

GRAFOS

TRABALHO PRÁTICO – 2018/1

1. Transporte Multimodal em Paris

Diante das diversas modalidades de transporte (ônibus, táxi, metrô, trem, carro, etc) existentes nos dias atuais, os passageiros de um sistema de transporte podem tomar diferentes rotas para sair de um ponto de origem e alcançar um ponto de destino. O transporte multimodal permite que os passageiros alternem entre tipos de transportes diferentes visando algum objetivo, como distância mais curta ou tempo de viagem.

A Figura 1 apresenta a rede do sistema de transporte multimodal da cidade de Paris, modelada através de um grafo de múltiplas camadas, onde cada camada corresponde a um tipo diferente de transporte, interconectadas por ligações entre essas camadas (indicam a mudança de modal). Cada vértice do grafo corresponde a uma interseção de vias, estação de metrô ou estação de trem. Cada aresta do grafo corresponde a vias/ruas, linha de metrô, linha de trem, mudança de modal. Este grafo contém 15.497 vértices e 27.092 arestas.

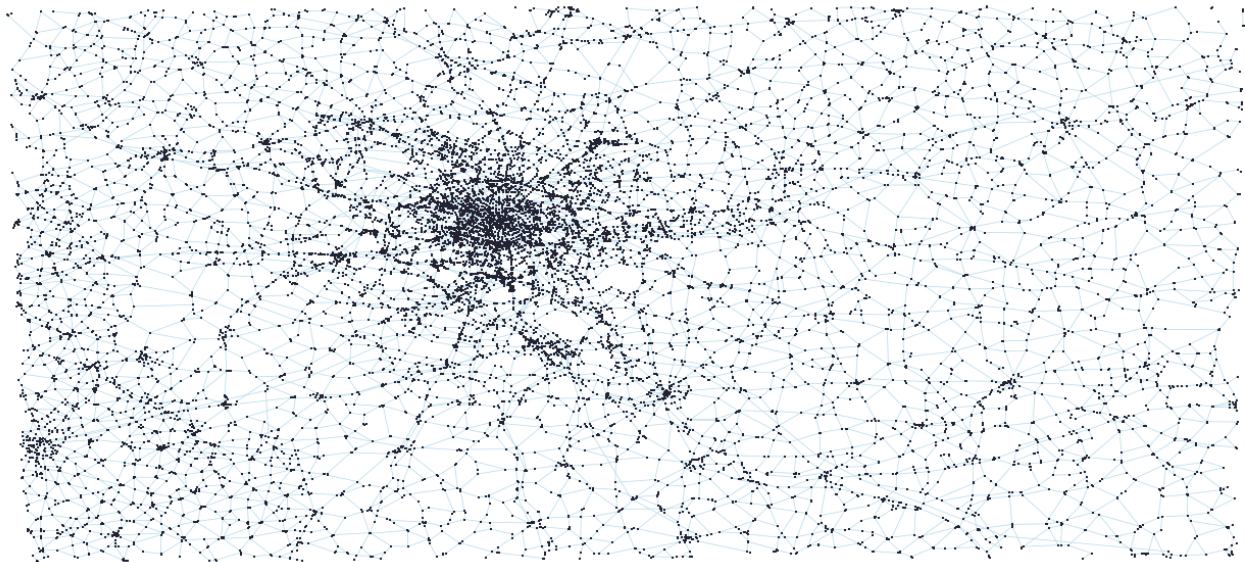
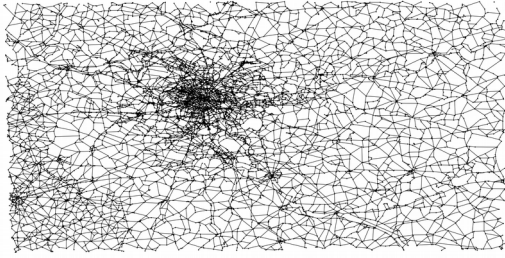


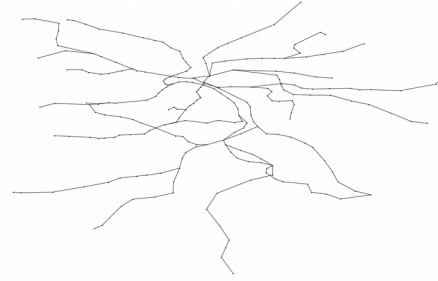
Figura 1: Sistema de Transporte Multimodal de Paris

A Figura 2 apresenta as diversas camadas do grafo desagregadas. Cada camada contém as seguintes características:

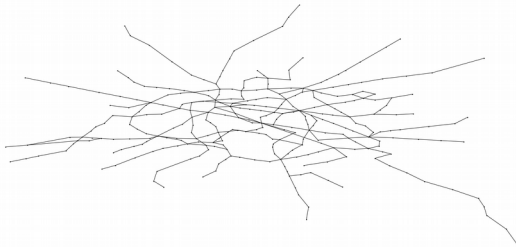
Rede	Número de vértices	Número de arestas
Rodovias	14.807	22.281
Trem	241	244
Metrô	303	356
Tram/RER	146	140



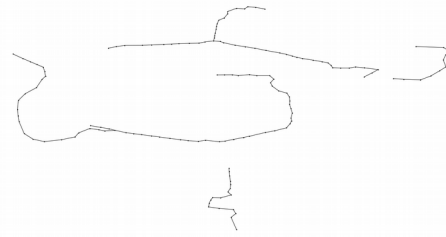
Rede de rodovias



Rede de trem



Rede de metrô



Rede de tram/RER

Figura 2: Redes de transporte multimodal desagregadas

2. Problema: Definir rotas alternativas

O problema ser solucionado no grafo de sistema de transporte multimodal de Paris consiste em definir diferentes rotas para o passageiro, de forma que este possa escolher a rota que mais lhe agrada. Dessa forma, o programa deve permitir ao passageiro a escolha do ponto de origem (algum vértice do grafo, dado por *origem*) e também seu ponto de destino (outro vértice do grafo, dado por *destino*). O passageiro também deve indicar quantas rotas (dado por *k*) alternativas devem ser ofertadas.

A saída consiste em devolver ao passageiro o conjunto das *k* rotas mais curtas entre origem e destino.

3. Solução por k-shortest paths [1]

Uma possível forma de solução do problema pode ser baseada no algoritmo de *k* caminhos mínimos. O algoritmo proposto por Yen [2] em 1971 emprega o algoritmo de caminho mínimo para encontrar o melhor caminho e posteriormente encontra *k-1* desvios do melhor caminho. Este algoritmo garante que os caminhos sejam elementares, ou seja, livres de laços.

4. Trabalho:

O trabalho consiste em implementar uma solução para o problema proposto através do algoritmo *k-shortest paths*. Assim, o usuário do sistema fornece os seguintes dados de entrada e obtém os seguintes dados de saída:

Entrada: origem, destino, *k*

Saída: *k* rotas

5. Entrega:

a. Documentação

Como o programa funciona, quais estruturas de dados utilizadas, como é a entrada do programa, como é a saída, em qual linguagem foi implementado, como se faz para executar o programa, resultados encontrados e os testes realizados para confirmar se o programa executa corretamente.

b. Código do programa na linguagem escolhida.

6. Pontos Extra:

a. Interface gráfica indicando as k rotas e modais identificados por cores diferentes

b. Implementação com uso de heap

7. Arquivos:

Os dados necessários para implementação do trabalho estão disponíveis no Portal Didático. Os arquivos são detalhados a seguir.

vertex.csv: contém informações sobre os vértices do grafo incluindo ID, latitude, longitude e camada a qual pertence.

edges.csv: contém informações sobre as arestas do grafo incluindo ID do vértice de origem, ID do vértice de destino, ID da aresta, direção e camada.

Atenção para os detalhes:

Grafo direcionado: OneWay (1 arco), TwoWay (2 arcos)

Camadas: road, metro, train, tram

Pesos: o grafo de entrada é não ponderado e os pesos deve ser criados utilizando a distância de *Haversine* [3] considerando no modelo a curvatura da Terra, com raio $R = 6371000$ metros.

Referências

[1] https://en.wikipedia.org/wiki/K_shortest_path_routing

[2] https://en.wikipedia.org/wiki/Yen%27s_algorithm

[3] https://pt.wikipedia.org/wiki/F%C3%B3rmula_de_Haversine