**CX2001 Algorithms** **2020/21 Semester 1**

# 8: Heuristic Algorithms

*School of Computer Science and Engineering* *Nanyang Technological University*

## 8.1 Heuristic Algorithms

Many real-word problems we encountered are related to some NP problems introduced in previous lecture. Although we cannot solve it in polynomial-time by using deterministic algorithm, we can

- Use smaller problem sizes
- Solve for a special instance of the problem
- Use heuristic algorithms to get a close to optimal solution

A heuristic is a common-sense rule drawn from experience rather than from a mathematically proved assertion. Approximation algorithms are usually based on some problem-specific heuristic. These are fast (polynomially bounded) algorithms that are not guaranteed to give the best solution but will give one that is close to the optimal.

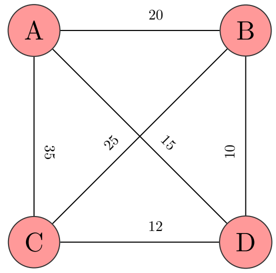
Here, we will introduce two approximation algorithm based on **Greedy strategies** for the Traveling Salesman Problem.

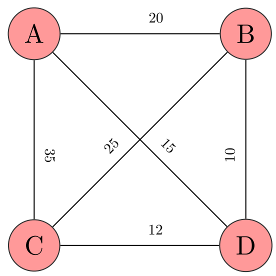
- The Nearest Neighbour Strategy
- The Shortest Link Strategy

### 8.1.1 Nearest Neighbour Algorithm

**Algorithm 1** Nearest Neighbour TSP Algorithm

nearestTSP (V,E)
Select an arbitrary vertex $s \in V$ to start the cycle $C$
$v \leftarrow s$
**while** there are vertices not in $C$ **do** ▷ Run $|V| - 1$ iterations
    select an edge vw of minimum weight AND w is not in C;
    add edge vw to C;
    v = w;
add edge vs to C; ▷ Return to the starting vertex
Return C;




cost =0

Nearest Neighbour Algorithm

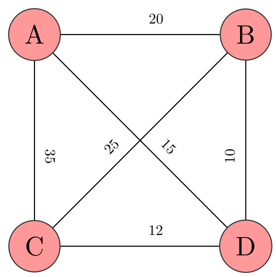
Starting from vertex A, the nearest-neighbour strategy results in total distance of 85. However, it is not the minimum distance. This is an example that a greedy strategy does not always give an optimum solution.

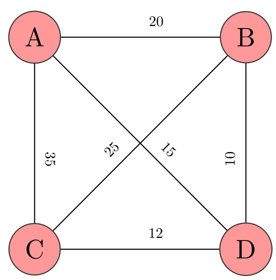
### 8.1.2 Shortest Link Algorithm

Shortest Link Algorithm aims to build a cycle C in an incremental manner. It iteratively selects the edge with the lightest weight but the edge cannot form a cycle with other selected edge and the edge cannot be the third selected edge of any vertices.

**Algorithm 2** Shortest Link TSP Algorithm

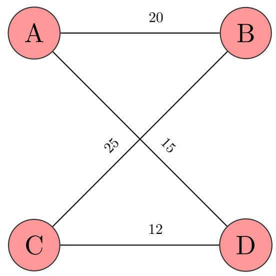
```
shortestLinkTSP (V,E)
R ← E
C ← Empty
while no of edges in C < |V| − 1 do                    ▷ Run |V| − 1 iterations
    remove the lightest edge vw from R;
    if vw does not form a cycle in C AND vw is not the third edge in C incident on v or w then
        add edge vw to C;
add edge connecting the end points to C;
Return C;
```




Shortest Link Algorithm

Nearest Neighbour or Shortest Link alogiritthm does not give the optimal solution but time complexity is polynomial bounded. The optimal solution is given below:




Optimal Solution

Cost = 12 + 15 + 25 + 12 = 72

## 8.2 Genetic Algorithm (GA)

Genetic algorithm (GA) is a class of heuristic algorithms used for optimisation and search problems based upon the mechanics of natural genetics. It combines the Darwinian's principle of survival-of-the-fittest with a random, yet structured information exchange among a population of artificial chromosomes. GAs mimick this process to repeatedly search for good solutions. The algorithm can be summarised in the following steps:

1. Encode candidate solutions as population of chromosomes.
2. Select good individuals based on some fitness function.
3. Reproduce new individuals through genetic operators of mutation, selection, crossover etc.
4. Repeat from Step 2 until a satisfactory result found
5. Decode the best chromosome found

For each individual chromosome in the population, its fitness function is used to evaluate the performance of the chromosomes and generate a probability of selection for reproduction.