

Simulação de Tráfego e Mobilidade Urbana

Grupo nº6: Henrique Ponte, Miguel Carvalho, Miguel Viveiros

Docentes: José Cascalho, Armando Mendes



1 Resumo

O nosso trabalho é baseado, em avaliar um ‘dataset’ em que contem informação sobre os congestionamentos na capital e retirar a informação sobre as horas e os motivos em que estes ocorrem.

Para tal foi usado o ‘notebook’ do ‘Jupiter’ e o ‘Python’ para além de processar os dados, ainda fizemos uma simulação para encontrar o caminho mais rápido de um ponto para outro utilizando o algoritmo A* este caminho é calculado a partir dos transito existente na determinada hora.

2 Introdução

O nosso problema consistia em analisar os dados de forma a saber em que ruas e quais horários acontecem os congestionamentos, tendo em conta o motivo e o impacto no trânsito.

Para a parte de explorar os dados e pré-processamento começou-se por remover os ‘outliers’ e elaborou-se tabelas para mostrar as relações entre colunas do ‘dataset’. De seguida procedeu-se com o estudo da aprendizagem não supervisionada e supervisionada do conjunto de dados.

Num contexto de simulação, usou-se os horários mais frequentes para simular uma ‘mini Lisboa’ para permitir obter um sistema de GPS utilizando o algoritmo A* de forma a combater o tráfego.

3 Simulação de Tráfego e Mobilidade Urbana

3.1 Trabalho Desenvolvido

Com este trabalho pretendemos procurar uma solução para combater ao tráfego na cidade de Lisboa de em que na simulação para encontrar o caminho mais rápido de um ponto para outro será utilizado o algoritmo A* que irá calculado a partir dos transito existente na determinada hora do dia o melhor caminho para chegar ao destino no menor tempo possível e no menor engarrafamento.

3.2 Aprendizagens

Com este trabalho podemos melhorar os nossos conhecimentos com o algoritmo A* e o nosso conhecimento em mexer em grafos, onde utilizamos tecnicas de cores referentes a pesos do tráfego.

3.3 Algoritmo A*

O algoritmo A* tratase de um método de busca de caminho mais eficiente, especialmente em situações de busca de caminhos, também conhecido como pathfinding. Ele busca o caminho em um grafo de um vértice inicial até um vértice final. O processo começa pelo nó inicial e aplica uma função de heurística a todos os vizinhos do nó atual. Esta função retorna um número que indica a distância estimada para o nó final, geralmente usando a distância euclidiana. O vizinho com o menor valor é considerado o mais próximo do nó final e se torna o nó atual. Este procedimento é repetido até que o nó atual seja o nó final.

4 Descrição

Para a nossa aplicação, começou-se por fazer um grafo com alguns pontos turísticos de Lisboa associando a estes um peso base e coordenadas. Para além disto, utilizou-se uma heurística com a distância em linha reta entre os pontos do grafo que pudesse modificar os valores de peso de cada conexão existente entre os vértices. De seguida, criou-se uma janela ‘tkinter’ de forma a mostrar o mapa gerado e os campos de inserção de origem e destino, bem como uma opção para trocar a hora atual do mapa. O algoritmo A* encontra o caminho mais curto em um grafo, começando pelo nó inicial, avaliando todos os vizinhos do nó atual com uma função de heurística, e escolhendo o vizinho com o menor valor para se tornar o novo nó atual, repetindo este processo até que o nó final seja alcançado.

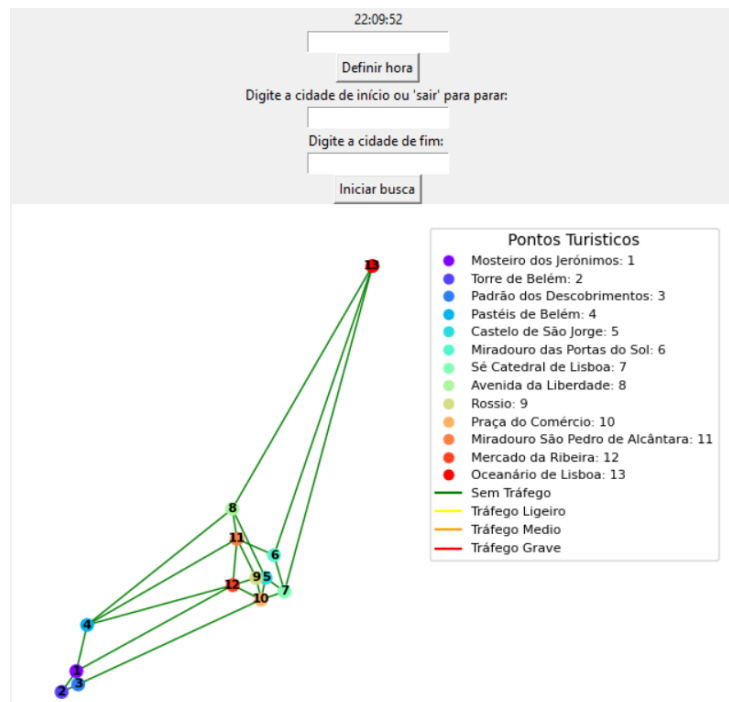


Fig. 1: Simulação de tráfego no momento sem tráfego

De acordo com a hora do dia as arestas do grafo vão mudando de cor tendo em conta o trânsito sendo verde o trânsito mais leve e o vermelho o mais acentuado passando pelo amarelo e laranja. Implementou-se o algoritmo A* que permite verificar o estado do mapa atual e indica o caminho com menor custo, não necessariamente o com menor distância. Através da figura 1 podemos ver que pela hora não existe tráfego que afete a deslocação entre pontos.

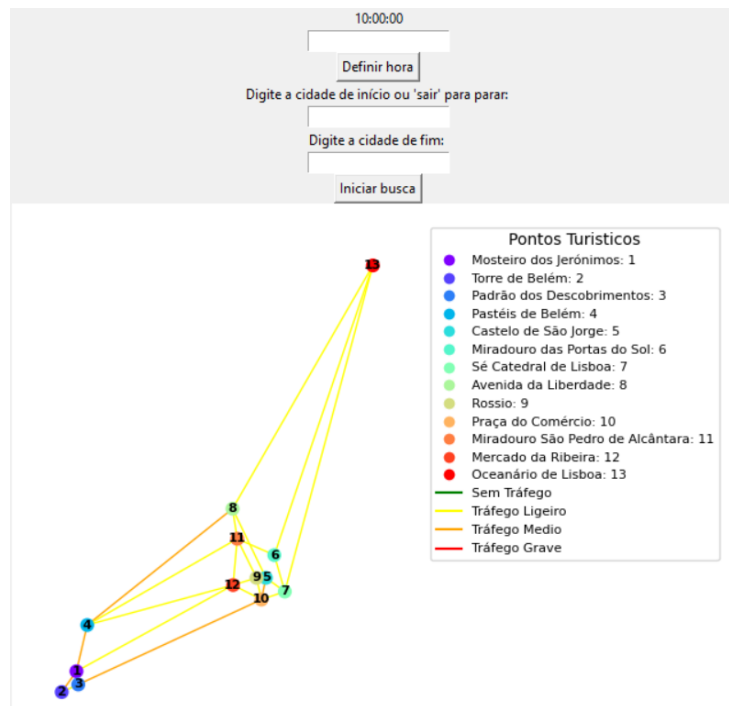


Fig. 2: Simulação de tráfego no momento com tráfego

Já na figura 2 representa uma hora de maior tráfego o que irá afetar a deslocação entre pontos, deste modo será aplicado o algoritmo A* para calcular o melhor caminho entre o ponto de partida escolhido e o destino pretendido.

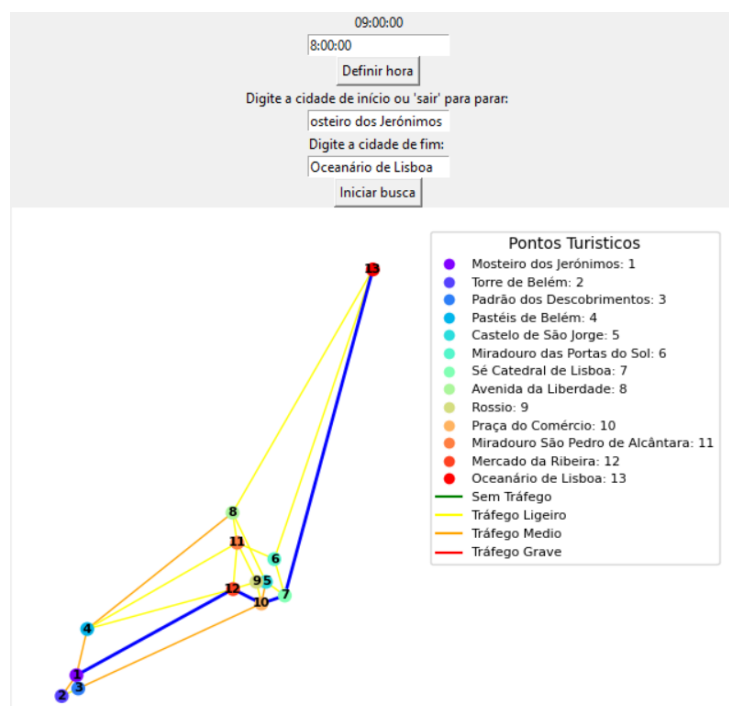


Fig. 3: Simulação da solução entre pontos

O algoritmo seleciona a rota de menor custo com base no volume de tráfego. Portanto, mesmo que uma rota seja mais longa em termos de distância, ela pode ser mais rápida considerando as condições de tráfego. Assim sendo através da figura 3 podemos observar a solução para a partida dos Mosteiro dos Jerónimos e chegada ao Oceanário de Lisboa.

5 Discussão

Apartir desse trabalho podemos verificar quais são as horas com maior e menor tráfego na cidade de Lisboa. Com a simulação foi possível observar que nem sempre o caminho mais curto é o caminho mais rápido devido ao tráfego provocado por vários motivos presentes no dataset e também ajudou a conhecermos melhor o algoritmo A*. No final, foi possível obter uma aplicação que é responsável por ilustrar o caminho mais curto, sendo capaz de receber a lista de pontos e ilustrar o caminho da solução através de uma faixa azul.

Em suma, temos uma aplicação que só em si já simula o trânsito em alguns pontos turísticos de Lisboa através de um sistema de horas e pesos, e conseguiu-se encontrar o caminho mais curto entre dois pontos consoante o trânsito na cidade.

Referências

DataSet: <https://dados.cm-lisboa.pt/en/dataset/condicionamentos-de-transito-historico>