**Faculdade de Engenharia da Universidade do Porto**



**Relatório Projeto CAL**

Parte 1

*RideSharing: partilha de viagens*

11 abril 2018

Grupo 10 – Turma 2

Gonçalo Santos 201603265 up201603265@fe.up.pt

João Vieira 201603190 up201603190@fe.up.pt

Susana Lima 201603634 up201603634@fe.up.pt

Índice

[Introdução 3](#_Toc510868427)

[Explicação do problema 4](#_Toc510868428)

[Descrição do problema 5](#_Toc510868429)

[1ªiteração: recolha de pessoas cujo percurso mais rápido se insira no percurso do condutor 5](#_Toc510868430)

[2ªiteração: recolha de pessoas cuja origem e destino estejam contidos no percurso do condutor 5](#_Toc510868431)

[Formalização do problema 6](#_Toc510868432)

[Dados de entrada 6](#_Toc510868433)

[Dados de saída 7](#_Toc510868434)

[Restrições 7](#_Toc510868435)

[Sobre os dados de entrada 7](#_Toc510868436)

[Sobre os dados de saída 7](#_Toc510868437)

[Funções objetivo 7](#_Toc510868438)

## Introdução

No âmbito da unidade curricular de Conceção e Análise de Algoritmos do 2º ano do MIEIC foi proposta a resolução de um problema da forma mais eficiente possível usando os conhecimentos adquiridos nas aulas.

Neste relatório são descritos o problema e a melhor estratégia encontrada para a sua resolução.

## Explicação do problema

O tema do projeto incide sobre *RideSharing*, partilha de viagens cujo objetivo seria a otimização do recurso automóvel, evitando que este se desloque com apenas um ocupante.

A ideia básica por detrás deste conceito é a partilha de automóveis por diferentes pessoas cuja origem e destino de viagem coincida parcial ou totalmente com as de um condutor.

O programa desenvolvido permite o cálculo de percursos mais curtos que permitam maximizar o número de ocupantes de um veículo entre uma origem e um destino, tendo em consideração as restrições de tempo impostas tanto pelo condutor como pelos possíveis passageiros, bem como as horas de chegada aos pontos de encontro.

Foi considerada também a familiaridade entre passageiros, ou seja, pessoas que viagem em conjunto.

### Descrição do problema

O problema em questão foi decomposto em duas iterações:

## 1ªiteração: recolha de pessoas cujo percurso mais rápido se insira no percurso do condutor

Numa primeira abordagem do problema foram apenas considerados como possíveis passageiros as pessoas cujo caminho mais curto coincidisse parcial ou totalmente com o percurso do condutor, tendo sempre em consideração as restrições temporais impostas e as horas de chegada dos passageiros e dos condutores aos pontos de encontro. A capacidade do veículo também foi uma restrição imposta.

Convém salientar que a familiaridade entre pessoas foi considerada sendo que grupos de pessoas que viajassem em conjunto não poderiam ser separados, ou seja, um grupo só seria transportado caso no veículo existissem lugares disponíveis para acomodar todos os elementos do mesmo.

No entanto, pessoas cuja origem e destino estivessem contidas no percurso do condutor não seriam transportadas, embora todas as outras restrições fossem cumpridas, se o percurso mais rápido da mesma não estivesse contido (em parte ou na totalidade) no itinerário do motorista.

## 2ªiteração: recolha de pessoas cuja origem e destino estejam contidos no percurso do condutor

Numa segunda fase, de modo a evitar a situação descrita na iteração anterior, foi acrescentado ao programa um pós-processamento que verificasse a existência de pessoas que cumprissem todas as restrições de tempo nos pontos de recolha contidos no percurso do condutor, tendo em consideração a lotação do carro nos sucessivos pontos, cuja origem e destino pertencessem ao caminho em questão. Se tal acontecesse essas pessoas seriam também transportadas.

Como consequência deste pós-processamento, pessoas cuja origem e destino estivessem contidas no percurso do condutor seriam transportadas se as condições impostas fossem cumpridas.

### Formalização do problema

O programa implementado permite, de uma forma geral, resolver o problema de *ride sharing* recebendo um ponto de partida, com a respetiva hora, e um ponto de chegada, um limite de tempo máximo para o percurso a efetuar, atendendo à capacidade do veículo, caso o utilizador se trate de um condutor ou o número de pessoas a viajar em grupo, caso o utilizador seja um passageiro.

## Dados de entrada

Os is são porque são os dados iniciais duhh

Pi – Sequência de passageiros que aguardam transporte, sendo Pi(i) o seu i-ésimo elemento. Cada um é caracterizado por:

* Número de pessoas com quem viaja.
* Ponto de partida.
* Ponto de chegada.
* Hora de partida.
* Limite de tempo.

Ci – Sequência de condutores disponíveis, sendo Ci(i) o seu i-ésimo elemento. Cada um é caracterizado por:

* Capacidade do automóvel que conduz.
* Ponto de partida.
* Ponto de chegada.
* Hora de partida.
* Limite de tempo.

Gi(Vi,Ei) – grafo dirigido pesado composto por:

* V – vértices – representam todos os pontos de encontro
* E– arestas – representam vias de ligação entre os pontos (estradas, ruas) com:
  + wt – corresponde ao tempo entre os dois vértices que a delimitam
  + wp – corresponde à sequência de pessoas que aguardam transporte entre os dois vértices que a delimitam
  + w – peso da aresta (calculado através da reta 3d???)

As origens e destinos dos passageiros e condutores estão contidos em V.

## Dados de saída

Gf(Vf,Ef) – grafo dirigido em que Vf e Ef têm os mesmos atributos que Vi e Ei. No entanto, é de referir que Ef diferencia- se de Ei caso tenham sido transportados passageiros (remoção dos passageiros que aguardavam transporte).

Cf – sequência ordenada de todos os condutores utilizados sendo Cf(i) o seu i-ésimo elemento. Cada um com:

* I – sequência de vértices a visitar entre o vértice de partida e o de chegada, correspondente ao itinerário do condutor.
* Número de pessoas transportadas ao longo do percurso??
* Distancia percorrida??
* Tempo de viagem??
* Hora de chegada??

Pf – sequência de passageiros transportados sendo Pf(i) o seu i-ésimo elemento. Cada um com:

* I – sequência de vértices visitados
* Pc – posição atual
* Hora atual

## Restrições

## Sobre os dados de entrada

## Sobre os dados de saída

## Funções objetivo

(não sei se é isto)

A solução ótima encontrada passa por minimizar o tempo de viagem e maximizar o número de passageiros transportados durante a mesma.

f - |wt|

g - |wp|

No programa implementado foi privilegiada a maximização do número de passageiros sobre a minimização do tempo. (explicar função reta 3d????)