Simulated Annealing

Arna Fariza – S2 TIK PENS

Introduction to Simulated Annealing

- **Definition:** Simulated Annealing (SA) is a probabilistic technique for approximating the global optimum of a function.
- **Inspiration:** Based on the process of annealing in metallurgy. Origin in metallurgy—annealing is the process of heating and slowly cooling metal to decrease defects.
- **Key Idea:** Gradually lower the "temperature" to reduce the system’s energy, leading to an optimal solution.
- **When to Use**: Discuss how it’s particularly helpful for combinatorial and non-convex optimization problems, where traditional methods might get stuck in local optima.

Why Use Simulated Annealing?

- **Optimization Problems:** Often used when the search space is large and complex.
- **Advantages:** Can escape local minima, efficient for certain types of problems.
- **Limitations:** Results depend on cooling schedule and can be computationally intensive.

Core Principles

- Metaphor of Energy and Temperature:
  - **Energy (E):** Represents the objective function or solution quality in optimization.
  - **Temperature (T):** Controls the probability of accepting worse solutions.
- Cooling Schedule: Highlight how temperature gradually decreases during the process.

How Simulated Annealing Works

1. **Start with an initial solution** (randomly generated).
2. **Initialize a high temperature.**
3. **Iterate:** For each temperature, make small changes to the solution.
   - If the change improves the solution, accept it.
   - If not, accept it with a probability dependent on the temperature.
4. **Reduce temperature gradually** following a cooling schedule until it reaches a threshold.

Key Components of Simulated Annealing

- **Temperature (T):** Controls the probability of accepting worse solutions.
- **Cooling Schedule:** Determines how fast temperature decreases.
- **Acceptance Probability:** Defines the likelihood of accepting worse solutions.

Algorithm step

1. **Initialization**: Choose an initial solution and set initial temperature.
2. **Iterative Process**:
    1. Generate a *neighbor solution*.
    2. Calculate the *change in energy* (ΔE).
    3. Decide whether to accept the new solution:
        1. If better, accept it.
        2. If worse, accept with a probability based on temperature and ΔE (use formula).
3. **Cooling**: Gradually lower the temperature following a predefined *cooling schedule* (e.g., geometric, linear).
4. **Stopping Criterion**: Repeat until the system "freezes" (temperature reaches minimum or no more improvement is possible).

Algorithm component

- **Neighborhood Selection:** Explain how to define neighboring solutions, relevant to specific problems.
- **Acceptance Probability:** Introduce the Metropolis criterion:

$$P(accept)=\exp(-\Delta E/T)$$

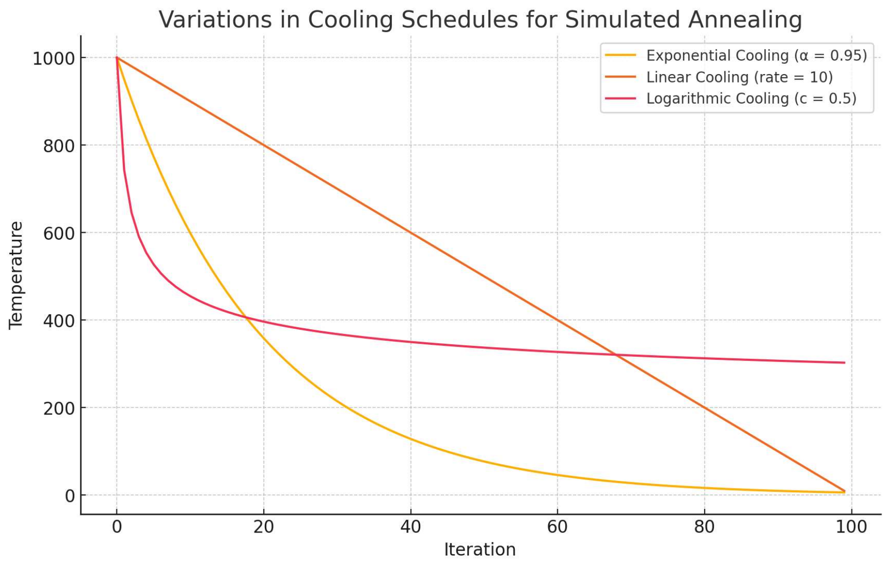
- **Cooling Schedule:** Various cooling strategies (e.g., exponential decay with Tnew=α×ToldT where α is between 0.8 and 0.99).

Acceptance Probability Formula

- **Formula:** $P(\Delta E)=\exp(-\Delta E/T)$
  - Where ΔE is the change in solution quality.
  - **Explanation:** Higher T means higher probability of accepting a worse solution.

Cooling Schedules

- **Linear Cooling:** Decrease temperature linearly.
- **Geometric Cooling:** Multiply temperature by a constant (e.g., 0.9).
- **Logarithmic Cooling:** Slower cooling, often provides better accuracy.



Application of Simulated Annealing

Traveling Salesman Problem (TSP)

Scheduling and Resource Allocation

Machine Learning: Hyperparameter Tuning

Advantages & Limitations

Advantages:

- Simple and flexible, Can handle large and complex spaces.

Limitations:

- Heavily dependent on the cooling schedule, slower than some other heuristics.

Summary

Wrap-Up: Simulated annealing offers a practical way to approach optimization problems by balancing exploration and exploitation.

Key Takeaway: Effectiveness depends on properly tuning parameters like temperature and cooling rate.