Artificial Intelligence Principles

6G7V0011 - 1CWK100

Dr. Peng Wang

Email: p.wang@mmu.ac.uk

Department of Computing and Mathematics

Tuesday, Dec. 3rd, 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.

Application - The Travelling Salesman Problem (TSP)

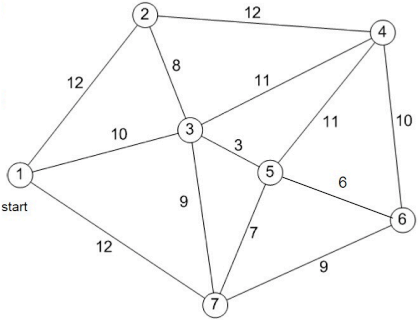


Figure 1: Nodes are places, arcs are accessibility, and numbers on arcs are costs.

- 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.

2-opt for TSP - I

In optimization, **2-opt** is a simple local search algorithm for solving the traveling salesman problem (TSP).

procedure 2optSwap(route, i, k)
{

- take route[0] to route[i] and add them in order to new_route
- take route[i+1] to route[k] and add them in reverse order to new_route
- take route[k+1] to route[end] and add them in order to new_route

return new_route
}

2-opt for TSP - I

In optimization, **2-opt** is a simple local search algorithm for solving the traveling salesman problem (TSP).

Example route: $a \rightarrow b \rightarrow e \rightarrow d \rightarrow c \rightarrow f \rightarrow g$

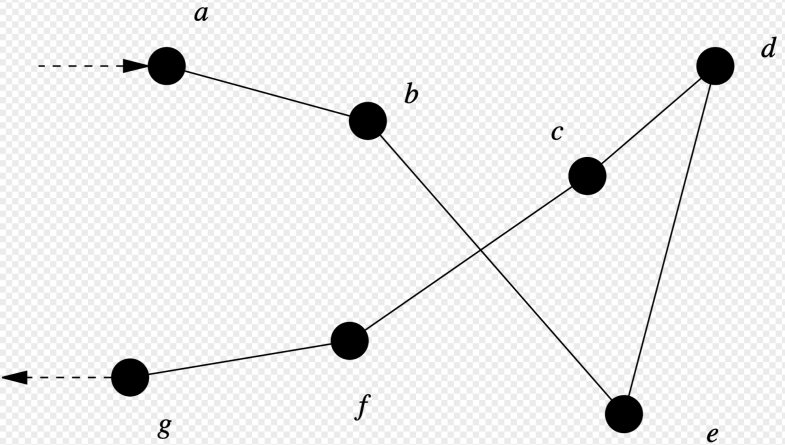
Example parameters: i = 1, k = 4 (starting index 0)

Contents of new_route by step:

- $(a \rightarrow b)$
- $a \rightarrow b \rightarrow (c \rightarrow d \rightarrow e)$
- $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow (f \rightarrow g)$

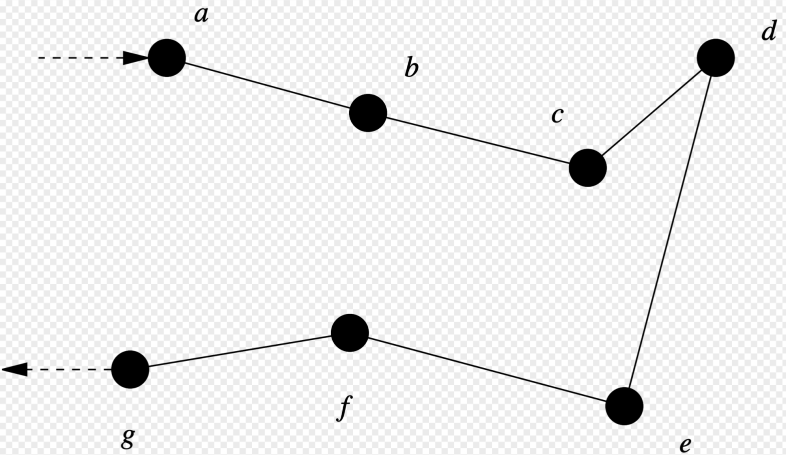
2-opt for TSP - II

In optimization, **2-opt** is a simple local search algorithm for solving the traveling salesman problem.



2-opt for TSP - III

In optimization, **2-opt** is a simple local search algorithm for solving the traveling salesman problem.



Outline

The Theory of Evolution

The **theory of evolution** was developed by Charles Darwin in On the Origin of Species by Means of Natural Selection (1859) and independently by Alfred Russel Wallace (1858). The central idea is simple: **variations occur in reproduction and will be preserved in successive generations approximately in proportion to their effect on reproductive fitness.**

Watch this video:
Theory of Evolution: How did Darwin come up with it?

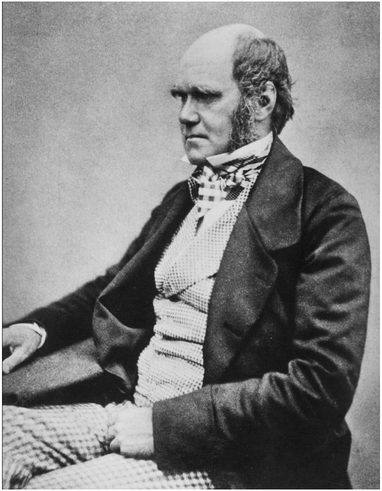


Figure 2: Darwin, when he was preparing **On the Origin of Species**.

Local Search - Genetic Algorithm

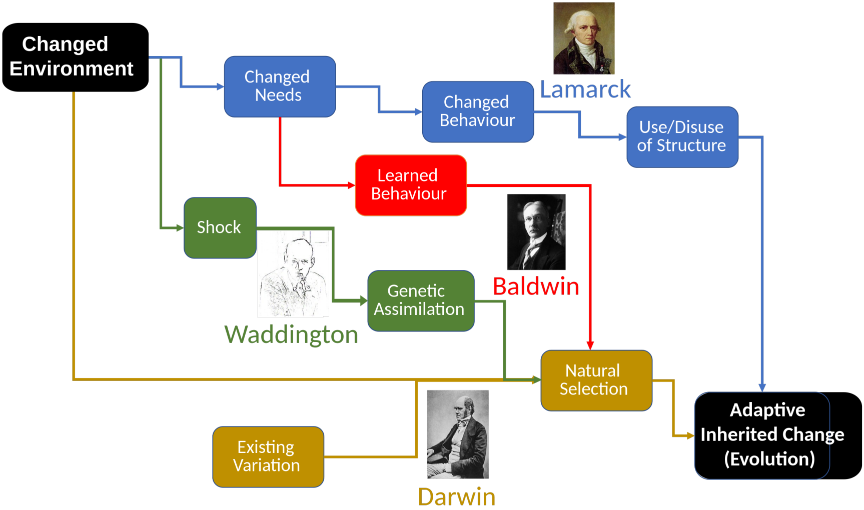
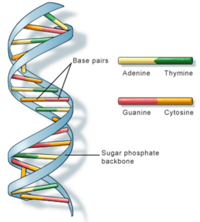


Figure 3: About Evolution!

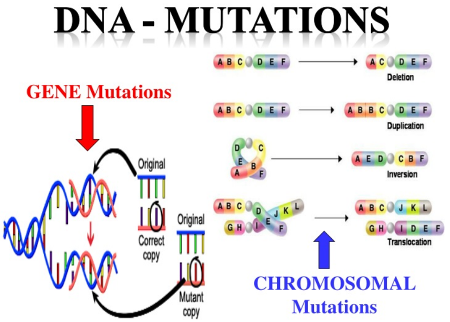
Local Search - Genetic Algorithm

- Individuals compete for resources
- Individuals with better genes (not necessarily the strongest but the ones that adapt to the environments) have a larger chance to produce offspring, and vice versa
- After many generations, the population consists of lots of genes from the superior individuals, and less from the inferior individuals
- Superiority defined by the fitness to the environment

  Gene or DNA encodes and passes down the changes, etc.



- **DNA is the hereditary material in humans and almost all other organisms.**
- **The information in DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T).** Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people.
- **The order of these bases determines the information available for building and maintaining an organism**, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences.
- **DNA bases pair up with each other**, A with T and C with G, to form units called base pairs.
- **An important property of DNA is that it can replicate**. Each strand of DNA in the double helix can serve as a pattern for duplicating the sequence of bases. This is critical when cells divide because each new cell needs to have an exact copy of the DNA present in the old cell.



- DNA Replication
- The fittest genes pass down, A (Adenine), C (Cytosine), T (Thymine) and G (Guanine)
- Crossover → diversity
- Mutation → could be good or bad
- Credits left right

Local Search - Genetic Algorithm

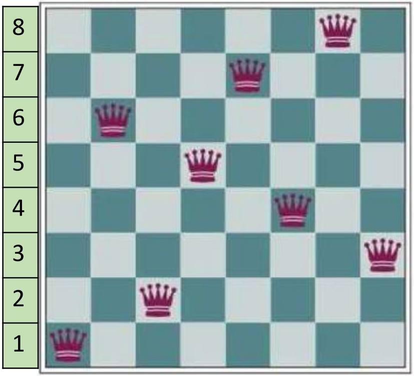


Figure 4: Coding a state

One way to code is:
[1 6 2 5 7 4 8 3] % this is a state
What about TSP?

- Each state s is called an **individual**.
- Often (carefully) coded up as a string.
- The score $f(s)$ is called the **fitness** of s.
- Our goal is to find the global optimum (fittest) state.
- At any time we keep a fixed number of states.
- They are called the **population**.

Algorithm 1: Pseudocode of the genetic algorithm

Input: Initial population $\mathbf{S} = \{\mathbf{s_0}, \mathbf{s_1}, \cdots, \mathbf{s_{n-1}}\}$ with n individuals $\mathbf{s_i}$, $i = 0, \cdots, n-1$, maximum generation N, a fitness function f and fitness threshold f_{th}

Output: An individual $\mathbf{s}$

```
for gen ∈ [0, N − 1] do
    for i ∈ [0, n − 1] do
        p_i = f(s_i) / Σ_{j=0}^{n−1} f(s_j)
    end for
    parent1 ← randomly pick from S according to p
    parent2 ← randomly pick from S according to p
    randomly select a crossover point, split and recombine parent1 and
    parent2 to generate a child % parameter
    mutate child according to mutation rate % parameter
    if found fittest_individual then
```

Local Search - Genetic Algorithm II

```
        return fittest_individual % fitness threshold
    end if
end for
return individual with the best fitness
```

Information of a population with 5 individuals

Individual	Fitness	Probability
A	5	10%
B	20	40%
C	11	22%
D	8	16%
E	6	12%

Table 1: An example

Proportional selection

- $p_i = f(\mathbf{s}_i) / \sum_{j=0}^{n-1} f(\mathbf{s}_j)$
- $\sum_{j=0}^{n-1} f(\mathbf{s}_j) = 5 + 20 + 11 + 8 + 6 = 50$
- $p_1 = 5/50 = 10\%$

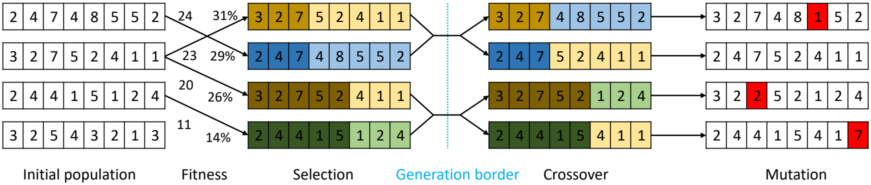


Figure 5: Illustration of genetic algorithm

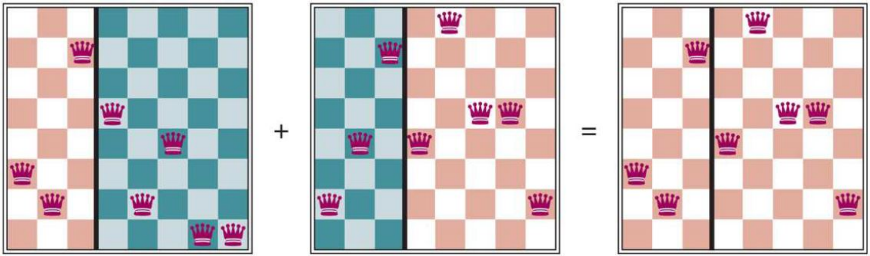


Figure 6: Illustration of genetic algorithm - results

Local Search - Genetic Algorithm

- State encoding is the real ingenuity, not the decision to use genetic algorithm.
- Lack of diversity can lead to premature convergence and non-optimal solution
- Not much to say theoretically
  - Cross over (sexual reproduction) much more efficient than mutation (asexual reproduction).
- Easy to implement.
- **A lot of variants out there!**

The eight queens problem - a nice video

Not familiar with the eight queens problem, check the followings

Click on Eight queens problem explanation to watch

or paste in your browser the following link

https://www.youtube.com/watch?v=jPcBU0Z2Hj8

Lab - The Travelling Salesman Problem (TSP)

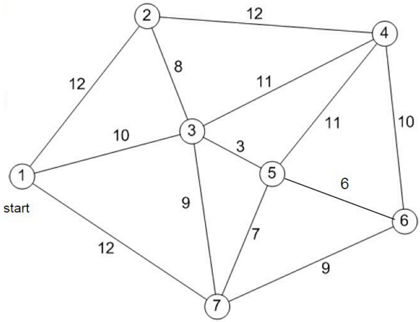


Figure 7: Nodes are places, arcs are accessibility, and numbers on arcs are costs.

- 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.

Outline

Quiz

Q1. What are the key steps in genetic algorithm.

Q2. What's the benefit of mutation?