Meat Wagons - Transporte de Prisioneiros

Turma 2 Grupo 3

up201806250@fe.up.pt	Diogo Samuel Gonçalves Fernandes
up201806490@fe.up.pt	Hugo Miguel Monteiro Guimarães
up201806554@fe.up.pt	Telmo Alexandre Espirito Santo Baptista

12 de Abril de 2020

Projeto CAL - 2019/20 - MIEIC

Professor das Aulas Práticas: Rosaldo José Fernandes Rossetti

U. PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Índice

1 Descrição do Problema **3**

2 Formalização do Problema **4**

2.1 Dados de Entrada 4
2.2 Dados de Saída 5
2.3 Restrições 5

1 Descrição do Problema

2 Formalização do Problema

2.1 Dados de Entrada

C_i - sequência de veículos, sendo $C_i(i)$ o seu i-ésimo elemento. Cada veículo é caraterizado por:

- *capacity* - número de prisioneiros que pode transportar
- *type* - tipo de veículo
- *loc* - localização atual do veículo

R_i - sequência de pedidos de transporte de prisioneiros, sendo $R_i(i)$ o seu i-ésimo elemento. Cada pedido é caraterizado por:

- *pickup* - local de recolha dos prisioneiros
- *dest* - local de destino dos prisioneiros
- $numPris$ - número de prisioneiros a serem transportados
- *type* - tipo de prisioneiros

$G_i = (V_i, E_i)$ - grafo dirigido pesado, composto por:

- V - vértices, representando pontos da rede viária, com:
  - ID - Identificador único do vértice
  - D - Densidade populacional no vértice
  - $Adj \subseteq E$ - arestas que saiem do vértice
- E - arestas, representando conexão entre dois pontos da rede viária, com:
  - ID - Identificador único da aresta
  - W_d - peso da aresta em relação à distância (representa a distância entre os dois vértices)
  - W_t - peso da aresta em relação ao tempo (representa o tempo médio que demora a percorrer a distância entre os dois vértices, considerando o tráfego normal naquela conexão da rede viária)
  - *open* - se a conexão entre os vértices está aberta, isto é, se a rua estiver cortada por alguma razão então não é possível utilizar esta conexão

S - vértice da central

2.2 Dados de Saída

$G_f = (V_f, E_f)$ - grafo dirigido pesado, tendo V_f e E_f os mesmos atributos que V_i e E_i, excluindo atributos específicos do algoritmo utilizado

C_f - sequência de veículos com os serviços a realizar, sendo $C_f(i)$ o seu i-ésimo elemento. Cada veículo é caraterizado por:

- S - sequência de serviços a realizar, sendo $S(i)$ o seu i-ésimo elemento. Cada serviço é caraterizado por:
  - $emptySeats$ - número de lugares vazios
  - R_f - sequência de pedidos atendidos, sendo $R_f(i)$ o seu i-ésimo elemento. Cada pedido atendido é caraterizado por:
    - $pickupHour$ - hora de chegada ao local de recolha
    - $destHour$ - hora de chegada ao local de destino
  - $P = e \in E_i$ - sequência de arestas a percorrer, sendo $P(i)$ o seu i-ésimo elemento
  - $dist$ - distância percorrida no serviço
  - $startHour$ - hora esperada de ínicio do serviço
  - $endHour$ - hora esperada de termino do serviço

2.3 Restrições

Sobre os dados de entrada

- $\forall i \in [0, |C_i|[$: $capacity(C_i(i)) > 0$, uma vez que não faz sentido os veículos não poderem transportar prisioneiros
- $\forall i \in [0, |C_i|[$: $loc(C_i(i)$ deve pertencer ao mesmo componente fortemente conexo do grafo G_i que o vértice S, dado que o veículo tem de ser capaz de voltar à central. $loc(C_i(i))$ ao chegar a um destino de entrega de prisioneiros
- $\forall r \in R_i, dest(r)$ deve pertencer ao mesmo componente fortemente conexo do grafo G_i que o vértice S, uma vez que o veículo tem de ser capaz de voltar à central
- $\forall r \in R_i, numPris(r) > 0$, uma vez que não faz sentido ter um pedido para transportar zero prisioneiros
- $\forall e \in E_i, W_d(e) > 0 \wedge W_t(e) > 0$, uma vez que o peso da aresta representa a distância ou o tempo médio necessário para percorrer a aresta, se esta distância ou tempo forem zero estaremos num ciclo no mesmo vértice
- $\forall e \in E_i, e$ deve ser uma rua ao qual os veículos possam utilizar, ruas que os veículos não tenham permissão para entrar não são incluídas no grafo G_i
- $S \in V_i$, uma vez que a central é um vértice do grafo G_i

Sobre os dados de saída

- $|C_f| \leq |C_i|$ - não se pode usar mais veículos que os disponíveis
- $\forall v_f \in V_f, \exists v_i \in V_i$ tal que v_i e v_f têm os mesmos valores para todos os atributos, com exceção de atributos especificos aos algoritmos utilizados
- $\forall e_f \in E_f, \exists e_i \in E_i$ tal que e_i e e_f têm os mesmo valores para todos os atributos, com exceção de atributos especificos aos algoritmos utilizados
- $\forall c \in C_f, \forall s \in S(c), 0 \leq emptySeats < capacity(c)$ pois cada serviço deve ter pelo menos um prisioneiro, e não pode haver sobrelotação do veículo
- $\forall c \in C_f, \forall s \in S(c), |R_f(s)| > 0$ uma vez que só faz sentido realizar um serviço se for houver um pedido de transporte de prisioneiros
- $\forall c \in C_f, \forall s \in S(c), endHour(s) > startHour(s)$
- $\forall c \in C_f, \forall s \in S(c), startHour(s) < pickupHour(\forall r \in R_f) < endHour(s) \wedge startHour(s) < destHour(\forall r \in R_f) \leq endHour(s)$