



Universidade do Minho

Braga, Portugal

# OTIMIZAÇÃO DE UMA FROTA DE TÁXIS

## RELATÓRIO DO TRABALHO PRÁTICO

### Inteligência Artificial

Departamento de Informática

Engenharia Informática 2025/26

Grupo 1:

A106936 - Duarte Escairo Brandão Reis Silva

A106932 - Luís António Peixoto Soares

A106856 - Tiago Silva Figueiredo

A104704 - Inês Ferreira Ribeiro

9 Janeiro 2026

# Índice

1. Avaliação por pares .....	1
2. Descrição do Problema .....	2
3. Formulação do Problema .....	3
4. Implementação .....	5
4.1. Cidade .....	5
4.2. Algoritmos de procura .....	6
4.3. Tempo .....	6
4.4. Gerar dados .....	6
4.5. Simulação .....	7
4.6. Gestor de pedidos .....	7
4.7. Estatísticas .....	9
5. Resultados .....	10
5.1. Modo básico .....	10
5.2. Modo rápido .....	12
5.3. Modo económico .....	14
5.4. Modo ambiental .....	16
6. Discussão .....	18

## 1. Avaliação por pares

A106936 Duarte Escairo Brandão Reis Silva  $\rightarrow$  DELTA = 0

A106932 Luís António Peixoto Soares  $\rightarrow$  DELTA = 0

A106856 Tiago Silva Figueiredo  $\rightarrow$  DELTA = 0

A104704 Inês Ferreira Ribeiro  $\rightarrow$  DELTA = 0

## 2. Descrição do Problema

No âmbito da Unidade Curricular de Inteligência Artificial, foi-nos proposto desenvolver algoritmos de procura que permitam otimizar a gestão da frota da empresa de táxis *TaxiGreen*, garantindo que o atendimento de todos os pedidos de boleia é feito num período de tempo aceitável, tomando em consideração as preferências dos clientes, as limitações de cada tipo de veículo, o seu impacto ambiental e o trânsito presente na rua. Os veículos elétricos têm menor autonomia e necessitam de carregar periodicamente em estações de recarga. Enquanto que os veículos a combustão podem reabastecer mais rapidamente, em postos de abastecimento, mas têm custos operacionais superiores e maior impacto ambiental (emissões de  $CO_2$ ).

Cada veículo da frota é caracterizado por:

- **Tipo de motorização:** elétrica ou a combustão;
- **Autonomia máxima:** distância, em quilómetros, que consegue percorrer, após encher o depósito/bateria;
- **Autonomia atual:** distância, em quilómetros, que consegue percorrer, até ter que voltar a abastecer/carregar;
- **Capacidade de passageiros:** quantos passageiros consegue levar de uma vez só;
- **Custo operacional:** relação entre o custo do abastecimento/carregamento e a autonomia (€/km);
- **Impacto ambiental:** emissões de  $CO_2$  (g/km);
- **Disponibilidade:** se está disponível para aceitar pedidos de boleia;
- **Localização atual:** onde o se encontra, de momento;
- **Tempo de reabastecimento:** tempo, em minutos, que demora a encher o depósito/carregar a bateria.

Cada pedido de boleia é caracterizado por:

- **Origem:** zona onde cliente pretende ser recolhido;
- **Destino:** zona onde cliente pretende ser deixado;
- **Número de passageiros:** quantas pessoas pretendem ir na viagem;
- **Horário pretendido:** limite de tempo, em minutos, para ser atendido;
- **Prioridade:** se é um cliente *premium* ou se trata de um pedido urgente;
- **Preferência ambiental:** se o cliente tem preferência por um veículo elétrico.

### 3. Formulação do Problema

**Representação do estado:** O estado do problema, a cada instante, pode ser representado pela expressão:  $E = (t, V_o, V_d, P_r, G)$ , em que:

- $t$  é o instante de tempo;
- $V_o$  é o conjunto de veículos ocupados;
- $V_d$  é o conjunto de veículos disponíveis;
- $P_r$  é o conjunto de pedidos de boleia por responder;
- $G$  é o grafo que representa a cidade.

O grafo pode ser representado por:  $G = (V, A)$ , em que:

- $V$  é o conjunto de nós/locais;
- $A$  é o conjunto de arestas (caminhos entre dois nós).

Cada nó pode ser representado por:  $(l, l_t) \in V$ , em que:

- $l$  é o local;
- $l_t \in \{\text{ponto de coleta, posto de combustível, estação de carregamento}\}$  é o tipo de local.

Cada veículo pode ser representado por:  $v = (m, a, a_{\max}, n_{\max}, c, i, l_v, t_a)$ , em que:

- $m \in \{\text{combustão, elétrico}\}$  é o tipo de motorização;
- $a$  é a autonomia atual;
- $a_{\max}$  é a autonomia máxima;
- $n_{\max}$  é a capacidade de passageiros;
- $c$  é o custo operacional;
- $i$  é o impacto ambiental;
- $l_v \in V$  é a sua localização atual;
- $t_a$  é o tempo de reabastecimento.

Cada veículo ocupado pode ser representado por:  $v_o = (t_d, v)$ , em que:

- $t_d$  é o instante de tempo em que ficará disponível;
- $v$  é o veículo.

Cada pedido pode ser representado por:  $p = (l_o, l_d, n, s, b)$ , em que:

- $l_o \in V$  é a origem;
- $l_d \in V$  é o destino;
- $n$  é o número de passageiros;
- $s \in \mathbb{B}$  é a prioridade;
- $b \in \mathbb{B}$  é a preferência ambiental.

Cada pedido por responder pode ser representado por:  $p_r = (t_r, p)$ , em que:

- $t_r$  é o instante de tempo até o qual o pedido pode ser respondido;
- $p$  é o pedido.

**Estado inicial:** Assumindo que, inicialmente, toda a frota se encontra disponível e ainda não existem pedidos, o estado inicial do problema vai ser:  $E = (0, \emptyset, V_d, \emptyset, G)$

**Tipo de problema:** O ambiente é totalmente observável, mas devido à variação dinâmica do trânsito e à chegada espontânea de pedidos, não é determinístico, isto é, não há garantia de que uma ação tenha sempre o mesmo resultado, pelo que se trata de um problema de contingência.

**Teste objetivo:** Como este é um problema de otimização, a solução ótima é desconhecida, sendo o objetivo responder ao máximo de pedidos possíveis.

**Operadores:** Sendo  $t_e$  a função que calcula o tempo esperado entre dois nós e  $d$  a função que calcula a distância entre dois nós:

1. Enviar veículo,  $v = (m, a, a_{\max}, n_{\max}, c, i, l_v, t_a)$ , para responder ao pedido de boleia,  $p = (l_o, l_d, n, s, b)$ .

Pré-requisitos:

- O pedido ainda está por responder:  
 $(t_r, p) \in P_r$
- O veículo está disponível ou estará disponível a tempo de responder ao pedido:  
 $v \in V_d \vee ((t_d, v) \in V_o \wedge (t_d + t_e(l_v, l_o) \leq t_r))$
- O veículo tem capacidade para levar todos os passageiros:  
 $n \leq n_{\max}$
- Se o pedido tiver preferência ambiental, o veículo terá de ser elétrico:  
 $b \rightarrow (m = \text{elétrico})$
- O veículo tem autonomia suficiente para ir buscar os passageiros e fazer a viagem:  
 $a > d(l_v, l_o) + d(l_o, l_d)$

Pós-requisitos:

- Pedido é dado por respondido:  
 $p \notin P_r$
- Veículo fica indisponível até terminar pedido:  
 $v \notin V_d \wedge (t_d, v) \in V_o \wedge t_d = t_d + t_e(l_v, l_o) + t_e(l_o, l_d)$
- Autonomia do veículo diminui:  
 $a = a - (d(l_v, l_o) + d(l_o, l_d))$

2. Abastecer/carregar veículo,  $v = (m, a, a_{\max}, n_{\max}, c, i, l_v, t_a)$ , ao nó,  $(l, l_t)$ .

Pré-requisitos:

- O veículo tem autonomia suficiente para chegar até ao nó:  
 $a \geq d(l_v, l)$
- O tipo de local tem de ser apropriado para o tipo de motorização do veículo:  
 $(m = \text{elétrico} \wedge l_t = \text{estação de carregamento}) \vee (m = \text{combustão} \wedge l_t = \text{posto de combustível})$

Pós-requisitos:

- O veículo fica com autonomia máxima:  
 $a = a_{\max}$
- O veículo fica indisponível até abastecer:  
 $v \notin V_d \wedge (t_d, v) \in V_o \wedge t_d = t_d + t_e(l_v, l) + t_a$

**Custo da solução:** O custo da solução é dado pela soma dos quilómetros percorridos multiplicados pelo custo operacional estimado de cada veículo.

## 4. Implementação

### 4.1. Cidade

Para a implementação da simulação, a cidade é representada por um grafo bidirecional em que cada nó corresponde a uma localização e cada aresta, um caminho entre duas localizações, tem um peso correspondente à distância entre elas. As localizações e as distâncias entre elas não variam.

Cada localização é caracterizada pelo seu nome e tipo. Para distinguir os diferentes tipos de localização foi criado o seguinte *Enum*:

- **0** (Estacao\_Carregamento): locais onde veículos de motorização elétrica podem carregar a bateria;
- **1** (Posto\_Combustivel): locais onde veículos com motorização de combustão podem encher o depósito;
- **2** (Ponto\_Coleta): locais onde os passageiros são podem ser recolhidos ou deixados, consoante o pedido.

Cada nó tem também associada uma heurística correspondente ao trânsito mínimo nessa localização, que é calculada como sendo metade do número de vizinhos/nós adjacentes a si. De forma a adicionar alguma variação do trânsito, para além de este duplicar, durante as horas de ponta, foi também adicionado o impacto das condições meteorológicas. Antes de cada simulação, é definido, aleatoriamente, o impacto do clima para cada hora, existindo 20% de probabilidade de mau tempo, i.e., de o trânsito ser o dobro.

O grafo contém também uma *cache* em que armazena, durante cada simulação, os caminhos encontrados para evitar gastar recursos computacionais a calcular as mesmas rotas múltiplas vezes.

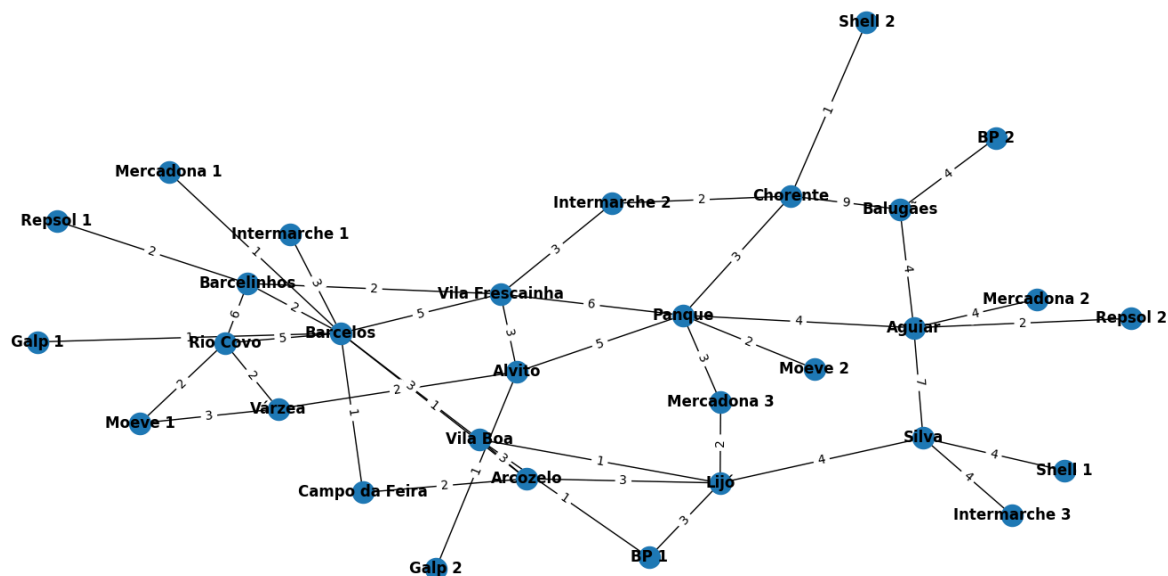


Figura 1: Grafo representante da cidade.

## 4.2. Algoritmos de procura

Para definir as rotas que os veículos vão tomar, foram implementados os seguintes algoritmos de procura:

- **Procura em Profundidade (DFS):** Avança recursivamente por um caminho (nó inicial → primeiro vizinho → primeiro vizinho do primeiro vizinho → etc.) e volta atrás para tentar outros, se não encontrar lá o nó de destino.
- **Procura em Largura (BFS):** Explora o grafo por camadas, visitando primeiro todos os vizinhos do nó inicial e, se nenhum deles for o nó de destino, explora de seguida os vizinhos destes. Encontra o caminho com menor número de arestas.
- **Procura Gulosa (Greedy):** Escolhe como o próximo nó, aquele que tiver melhor heurística local, sem considerar custos acumulados.
- **Procura A\*:** Usa a soma do custo acumulado e da heurística para escolher o próximo nó a explorar. Encontra o caminho com menor custo, se a heurística usada for válida.

## 4.3. Tempo

Para criar uma noção de tempo semelhante à de um dia, foi criada a classe *Hora*, que consiste em dois inteiros que representam as horas e os minutos, respetivamente. Esta classe simples tem um método para incrementar o tempo, que é usado ciclicamente durante a simulação para simular a passagem de tempo. Existe também um método que verifica se a simulação se encontra em hora de ponta, definidas como sendo das 7:00h às 9:00h e das 17:00h às 19:00h, usado para aumentar, tanto o trânsito em cada nó, como o volume de pedidos que são feitos, durante esses períodos.

## 4.4. Gerar dados

As simulações com os diferentes algoritmos são executadas sequencialmente e, para a comparação entre eles ser minimamente justa, são previamente gerados 15 veículos elétricos, 30 veículos a combustão e 1000 pedidos para serem usados por todas. As funções usadas para gerar estes objetos, embora tenham um fator aleatório, tomam em consideração as regras definidas no enunciado. Por exemplo:

- **Autonomia:** entre 300 e 500km (elétricos) vs. entre 500 e 800km (combustão);
- **Custos operacionais:** entre 1 e 2€/km (elétricos) vs. entre 3 e 5€/km (combustão);
- **Emissões de CO<sub>2</sub>:** entre 0 e 1g/km (elétricos) vs. entre 5 e 15g/km (combustão);
- **Tempo de reabastecimento:** entre 10 e 25 minutos (elétricos) vs. entre 1 e 2 minutos (combustão).

As outras características, comuns entre os dois tipos, variam igualmente, como a capacidade, que fica entre 1 e 7 passageiros, e a localização, que pode ser qualquer ponto de coleta da cidade.



Os pedidos são também gerados com algum critério, sendo cerca de 20% deles com preferência ambiental e outros 20% prioritários, tendo estes um tempo máximo de espera mais reduzido que os pedidos normais (entre 5 e 15 minutos vs. entre 15 e 30 minutos). É também tomado o cuidado de não gerar pedidos escusados, isto é, cuja origem e destino sejam a mesma localização.

Ao longo da realização do trabalho, de forma a ter uma noção do impacto das mudanças feitas na implementação, foram gerados veículos e pedidos que foram armazenados em dois ficheiros *CSV* (*Veiculos.csv* e *Pedidos.csv*) para poderem ser reutilizados, ao invés de testar sempre com dados diferentes. Antes de cada simulação é dada a opção de usar os dados que estão nesses ficheiros ou gerar novos, que os vão substituir.

## 4.5. Simulação

Os parâmetros iniciais dados ao simulador são os seguintes:

- uma lista com os 45 veículos, inicialmente disponíveis, da empresa;
- uma lista com os 1000 pedidos a serem feitos, ao longo do dia;
- a hora a que começa a simulação, definida para as 6:00h;
- um grafo com os locais da cidade e os caminhos entre eles;
- um número que identifica o algoritmo de procura a ser usado;
- um número que identifica o modo da simulação.

Para simular a passagem do tempo, a *thread* principal do programa funciona num ciclo em que cada iteração simula a passagem de um minuto, em que são retirados pedidos da lista fornecida e adicionados a um dicionário de pedidos por responder. Este dicionário mantém duas listas ordenadas por ordem crescente do tempo máximo de espera, uma para os pedidos prioritários e outra para os normais. O número de pedidos adicionados por minuto varia entre 1 e 4, se for hora de ponta, ou entre 0 e 1, caso contrário. Depois, é incrementada a hora, sendo por isso decrementados o tempo de indisponibilidade de todos os veículos ocupados e, também, o tempo de espera máximo de todos os pedidos por responder. Caso o tempo de espera de um pedido chegue a 0, esse pedido é removido e dado como falhado. Este ciclo repete-se até já não existirem mais pedidos para adicionar.

## 4.6. Gestor de pedidos

Para os pedidos que são adicionados pela *thread* principal, ao longo da simulação, há outra *thread*, que funciona também num ciclo e determina a resposta a cada um deles. Esta *thread* escolhe o pedido mais urgente, dando preferência aos prioritários, e percorre a lista de veículos à procura de um que ou consiga encaixar o pedido na sua rota atual ou que cumpra as seguintes condições:

- O veículo tem capacidade para levar o número de passageiros indicado no pedido;
- O veículo, esteja ocupado ou não, consegue chegar ao cliente antes de se esgotar o tempo de espera máximo;

- O veículo tem autonomia suficiente para ir buscar os passageiros, fazer a viagem e ainda sobrar para ir a um posto de abastecimento/carregamento, se necessário.

O valor de autonomia definido como “suficiente” é a soma das distâncias desde a localização atual do veículo ao local de coleta dos passageiros até ao seu destino + 20 quilómetros. Para o abastecimento/carregamento, foi definido que, se um veículo terminar um pedido com autonomia abaixo dos 50 quilómetros, ele vai de seguida ao primeiro posto de combustível/estação de carregamento que encontrar usando uma versão alterada do algoritmo a ser testado na simulação. Este algoritmo alterado, em vez de procurar o caminho entre a origem e um determinado destino, procura até uma localização de um dado tipo.

As boleias partilhadas ocorrem quando a rota de resposta a um pedido encaixa na rota atual de um veículo ocupado, se este tiver a capacidade de levar os passageiros todos. Este tipo de boleia tem prioridade sobre a boleia individual, porque não tem custos ou impacto ambiental extra. Para permitir esta funcionalidade, cada carro tem 3 listas, do mesmo tamanho, que usa para simular a travessia da sua rota. Uma lista com as localizações por onde passa, outra com o tempo que demora em cada uma delas e uma terceira com o número de passageiros que leva ao passar essa localização. O tempo que demora na sua localização atual é decrementado pela *thread* principal e, quando chega a 0, é retirado o primeiro elemento das 3 listas e atualizada a localização do veículo. O número de passageiros é 0, até chegar ao ponto de coleta dos passageiros, e volta a 0 no destino. No caso de uma boleia partilhada, o número de passageiros é simplesmente incrementado na porção da viagem partilhada.

Os pedidos com preferência ambiental não são necessariamente respondidos por veículos elétricos, no entanto, é-lhes dada prioridade na procura. Tirando proveito do facto de que a frota está organizada numa lista com os veículos elétricos primeiro, separados dos veículos a combustão, os pedidos com preferência ambiental começam a procura pelo início, enquanto que os outros começam pelo fim.

A escolha do veículo, no entanto, é feita em função do modo de simulação, que é escolhido antes das simulações se iniciarem. Cada modo, exceto o básico, visa focar a otimização do sistema em diferentes fatores, como o tempo de resposta, os custos operacionais, o impacto ambiental, etc. Os modos disponíveis são os seguintes:

- **Básico:** é escolhido o “primeiro” veículo disponível;
- **Rápido:** é escolhido o veículo que conseguir responder ao pedido no menor tempo possível;
- **Económico:** é escolhido o veículo que tenha o mínimo custo para responder ao pedido;
- **Ambiental:** é escolhido o veículo que liberte a menor quantidade de  $CO_2$ , ao responder ao pedido;

## 4.7. Estatísticas

Durante cada simulação, são atualizados valores usados para calcular estatísticas que permitem comparar o desempenho dos diferentes algoritmos e perceber quais as vantagens e desvantagens de cada um. Quando um pedido é respondido por um veículo, para além de se incrementar o número de pedidos cumpridos, é também calculado o número de quilómetros feitos sem passageiros (distância até ao ponto de recolha dos passageiros), a quantidade de emissões de  $CO_2$  e o custo da viagem. No final, são apresentadas os seguintes dados:

- Taxa de sucesso (% de pedidos que foram cumpridos);
- Número de pedidos cumpridos;
- Número de pedidos rejeitados ou falhados (total, prioritários, em hora de ponta);
- Tempo médio de resposta, em minutos;
- Pedidos com preferência ambiental respeitados (% deles que foram efetivamente respondidos por um veículo elétrico);
- Número de boleias partilhadas por clientes;
- Custos operacionais, em euros;
- Impacto ambiental, em quilogramas;
- Distância percorrida sem passageiros, em quilómetros.

## 5. Resultados

Embora seja muito difícil ter duas simulações iguais, devido à natureza aleatória da adição dos pedidos, para obter resultados que possam ser comparados, todos os modos foram testados com o mesmo *dataset*. Um fator a ter em consideração durante a análise destes resultados é que, quanto mais tarde/cedo terminar a simulação, menor/menor a densidade de pedidos adicionados ao longo desta, o que pode explicar melhorias/quedas inesperadas no desempenho.

### 5.1. Modo básico

```
Simulação terminada às 22:33h  
  
Taxa de sucesso: 88.7%  
Nº de pedidos cumpridos: 887  
Nº de pedidos rejeitados: 113  
- 54 prioritários  
- 111 em hora de ponta  
Tempo médio de resposta: 6.13 minutos  
Pedidos com preferência ambiental respeitados: 77.6%  
Boleias partilhadas: 188  
Custos operacionais: 31238.0€  
Emissões de CO2: 68.4kg  
Distância percorrida sem passageiros: 2609km
```

Figura 2: Resultados do algoritmo BFS no modo básico.

```
[12:46] Carro 44 desativado.  
[13:39] Carro 42 desativado.  
[14:04] Carro 0 desativado.  
[14:30] Carro 3 desativado.  
[14:36] Carro 39 desativado.  
[14:46] Carro 35 desativado.  
[15:05] Carro 4 desativado.  
[17:09] Carro 1 desativado.  
[20:51] Carro 38 desativado.  
[22:22] Carro 9 desativado.  
[22:46] Carro 24 desativado.  
Simulação terminada às 22:51h  
  
Taxa de sucesso: 63.8%  
Nº de pedidos cumpridos: 638  
Nº de pedidos rejeitados: 362  
- 100 prioritários  
- 304 em hora de ponta  
Tempo médio de resposta: 4.53 minutos  
Pedidos com preferência ambiental respeitados: 40.5%  
Boleias partilhadas: 182  
Custos operacionais: 41756.0€  
Emissões de CO2: 94.3kg  
Distância percorrida sem passageiros: 1860km
```

Figura 3: Resultados do algoritmo DFS no modo básico.

```
Simulação terminada às 23:15h  
  
Taxa de sucesso: 88.3%  
Nº de pedidos cumpridos: 883  
Nº de pedidos rejeitados: 117  
- 53 prioritários  
- 114 em hora de ponta  
Tempo médio de resposta: 6.06 minutos  
Pedidos com preferência ambiental respeitados: 72.7%  
Boleias partilhadas: 204  
Custos operacionais: 28016.0€  
Emissões de CO2: 62.1kg  
Distância percorrida sem passageiros: 2489km
```

Figura 4: Resultados do algoritmo A\* no modo básico.

```
Simulação terminada às 22:28h  
  
Taxa de sucesso: 86.4%  
Nº de pedidos cumpridos: 864  
Nº de pedidos rejeitados: 136  
- 49 prioritários  
- 128 em hora de ponta  
Tempo médio de resposta: 5.18 minutos  
Pedidos com preferência ambiental respeitados: 72.7%  
Boleias partilhadas: 280  
Custos operacionais: 36953.0€  
Emissões de CO2: 81.5kg  
Distância percorrida sem passageiros: 2168km
```

Figura 5: Resultados do algoritmo Greedy no modo básico.

**Sumário:** DFS obteve uma taxa de sucesso de 63.8%, enquanto que o resto dos algoritmos esteve entre 86.4% e 88.7%, e desativou 11 veículos. DFS respondeu a pedidos apenas a 40.5% dos pedidos com preferência ambiental com veículos elétricos, enquanto que o resto dos algoritmos ficou entre 72.7% e 77.6%. DFS teve o tempo de resposta médio mais rápido, 4.53 minutos, e BFS obteve mais demorado, 6.13 minutos. DFS teve os custos operacionais mais elevados, 41.756€, e A\* os menores, 28.016€. DFS emitiu a maior quantidade de CO<sub>2</sub>, 94.3 kg, e A\* o mínimo, 62.1 kg. DFS percorreu o mínimo de quilómetros sem passageiros, 1860, e BFS o máximo, 2609. DFS deu o menor número de boleias partilhadas, 182, e o algoritmo guloso o máximo, 280. Em todas as simulações, grande parte dos pedidos recusados aconteceram durante hora de ponta e menos de metade deles eram pedidos prioritários.

## 5.2. Modo rápido

```
Simulação terminada às 23:02h  
  
Taxa de sucesso: 86.4%  
Nº de pedidos cumpridos: 864  
Nº de pedidos rejeitados: 136  
- 50 prioritários  
- 131 em hora de ponta  
Tempo médio de resposta: 1.6 minutos  
Pedidos com preferência ambiental respeitados: 53.7%  
Boleias partilhadas: 194  
Custos operacionais: 23000.0€  
Emissões de CO2: 50.9kg  
Distância percorrida sem passageiros: 513km
```

Figura 6: Resultados do algoritmo BFS no modo rápido.

```
[10:03] Carro 0 desativado.  
[13:49] Carro 7 desativado.  
[14:07] Carro 2 desativado.  
[16:13] Carro 41 desativado.  
[16:38] Carro 37 desativado.  
[16:58] Carro 40 desativado.  
[17:17] Carro 12 desativado.  
[20:23] Carro 11 desativado.  
[21:04] Carro 39 desativado.  
[21:52] Carro 1 desativado.  
[22:04] Carro 9 desativado.  
[22:25] Carro 42 desativado.  
[22:51] Carro 27 desativado.  
[22:59] Carro 35 desativado.  
[23:55] Carro 6 desativado.  
Simulação terminada às 00:16h  
  
Taxa de sucesso: 63.1%  
Nº de pedidos cumpridos: 631  
Nº de pedidos rejeitados: 369  
- 106 prioritários  
- 300 em hora de ponta  
Tempo médio de resposta: 2.01 minutos  
Pedidos com preferência ambiental respeitados: 30.7%  
Boleias partilhadas: 181  
Custos operacionais: 44280.0€  
Emissões de CO2: 98.7kg  
Distância percorrida sem passageiros: 819km
```

Figura 7: Resultados do algoritmo DFS no modo rápido.

```
Simulação terminada às 23:52h  
  
Taxa de sucesso: 87.3%  
Nº de pedidos cumpridos: 873  
Nº de pedidos rejeitados: 127  
- 61 prioritários  
- 120 em hora de ponta  
Tempo médio de resposta: 1.53 minutos  
Pedidos com preferência ambiental respeitados: 51.2%  
Boleias partilhadas: 214  
Custos operacionais: 21391.0€  
Emissões de CO2: 47.8kg  
Distância percorrida sem passageiros: 599km
```

Figura 8: Resultados do algoritmo A\* no modo rápido.

```
Simulação terminada às 22:21h  
  
Taxa de sucesso: 88.7%  
Nº de pedidos cumpridos: 887  
Nº de pedidos rejeitados: 113  
- 47 prioritários  
- 106 em hora de ponta  
Tempo médio de resposta: 1.67 minutos  
Pedidos com preferência ambiental respeitados: 52.2%  
Boleias partilhadas: 362  
Custos operacionais: 31117.0€  
Emissões de CO2: 68.9kg  
Distância percorrida sem passageiros: 541km
```

Figura 9: Resultados do algoritmo Greedy no modo rápido.

**Sumário:** No modo rápido, o algoritmo que obteve o menor tempo de resposta médio foi A\*, 1.53 minutos, e DFS obteve o pior, 2.01 minutos. DFS obteve novamente a taxa de sucesso mais baixa, 63.1%, enquanto que o resto dos algoritmos esteve entre 86.4% e 88.7%, e desativou 15 veículos. DFS respondeu a 30.7% dos pedidos com preferência ambiental com veículos elétricos, enquanto que o resto dos algoritmos ficou entre 51.2% e 53.7%. DFS teve os custos operacionais mais elevados, 44.280€, e A\* os menores, 21.391€. DFS emitiu a maior quantidade de  $CO_2$ , 98.7 kg, e A\* o mínimo, 47.8 kg. BFS percorreu o mínimo de quilómetros sem passageiros, 512, e DFS o máximo, 819. DFS deu o menor número de boleias partilhadas, 181, e o algoritmo guloso o máximo, 362. Em todas as simulações, grande parte dos pedidos recusados aconteceram durante hora de ponta e menos de metade deles eram pedidos prioritários.

### 5.3. Modo económico

```
Simulação terminada às 23:56h  
  
Taxa de sucesso: 94.5%  
Nº de pedidos cumpridos: 945  
Nº de pedidos rejeitados: 55  
- 34 prioritários  
- 51 em hora de ponta  
Tempo médio de resposta: 3.81 minutos  
Pedidos com preferência ambiental respeitados: 64.4%  
Boleias partilhadas: 264  
Custos operacionais: 19046.0€  
Emissões de CO2: 32.7kg  
Distância percorrida sem passageiros: 1577km
```

Figura 10: Resultados do algoritmo BFS no modo económico.

```
[07:21] Carro 5 desativado.  
[08:04] Carro 0 desativado.  
[08:36] Carro 9 desativado.  
[11:07] Carro 3 desativado.  
[11:32] Carro 8 desativado.  
[11:51] Carro 2 desativado.  
[12:12] Carro 11 desativado.  
[13:45] Carro 12 desativado.  
[14:26] Carro 7 desativado.  
[18:11] Carro 24 desativado.  
[19:21] Carro 17 desativado.  
[19:33] Carro 4 desativado.  
[19:48] Carro 40 desativado.  
[21:23] Carro 1 desativado.  
[21:32] Carro 37 desativado.  
[23:08] Carro 35 desativado.  
Simulação terminada às 23:46h  
  
Taxa de sucesso: 68.5%  
Nº de pedidos cumpridos: 685  
Nº de pedidos rejeitados: 315  
- 87 prioritários  
- 274 em hora de ponta  
Tempo médio de resposta: 3.92 minutos  
Pedidos com preferência ambiental respeitados: 21.0%  
Boleias partilhadas: 203  
Custos operacionais: 39125.0€  
Emissões de CO2: 92.9kg  
Distância percorrida sem passageiros: 1498km
```

Figura 11: Resultados do algoritmo DFS no modo económico.



```
Simulação terminada às 22:20h  
Taxa de sucesso: 93.9%  
Nº de pedidos cumpridos: 939  
Nº de pedidos rejeitados: 61  
- 31 prioritários  
- 58 em hora de ponta  
Tempo médio de resposta: 3.79 minutos  
Pedidos com preferência ambiental respeitados: 58.5%  
Boleias partilhadas: 306  
Custos operacionais: 17112.0€  
Emissões de CO2: 32.3kg  
Distância percorrida sem passageiros: 1581km
```

Figura 12: Resultados do algoritmo A\* no modo económico.

```
Simulação terminada às 22:27h  
Taxa de sucesso: 94.1%  
Nº de pedidos cumpridos: 941  
Nº de pedidos rejeitados: 59  
- 32 prioritários  
- 55 em hora de ponta  
Tempo médio de resposta: 3.59 minutos  
Pedidos com preferência ambiental respeitados: 53.7%  
Boleias partilhadas: 413  
Custos operacionais: 22332.0€  
Emissões de CO2: 40.6kg  
Distância percorrida sem passageiros: 1402km
```

Figura 13: Resultados do algoritmo Greedy no modo económico.

**Sumário:** No modo económico, o algoritmo que obteve os menores custos operacionais foi A\*, 17.112€, e DFS obteve os piores, 39.125€. DFS obteve novamente a taxa de sucesso mais baixa, 68.5%, enquanto que o resto dos algoritmos esteve entre 93.9% e 94.5%, e desativou 16 veículos. DFS respondeu a 21% dos pedidos com preferência ambiental com veículos elétricos, enquanto que o resto dos algoritmos ficou entre 53.7% e 64.4%. DFS emitiu a maior quantidade de CO<sub>2</sub>, 92.9 kg, e A\* o mínimo, 32.3 kg. O algoritmo Greedy teve o tempo de resposta médio mais rápido, 3.59 minutos, e DFS obteve mais demorado, 3.92 minutos. Greedy percorreu o mínimo de quilómetros sem passageiros, 1402, e A\* o máximo, 1581. DFS deu o menor número de boleias partilhadas, 203, e o algoritmo guloso o máximo, 413. Em todas as simulações, grande parte dos pedidos recusados aconteceram durante hora de ponta e, exceto para DFS, mais de metade deles eram pedidos prioritários.

## 5.4. Modo ambiental

```
Simulação terminada às 22:56h  
  
Taxa de sucesso: 91.7%  
Nº de pedidos cumpridos: 917  
Nº de pedidos rejeitados: 83  
- 39 prioritários  
- 80 em hora de ponta  
Tempo médio de resposta: 5.26 minutos  
Pedidos com preferência ambiental respeitados: 61.5%  
Boleias partilhadas: 262  
Custos operacionais: 20607.0€  
Emissões de CO2: 27.7kg  
Distância percorrida sem passageiros: 2162km
```

Figura 14: Resultados do algoritmo BFS no modo ambiental.

```
[07:35] Carro 9 desativado.  
[07:44] Carro 5 desativado.  
[08:06] Carro 0 desativado.  
[10:25] Carro 4 desativado.  
[11:31] Carro 11 desativado.  
[14:50] Carro 8 desativado.  
[15:09] Carro 12 desativado.  
[16:08] Carro 3 desativado.  
[17:04] Carro 6 desativado.  
[17:30] Carro 7 desativado.  
[17:39] Carro 14 desativado.  
[17:53] Carro 1 desativado.  
[18:25] Carro 10 desativado.  
[19:44] Carro 2 desativado.  
Simulação terminada às 22:55h  
  
Taxa de sucesso: 65.4%  
Nº de pedidos cumpridos: 654  
Nº de pedidos rejeitados: 346  
- 87 prioritários  
- 289 em hora de ponta  
Tempo médio de resposta: 3.93 minutos  
Pedidos com preferência ambiental respeitados: 15.6%  
Boleias partilhadas: 227  
Custos operacionais: 38031.0€  
Emissões de CO2: 76.4kg  
Distância percorrida sem passageiros: 1254km
```

Figura 15: Resultados do algoritmo DFS no modo ambiental.

```
Simulação terminada às 23:01h  
  
Taxa de sucesso: 92.5%  
Nº de pedidos cumpridos: 925  
Nº de pedidos rejeitados: 75  
- 35 prioritários  
- 73 em hora de ponta  
Tempo médio de resposta: 4.88 minutos  
Pedidos com preferência ambiental respeitados: 58.5%  
Boleias partilhadas: 294  
Custos operacionais: 18203.0€  
Emissões de CO2: 26.7kg  
Distância percorrida sem passageiros: 2015km
```

Figura 16: Resultados do algoritmo A\* no modo ambiental.

```
Simulação terminada às 01:09h  
  
Taxa de sucesso: 93.5%  
Nº de pedidos cumpridos: 935  
Nº de pedidos rejeitados: 65  
- 35 prioritários  
- 57 em hora de ponta  
Tempo médio de resposta: 4.43 minutos  
Pedidos com preferência ambiental respeitados: 60.0%  
Boleias partilhadas: 360  
Custos operacionais: 24675.0€  
Emissões de CO2: 34.0kg  
Distância percorrida sem passageiros: 1924km
```

Figura 17: Resultados do algoritmo Greedy no modo ambiental.

**Sumário:** No modo ambiental, o algoritmo que emitiu a menor quantidade de  $CO_2$  foi A\*, 26.7 kg, e DFS emitiu o máximo, 76.4 kg. DFS obteve novamente a taxa de sucesso mais baixa, 65.4%, enquanto que o resto dos algoritmos esteve entre 91.7% e 93.5%, e desativou 14 veículos. DFS respondeu a 15.6% dos pedidos com preferência ambiental com veículos elétricos, enquanto que o resto dos algoritmos ficou entre 58.5% e 61.5%. DFS teve os custos operacionais mais elevados, 38.031€, e BFS os menores, 20.607€. DFS teve o tempo de resposta médio mais rápido, 3.93 minutos, e BFS obteve mais demorado, 5.26 minutos. DFS percorreu o mínimo de quilómetros sem passageiros, 1254, e BFS o máximo, 2162. DFS deu o menor número de boleias partilhadas, 227, e o algoritmo guloso o máximo, 360. Em todas as simulações, grande parte dos pedidos recusados aconteceram durante hora de ponta e, exceto para DFS, perto de metade deles eram pedidos prioritários.

## 6. Discussão

Nos resultados obtidos, existe uma óbvia diferença entre o desempenho do algoritmo de procura em profundidade (DFS) e o desempenho do resto. Para além de resultar numa taxa de sucesso até 28.1% inferior às outras, com o número de pedidos rejeitados sempre acima dos 30%, entre todas as simulações feitas, as que usam este algoritmo são as únicas em que veículos têm que ser desativados. Nestas simulações, são desativados, em média, 14 veículos, porque não são encontradas estações de carregamento/postos de abastecimento a uma distância alcançável com a autonomia que lhes resta. As soluções deste algoritmo mostram-se também muito custosas, pelo facto de que, mesmo sendo as simulações em que menos pedidos são respondidos, logo são feitas menos viagens, apresentam valores elevados de custos operacionais, emissões de  $CO_2$  e quilómetros percorridos sem passageiros. Nos modos especializados, DFS obtém sempre o pior resultado no parâmetro em que a simulação está a focar a otimização. Estes resultados destacam a desadequação do algoritmo para problemas de otimização, em que o custo/distância das soluções é importante.

Os restantes algoritmos apresentam todas taxas de sucesso muito superiores e próximas umas das outras. Embora não tenha exatamente a maior taxa de sucesso, o algoritmo de procura  $A^*$  prova-se o melhor para problemas de otimização, apresentando em todos os modos especializados os menores valores para os parâmetros específicos, sem comprometer outros critérios. Apesar de não usar heurísticas ou custos acumulados, BFS tem resultados surpreendentemente bons e estáveis e apresenta resultados próximos de  $A^*$  nos parâmetros a otimizar, talvez pela diferença dos custos das arestas do grafo não ser muito grande, mas desleixa-se por vezes com os outros critérios. O algoritmo Greedy, apesar de não garantir a solução ótima, mostra-se mais eficiente em termos de tempo de resposta e maximização de boleias partilhadas, mas, por ignorar custos acumulados, conduz a resultados abaixo de um algoritmo não-informado, como BFS, nos parâmetros que deviam ser otimizados.