Relatório CAL - 1ª Parte Trabalho Prático
11/4/2019

Tema 8 — SecurityVan: Entrega de Valores

(falta letra do grupo)

Eduardo Ribeiro —> up201705421@fe.up.pt

Eduardo Macedo —> up201703658@fe.up.pt

Diogo Machado —> up201706832@fe.up.pt

Descrição do Tema

O trabalho está relacionado com o conceito de veículos especializados em transporte e entrega de valores, que transportam grandes quantias de dinheiro e outros objetos valiosos, de um ponto para outro.

O trabalho consiste em implementar um sistema que permite a identificação de rotas ótimas para tais transportes, de modo a ser utilizado por uma empresa que se especializa em transporte de valores. Inicialmente, deve ser considerado que a empresa apenas dispõe de um veículo, a fazer todos os trajetos/entregas. Numa fase mais posterior, considerar-se-à que a empresa tem vários veículos e que os seus trabalhos/trajetos poderão ser especializados por tipo de cliente (bancos, museus, etc).

Estes transportes podem também implicar a recolha prévia dos valores em certos pontos, para depois os entregar. Por isso, são considerados dois tipos de transporte: um, em que é necessário sair da central onde estão os veículos, colectar os bens a um cliente, e depositá-los noutro; e o outro, em que se considera que no início o veículo já tem os bens, e basta ir para o destino.

Será considerado que os trajetos que o(s) veículo(s) poderão fazer serão do tipo:

central —> origem —> origem —> (….) —> destino —> destino —> (….) —> central

sendo que algumas origens poderão ser omitidas, devido à razão explicada em cima.

A entrega só poderá ser efetuada se existirem caminhos que liguem a central dos camiões, as origens (se existirem), os destinos, e de volta à central, dois a dois. Ou seja, para a entrega se poder concretizar, é necessário que os pontos façam parte do mesmo componente fortemente conexo do grafo, pelo que é de extrema importância a avaliação da conectividade do mesmo.

Por último, será também necessário considerar que obras nas vias públicas poderão tornar certas zonas do mapa inacessíveis, podendo ser impossível chegar a certos clientes; devem ser identificados os pontos de recolha/entrega com acessibilidade reduzida.

Identificação e Formalização do Problema

Dados de Entrada

- Ti —> sequência de trajetos/entregas que será necessário fazer, sendo i o i-ésimo elemento. Cada entrega será caracterizada por:

  - type —> tipo de cliente (museu, banco, loja, …) (na 1ª parte do problema, isto não vai ser considerado, uma vez que há apenas um veículo para todo o tipo de transportes)

  - origem —> vértice em que será necessário ir colectar os bens (poderá ser omitido)

  - destino —> vértice em que será necessário entregar os bens, depois de estarem dentro do veículo.

  - ID —> um ID único e específico, para cada trajeto/entrega.

- Ci —> veículo número i, que está disponível para fazer um percurso (na 1ª parte do problema, apenas vai ser considerado um único veículo). Cada um tem:

  - type —> tipo de entrega/trajeto (museu, banco, loja, …) em que o veículo é especializado (na 1ª parte, o único veículo irá fazer todos os trajetos, pelo que não terá especialidade).

- G = (V, E) —> grafo dirigido(*) pesado, representando o mapa da cidade. Este grafo irá ser constituído por vértices (V) e arestas (E).

  Cada vértice (representam pontos de importância no mapa) é caracterizado por:

  - type —> o vértice representa um museu, banco, loja, ou nenhum dos anteriores

  - Adj $\subseteq$ E —> arestas que saem desse vértice

  Cada aresta (representam ruas, pontes e outras vias) é caracterizada por:

  - w —> peso da aresta, que representa a distância entre os dois vértices que esta liga

  - ID —> identificador único de uma aresta

  - dest $\in$ V —> vértice de destino da aresta

- S $\in$ V —> vértice inicial, que representa a central de onde os veículos saem

(*) —> é considerado dirigido, pois a maior parte das ruas é só num sentido.

Dados de Saída

- Cf —> sequência ordenada de todos os veículos usados, cada um com:

  - type —> tipo de serviço/trajeto que esse veículo fez (igual ao desse veículo no início do algoritmo) (na 1ª parte do problema, não vai ser considerado, porque será um veículo a fazer todos os trajetos).

  - D —> vetor ordenado com os IDs das arestas pelo qual o veículo terá de passar. Pode haver repetidos.

  - Tf —> vetor com os IDs dos trajetos/serviços que esse veículo fez.

Restrições

Restrições nos dados de entrada (Pré condições):

- $\forall$ i $\in$ [1; T.size()] :

  - type(T[i]) = "BANK" v "MUSEUM" v "SHOP" —> o tipo de transporte/cliente terá de ser: ou agência bancária, ou museu, ou loja.

  - origem(T[i]) $\in$ V v origem(T[i]) = NULL —> a origem terá de ser um dos vértices do grafo. No entanto, e como já foi explicado acima, o trajeto poderá não ter origens, pelo que bastará ir da central para os destinos.

  - destino(T[i]) $\in$ V —> o destino terá de ser um dos vértices do grafo. Ao contrário da origem, não pode ser omitido.

  - type(origem(T[i])) = type(destino(T[i])) = type(T[i]) —> será considerado apenas trajetos de museu para museu, banco para banco, etc. O tipo do vértice de destino terá de ser igual ao tipo do trajeto, bem como o tipo da origem, se esta existir. Esta restrição também assegura que a origem e o destino serão sempre ou um banco ou um museu ou uma loja, e não vértices normais.

  - origem(T[i]) ≠ destino(T[i]) —> os vértices de destino e origem (este último, quando for especificado) terão de ser diferentes.

- $\forall$ V, type(V) = "BANK" v "MUSEUM" v "SHOP" v "NONE" —> para além de termos vértices a representar os possíveis clientes, também temos vértices "normais".

- $\forall$ C $\in$ Ci, type(C) = "BANK" v "MUSEUM" v "SHOP", exceto na 1ª parte do problema, em que apenas é considerado um veículo para tudo.

- type(S) = "NONE" —> a central dos veículos será um vértice do tipo "NONE".

- $\forall$ E, w(E) > 0 —> o peso das arestas terá de ser positivo uma vez que representam distâncias.

- $\forall$ E, E deve ser utilizável pelo(s) veículo(s). As arestas que não cumprirem isto serão removidas por um pré-processamento, e não serão incluídas no grafo G inicial.

- Seja L o conjunto de vértices constituído por S (central), todos as origens, e todos os destinos. É necessário que todos os elementos de L estejam no mesmo componente fortemente conexo do grafo (CFC).

  Deste modo, a partir de qualquer elemento do conjunto, é possível aceder qualquer outro, no grafo dirigido. Dado que todos os percursos começam e acabam no mesmo ponto (central), então todos os vértices do percurso devem pertencer ao mesmo CFC. Cada percurso então poder ser considerado como um CFC, sendo que como terão pelo menos um vértice em comum (central), irão originar um grande CFC, composto pela central, todas as origens, e todos os destinos (conjunto L).

Restrições nos dados de saída (Pós condições):

- Cf.size() <= Ci.size() —> obviamente, não será possível utilizar mais veículos do que os disponíveis no início.

- $\forall$ C $\in$ Cf:

  - $\forall$ i $\in$ [1; Tf.size()], type(C) = type (Tf[i]) —> um veículo de um dado tipo só pode fazer trajetos/entregas desse tipo (não aplicável à 1ª parte do projeto).

  - Seja di o primeiro elemento de D. É preciso que di $\in$ Adj(S), uma vez que todos os percursos começam a sair da central dos veículos.

  - Seja df o último elemento de D. É necessário que dest(df) = S, uma vez que todos os percursos devem acabar por retornar à central dos veículos.

  - $\forall$ i $\in$ [1; Tf.size()], $\exists$ d1, d2 $\in$ D tal que dest(d1) = origem(Tf[i]) ^ dest(d2) = destino(Tf[i]) ^ d1 < d2 —> se um trajeto foi incluído no vetor de trajetos desse veículo, significa que este concluiu o dado trajeto, ou seja, passou pelo menos uma vez na origem, e depois no destino do trajeto em causa. Por causa disto, o vetor de arestas pelo qual o veículo terá de passar terá de possuir pelo menos duas arestas, cujo destino sejam a origem e o destino do trajeto, respetivamente, sendo que esta última terá de vir depois da primeira, no vetor ordenado (isto não elimina a possibilidade de se passar no destino antes da origem).

- $\min(\sum_{c \in Cf} (\sum_{d \in D} w(d)))$ —> os caminhos percorridos pelos veículos serão os que permitirem fazer as entregas na menor distância possível.

Função objetivo

A solução ótima do problema reside em, como já sabemos, encontrar as rotas ótimas para a entrega dos valores por parte dos veículos. Ou seja, queremos que eles terminem as suas tarefas/trajetos na menor distância total possível. Assim, a solução ótima irá passar pela minimização da soma das distâncias percorridas por todos os veículos utilizados:

$$\min\left(\sum_{c \in C} f \left(\sum_{d \in D} w(d) \right) \right)$$

Perspetiva de solução