MATH 214 - Computational Stochastics Final Course Project

Roshan Naik Jarupla(1)

(1) University of California, San Diego
Masters CSE, email: rjarupla@eng.ucsd.edu, PID: A53301504

Keywords: Markov chain Monte Carlo simulation, knapsack, computational stochastics.

Abstract. This paper is a part of the course project in MATH 214. It is a detailed study and analysis of Simulated Annealing problem applied to Knapsack which is known to be NP-Hard. Due to the simplicity of Simulated Annealing, this algorithm has been greatly studied and applied to various challenging problems to find an approximate optimal solution. This project report is organized into four sections. First, the mathematical formulation of Knapsack. Second, the pseudo-code for general Simulated Annealing. Third, the experiments and analysis. Finally, the conclusion.

1 Problem Description

$$
\begin{aligned}
\text{Given: } & m \geq 1 \text{ an integer} \\
& \overrightarrow{v} = (v_1, \dots, v_m) \in \mathbb{R}^m, v_i \text{ each } v_i \geq 0 \\
& \overrightarrow{w} = (w_1, \dots, w_m) \in \mathbb{R}^m, v_i \text{ each } w_i \geq 0 \\
& b \in \mathbb{R}^+ \\
\text{Objective: } & \max \overrightarrow{v} \cdot \overrightarrow{z} = \max \sum_{i=1}^{m} v_i z_i \\
& \text{s.t.} \overrightarrow{w} \cdot \overrightarrow{z} = \sum_{i=1}^{m} w_i z_i \leq b \\
& z_i = 0 \vee z_i = 1 \quad \forall i = \{1, 2, \dots, m\}
\end{aligned}
\tag{1}
$$

We denote S the set of all z which satisfy the constraints $\sum_{i=1}^{m} w_i z_i \leq b$, and $z_i = 0 \vee z_i = 1 \quad \forall i = \{1, 2, \dots, m\}$. Therefore,

$$
S = \left\{ \overrightarrow{z} \in \mathbb{R}^m | \sum_{i=1}^{m} w_i z_i \leq b . z_i = 0 \vee z_i = 1 \quad \forall i = \{1, 2, \dots, m\} \right\} \tag{2}
$$

2 Algorithm Description

The algorithm is a follows:
Define: $\pi_{\beta(t)}(\overrightarrow{z}) = \frac{1}{Z_{\beta(t)}} e^{\beta(t) \cdot \overrightarrow{v} \cdot \overrightarrow{z}}, \quad \forall z \in S$
Here, $\pi_{\beta(t)}(\overrightarrow{z})$ is the Boltzmann-Gibbs distribution, and $\beta(t)$ is the annealing schedule. It is the gradually increased with t and is the modification over Metropolis-Hastings MCMC method.

Algorithm 1 Simulated Annealing for Knapsack

Input: $\overrightarrow{v} = (v_1, \ldots, v_m), \overrightarrow{w} = (w_1, \ldots, w_m), b$ non-negative

$X_0 = \overrightarrow{z} = (z_1, \ldots, z_m), k := 0, steps := 20000$

while $k \leq steps$ **do**

  Select $J \in \{1, \ldots, m\}$ uniformly at random

  $X_k = \overrightarrow{z} \in S.\ \ \overrightarrow{z} = (z_1, \ldots, z_m)$

  $Y = (z_1, \ldots, z_{J-1}, 1 - z_J, z_{J+1}, \ldots, z_m) = \overrightarrow{y}$;   `// Flip` $\overrightarrow{z}$ `at index J`

  if $\overrightarrow{y} \notin S$ **then**

    $X_{k+1} \leftarrow X_k$;   `// Reject Y`

  else if $\overrightarrow{y} \in S$ **then**

    Pick $u \sim \mathcal{U}[0, 1]$

    $\alpha = min\left(1, \frac{\pi_{\beta(k)}(\overrightarrow{y})}{\pi_{\beta(k)}(\overrightarrow{z})}\right)$

    if $u \leq \alpha$ **then**

      $X_{k+1} \leftarrow Y$;   `// Accept Y`

    else

      $X_{k+1} \leftarrow X_k$;   `// Reject Y`

    end

  $k = k + 1$

end

3 Simulation Results and Analysis

Simulated Annealing is a branch of methods for simulating a Markov Chain that attains a desired invariant distribution π of probability values for each state in the state space S. Markov Chain Monte Carlo (MCMC) simulation methods were greatly successful in obtaining a sequence of random samples from a probability distribution from which direct sampling is difficult. Metropolis Hastings algorithm [1] is one such MCMC method for sampling from a distribution that is widely recognized. The Simulated Annealing [2, 3] relaxes the time-homogeneous assumption of the underlying Markov Chain and introduces an annealing schedule (temperature) for better convergence. This method is used to find approximate optimal solutions to complicated NP-Hard problems like Knapsack [4], Traveling Salesman, Bin-Packing [5], etc., and many other real world hard problems like Airline Crew Scheduling, Railway Crew Scheduling, Vehicle Routing Problem, Layout-Routing of Electronic Circuits, Large Scale Aircraft Trajectory Planing, Complex portfolio problem, Graph coloring problem, High-dimensionality minimization problems, etc.

In the context of this project, we look at the application of Simulated Annealing optimization to find optimal solutions for 0/1 Knapsack and perform some tests and experiments with different annealing schedules, convergence rates, and plot the simulation results on various examples. Knapsack often arises as a sub-problem in resource allocation applications where there are financial constraints, such as:

• Cargo loading (truck, boat, cargo aircraft)
• Satellite channel assignment
• Portfolio optimization

This problem is easy to formulate but hard to solve due to the associated combinatorics:

We present 4 examples with increasing complexity and study how different annealing schedules results affects the simulation results. Three annealing schedules are tested:

(a) Linear: $\beta(t) = c(1 + 0.002 * k)$
(b) Logarithmic: $\beta(t) = c(1 + log(1 + 0.002k))$
(c) Geometric: $\beta(t) = c(1 + 0.0002)^k$

Here c is just a normalizing constant to handle cases when $\overrightarrow{v} \cdot \overrightarrow{z}$ is large. We divide

Table 1: Complexity of Search Space

n	2^n
10	$1.024 * 10^3$
20	$1.048 * 10^6$
40	$1.099 * 10^12$
60	$1.152 * 10^18$
80	$1.208 * 10^24$
100	$1.267 * 10^30$

the simulation into two parts. For the first 10000 iterations, we use constant β, and for the next 10000 iterations, we gradually increase β according to the annealing scheme.

We know that Simulated Annealing converges to one of the optimal solutions by theory. Let $M = \underset{\vec{z} \in S}{max} \vec{v} \cdot \vec{z}$ and $S_{max} = \{\vec{z} \in S : \vec{v} \cdot \vec{z} = M\}$. Then,

$$\underset{\beta \to \infty}{lim} \left(e^{-\beta M} \sum_{\vec{z} \in S} e^{\beta \vec{v} \cdot \vec{z}} \right) = \underset{\beta \to \infty}{lim}\, e^{-\beta M} Z_\beta = |S_{max}|. \tag{3}$$

Hence,

$$\underset{\beta \to \infty}{lim}\, \pi_\beta(\vec{z}) = \begin{cases} \frac{1}{|S_{max}|} & : \vec{z} \in S_{max} \\ 0 & : \vec{z} \notin S_{max} \end{cases} \tag{4}$$

We know that $F_\beta = \frac{1}{\beta} log(Z_\beta) \to \underset{\vec{z} \in S}{max} \vec{v} \cdot \vec{z} = M$, and $E_\beta = \sum_{\vec{z} \in S} \vec{v} \cdot \vec{z} \pi_\beta(\vec{z}) \to M$. So we can approximate Z_β from F_β due to it's convergence and thereby also approximate $|S_{max}|$.

3.1 *Example 1 - Equal Weights*

We start by testing for a simple problem with equal weights.
$\vec{v} = [1, 2, 3, 4, 5, 3, 2, 1, 4, 8, 0, 1, 4, 6, 3]$, $\vec{w} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$, $b = 5$. Under this condition, we can solve the problem efficiently in $O(nlog(n))$ by sorting the items by decreasing v and the top b items. Simulated Annealing results for the same are shown below:

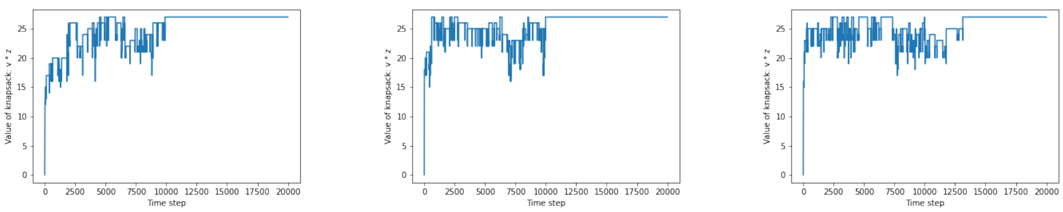

Figure 1: Annealing Schedules - (a) Linear (b) Logarithmic (c) Geometric

The value converges to 27 which is the optimal result. Therefore, $M = 27$ and $Z_\beta = e^{\beta F_\beta}$. Finally, $|S_{max}| = e^{-\beta M} Z_\beta = 1$.

3.2 *Example 2 - Simple*

We set $\vec{v} = [5, 7, 5, 8, 11]$, $\vec{w} = [3, 4, 5, 8, 10]$,
$b = 20$. Under this problem size, we can solve efficiently quickly. Simulated Annealing converges in less than 100 steps. The results for the same are shown below:

The value converges to 25 which is the optimal result with $\vec{z} = [1, 1, 1, 1, 0]$. Therefore, $M = 25$ and $Z_\beta = e^{\beta F_\beta}$. Finally, $|S_{max}| = e^{-\beta M} Z_\beta = 1$.

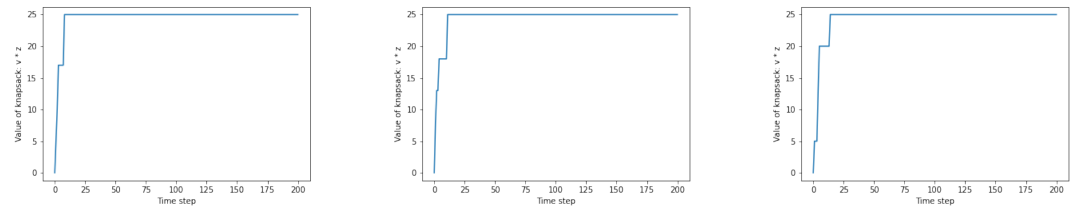

Figure 2: Annealing Schedules - (a) Linear (b) Logarithmic (c) Geometric

3.3 *Example 3 - Complex*

We set $\overrightarrow{v} = [360, 83, 59, 130, 431, 67, 230, 52, 93, 125, 670, 892, 600, 38, 48, 147, 78, 256, 63, 17, 120, 164, 432, 35, 92, 110, 22, 42, 50, 323, 514, 28, 87, 73, 78, 15, 26, 78, 210, 36, 85, 189, 274, 43, 33, 10, 19, 389, 276, 312]$,
$\overrightarrow{w} = [7, 0, 30, 22, 80, 94, 11, 81, 70, 64, 59, 18, 0, 36, 3, 8, 15, 42, 9, 0, 42, 47, 52, 32, 26, 48, 55, 6, 29, 84, 2, 4, 18, 56, 7, 29, 93, 44, 71, 3, 86, 66, 31, 65, 0, 79, 20, 65, 52, 13]$,
$b = 850$.

Under this problem size, we can only find approximate solutions in reasonable time. Simulated Annealing is run for 20000 steps. The results for the same are shown below:

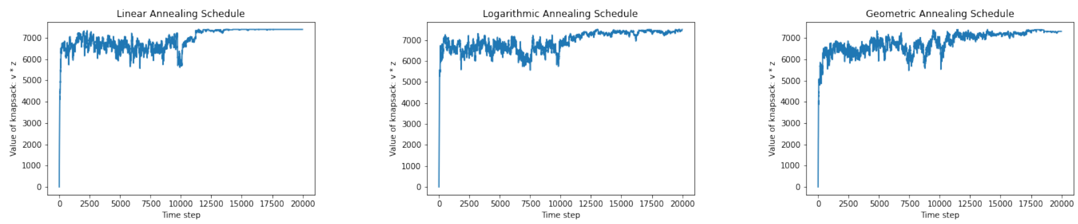

Figure 3: Annealing Schedules - (a) Linear (b) Logarithmic (c) Geometric

The approximate optimal value is 7454.215 under geometric annealing schedule. It seems like geometric and logarithmic schedule perform a little better linear schedule. Logarithmic schedule increases β by fewer amount each time step. Geometric schedule also increases β slowly, but Linear schedule increases β by a constant amount in every iteration.

3.4 *Example 4 - Very Hard*

Number of items = 300. We generate 300 dimensional vectors. $\overrightarrow{v}$ randomly from a uniform distribution between 10 to 1000 and $\overrightarrow{w}$ randomly from a uniform distribution between 0 to 100, $b = 2500$.

Under this problem size, we can only find approximate solutions. Simulated Annealing is run for 20000 steps. The results for the same are shown below:

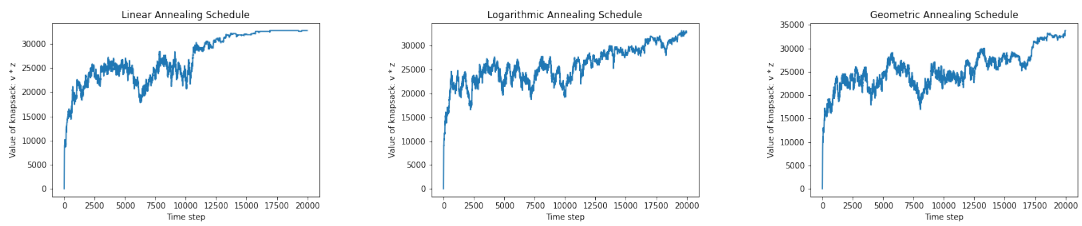

Figure 4: Annealing Schedules - (a) Linear (b) Logarithmic (c) Geometric

The approximate optimal value is 32663.837 under geometric annealing schedule. It seems like geometric and logarithmic schedule perform a little better linear schedule. Logarithmic schedule increases β by fewer amount each time step. Geometric schedule also increases β slowly, but Linear schedule increases β by a constant amount in every iteration.

4 Conclusion

We presented Simulated annealing (SA), a global optimization metaheuristic. The main advantage of SA is its simplicity. SA is based on an analogy with the physical annealing of materials that avoids the drawback of the Monte-Carlo approach (which can be trapped in local minima), thanks to an efficient Metropolis acceptance criterion. When it is hard to directly evaluate the probability of a state in very large state space, simulated annealing is the right strategy to apply. This is due to the fact that we do not need to know the exact probability vector π but only the ratios between the probability of two states. The analysis shows that Logarithmic or Geometric performs better empirically compared to simple Linear annealing schedule. This algorithm has vast applications in solving other problems as well which makes it's study and understanding very important.

References

[1] W. K. Hastings. " Monte Carlo sampling methods using Markov chains and their applications". *Biometrika*, Volume 57, Issue 1, April 1970, Pages 97–109.

[2] Martin Pincus. "A Monte-Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems". *Journal of the Operations Research Society of America*, 18 (6): 967–1235, Nov–Dec 1970.

[3] Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. "Optimization by Simulated Annealing". *Science*. 220 (4598): 671–680, 1983.

[4] Mathews, G. B. "On the partition of numbers". *Proceedings of the London Mathematical Society*, 28: 486–490, 1897.

[5] Skiena, S. S. "Who is Interested in Algorithms and Why? Lessons from the Stony Brook Algorithm Repository". *ACM SIGACT News*. 30 (3): 65–74, 1999.