# Artificial Intelligence Principles

6G7V0011 - 1CWK100

Dr. Peng Wang
Email: p.wang@mmu.ac.uk

Department of Computing and Mathematics

Tuesday, Nov. 25th, 2024

# Outline

# Outline

# Quiz

**Q1**. Time and space complexity of BFS and DFS.

**Q2**. Pros and cons of hill climbing.

**Q3**. The advantages of Simulated annealing compared to hill climbing.

# Outline

Quiz

Search Algorithms

- Simulate Annealing

# Local Search - Simulated Annealing

Figure 1: Annealing example, image credits

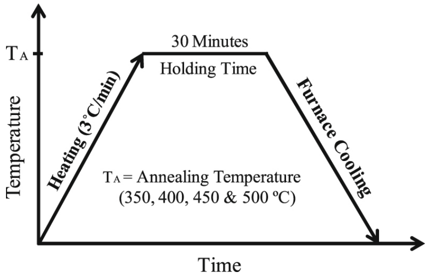


Figure 2: Annealing temperature, image credits

**'In metallurgy, annealing is the process used to temper or harden metals and glass by heating them to a high temperature and then gradually cooling them, thus allowing the material to reach a low-energy crystalline state.'**

**Note**: From **hill climbing** to **gradient descent**. Essentially the same, a '-' difference, i.e., $\min f(x) \iff \max -f(x)$

## Takeaway from annealing

- A cool section is not good enough (otherwise why bother to reheat it?)
- Reheat it with a strategy
- Cool it down
- A better section (the same one, but good quality)

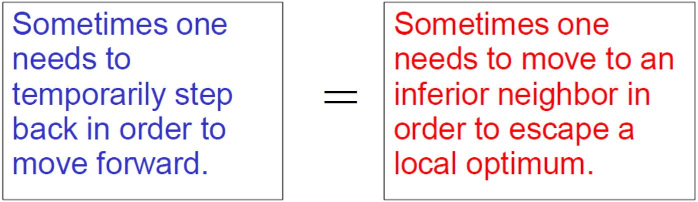


Figure 3: Revisited history for a brighter future (Xiaojin Zhu)

Heated $\rightarrow$ Cool $\rightarrow$ Bad $\rightarrow$ **Reheated** $\rightarrow$ **Cool** $\rightarrow$ **Better** $\rightarrow \cdots$

# Local Search - Simulated Annealing I



Algorithm 1: Pseudocode of the simulated annealing algorithm

**Input:** Initial temperature **Temp**, cost function $f$, scheduling function *schedule(current.state, neighbor.state, T)*=$\exp(-\frac{|\Delta E|}{T})$, time $t$

**Output:** A local maximum state **g**

*current* ← initial *node* with **s** as state
**for** $t \in [1, sys.maxsize]$ **do**
  $T = Temp * \exp(-\lambda * t)$, $k = 20$, $\lambda = 0.05$
  **if** $T \simeq 0$ **then**
    **return** *current.state*
  **end if**
  *neighbor* ← random choice from *neighbors*
  $\Delta E = f(current.state) - f(neighbor.state)$
  **if** $\Delta E > 0$ **then**
    *current* ← *neighbor*   neighbor is better, always accept.
  **else**

# Local Search - Simulated Annealing II

```
        if exp(−|ΔE|/T) ≥ random(0,1) then
            current ← neighbor        neighbor is worse, accept with a probability.
        else
            current ← current
        end if
    end if
end for
```

On choosing temperature $T$

- High: almost always accept any new state
- Low: First-choice hill climbing

**Tricks**: Determine them via experiments!

# Understanding Scheduling

Codes on Moodle!

# Application - The Travelling Salesman Problem (TSP)

Manchester Metropolitan University

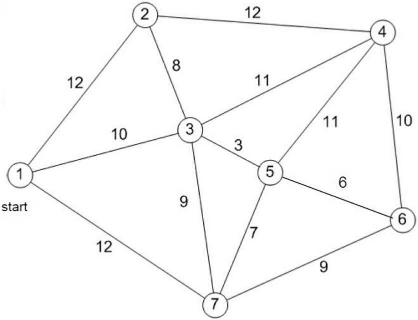


Figure 4: Nodes are places, arcs are accessibility, and numbers on arcs are costs.

- Starting from city 1, the salesman must travel to all cities once before returning home
- The distance between each city is given, and is assumed to be the same in both directions
- Only the links shown are to be used
- Objective - Minimise the total distance to be travelled
- Current state: **1234567**
  Neighbors: **1254367**, **1256347**, etc.

Wikipedia has some interesting applications!

**Have look at this video on how simulated annealing works!**

# Pros and Cons - Simulated Annealing

## Pros

- Not much to say theoretically
  - With infinitely slow cooling rate, finds global optimum with probability 1
  - Proposed by Metropolis in 1953 based on the analogy that alloys manage to find a near global minimum energy state, when annealed slowly.
- Easy to implement.

## Cons

- Cooling scheme important
- Neighborhood design is the real ingenuity, not the decision to use SA
  1. Permutation
  2. Combinatorial problem
  3. E.g. TSP

# Summary

## Summary

- Local search in general
- Hill climbing

  1. Maximise **objective** function
  2. Could stuck in local optima, variants are less efficient

- Simulated annealing

  1. Minimise **cost** function
  2. Annealing guarantees to find global optima with probability 1