Design of Algorithms Programming Project II - Routing Algorithm for Ocean Shipping and Urban Deliveries

MADE BY GROUP G03_2 :

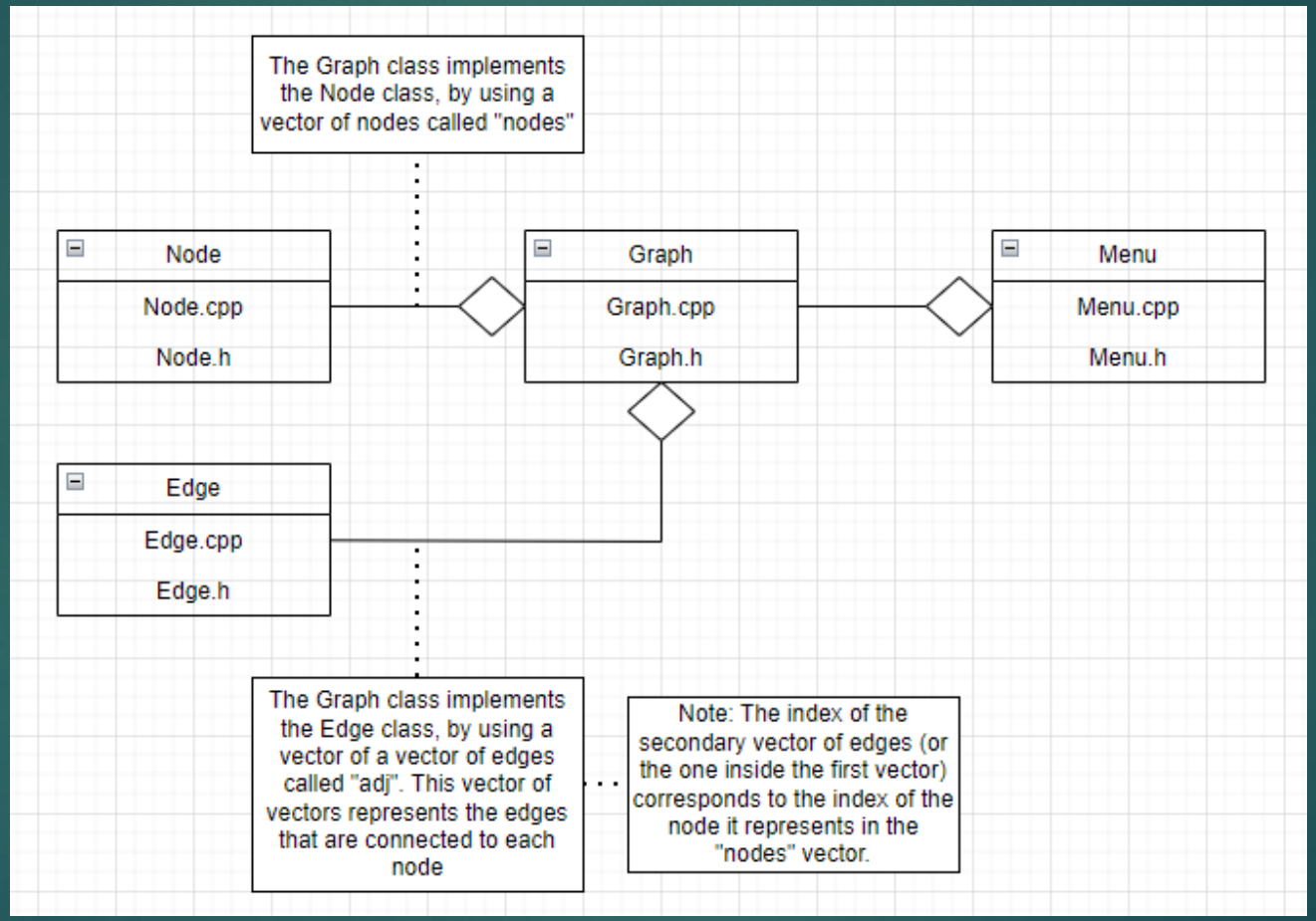
- MANSUR MUSTAFIN - 202102355
- JOSÉ NUNO BARBOSA QUINTAS – 202108712
- FRANCISCO GONÇALVES DE SOUSA - 202108838

MAY 4 2023

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Table of Content

Class Diagram



Dataset Reading

- When an object of the class Graph is created, two string containing the names of the files from which the data will be loaded are passed to the Graph Constructor.
- The Graph has a vector of Nodes called "nodes", where the function "input_vertex" inserts the objects of the class Node it creates when the Graph constructor is called.
- The Graph also has a vector of vectors of Edges called "adj", where the function "input_edge" inserts the objects of the class Edge it creates when the Graph constructor is called.
- Note: The index of the "inside" vector in "adj" corresponds to the index of the node in "nodes" whose edges are in the vector.

Graph description

This is how the Graph has been organized:

- Objects of the class Node are the vertices of the Graph and stored in a vector named “nodes”
- Objects of the class Edge are the edges of the Graph and stored in a vector of vectors named “adj”

List of features and algorithms

- Backtracking Algorithm:
  - Time Complexity: O(N!), where N is the number of Nodes in the Graph being analysed
- Ant Colony Optimization Algorithm:
  - Time Complexity: O(max_iter * num_ants * n^2), where max_iter is the maximum number of iterations possible, num_ants is the number of ants given and n is the size of the distance matrix. All of these variables are passed to the ACO() function.
- Simulated Annealing Algorithm:
  - Time Complexity: O(N * A), where N is the size of the permutation used and A is the number of adjacent nodes of each node of the permutation

List of features and algorithms

- Cristofides Algorithm:
  - Time Complexity: O(E * log(V) + N^3), where E is the number of edges in the Graph, V is the number of vertices (or nodes) and N is the size of the path
- Triangular Approximation and…
- Triangular Approximation using a Distance Matrix:
  - Used algorithms/functions:
    - Prim's Algorithm: Time complexity: O(E * log(V)), where E is the number of Edges and V is the number of vertices (or nodes) of the Graph being analysed
    - Preorder Walk: Time complexity: O(E * log(E) + N * E), where E is the number of edges of the Minimum Spanning Tree (MST) and N is the number of adjacent nodes of each node.

Comparison of Algorithms and their Minimum Cost

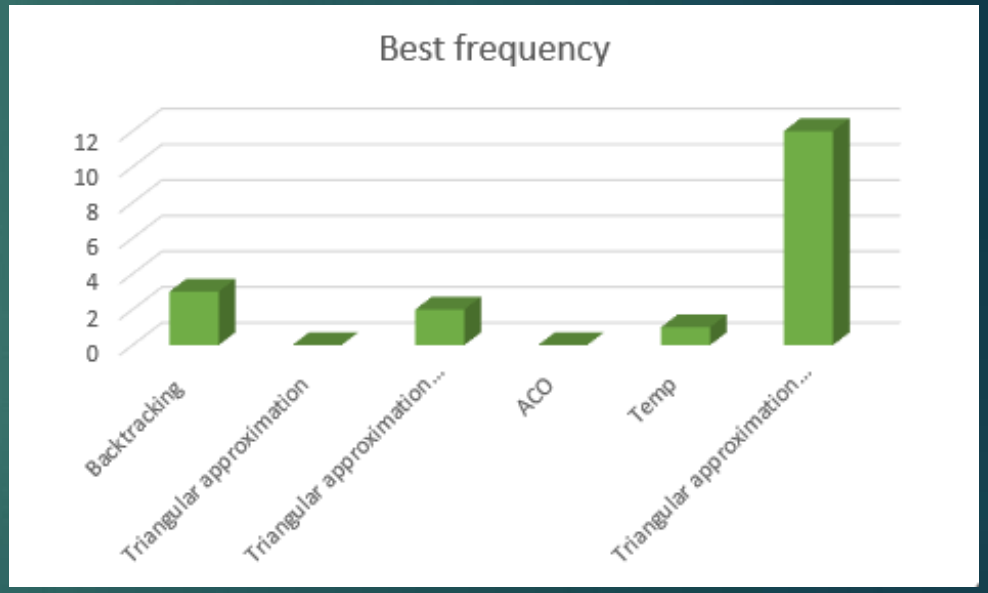
MINIMUM COST	Backtracking	Triangular approximation	Triangular approximation (matrix)	ACO	Temp	Christofides Algorithm	
edges_25	N/A	349 573	349 573	295 586	287 329	307668	Temp
edges_50	N/A	554 134	554 134	486 977	523 934	482329	Christofides Algorithm
edges_75	N/A	627 035	627 035	616 767	617 110	557708	Christofides Algorithm
edges_100	N/A	681 458	681 458	623 216	640 287	560190	Christofides Algorithm
edges_200	N/A	909 414	909 414	846 597	1 095 819	769966	Christofides Algorithm
edges_300	N/A	1 196 893	1 196 893	1 144 784	1 625 562	991187	Christofides Algorithm
edges_400	N/A	1 344 211	1 344 211	1 389 038	2 202 735	1142352	Christofides Algorithm
edges_500	N/A	1 496 185	1 496 185	1 388 547	2 753 043	1190010	Christofides Algorithm
edges_600	N/A	1 618 207	1 618 207	1 526 154	3 526 691	1354084	Christofides Algorithm
edges_700	N/A	1 757 669	1 757 669	1 863 896	4 248 122	1490713	Christofides Algorithm
edges_800	N/A	1 864 895	1 864 895	2 009 223	4 676 631	1567355	Christofides Algorithm
edges_900	N/A	2 052 974	2 052 974	2 110 012	5 501 089	1681026	Christofides Algorithm
shipping	87	N/A	N/A	91	N/A	N/A	Backtracking
tourism	2 600	2 600	2 600	2 600	2 600	2600	Backtracking
stadiums	341	398	398	360	341	370	Backtracking
edges_1	N/A	1 125 735	1 125 735	1 132 233	3 521 262	940996	Christofides Algorithm
edges_2	N/A	3 368 300	2 286 682	6 396 266	29 781 064	4273491	Triangular approximation (matrix)
edges_3	N/A	5 837 626	3 230 269	11 615 265	71 733 674	8702097	Triangular approximation (matrix)

Comparison of Algorithms and their Execution Times

EXECUTION TIME	Backtracking	Triangular approximation	Triangular approximation (matrix)	ACO	Temp	Christofides Algorithm
edges_25	N/A	<0.1	<0.1	<0.1	15,6	<0.1
edges_50	N/A	<0.1	<0.1	0,14	16	<0.1
edges_75	N/A	<0.1	<0.1	0,3	16,3	<0.1
edges_100	N/A	<0.1	<0.1	0,5	16,7	<0.1
edges_200	N/A	<0.1	<0.1	2,2	18,2	<0.1
edges_300	N/A	<0.1	<0.1	4,9	19,7	<0.1
edges_400	N/A	<0.1	<0.1	8,6	21,1	<0.1
edges_500	N/A	<0.1	<0.1	13,6	22,6	<0.1
edges_600	N/A	<0.1	<0.1	19,8	24,2	<0.1
edges_700	N/A	<0.1	<0.1	26,8	25,8	0,1
edges_800	N/A	<0.1	<0.1	35,1	27,4	0,1
edges_900	N/A	<0.1	<0.1	44,5	28,6	0,2
shipping	<0.1	N/A	N/A	<0.1	N/A	N/A
tourism	<0.1	<0.1	<0.1	<0.1	14,4	<0.1
stadiums	0,7	<0.1	<0.1	<0.1	15,3	<0.1
edges_1	N/A	<0.1	<0.1	55,4	26,6	0,2
edges_2	N/A	0,6	1,9	23min	45,2	11,7
edges_3	N/A	1,7	8	1h32min	57,4	76,3

Comparison of Algorithms (Conclusion)

Algorithm	Number of times this algorithm is the best option
Backtracking	3
Triangular Approximation	0
Triangular Approximation with Distance Matrix	2
ACO	0
Temp	1
Triangular Approximation using Cristofides	12



User interface


```
|============================================================|
|                       TSP Algorithms                       |
|============================================================|
| Backtracking Algorithm                                [21] |
| Triangular Approximation                              [22] |
| Triangular Approximation using matrix                 [23] |
| ACO                                                   [24] |
| Temp                                                  [25] |
| Triangular approximation using Christofides Algorithm [26] |
|============================================================|
|                     Other operations                       |
|============================================================|
| Run all agorithms                                     [11] |
| Change files                                          [12] |
| Exit                                                  [0]  |
|============================================================|

Please choose an option:22

         Do you want to print the path ?
|================================================|
|       Yes [1]                  No  [2]         |
|================================================|
2
```

```
mansur@mansur-g14:~/Рабочий стол/DA-tsp/cmake-build-debug$ cat output.txt
File: edges_25.csv ____________________________________________________

--** Backtracking **--
This algorithm is inefficient for this size of graphs
I have one week left to complete the project, algorithm takes more.

--** Triangular approximation **--
Minimum cost: 349573.20
Execution time: 0.00 seconds

--** Triangular approximation using matrix **--
Minimum cost: 349573.20
Execution time: 0.00 seconds

--** ACO **--
Minimum cost: 295586.80
Execution time: 0.04 seconds

--** Test2 **--
Minimum cost: 311681.10
Execution time: 15.61 seconds

--** Triangular approximation using Christofides Algorithm **--
Minimum cost: 307668.00
Execution time: 0.00 seconds

Path: 0->22->12->23->9->15->5->4->8->24->17->7->11->20->6->13->14->1->2->16->3->
19->10->18->21->0
```

Main difficulties

Our main difficulties were the following:

- Analyzing the input arguments in ACO and Annealing algorithms.
- Debug of program.

Effort of each member

- Mansur Mustafin – Graph Algorithms, Main Functions
- Francisco Gonçalves de Sousa – Graph Construction, User Interface, Menu
- José Nuno Barbosa Quintas – Graph Structure and Population, Documentation, Presentation.