AirShuttle - Transfer em aeroportos

Turma 4 Grupo F

24/04/2019

Afonso Sá – up201604605@fe.up.pt

André Serralheiro – up201604566@fe.up.pt

Carlos Vieira – up201606868@fe.up.pt

Índice

Descrição do tema 3

Formalização do problema 5

* Dados de entrada 5
* Dados de saída 5
* Restrições 6
* Função Objetivo 6

Solução Implementada 7

* Dijkstra 7

Casos de utilização 8

Dificuldades 9

Conclusão 9

Descrição do tema

A empresa AirShuttle presta serviços de transfer entre o aeroporto Francisco Sá Carneiro e hotéis, ou outros locais da região, por um preço bastante acessível.

Para o efeito dispõe de um certo número de carrinhas, uma ou mais, com capacidade para transportar um certo número de passageiros cada. Ao viajar para o porto, os passageiros têm a opção de realizar uma reserva do serviço pela Internet, indicando hora de chegada do voo, e destino para onde desejam ir.

Dado um conjunto de pedidos de serviços a prestar num determinado dia, e respetivos destinos, a empresa deve elaborar o seu plano de deslocações. A ideia é reunir o maior número possível de passageiros por deslocação, de forma a evitar que passageiros que chegaram mais cedo fiquem muito tempo à espera dos próximos.

Primeira Parte - Uma carrinha

Primeiramente no problema que vamos abordar a empresa tem uma só carrinha, como tal, de modo a maximizar a quantidade de deslocações estas devem ser breves.

De tal modo, ordenamos os passageiros numa lista de espera em ordem da sua chegada. Criamos assim um grupo de passageiros de forma a que a primeira pessoa não espere mais de uma hora pelo início da viagem. Quando a carrinha estiver de novo no aeroporto realizamos de novo o mesmo processo com a lista atual. Conseguindo assim que o passeiro que está na lista de espera há mais tempo, fique o mínimo de tempo possível no aeroporto. Os percursos têm duração máxima de uma hora, e apenas nos casos de sobrelotação poderá um passageiro esperar mais do que o tempo estipulado.

Segunda Parte – Frota de carrinhas

Na segunda parte a empresa possui uma frota de carrinhas, como tal as viagens efetuadas por cada carrinha podem ser maiores, uma vez que há mais carrinhas para preencher o horário.

O tempo de espera máximo para os passageiros pode ser diminuído passando a ser de 30 minutos. Segue-se então a mesma logica da primeira parte, mas tem-se agora em conta o destino dos passageiros, de tal modo, agrupam-se os passageiros de forma mais eficiente.

Formalização do problema

Dados de entrada:

G (V, E) – Grafo Não Dirigido Pesado

V – Vértices, representam paragens entre o caminho realizado pela carrinha. Estas paragens são o aeroporto, hotéis e caminho predefinidos.

E – Arestas, são o caminho a ser percorrido entre vértices.

* T – Peso da aresta, que é medido em minutos (Tempo)

I – Vértice inicial, indica o local onde o caminho vai ser iniciado (Aeroporto).

Dados de saída:

Os dados de saída são todas as paragens pelo qual a carrinha passa durante a viagem, a quantidade de tempo que esta demorou, bem como a quantidade de pessoas transportadas.

Caminho percorrido: {Vi}, i = 1, …, n ; n≠0

Tempo da viagem: (soma do tempo de cada arresta percorrida)

Restrições:

Em relação à viagem, na primeira parte, há a restrição de tempo de viagem, sendo este uma hora e para a segunda parte, quatro horas.

Quanto à quantidade de pessoas a serem transportadas numa carrinha de uma só vez, a restrição é de 10 pessoas, sem contar com o condutor, como tal, durante uma viagem não é possível transportar mais do que 10 pessoas ao mesmo tempo.

A viagem acaba sempre quando se chega de volta ao aeroporto.

Função Objetivo:

De modo a otimizar a resolução do problema vamos necessitar que as viagens sejam as mais curtas possíveis, tanto em tempo como em distância, e com o maior número de passageiros transportados. Como tal a função objetivo deste problema requer a otimização de:

, logo queremos o minimizar o

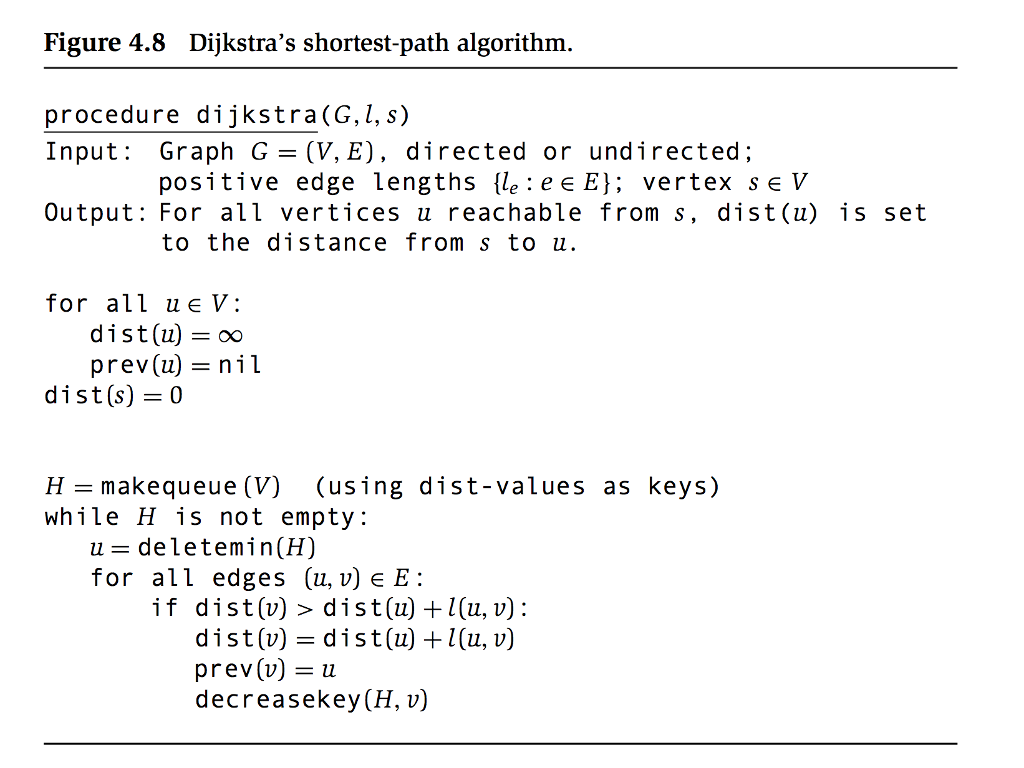
Solução Implementada

Algoritmos a utilizar:

Seguindo a lógica descrita em cima, o nosso problema vai exigir o caminho mais curto, consequentemene o algoritmo que vamos implementar na resolução do problema é o:

* Dijkstra: Este algoritmo vai ser implementado de modo a obter o caminho mais curto, de tal modo obtendo assim o caminho mais rápido e maximizando a eficiência de cada transporte.

Como o grafo que é estudado é pesado é então necessário usar a variação gananciosa do algoritmo de Dijkstra, assim é possível obter uma maximização da eficiência em cada passo. O algoritmo possui então um tempo de execução de .

Exemplo em pseudo-código:

Casos de utilização

Ao serem realizadas as marcações a empresa guarda os dados que interessam para a viagem, o hotel de destino, as estradas a serem percorridas e o tempo de viagem até ao mesmo. De tal modo o nosso programa necessita de realizar:

* Leitura de dados de ficheiros relativos aos caminhos e passageiros;
* Escolha do melhor percurso em termos de tempo de viagem, em relação a uma carrinha ou uma frota;
* Visualização através do GraphViewer dos caminhos entre o aeroporto e hotéis;
* Visualização do melhor percurso através do GraphViewer aliado com o openstreatmap.

Dificuldades

A nossa principal dificuldade foi a escolha de um algoritmo que realizasse as tarefas pretendidas, uma vez temos um grande número de variáveis que vão todas influenciar a escolha do percurso a ser realizado. Como não há nenhum algoritmo que tenha em conta todas essas variáveis, o tempo de viagem, tempo de espera do cliente e quantidade de clientes a transportar foi-nos difícil subdividir o problema de modo a encontrar algoritmos que nos ajudassem na escolha do caminho ideal.

Como tal, apostamos, por agora, na escolha do caminho mais rápido, sendo que como tal dividimos a lógica da escolha dos passageiros e do caminho.

Conclusão

A realização do trabalho envolveu primeiramente uma divisão da lógica do mesmo, sendo que em relação aos passageiros decidimos criar uma fila, no qual estes se agrupam e têm um tempo de espera para poderem entrar na carrinha. Assim, quanto à logica dos passageiros o problema está resolvido uma vez que a espera é pequena e temos em conta a prioridade de quem está à espera durante mais tempo.

Em relação ao caminho mais curto entre paragens para deixar os clientes nos hotéis, escolhemos o algoritmo de Dijkstra que nos fornece o caminho mais rápido (peso das arestas medido em minutos).

No entanto, pretendemos futuramente conseguir resolver o problema de uma forma mais eficaz, uma vez que sabemos que a nossa resolução do problema não seja a mais prática para os clientes.

Esforço de cada elemento

Afonso Sá - %

André Serralheiro – %

Carlos Vieira – %