CITY SIGHTSEEING: TRILHAS TURÍSTICAS URBANAS

Bruno Micaelo (up201706044@fe.up.pt)

Inês Alves (up201605335@fe.up.pt)

João Campos (up201704982@fe.up.pt)

CONCEPÇÃO E ANÁLISE DE ALGORITMOS - 01/05/2019

Indice

Introdução

Este trabalho tem como propósito servir como implementação de um sistema de planeamento de pequenas viagens turísticas em autocarros.

Uma viagem de autocarro começa num certo ponto de encontro, onde os turistas se reúnem, e estes, por sua vez, têm vários pontos de interesse (POIs), que são locais da cidade que gostariam de visitar. O autocarro deve, então, tentar passar pelos pontos de interesse dos turistas que transporta, através do melhor caminho possível. Para isto, utilizar-se-ão vários algoritmos de forma a proporcionar uma solução ideal, que tanto minimize o número de autocarros utilizados pela empresa, como a distância total percorrida nas viagens, sendo preferida a redução da distância total, visto que uma menor distância equivale a um menor tempo de viagem para os turistas.

Formalização do problema

Dados de entrada

Gi = (Vi, Ei)
Grafo dirigido pesado, este é composto por:

- Vértices: são representativos de pontos da cidade, cada um tem:
  - id - Identificador do vértice;
  - name - Nome do ponto da cidade representado;
  - x - Componente X da coordenada geográfica do ponto;
  - y - Componente Y da coordenada geográfica do ponto;
  - adj - Vetor de arestas que partem do vértice.
- Arestas: são representativos das vias da cidade, cada uma tem:
  - id - Identificador da aresta;
  - name - Nome da via representada;
  - weight - Distância a percorrer na aresta;
  - dest - Vértice destino da aresta.

Ai

Vetor de autocarros, cada um composto por:

- id - Identificador do autocarro;
- cap - Capacidade do autocarro (sem contar com o motorista).

Pi
Vetor de passageiros, cada um composto por:

- id - Identificador do passageiro;
- name - Nome do passageiro;
- pois - Vetor de pontos de interesse (referências a Vértices).

S - Vértice inicial (tipicamente o ponto de encontro para os turistas).

Ti - Vetor de vértices finais (tipicamente é apenas um, correspondente ao ponto de encontro).

Dados de saída

Gf = (Vf, Ef)

Grafo dirigido pesado correspondente ao grafo inicial sem os vértices e arestas inacessíveis pelos autocarros.

Af

Vetor de autocarros utilizados, cada um composto por:

- pass - Vetor de referências aos passageiros que viajam no autocarro;
- path - Vetor de referências a arestas pelas quais o autocarro passa.

Restrições

Dados de entrada

- O número de autocarros é superior a 0;
- O número de turistas é superior a 0;
- ∀ a ∈ Ai, a.cap > 0;
- ∀ e ∈ Ei, e.weight > 0;
- Todos os identificadores têm que ser únicos em:
  - Vértices;
  - Arestas;
  - Autocarros;
  - Turistas;
- S ∈ Vi;
- Ti ⊆ Vi;
- O grafo deve ser fortemente conexo;
- Todos os vértices não acessíveis pelos autocarros são ignorados.

Dados de saída

- Vf ⊆ Vi, tendo em conta a remoção dos vértices inacessíveis reduz o conjunto de vértices original;
- Ef ⊆ Ei, tendo em conta a remoção das arestas indisponíveis reduz o conjunto de arestas original;
- O tamanho de Af deve de ser menor ou igual ao tamanho de Ai, visto que os autocarros utilizados pertencem também ao conjunto de autocarros disponíveis.
- Para cada autocarro, o path deve ter como o primeiro elemento uma aresta proveniente de S, e o último elemento deve ter uma aresta cujo destino pertence a Ti (normalmente S).

Funções objetivo

De forma a ser utilizado o menor número de autocarros e a que estes percorram a menor distância possível entre pontos de interesse, pretendemos minimizar as seguintes funções objetivo, privilegiando a da função g em relação à da função f:

- f = |Ai| (minimização do número de autocarros);
- g = $\sum_{a \epsilon Af}$ [$\sum_{p \epsilon a.path}$ $w(p)$] (minimização da distância percorrida);

Solução do problema

Decompomos este problema em três iterações:

1. **Autocarro com capacidade infinita, sem organização dos turistas**

Numa primeira fase, o objetivo é apenas encontrar o menor caminho possível entre os vários pontos de interesse na viagem de autocarro, não se agrupando, então, os turistas pelos seus POIs (pontos de interesse) semelhantes. A viagem começa num ponto de encontro definido, num autocarro com capacidade de passageiros infinita, passando por todos os POIs dos turistas que transporta, sempre através do melhor caminho possível.

É, também, necessário que os vértices do grafo, que representam os pontos de interesse da cidade, façam parte da mesma componente fortemente conexa, ou seja, terão que existir caminhos de acesso a qualquer POI do grafo. Por exemplo, se o autocarro se encontrar num certo ponto de interesse, deve tanto poder chegar a qualquer outro POI, como regressar, se necessário, ao local inicial, seguindo vários caminhos do mapa.

Para além disto, é necessário ter em conta que algumas vias de comunicação da cidade podem estar inacessíveis (no caso de obras, por exemplo), procedendo-se à remoção das arestas que as representam.

Utilizaremos, para garantir a melhor solução possível, o algoritmo de Dijkstra.

2. **Autocarros com capacidades infinitas, com organização dos turistas**

Neste caso, ao contrário da primeira iteração, os turistas irão ser distribuídos por vários autocarros, tendo, agora, em conta, os seus pontos de interesse, de modo a que cada autocarro tenha passageiros com POIs iguais ou próximos entre si no mapa. Assim, em cada viagem, haverá um menor número total de POIs, de forma a otimizar o caminho a percorrer. A capacidade do autocarro é, ainda, infinita.

Para isto, utilizaremos uma implementação gananciosa para a distribuição dos turistas.

3. **Autocarros com capacidades finitas, com organização dos turistas**

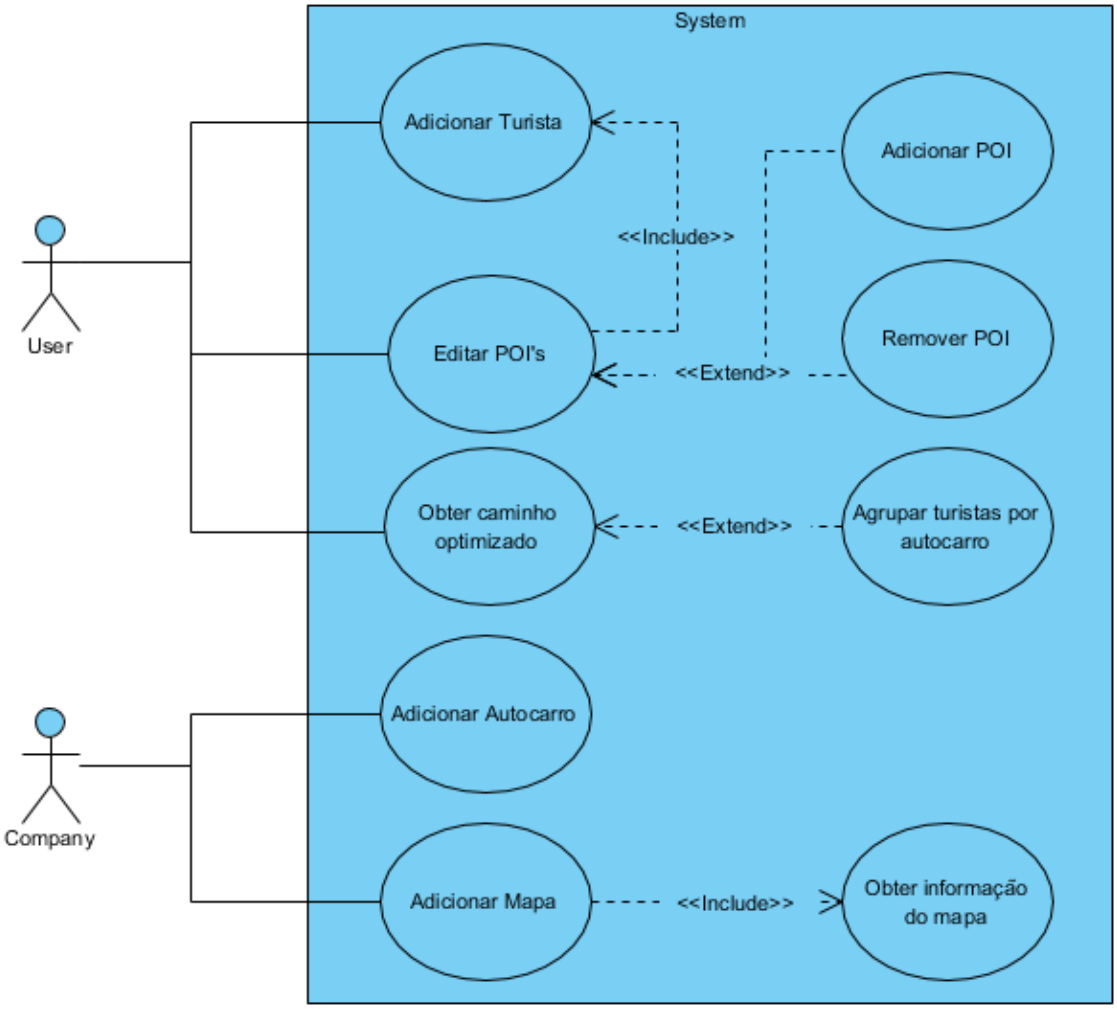
A terceira fase considera, finalmente, autocarros com capacidades finitas, encontrando a solução final para o problema. Neste caso, se não for possível transportar certo grupo de turistas por excederem a capacidade máxima de passageiros, terá de ser utilizado outro autocarro.

Pretende-se, então, minimizar tanto o número total de autocarros necessários como a distância total percorrida nas viagens, dando prioridade a esta última.

Para isto, utilizaremos um algoritmo de retrocesso, alcançando a solução ideal através de sucessivas tentativas.

Casos de Utilização

Os seguintes casos de utilização têm como base a última implementação do algoritmo de minimização de caminhos e organização de turistas, sendo resumidos pelo seguinte diagrama.



Os casos uso do sistema envolvem dois atores, User e Company, representando respetivamente o utilizador (que adiciona um ou mais turistas e organiza as viagens) e a empresa responsável pelo sistema (que disponibiliza o transporte nas viagens).

Use Case 1	Adicionar Turista
Actor	User
Fluxo básico	O utilizador especifica o nome do turista a adicionar, podendo especificar de seguida os respetivos pontos de interesse (Ver Use Case 2).

Use Case 2	Editar POI's
Actor	User
Fluxo básico	O utilizador seleciona um turista e altera os POI's associados ao mesmo, podendo também adicionar e remover POI's *(Ver Use Case 2.1 e 2.2)*

Use Case 2.1	Adicionar POI
Actor	User
Fluxo básico	O utilizador seleciona do conjunto de vértices disponíveis um ponto de interesse para ser adicionado à lista de POI's de um turista.

Use Case 2.2	Remover POI
Actor	User
Fluxo básico	O utilizador seleciona da lista de POI's de um turista um ponto de interesse para remover da mesma.

Use Case 3	Obter caminho optimizado
Actor	User
Fluxo básico	O utilizador obtém o caminho optimizado tendo em conta a disponibilidade dos autocarros e a combinação dos interesses dos turistas.

Use Case 3.1	Agrupar turistas por autocarro
Actor	User
Fluxo básico	O utilizador obtém o conjunto de autocarros com os respetivos grupos de turistas a transportar, considerando a compatibilidade entre os seus interesses.

Use Case 5	Adicionar Autocarro
Actor	Company
Fluxo básico	A empresa de transportes adiciona um autocarro ao conjunto de transportes disponíveis.

Use Case 6	Adicionar Mapa
Actor	Company
Fluxo básico	A empresa adiciona um conjunto de locais e vias ao mapa da zona disponível para visitas turísticas (a partir do tratamento de info informação relativa mesmo *(Ver Use Case 6.1)*).

Use Case 6.1	Obter informação do mapa
Actor	Company
Fluxo básico	A empresa fornece um ficheiro com informação relativamente ao mapa para o sistema a tratar e obter um conjunto de locais e vias, mais facilmente visualizáveis pelos utilizadores e tratáveis pelo sistema.

Conclusão

O objetivo deste trabalho é a criação de um algoritmo que permita organizar turistas por autocarros, por forma a permitir a visita de diferentes pontos de interesse numa dada área.

Ao realizar este relatório tivemos em consideração a elevada complexidade da preferência de POIs pelos turistas. Decidimos então, primeiramente, arranjar uma solução que não envolvesse esta parte do projeto, construindo, mais tarde, um algoritmo que categorizasse os turistas pelos seus gostos, de forma a dividi-los nos diferentes autocarros.

Quanto ao esforço realizado por cada membro do grupo, cada um trabalhou de forma igual, contribuindo para cada secção do trabalho.