



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEIC - Conceção e Análise de Algoritmos

Planeamento de itinerários multimodais

Relatório Intercalar

realizado por:

2MIEIC01 - Grupo B - Tema 4

João Nogueira, up201303882;

António Pedro Fraga, up201303095;

Filipa Barroso, up201307852.

Índice

Introdução.....	3
Explicação do Problema	4
Descrição do Problema.....	5
<i>Input</i>	5
<i>Introdução de dados</i>	5
<i>Output</i>	5
<i>Objetivo</i>	5
<i>Restrição</i>	5
Formalização do Problema	6
<i>Input</i>	6
<i>Output</i>	6
<i>Objetivo</i>	6
<i>Restrição</i>	6
Perspectivas de solução	7
<i>Algoritmos</i>	7
Métricas de avaliação	8
<i>Avaliação empírica do seu desempenho</i>	8
<i>Complexidade temporal</i>	8
<i>Complexidade espacial</i>	8
Conclusão.....	9

Introdução

Foi no âmbito da Unidade Curricular de Concepção e Análise de Algoritmos do 2º ano do MIEIC que nos foi apresentado um problema que nos foi pedido para resolver da forma mais eficiente possível.

Neste relatório está explicado o problema em si, por forma a que se entenda exatamente qual o nosso objetivo e como o pretendemos atingir.

Depois de explicitar qual o problema que procuraremos resolver, explicamos neste relatório os métodos que usaremos para a resolução do problema em questão.

Explicação do Problema

É uma preocupação cada vez maior, o facto de nas grandes cidades o transporte individual ser demasiado utilizado, em detrimento dos. Um problema presente na maior parte das cidades com vários serviços de transportes públicos é o facto de não haver plataformas que facilitem ao utilizador a escolha do melhor itinerário tendo em conta os vários tipos de transporte. Estas plataformas teriam em conta os transportes com paragens em comum e os horários dos vários transportes e linhas diferentes.

Nesta plataforma, caberia ao utilizador apenas introduzir a origem e o destino do percurso. A plataforma calcularia os vários itinerários possíveis, e mostraria ao utilizador o itinerário multimodal mais curto/rápido.

Para avaliar o itinerário podem ser utilizados vários critérios: custo da viagem, tempo de viagem, número de transbordos ou distância.

Descrição do Problema

De uma forma geral, pretendemos resolver o problema de itinerários intermodais através de uma aplicação que, apenas recebendo a estação de origem e a de destino, consiga devolver o itinerário mais rápido, curto ou barato dependendo das preferências do utilizador.

Input

Construção de um grafo, $G = (V, E)$, de estações e linhas no qual:

- V – vértices – representam todas as paragens/estações dos meios de transportes disponíveis;
- E – arestas – representam todas as distâncias, tempos de viagens e custos;
- Nó de início de viagem e nó de destino.

Introdução de dados

Um ficheiro das estações com respetivas linhas/tipos de transportes que nelas passam, tempo médio entre estações tal como custo e duração.

Output

Todas as estações em que o utilizador passou por forma a otimizar o seu trajeto e o valor final da distância, custo ou duração da viagem dependendo da preferência do utilizador.

Objetivo

Facilitar aos utilizadores a escolha dos melhores trajetos consoante os critérios por estes preferidos.

Restrição

No mesmo caminho não se passa pela mesma estação mais do que uma vez.

Formalização do Problema

Formalizamos agora o problema, de acordo com aquela que achamos ser a melhor forma para resolver aquilo a que nos propusemos.

Input

$G < V, E >$,

V : estações/paragens

E : ligações entre pontos

(tempo de viagem);

(custo da viagem);

(distância da viagem).

I : ponto inicial

F : ponto final

Output

$Caminho = \{V_i\}, I = 1 \dots n$

$Valor$

Objetivo

$Min(valor)$:

$$valor = \sum_{i=1}^n (E_{ij}),$$

$ij \in Caminho$

Restrição

$\forall_i \in Caminho = \{V_i\}$

$\sim \exists_j \in Caminho : V_i = V_j \wedge i \neq j$

Perspectivas de solução

Algoritmos

Pensamos que o melhor algoritmo a ser usado neste trabalho é o **algoritmo de Dijkstra**, que encontra o caminho mais curto num grafo dirigido ou não dirigido em **tempo computacional $O([arestas + vértices]\log n)$** , este algoritmo não serve para grafos com pesos negativos, mas visto que este não é o caso, o algoritmo é perfeitamente aplicável.

Este algoritmo é um algoritmo ganancioso, tomando decisões que parecem óptimas no momento, determinando assim o conjunto de melhores caminhos intermediários. O valor de cada aresta está associado a uma classe com os campos “distância”, “preço” e “tempo”, portanto são estes valores que são utilizados para o cálculo de melhor caminho, dependendo das preferências do utilizador. Podemos também adaptar este algoritmo ao **algoritmo de A***, este algoritmo não faz apenas a pesquisa de melhores caminhos, mas também considera caminhos mais direccionados.

Métricas de avaliação

Avaliação empírica do seu desempenho

Para avaliar e testar a complexidade temporal dos algoritmos vamos medir os diferentes tempos de execução do programa utilizando diferentes dados de entrada.

Complexidade temporal

Como já foi referido anteriormente, a complexidade temporal do algoritmo que estamos a pensar utilizar é de **$O([\text{arestas} + \text{vértices}] \log n)$** .

Complexidade espacial

Na resolução do problema será utilizada programação dinâmica, que tem como complexidade temporal $O(|V| * T_{\text{total}})$.

Conclusão

Após um grande debate dentro do grupo acerca daquilo que o problema exigia por forma a atingir um resultado ótimo, tendo em conta todos os pormenores e possibilidades de resolução, acreditamos que os algoritmos especificados nas Perspetivas de Solução são os mais adequados para uma boa resolução do problema.

A falta de experiência em todos os algoritmos relacionados com a estrutura de dados que iremos implementar neste trabalho foi provavelmente a nossa maior dificuldade. A parte que se segue será um desafio. No entanto, estamos confiantes que iremos superar esse desafio e contamos resolver o problema da forma mais adequada possível.