DA Assign2- Routing Algorithm for Ocean Shipping and Urban Deliveries

Group 6_4:

-Luis Filipe Pinto Cunha

-Pedro Miguel Martins Romão

-Inês Martin Soares

Introduction

- In this programming project, the goal is to tackle the challenging problem of finding optimal routes for vehicles in various shipping and delivery scenarios, ranging from urban deliveries to ocean shipping. This problem is commonly referred to as the Traveling Salesperson Problem (TSP). The TSP is known to be an intractable problem, meaning that there are no known efficient algorithms to solve it optimally for large instances.

Functionalities

All requested features have been implemented!

- -Menu
- -Backtracking Algorithm
- -Triangular Approximation Heuristic(Christofides)
- -Two Optimal Heuristic
- -Nearest Neighbor Heuristic

```
|                              Main Menu                              |
+---------------------------------------------------------------------+
|                                                                     |
|  Option 1: Backtracking Heuristic                                   |
|  Option 2: Triangular Approximation Heuristic                       |
|  Option 3: Other Heuristics                                         |
|                                                                     |
|  Option 4: About                                                    |
|                                                                     |
|  Option 0: Exit                                                     |
+---------------------------------------------------------------------+
Choose an Option between 0 and 4:
```

```
+------------------------------------------------------------------
|                         Other Heuristics
|                            Main Menu
+------------------------------------------------------------------
|
|  Option 1: Two Optimal Heuristic
|  Option 2: Nearest Neighbour Heuristic
|
|  Option 0: Exit
+------------------------------------------------------------------
Choose an Option between 0 and 3:
```

```
+---------------------------------------------------------------------+
|                          Reading the Data                           |
|                        Two Optimal Heuristic                        |
+---------------------------------------------------------------------+
|                                                                     |
|  Option 1: Real-Word Graphs                                         |
|  Option 2: Toy Graphs                                               |
|                                                                     |
|  Option 0: Back                                                     |
+---------------------------------------------------------------------+
Choose an Option between 0 and 2:
```

Backtracking Algorithm

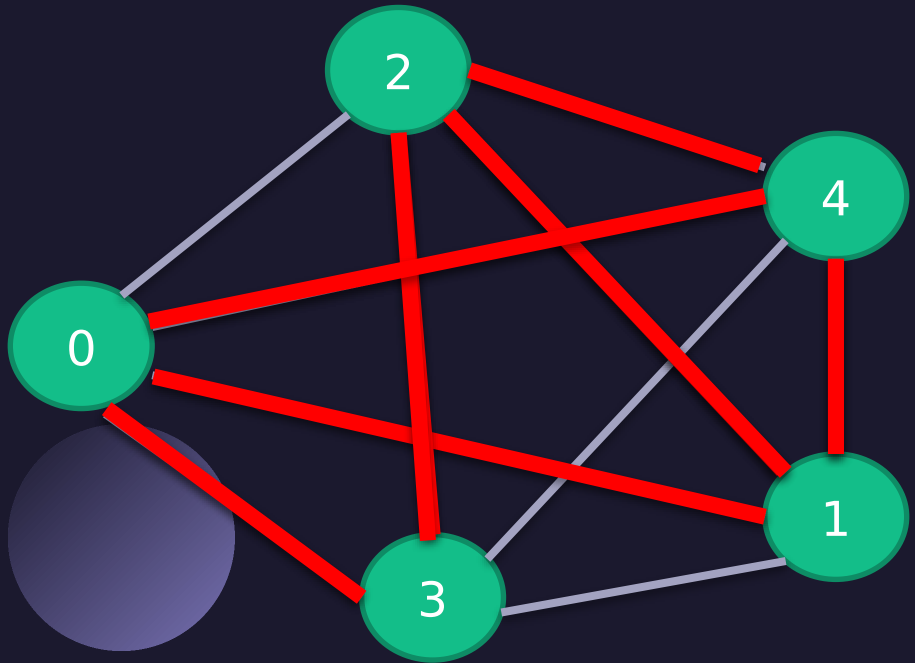
Backtracking is a general algorithmic technique used to systematically explore all possible solutions to a problem by incrementally building a solution candidate and backtracking when a dead end is reached.

```
Choose an Option between 0 and 6:
3
Path: 0 1 2 3 4 0
Path: 0 1 2 4 3 0
Path: 0 1 4 2 3 0
Path: 0 2 1 4 3 0
Path: 0 3 2 1 4 0
Distance for the PATH exactly above: 2600.00
Execution time: 0.01 seconds
```

File: tourism.csv

Time Complexity: O(n!)
n is the number of nodes

Space Complexity: O(n)
Requires storing the current path and the visited status of each node during the recursion



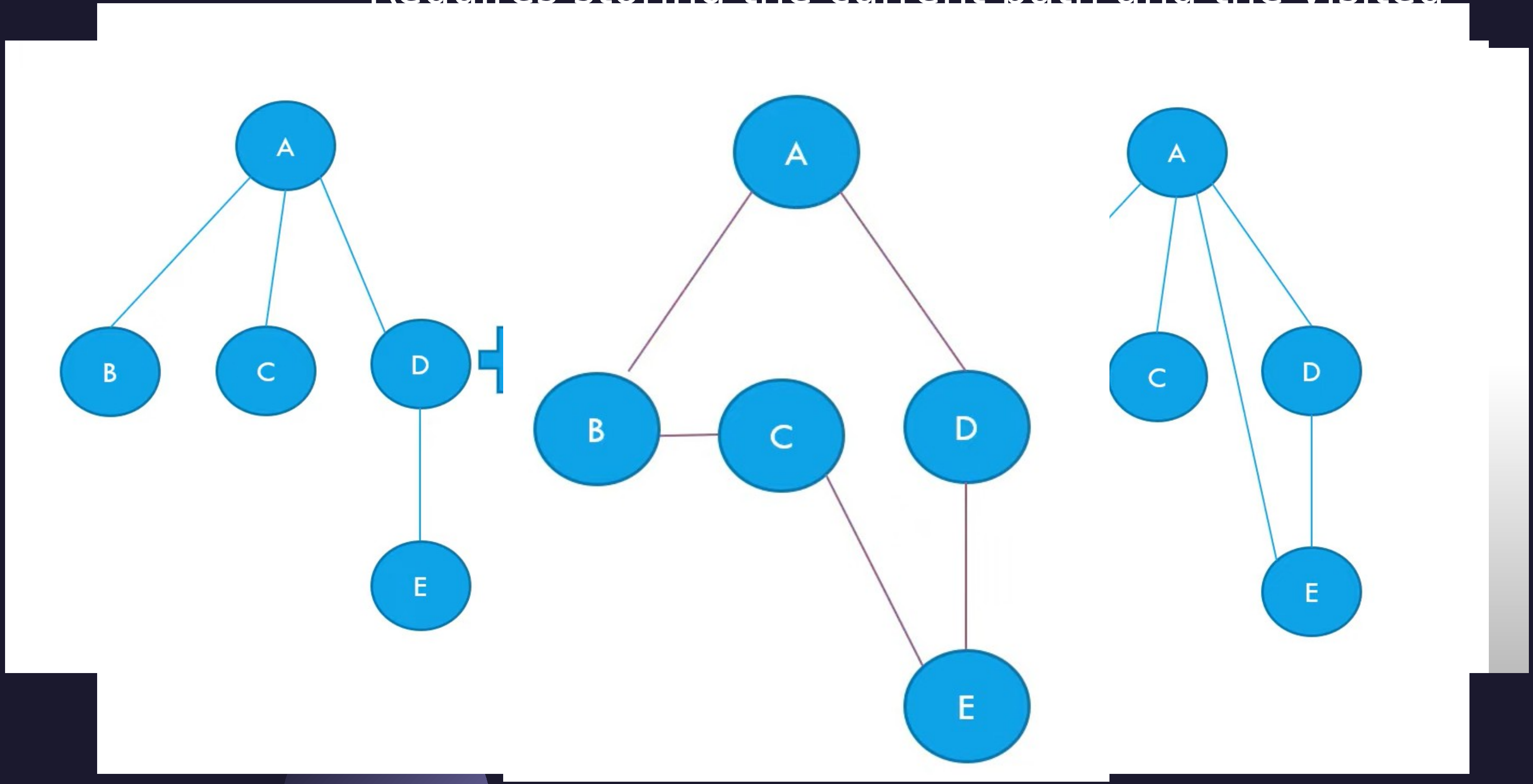
Critophides:

Critophides is an approach used to approximate the solution of the Traveling Salesman Problem (TSP) (3/2).

The process is such:

Create a minimum Spanning Tree
Make a Set of Vertices that have an odd degree
Find a Minimum-Cost, Perfect Matching
Combine Perfect Match to minimum Spanning Tree
Shortcut by removing all repeated Vertices

Time Complexity: $O(n^2)$
n is the number of nodes
Space Complexity: $O(n^2)$
Requires storing the current path and the visited



Two Optimal Heuristic

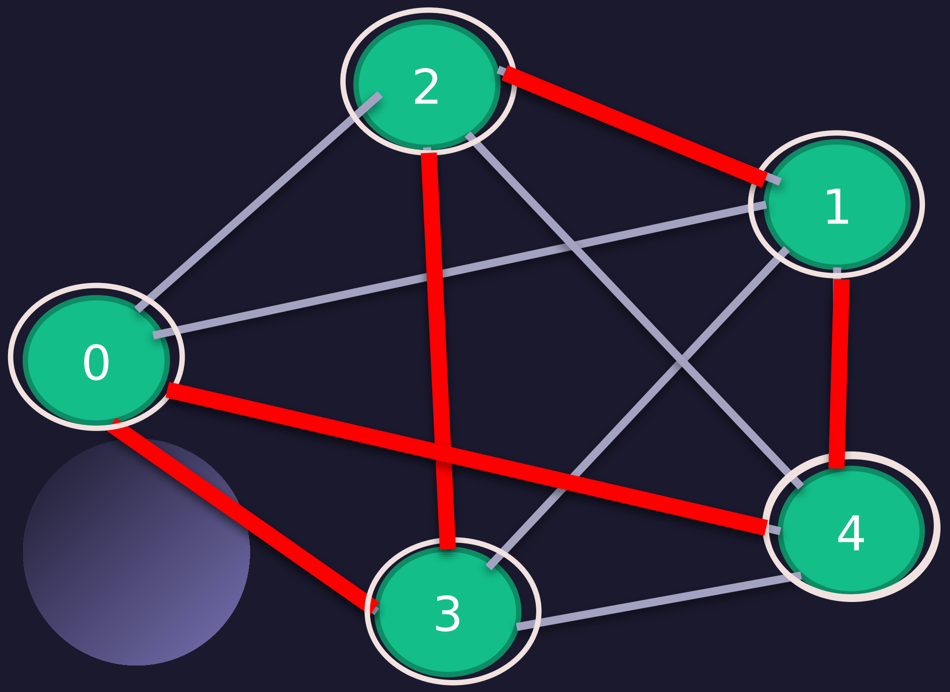
```
Two-Optimized Path: 0 3 2 1 4 0
Distance of the Path: 2600.00
Execution Time: 0.00 seconds
Choose an Option between 0 and 3:
```

File: tourism.csv

Time Complexity: O(n^2*k*(n-k))
n is the number of nodes
k is the number of iterations or swaps

Space Complexity: O(n)
Requires storing the current path and the visited status

The Two-Optimal Heuristic is an iterative improvement algorithm used to refine an initial solution to the Traveling Salesman Problem (TSP). It operates by iteratively removing two edges from the current tour and reconnecting them in a different way, aiming to reduce the total distance traveled.



Nearest neighbor Heuristic

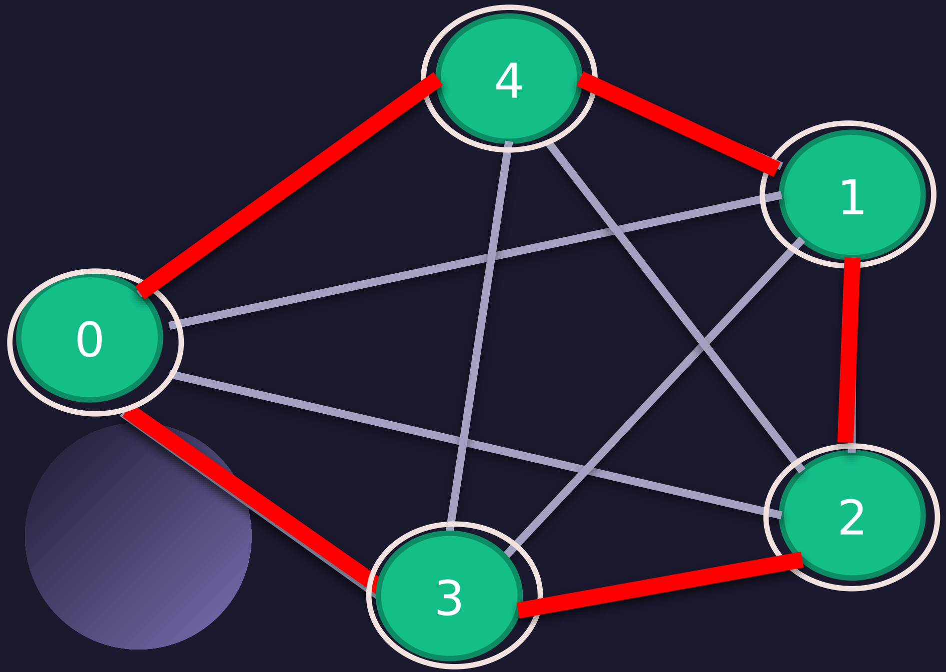
The Nearest Neighbor algorithm is a simple and efficient heuristic approach used to find an approximate solution for the Traveling Salesman Problem (TSP). It starts with an arbitrary node and repeatedly selects the nearest unvisited node as the next node to visit. This process continues until all nodes are visited, forming a tour.

```
Nearest Neighbor Path: 0 3 2 1 4 0
Distance of the Path: 1850.00
Execution Time: 0.00 seconds
Choose an Option between 0 and 3:
```

File:
tourism.csv

Time Complexity: O(n^2)
n is the number of nodes

Space Complexity: O(n)
Requires storing the current path and the visited status



Conclusions

The Backtracking algorithm guarantees an optimal solution by exhaustively exploring all possible permutations of nodes, but its time complexity grows more than exponentially with the number of nodes, making it impractical for large problem instances. However, it serves as a benchmark for evaluating the performance of other heuristics.

The Two-Optimal algorithm, based on local search, iteratively improves an initial solution by swapping pairs of edges to reduce the total tour distance. It provides good solutions for moderate-sized instances, making it more efficient than Backtracking.

Conclusions

The Nearest Neighbor heuristic is a simple and fast algorithm that constructs a tour by iteratively selecting the nearest unvisited city. While it can produce suboptimal solutions, it has a time complexity of O(n^2) and performs well on certain TSP instances, particularly for smaller problem sizes.

Christofides provides a 1,5 approximation of the optimal solution and has a time complexity of O(n^3), making it more efficient and reliable for larger problem instances.