb{R}^{10}$, which is ill-conditioned with a unique minimizer that is difficult to find in a large banana-shaped valley:

$$f_1(x) \coloneqq \sum_{i=1}^{9} 5(x_{i+1} - {x_i}^2)^2 + (1-x_i)^2, \forall x \in \mathbb{R}^{10}, \tag{11}$$

minimized at $x_* = (1, 1, ...1)^T$. Initial value is set as $x_0 = 0$.

5.2 Rastrigin function

The second problem (P2) aims at minimizing the Rastrigin function which is highly multimodal with distributed local minima:

$$f_2(x) \coloneqq 10 + \sum_{i=1}^{10} {x_i}^2 - \cos(2\pi x_i), \forall x \in \mathbb{R}^{10}, \tag{12}$$

minimized at $x_* = 0$. Initial value is set at $x_0 = (1, 1, ...1)^T$.

5.3 Results

All self-implemented numerical results are shown in Table 2. The original output from the paper is exhibited in Figure 1. As we can see, though the numbers are largely different, my result still agrees with the original paper result on the performance improvement of the Curious Simulate Annealing.

For these value differences, some potential causes include the difference in implementation, as the original paper uses Julia, while I am using Python. Some other details like sampling approaches may also be slightly different. Also, since the simulated annealing algorithm itself holds a lot of randomness, it is expected that the results may vary among various attempts. Here I basically picks the output that best reflects the performance of the algorithm.

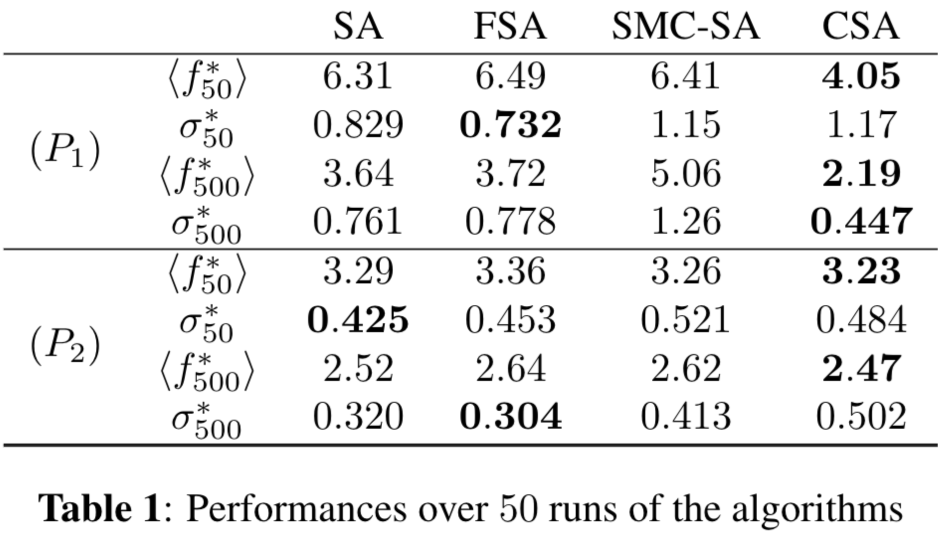
	SA	FSA	SMC-SA	CSA
$\langle f^*_{50} \rangle$	8.03	8.57	**4.37**	**4.37**
σ^*_{50}	1.24	1.60	0.98	**0.73**
$\langle f^*_{500} \rangle$	3.59	4.04	**1.87**	2.00
σ^*_{500}	1.12	1.01	**0.54**	0.64

(a) Test program 1 results

	SA	FSA	SMC-SA	CSA
$\langle f^*_{50} \rangle$	10.12	10.51	9.87	**9.86**
σ^*_{50}	0.53	1.35	0.12	**0.11**
$\langle f^*_{500} \rangle$	9.97	9.63	9.99	**9.58**
σ^*_{500}	**0.20**	1.40	0.40	1.57

(b) Test program 2 results

Table 2: Self-Implemented Numerical Results



		SA	FSA	SMC-SA	CSA
(P_1)	$\langle f^*_{50} \rangle$	6.31	6.49	6.41	**4.05**
	σ^*_{50}	0.829	**0.732**	1.15	1.17
	$\langle f^*_{500} \rangle$	3.64	3.72	5.06	**2.19**
	σ^*_{500}	0.761	0.778	1.26	**0.447**
(P_2)	$\langle f^*_{50} \rangle$	3.29	3.36	3.26	**3.23**
	σ^*_{50}	**0.425**	0.453	0.521	0.484
	$\langle f^*_{500} \rangle$	2.52	2.64	2.62	**2.47**
	σ^*_{500}	0.320	**0.304**	0.413	0.502

Table 1: Performances over 50 runs of the algorithms

Figure 1: Original Paper Result

6 Link to Source Code

All source code for my own implementation can be found at the following link:
https://github.com/happytree718/ECE490_Project

References

[1] Thomas Guilmeau, Emilie Chouzenoux, and Víctor Elvira, “Simulated annealing: a review and a new scheme,” in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, 2021, pp. 101–105.

[2] Heikki Haario and Eero Saksman, “Simulated annealing process in general state space,” *Advances in Applied Probability*, vol. 23, no. 4, pp. 866–893, 1991.

A SA Pseudo-code

```
Algorithm 1: SA
Initialization with $x_0 \sim \mu_0,\ \mu_0 \in \mathcal{M}(\mathcal{X})$
for $k = 1, ...$ do
    Generate a candidate $y_k \sim G(x_k, dy)$
    Compute the acceptance probability
        $p_k = \exp\left(-\left(\frac{f(y_k) - f(x_k)}{T_k}\right)_+\right)$    (3)
    Set $x_{k+1} = \begin{cases} y_k \text{ with probability } p_k \\ x_k \text{ with probability } 1 - p_k \end{cases}$
end
```

Figure 2: SA Algorithm

B SMC-SA Pseudo-code

```
Algorithm 2: SMC-SA
Initialize the algorithm $x_k^{(n)} \sim \mu_0$ for $1 \leq n \leq N_0$;
for $k = 1, ...$ do
    Compute the self-normalized weights
      $w_k^{(n)} \propto \frac{\pi_k}{\pi_{k-1}}(x_{k-1}^{(n)})$
    Resample $\{\tilde{x}_k^{(n)}\}_{n=1}^{N_k}$ from $\{x_{k-1}^{(n)}, w_k^{(n)}\}_{n=1}^{N_k}$
    Generate $\{x_k^{(n)}\}_{n=1}^{N_k}$ propagating the points
      $\{\tilde{x}_k^{(n)}\}_{n=1}^{N_k}$ with the MH kernel $P_k(x, dy)$
end
```

Figure 3: SMC-SA Algorithm