Faculdade de Engenharia de Universidade do Porto

CAL - Conceção e Análise de Algoritmos

Segunda, 25 de maio de 2020

Tema 6: CityTransfers

Report - Parte 2

Turma 5, Grupo 7:

António Melo Cabral Lima Braga	up201708995@fe.up.pt
João Nuno Diegues Vasconcelos	up201504397@fe.up.pt
Tiago Miguel Barbosa Marques	up201704733@fe.up.pt

Índice

Principais casos implementados

No projeto estão implementados os casos de calcular a melhor rota para o transporte de um cliente, sendo possível escolher entre vários algoritmos, os quais iremos discutir mais à frente, como se pode ver nesta imagem do menu correspondente ao primeiro de implementação:

```
|| ================= First Problem ================
|| (1) Dijkstra / Calculate and show route for one client
|| (2) A Star / Calculate and show route for one client
|| (3) Floyd-Warshall / Calculate and show route between any two nodes
||     Be aware it will take too long to calculate
|| (4) Return to main menu
```

Foi também implementado um algoritmo que permite calcular uma rota que transporta mais que um cliente, tendo por isso vários destinos, correspondente à segunda iteração de implementação:

```
|| ================== Main Menu ==================
|| (1) Nearest neighbour / using Dijkstra
|| (2) Return to main menu
```

Foi também feita uma analise dos vértices não alcançáveis a partir do vértice da estação de chegada dos clientes:

```
|| (6) Connectivity from station node / Depth First Search (DFS)
```

É também possível adicionar novos veículos da empresa e novos clientes, para se poder testar o programa com diferentes destinos:

```
|| (4) Add/Eliminate Client
|| (5) Add/Eliminate Car
```

Estruturas de dados utilizadas

Graph / Vertex / Edge

A classe graph presente no projeto é uma adaptação do que nos foi fornecido nas aulas, sendo também usadas as classes edge e vertex que foram ligeiramente alteradas para este projeto, tendo sido adicionado o seu ID no grafo.

Person

A classe person foi criada para se poder guardar os clientes, para ser ter acesso ao ID do vértices de destino e à sua hora de chegada à estação.

CityTransfers

Foi criada classe de forma a agregar todos os atríbutos da empresa num só sitio, de forma a estar de fácil acesso em todas as partes do programa.

Algoritmos utilizados e breve analise

Algoritmo de pesquisa em profundidade (DFS)

O algoritmo de pesquisa em profundidade faz uso da recursividade para descobrir que vértices podem ser alcançados a partir de um vértice inicial. Apresentando uma complexidade temporal de O(|V| + | E|).

Algoritmo de Floyd-Warshall

Este algoritmo é aplicado a grafos pesados dirigidos, calculando a distância mínima entre todos os pares de vértices do grafo. O algoritmo usa uma matriz de adjacências que contem as distâncias mínimas entre os pares de vértices, no entanto tempo uma complexidade temporal elevado, de O ($|V|^3$), o que torna muito evidente em grafos pesados.

Algoritmo de Dijkstra

O algoritmo de Dijkstra encontra o cainha mais curto desde um vértice até todos os outros em grafos dirigidos. Sendo utilizada uma fila de prioridade para se obter o vértice de menor distância obtém-se uma complexidade temporal de
O((|V| + |E|) * log(|V|);

Algoritmo A Star

O algoritmo calcula a distancia e o caminho entre dois pontos.
Este algoritmo é bastante semelhante ao Dijsktra, sendo que calcula a menor distância entre dois vértices através de um algoritmo ganancioso que avalia em primeiro lugar os vértices que considera mais promissores. A ordenação da fila de prioridade é ordenada de forma diferente, usando uma função heurística.
No entanto a utilizar a condição de paragem, acabar quando encontrar o vértice pretendido, pode não garantir a solução ótima.

Algoritmo do vizinho mais próximo

Este algoritmo é um algoritmo ganancioso que escolhe como vértice seguinte aquele que se encontrar mais próximo.

Para o nosso caso o percurso começa e acaba no mesmo local.

Analise da complexidade dos algoritmos implementados

Algoritmo de pesquisa em profundidade (DFS)

Ao analisar os dados obtidos nos mapas utilizados podemos ver que existe uma relação relativamente linear entre o tempo e o número de vértices.

Algoritmo de Floyd-Warshall

Analisando os dados obtidos com a implementação deste algoritmo percebe-se que este é bastante moroso para os mapas que utilizamos, para o mapa de Penafiel, 3964 vértices, demora cerca de 40 minutos o que não é de todo ideal, tornando a sua utilização difícil.

Algoritmo do vizinho mais próximo

Este algoritmo foi implementado usando como processamento o algoritmo de Dijkstra, tendo sido obtidos tempos de execução relativamente linear entre o tempo e o número de vértices. Sendo por isso de fácil uso projeto.

Algoritmo A Star e Dijkstra

O algoritmo de A Star apesar de rápido não garante a solução ideal, como foi comprovado pelos resultados obtidos, uma vez que comparando as distâncias calculadas, as do A Star são ligeiramente maiores que as de Dijkstra.

No entanto é ligeiramente mais rápido que Dijkstra, devido à heurística.

Os dois obtiveram tempos de execução bastante rápidos para todos os mapas.

Conectividade dos grafos utilizados

Em relação à conectividade do grafo, foi aplicado o algoritmo DFS(Depth First Search) abordado nas aulas teóricas, de modo a analisar os mapas com componentes não conexas, podendo ser realizada uma DFS a partir do vértice a estação de partida e descobrir a que vértices não é possível chegar.

Verificou-se que nos mapas completos fornecidos não é possível aceder:

Porto : 17837 vértices

Espinho: 3692 vértices

Penafiel: 3676 vértices

Conclusão

Na realização deste trabalho foi bastante interessante ver em ação os algoritmos estudados nas aulas em mapas obtidos com base em cidades reais.

No entanto não foi possível cumprir todos os objetivos do trabalho, ficando por implementar os casos de garantir o menor tempo de espera possível por parte dos clientes, não tendo sido implementada a terceira fase.

No decorrer deste trabalho foram encontradas várias dificuldades, sendo a principal não ter existido empenho de forma igual de todos os elementos do grupo, não tendo assim sido possível concluir todas a etapas planeadas para este projeto.

Esforço dedicado por cada elemento

António Melo Cabral Lima Braga:	10%
João Nuno Diegues Vasconcelos:	80%
Tiago Miguel Barbosa Marques:	10%