

Health Planet - Problema de Procura

INTELIGÊNCIA ARTIFICIAL

DIOGO PINTO A100551

JOÃO MAGALHÃES A100740

JORGE RODRIGUES A101758

MARIANA PINTO A100756

RODRIGO GOMES A100555

A100551 Diogo Pinto Delta = 0

A100740 João Magalhães Delta = 0

A101758 Jorge Rodrigues Delta = 0

A100756 Mariana Pinto Delta = 0

A100555 Rodrigo Gomes Delta = 0

Índice

Introdução.....	1
O Problema	2
Formulação do Problema.....	3
Representação do Estado	3
Representação do Estado Inicial	4
Representação do estado objetivo	5
Operadores	6
Custo da solução	6
Outras abordagens.....	7
Tarefas	8
Circuitos	8
Representar pontos de entrega	10
Estratégias de Procura	11
Resultados.....	12
Procura com ações não determinísticas	20
Extra: O mTSP: Como se liga ao problema de procura e possível solução	21
Conclusões	23
Bibliografia	24

Introdução

A navegação é um problema no qual o homem procura tornar mais eficiente desde o começo dos tempos. Desde a construção de rotas marítimas e terrestres, o comércio sempre requisitou aos comerciantes que, para conseguir lucrar, devem ser eficientes no que toca ao custo das suas travessias. Nos dias de hoje, as estradas estão muito bem delimitadas e seguem diferentes planeamentos que, por vezes, ultrapassam as centenas de anos. Também existem muitas mais possibilidades para os viajantes, surgindo diversas rotas com diversas características diferentes. Comparativamente à antiguidade, o problema de encontrar a melhor rota prevalece, mas, de forma diferente: Antigamente sofria-se por falta de escolha, hoje em dia sofre-se pelo crescimento das possibilidades, diminuindo a satisfação com a escolha feita.

Com as tecnologias atuais, nomeadamente a Inteligência Artificial, é facilmente concebida uma ferramenta de navegação exímia que pode ter diversos aspetos em consideração. No presente trabalho iremos desenvolver um modelo que leva em consideração requisitos ambientais a pedido da *Health Planet*, uma transportadora que procura sucesso no mercado atual. Para esse efeito, vamos cobrir as diferentes etapas da concessão, desde a formulação do problema e a representação de estado, até aos algoritmos concebidos e os respetivos resultados e conclusões. Vamos ainda apresentar algum conteúdo relativo à vida real, onde os cenários não costumam ser determinísticos e muito menos estáticos. Por fim, iremos expor um outro problema importante associado ao uso de rotas, indispensável para o negócio da transportação, juntamente com a aplicação de um algoritmo para resolver essa questão.

O Problema

A primeira etapa da resolução é perceber exatamente do que se trata o problema. Sem esta percepção, será impossível de avançar com a formulação do mesmo e as restantes componentes. Como referido no enunciado, a *Health Planet* é uma distribuidora e sabemos as seguintes informações:

- Existem diversos estafetas e meios de transportes diferentes.
- As entregas têm um peso e estão atribuídas a um estafeta.
- Um estafeta está apenas associado cidade/freguesia e possuem uma avaliação.
- Clientes podem requisitar o tempo máximo que estão dispostos a esperar pela entrega.
- A empresa tem como objetivo minimizar as suas emissões de CO₂.
- Cada veículo possui uma velocidade média, assim como um limite de peso e uma penalização na velocidade dependendo do volume carregado.

Com base nestes pontos, percebemos que o objetivo é encontrar formas de calcular trajetos eficientes para cada estafeta, de modo a que a entrega seja garantida e o tempo de entrega minimizado. Outras considerações importantes que retiramos são que, muito provavelmente, o problema que nós temos é apenas uma parte do que envolve uma distribuidora, onde seriam associados outros problemas, como por exemplo, o mTSP (multiple Travelling Salesman Problem), a parte económica de controlo de despesas e ainda a questão de transcrever as cidades para a abstração desenvolvida. Deste modo, teremos em consideração todos estes aspetos para as restantes fases do projeto.

Formulação do Problema

A formulação formal de um problema é fulcral para encontrar uma solução. Nesta etapa é esperado que o problema da distribuição sustentável seja formalizado de forma a responder aos seguintes 5 componentes: Representação do Estado; Estado Inicial; Estado Objetivo; Operadores; Custo da Solução. Nesta etapa, é expectável encontrar uma abstração capaz de compreender o problema em mãos sob a forma de um problema de procura.

Representação do Estado

Considerando o problema descrito, facilmente podemos identificar o seu tipo como um problema de estado único, dado que no momento antes da entrega começar, o ator, que neste contexto, é o estafeta, tem acesso ao ambiente, ou seja, conhece as diferentes ruas que pode ou não seguir para realizar a entrega e também sabe de onde vai partir, que será um posto da “*Health Planet*”. Além disso, a posição do estafeta é apenas alterada caso este concretize uma operação, daí ser determinístico. Contudo, podemos também considerar que, em certas situações, onde, por exemplo, uma rua esteja cortada por causa de obras ou por causa de um acidente, seja preciso calcular uma nova solução, estando assim perante um problema de contingência.

Resta agora definir como iremos representar o problema. Tendo em conta a descrição anterior, a representação escolhida será um grafo direcionado onde os nós representam pontos de interesse com coordenadas, sendo divididos em tipos para ficar mais expressivo graficamente.

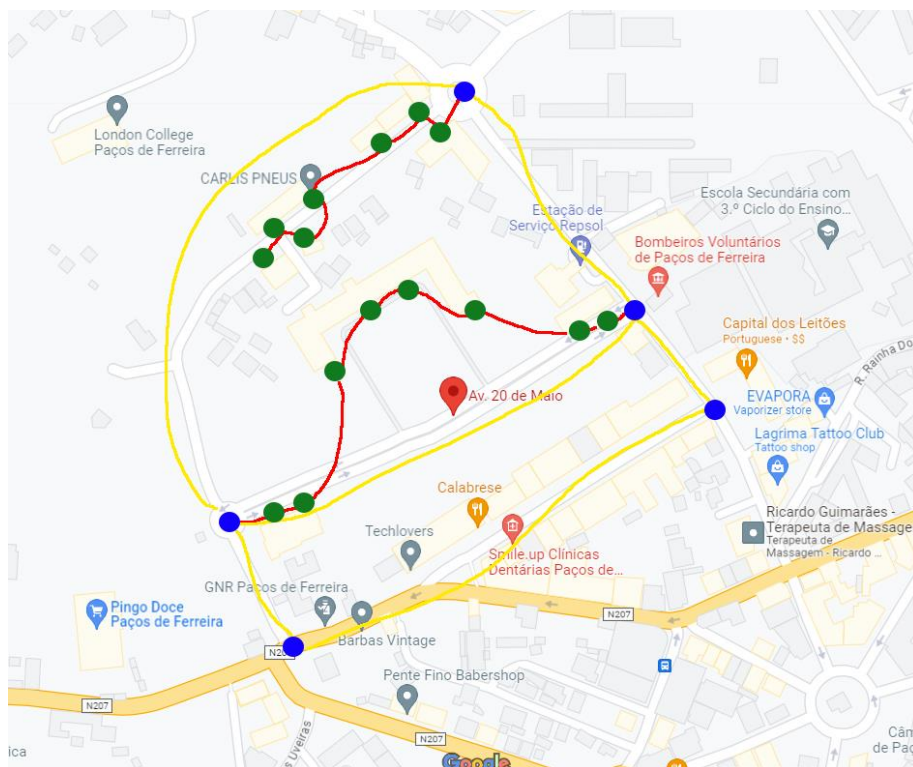


Figura 1- Exemplo de Grafo

Na figura em cima, está representado um exemplo do que será a nossa abstração aplicada a um bairro de uma cidade. Relativamente aos nodos, encontramos essencialmente dois tipos: os nodos azuis que representam as extremidades das ruas, cruzamentos e rotundas e os nodos verdes que representam os diferentes pontos de entrega possíveis numa determinada rua. Além dos nodos, existem também as arestas que, tal como os nodos, são divididas em dois tipos: as arestas amarelas que conectam os nodos azuis e representam as diferentes ruas de uma cidade; e as arestas vermelhas que representam as ligações entre os pontos de entrega de uma determinada rua. Com esta abstração, podemos assegurar um serviço funcional para todo o tipo de cidades com a possibilidade da utilização de algoritmos de procura para encontrar trajetos de entrega.

Resta agora atribuir as componentes que vão ser usadas futuramente para o cálculo da solução. Estas componentes vão estar associadas aos arcos do grafo e serão capazes de caracterizar uma determinada rua e, são estas: a distância, o coeficiente de congestão e o coeficiente de emissões. A componente distância será um valor numérico expresso em quilómetros, que servirá para calcular as emissões feitas, o tempo percorrido e o desgaste das viaturas. A congestão será um indicador para estimar um atraso em média que um estafeta terá em transitar na via, e o coeficiente das emissões expressa a quantidade extra de emissões de dióxido de carbono em gramas feitas por transitar numa via.

Representação do Estado Inicial

O estado inicial, no contexto do problema, maioritariamente será sempre algo simples. Em todos os estados iniciais do problema, encontraremos o nosso ator sempre num dado nodo, sendo isto importante porque, neste tipo de serviço, é conveniente existir um plano de contingência para problemas nas ruas. Se decidirmos ignorar as situações excecionais, o nosso estado inicial será sempre um posto onde o estafeta levanta uma ou mais encomendas para realizar a entrega. Nos casos excecionais, inicialmente partimos de uma estação, como nos casos descritos anteriores, porém, podemos ter um novo estado inicial caso seja preciso recalcular o trajeto.

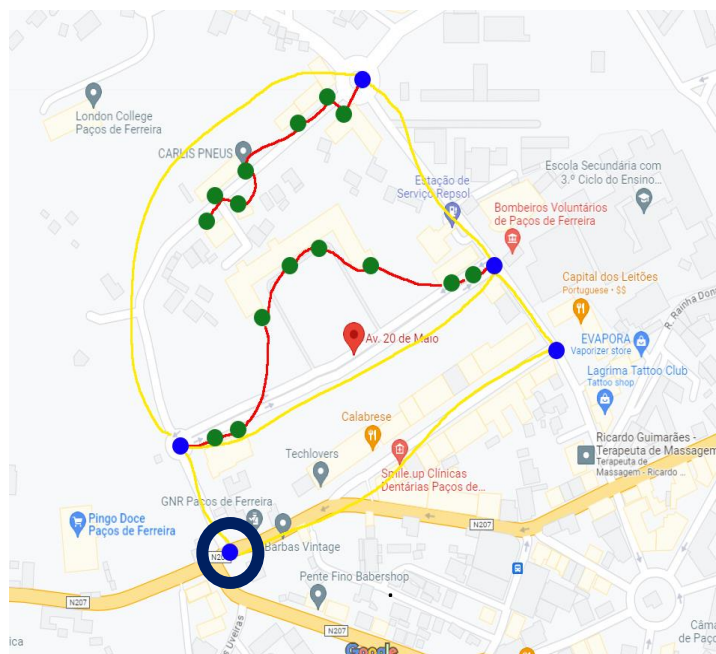


Figura 2- Estado Inicial no Nodo "A"

Representação do estado objetivo

O estado objetivo, contrariamente ao estado inicial, será sempre o mesmo num determinado serviço, independentemente do que acontece durante uma entrega. Considerando uma entrega, o estado objetivo será quando o ator se encontra no nodo representante do número da porta de entrega.

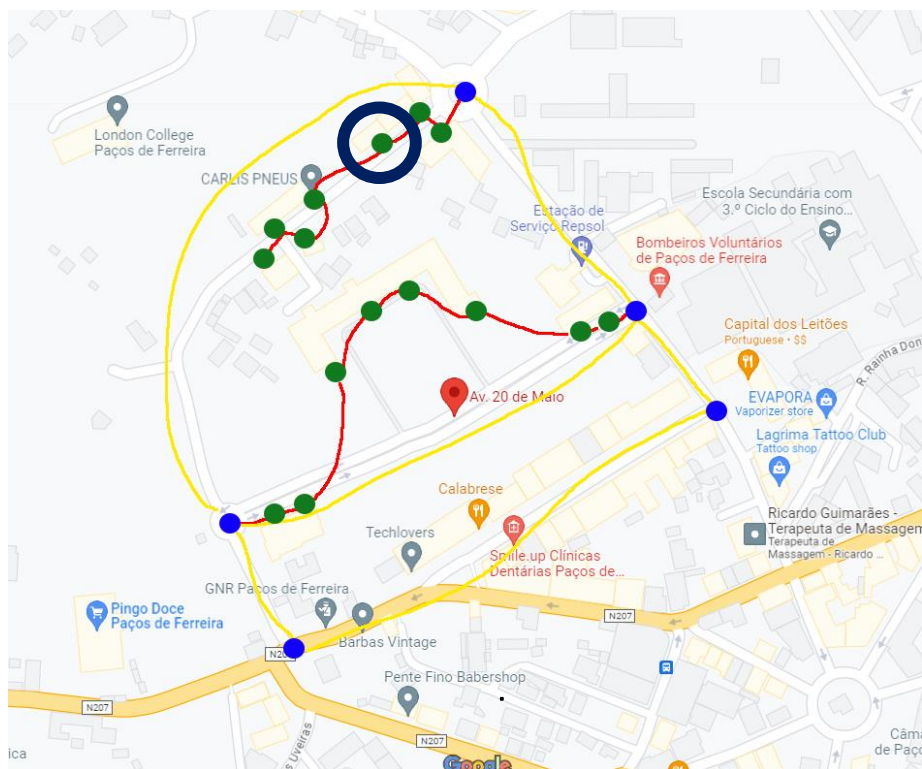


Figura 3- Estado Inicial no Nodo “B”

Operadores

O operador disponível para o problema vai ser dependente do estado atual do ator, mas, resume-se a trocar de nodo para nodo. Esta ação é independente do tipo do nodo, por isso, para simplificar, deveremos apenas representar os nodos dos números das portas nas ruas que importam.

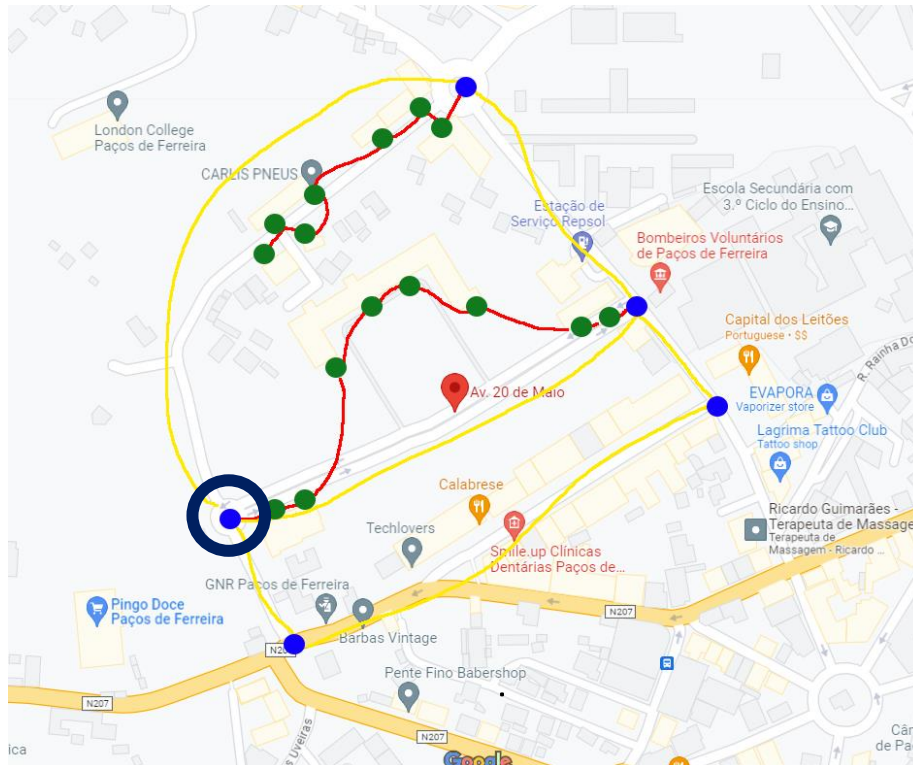


Figura 4- Ator encontra-se no nodo “D”

Na figura 4 temos um estado onde o ator se encontra no nodo “D”. Nesta situação, os atores têm 4 possibilidades de movimentação, que correspondem aos nodos adjacentes do próprio.

Cada operação terá um custo que se divide em três componentes: a distância, o tempo e as emissões. Por operação existe uma distância, que corresponde à distância física entre os dois pontos e duas taxas que, baseadas nesta distância, calculam penalizações temporais (representa o trânsito) e as emissões (representam mais ou menos esforço do motor). Todos estes aspetos são relevantes para o custo da solução final.

Custo da solução

A determinação do custo da solução é um elemento fundamental na avaliação da eficiência e do impacto ambiental das estratégias de entrega no contexto da *Health Planet*. A nossa estratégia visa calcular o custo total da pegada ecológica, a distância percorrida e o tempo das entregas.

No caso da nossa distribuidora, a componente de tempo e a da pegada são as mais importantes. Relativamente ao tempo, apenas temos que garantir que a entrega é feita antes do prazo. Em contrapartida, a pegada é suposta ser a mínima possível.

Outras abordagens

Antes de mais, deixamos esta pequena secção para referir que existem outras formas de formular o problema. Uma outra abordagem que foi considerada pelo grupo resumia-se a representar cada nó como uma rua em si, onde os nós sucessores seriam as ruas que teríamos acesso. O que levou à escolha da abstração que irá ser exposta neste documento foram os seguintes fatores:

- Representação espacial, visto ser mais fácil atribuir uma coordenada a um ponto só do que a um conjunto de pontos, garantindo precisão às nossas soluções.
- Pode ser otimizado, isto é, com o tempo os grafos podem ser podados, onde representamos, por exemplo, apenas as portas das ruas alvo.
- Representação mais humana das cidades, quase que é possível avaliar o planeamento urbano de uma determinada localidade.
- Granularidade, pois esta solução mostra muito detalhe e opções com as procuras.
- Semelhante a um GPS, uma representação conhecida de todos nós que pode ser utilizada de diversas formas

Contudo, nem tudo é positivo e existem alguns problemas com a abstração, como por exemplo, os seguintes:

- Complexidade, ou seja, é difícil e trabalhoso representar cidades muito grandes com muitas interseções e ruas.
- Difícil de manter redes dinâmicas grandes, logo torna-se mais difícil de atualizar corretamente as ruas.
- A performance dos algoritmos de procura pode ser prejudicada.

Tarefas

Nesta secção vamos desenvolver à cerca de todas as tarefas propostas para os alunos. A primeira tarefa (Formulação do Problema) já está descrita no ponto anterior, pois é determinante para a forma como se resolve as outras etapas.

Circuitos

Uma das partes mais difíceis de codificar foram os circuitos. Para não depender só de ferramentas externas, decidimos criar também as nossas metodologias para gerar grafos para testes. Esta questão não foi tão fácil assim, visto que existem muitas considerações que fogem às nossas competências, por envolverem questões de urbanismo. Por isso, inicialmente começamos com uma abordagem que gerava grafos totalmente aleatórios que acabavam por quebrar as regras da abstração, e, rapidamente foram descartados.

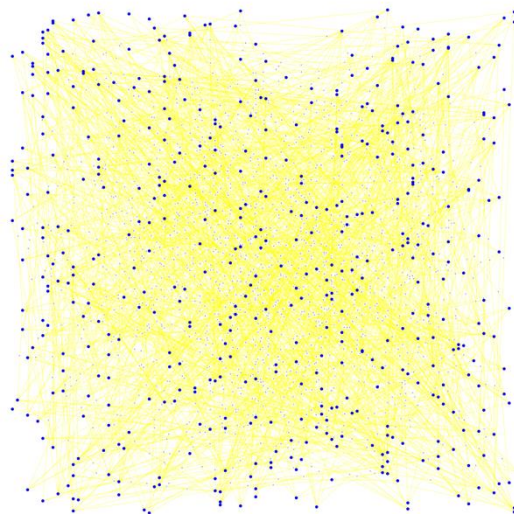


Figura 1 Grafo Aleatório

As próximas formas de construção de grafo escolhidas são bastante mais parecidas com casos reais. A segunda abordagem foi de um grafo que representa o famoso “*grid system*”, utilizado em diversas cidades espalhadas pelo mundo, como por exemplo, Nova York, São Francisco, Atenas, que aparecem numa grelha muito perfeita, e algumas variantes como em Valência e Belo Horizonte.



Figura 3 Vista Satélite de Atenas-Grécia

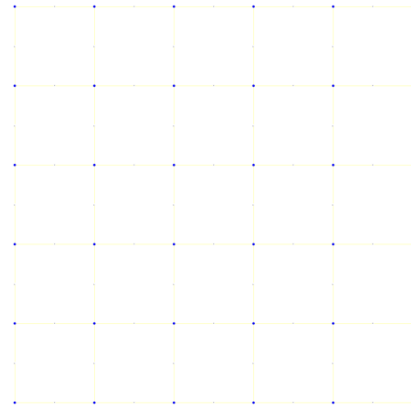


Figura 2 Grafo "grid system"

Por fim, precisávamos de uma representação mais desorganizada em que fosse possível representar uma vasta quantidade de cidades que, graças à sua antiguidade, não foram propriamente planeadas. Para isso, modificámos a função descrita em cima para tornar imperfeita a representação, onde surgem estruturas parecidas com rotundas e ruas com dimensões diferentes.

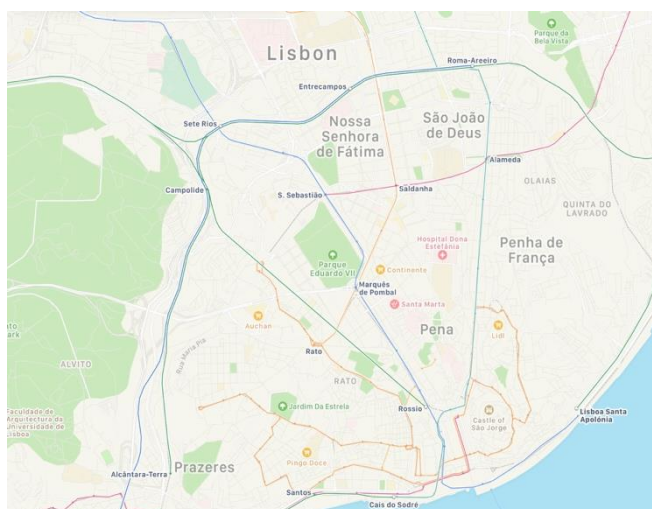


Figura 5 Parte do Mapa de Lisboa

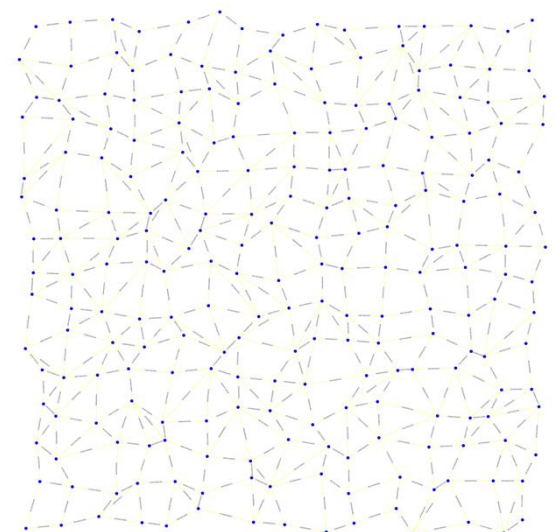


Figura 4 Grafo irregular

Com estas variantes podemos testar os algoritmos desenvolvidos em cenários diferentes e representativos de cidades reais. Os grafos aleatórios foram também usados para testes para ajudar o grupo a compreender o impacto que as heurísticas têm na busca de soluções, tendo sido essa uma das razões que nos levaram a encontrar e a explorar os circuitos de formas diferentes. Nenhum dos circuitos exibidos possui “Portas” representadas, pois são apenas exemplos para explicar os circuitos usados. Esta questão não influencia em nada os algoritmos de procura (desde que o alvo também não seja uma “Porta”).

Depois de desenvolver estas funcionalidades, decidimos recorrer à biblioteca “osmnx”, que se baseia no “OpenStreetMap” e permite visualizar as diferentes redes urbanas com imensa precisão. Com muita facilidade, conseguimos transferir esta informação para o restante código, tendo apenas que criar um valor de congestão e de pegada (porque esta informação não está disponível).

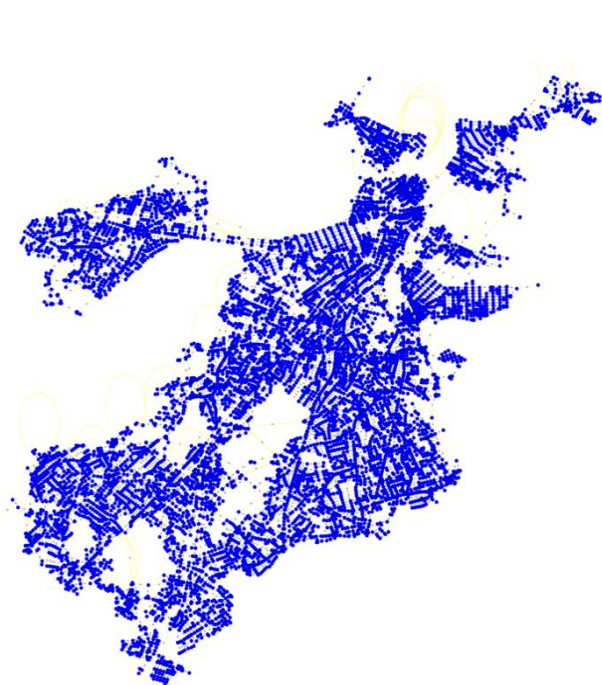


Figura 7 Boston, Massachusetts, EUA

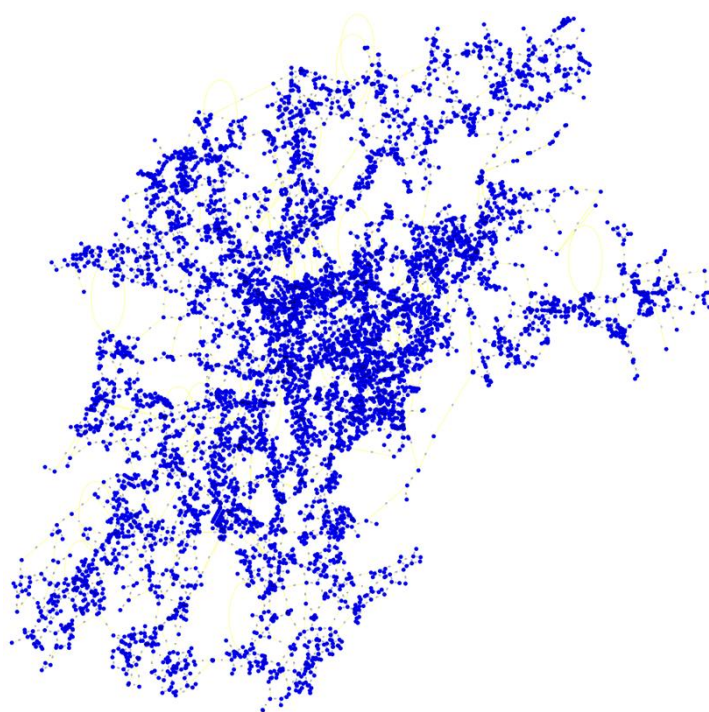


Figura 6 Braga, Portugal

Representar pontos de entrega

Esta tarefa já foi referida na formulação do problema, mas podemos utilizar este espaço para reforçar as vantagens de apenas representar as portas onde as entregas devem ser entregues. Ao evitar portas que não vão ser usadas, podemos reduzir significativamente o número de nodos e de arestas do grafo. O ideal para um sistema deste género é manter uma base de dados com todos os nós (Rua e Porta) que, quando necessário, são colocados no grafo.

Estratégias de Procura

Agora que temos uma forma de representar uma cidade, resta-nos executar as procuras sobre o grafo para encontrar as soluções. As procuras não são novidade, ou seja, já existem diversos algoritmos conhecidos com características diferentes. Apesar de, antes de mais, ser possível antecipar o desempenho dos diferentes algoritmos, decidimos implementar algoritmos variados para entender quais compensam e em que sentido. Nesta secção, vamos apresentar os algoritmos para depois discutir os seus resultados

DFS (procura em profundidade) -> Este é um algoritmo conhecidíssimo para encontrar soluções em grafos finitos que se destaca pela sua simplicidade na parte da programação e na sua logística. Daqui não esperamos grandes soluções, mas admitimos que possa causar alguma surpresa, pois a sua implementação gera alguma imprevisibilidade. Na parte da implementação, optamos por não seguir a forma mais conhecida e também fornecida nas aulas que usa recursão, pois o projeto foi desenvolvido em *python* que tem um limite bastante pequeno de chamadas recursivas. Por isso, recorreremos a uma alternativa que utiliza uma *stack* auxiliar que simula a recursão, o que já nos permite correr em grafos maiores.

BFS (procura em largura) -> Este é um algoritmo antigo que gera resultados relativamente bons. Graças à sua implementação, conseguimos soluções que usam poucos saltos entre nós, o que costuma indicar uma solução razoável. Este algoritmo forma a sua expansão recorrendo a uma fila de espera, o que pode ser custoso em termos de tempo e espaço. A implementação usada não difere da usada nas aulas (salvo a parte de guardar a expansão), pois é uma procura não informada.

Iterativa -> O conceito deste algoritmo é simples: Fazer procura em profundidade avançando apenas até um valor que é incrementado iterativamente. Para isso usamos a procura DFS alterada para terminar quando alcança uma determinada profundidade. De modo geral, este algoritmo é uma melhoria para a procura em profundidade, ficando relativamente semelhante à procura em largura. Uma das grandes vantagens deste algoritmo é o uso do seu espaço, sendo este consistentemente baixo.

Custo uniforme -> Esta é a última procura não informada que desenvolvemos e é, de longe, a favorita, pois consegue resultados ótimos. Tem um funcionamento similar ao da largura, mas procura sempre expandir nós cujo o ganho é melhor. Esta simples alteração leva a soluções excelentes. Relativamente à implementação, é preciso ter em atenção que as nossas arestas possuem 2 valores, sendo necessário atribuir mais ou menos peso a cada um desses valores. No nosso caso, achamos que atribuir mais peso à pegada do que ao tempo melhorou a qualidade da nossa solução. De modo geral, esta costuma ser a procura preferencial (a única desvantagem costuma ser o espaço requerido).

Gulosa Primeira pesquisa informada desenvolvida e de longe a mais simples. A gulosa expande sempre o nó cuja heurística é mais apetecível, o que pode resultar em poucas expansões e numa estranha semelhança (apesar de muito melhor) à procura em profundidade. Não é esperado que os resultados sejam maus e a nível de implementação é apenas necessário ter cuidado com a possibilidade de ciclos.

A* Uma boa candidata ao lugar de melhor procura, dado que combina a noção de ganho explorado em cima na procura em custo uniforme com a heurística, sendo que converge muito rapidamente para uma solução perto do ótimo. A implementação é parecida com a da gulosa, mas agora introduzimos à escolha o ganho. Com uma boa heurística, esperamos que as soluções sejam bastante perto do ótimo e da procura custo uniforme.

Por fim, realçar o motivo de não termos explorado com a procura Bidirecional. Esta técnica é muito difícil de aplicar em grafos que representam cidades, pois seriam excluídos caminhos que não pudessem ser feitos nos dois sentidos, o que não é realista. Pela mesma lógica, calcular a heurística também não dá para ser feito uma só vez, pois não conseguimos inverter o trajeto de modo a que o fim seja sempre o posto. Não há garantias que o trajeto inverso seja o melhor.

Resultados

No âmbito deste projeto, dedicamo-nos à avaliação de algoritmos de busca em diferentes tipos de grafos, variando não apenas em tamanho, mas também em forma, e representando, assim, diferentes desafios para os métodos de procura. Focando em métricas cruciais ao tema proposto, tais como distância percorrida, tempo de entrega e pegada de CO₂ associada, bem como número de nós expandidos e a quantidade de nós pertencentes à solução, pretendemos oferecer uma visão abrangente e criteriosa sobre o desempenho desses algoritmos em cenários variados.

Notar que compreender como os algoritmos de procura respondem a diferentes topologias de grafos é essencial para otimizar a resolução de problemas específicos e, por isso, a nossa abordagem visa avaliar a eficiência no uso de recursos, tempo e espaço.

Com os testes realizados esperamos apresentar uma análise capaz de orientar a escolha adequada de algoritmos em diferentes contextos. Seguem-se os testes realizados, acompanhados dos resultados obtidos:

Algoritmo / Teste	Origem: Node_120 Destino: Node_144 Veículo: Bicicleta Peso: 2Kg	Origem: Node_197 Destino: Node_45 Veículo: Mota Peso: 4Kg	Origem: Node_77 Destino: Node_73 Veículo: Carro Peso: 72Kg	Origem: Node_164 Destino: Node_61 Veículo: Bicicleta Peso: 5Kg	Origem: Node_182 Destino: Node_190 Veículo: Mota Peso: 17Kg	Origem: Node_142 Destino: Node_55 Veículo: Carro Peso: 76Kg
DFS	Distância(Km): 104.69 Tempo: 17.48 Emissão CO2 (g): 0.0 Nodos expandidos: 23 Nodos solução: 23	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 32 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 26 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 10 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 4 Nodos solução: 4	Distância(Km): 103.10 Tempo: 3.51 Emissão CO2 (g): 18564 Nodos expandidos: 22 Nodos solução: 22
BFS	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 139 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 196 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 182 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 111 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 20 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10458 Nodos expandidos: 110 Nodos solução: 15
A*	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 14 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 77 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 85 Nodos solução: 26	Distância(Km): 63.29 Tempo: 13.67 Emissão CO2 (g): 0.0 Nodos expandidos: 37 Nodos solução: 13	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 14 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10458 Nodos expandidos: 15 Nodos solução: 15
Gulosa	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 14 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 32 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 26 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 10 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 4 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10458 Nodos expandidos: 15 Nodos solução: 15
Custo Uniforme	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 105 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 195 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 185 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 119 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 42 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10458 Nodos expandidos: 86 Nodos solução: 15
Iterativa	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 14 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13426 Nodos expandidos: 32 Nodos solução: 32	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23270 Nodos expandidos: 26 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 10 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 4 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10458 Nodos expandidos: 15 Nodos solução: 15

Figura 8 - Testes aleatórios em grafo pequeno e aleatório

Algoritmo / Teste	Origem: Node_2487 Destino: Node_4128 Veículo: Bicicleta Peso: 4Kg	Origem: Node_21 Destino: Node_7135 Veículo: Mota Peso: 8Kg	Origem: Node_5735 Destino: Node_3442 Veículo: Carro Peso: 44Kg	Origem: Node_4276 Destino: Node_366 Veículo: Bicicleta Peso: 1Kg	Origem: Node_7452 Destino: Node_7222 Veículo: Mota Peso: 10Kg	Origem: Node_3790 Destino: Node_611 Veículo: Carro Peso: 91Kg
DFS	Distância(Km): 73565.79 Tempo: 14009.02 Emissão CO2 (g): 0.0 Nodos expandidos: 5507 Nodos solução: 5507	Distância(Km): 22366.99 Tempo: 2032.53 Emissão CO2 (g): 1964239 Nodos expandidos: 1699 Nodos solução: 1699	Distância(Km): 41318.0 Tempo: 1314.05 Emissão CO2 (g): 7407416 Nodos expandidos: 3015 Nodos solução: 3015	Distância(Km): 37218.49 Tempo: 5738.95 Emissão CO2 (g): 0.0 Nodos expandidos: 2815 Nodos solução: 2815	Distância(Km): 75501.5 Tempo: 7295.9 Emissão CO2 (g): 6621028 Nodos expandidos: 5635 Nodos solução: 5635	Distância(Km): 68247.3 Tempo: 2420.66 Emissão CO2 (g): 12378610 Nodos expandidos: 5071 Nodos solução: 5071
BFS	Distância(Km): 83.4 Tempo: 16.01 Emissão CO2 (g): 0.0 Nodos expandidos: 411 Nodos solução: 6	Distância(Km): 11.6 Tempo: 1.03 Emissão CO2 (g): 992.73 Nodos expandidos: 1664 Nodos solução: 6	Distância(Km): 50.0 Tempo: 1.59 Emissão CO2 (g): 8962.20 Nodos expandidos: 4294 Nodos solução: 7	Distância(Km): 116.1 Tempo: 17.99 Emissão CO2 (g): 0.0 Nodos expandidos: 7729 Nodos solução: 8	Distância(Km): 129.3 Tempo: 12.87 Emissão CO2 (g): 11677.40 Nodos expandidos: 5777 Nodos solução: 8	Distância(Km): 112.70 Tempo: 4.05 Emissão CO2 (g): 20719.2 Nodos expandidos: 9250 Nodos solução: 8
A*	Distância(Km): 192.2 Tempo: 37.54 Emissão CO2 (g): 0.0 Nodos expandidos: 4428 Nodos solução: 20	Distância(Km): 173.79 Tempo: 15.96 Emissão CO2 (g): 15422.53 Nodos expandidos: 1876 Nodos solução: 19	Distância(Km): 471.4 Tempo: 15.1 Emissão CO2 (g): 85097.2 Nodos expandidos: 1176 Nodos solução: 30	Distância(Km): 391.7 Tempo: 59.6 Emissão CO2 (g): 0.0 Nodos expandidos: 1092 Nodos solução: 32	Distância(Km): 479.3 Tempo: 47.08 Emissão CO2 (g): 42725 Nodos expandidos: 387 Nodos solução: 42	Distância(Km): 152.9 Tempo: 5.15 Emissão CO2 (g): 26326.6 Nodos expandidos: 4796 Nodos solução: 17
Gulosa	Distância(Km): 793.4 Tempo: 154.14 Emissão CO2 (g): 0.0 Nodos expandidos: 4436 Nodos solução: 57	Distância(Km): 367.19 Tempo: 33.22 Emissão CO2 (g): 32102.1 Nodos expandidos: 1863 Nodos solução: 33	Distância(Km): 471.4 Tempo: 15.1 Emissão CO2 (g): 85097.2 Nodos expandidos: 1279 Nodos solução: 30	Distância(Km): 847.5 Tempo: 130.47 Emissão CO2 (g): 0.0 Nodos expandidos: 1165 Nodos solução: 65	Distância(Km): 576.09 Tempo: 56.07 Emissão CO2 (g): 50883 Nodos expandidos: 346 Nodos solução: 47	Distância(Km): 483.09 Tempo: 16.82 Emissão CO2 (g): 85999.8 Nodos expandidos: 4809 Nodos solução: 40
Custo Uniforme	Distância(Km): 48.0 Tempo: 9.78 Emissão CO2 (g): 0.0 Nodos expandidos: 5566 Nodos solução: 12	Distância(Km): 11.6 Tempo: 1.03 Emissão CO2 (g): 992.73 Nodos expandidos: 10 Nodos solução: 6	Distância(Km): 50.0 Tempo: 1.59 Emissão CO2 (g): 8962.20 Nodos expandidos: 3006 Nodos solução: 7	Distância(Km): 63.9 Tempo: 10.19 Emissão CO2 (g): 0.0 Nodos expandidos: 8959 Nodos solução: 10	Distância(Km): 35.1 Tempo: 3.63 Emissão CO2 (g): 3291 Nodos expandidos: 326 Nodos solução: 9	Distância(Km): 62.1 Tempo: 1.95 Emissão CO2 (g): 9972.51 Nodos expandidos: 6727 Nodos solução: 11
Iterativa	Distância(Km): 83.4 Tempo: 16.01 Emissão CO2 (g): 0.0 Nodos expandidos: 6 Nodos solução: 6	Distância(Km): 11.6 Tempo: 1.03 Emissão CO2 (g): 992.73 Nodos expandidos: 6 Nodos solução: 6	Distância(Km): 83.0 Tempo: 2.53 Emissão CO2 (g): 14249.05 Nodos expandidos: 8 Nodos solução: 8	Distância(Km): 144.1 Tempo: 22.0 Emissão CO2 (g): 0.0 Nodos expandidos: 11 Nodos solução: 11	Distância(Km): 120.49 Tempo: 10.96 Emissão CO2 (g): 9949.1 Nodos expandidos: 8 Nodos solução: 8	Distância(Km): 99.3 Tempo: 3.54 Emissão CO2 (g): 18079.71 Nodos expandidos: 8 Nodos solução: 8

Figura 9 - Testes aleatórios em grafo grande e aleatório

Algoritmo / Teste	Origem: Node_11_63 Destino: Node_95_30 Veículo: Bicicleta Peso: 1Kg	Origem: Node_61_88 Destino: Node_52_68 Veículo: Moto Peso: 14Kg	Origem: Node_61_24 Destino: Node_24_8 Veículo: Carro Peso: 68Kg	Origem: Node_60_92 Destino: Node_42_86 Veículo: Bicicleta Peso: 4Kg	Origem: Node_65_32 Destino: Node_54_77 Veículo: Moto Peso: 11Kg	Origem: Node_25_42 Destino: Node_64_17 Veículo: Carro Peso: 93Kg
DFS	Distância(Km): 3699 Tempo: 570.83 Emissão CO2 (g): 0.0 Nodos expandidos: 3700 Nodos solução: 3700	Distância(Km): 4789 Tempo: 534.02 Emissão CO2 (g): 421393 Nodos expandidos: 4790 Nodos solução: 4790	Distância(Km): 7591 Tempo: 254.79 Emissão CO2 (g): 1368605 Nodos expandidos: 7592 Nodos solução: 7592	Distância(Km): 5636 Tempo: 1075.68 Emissão CO2 (g): 0.0 Nodos expandidos: 5637 Nodos solução: 5637	Distância(Km): 4500 Tempo: 449.8 Emissão CO2 (g): 394914 Nodos expandidos: 4501 Nodos solução: 4501	Distância(Km): 5936 Tempo: 211.3 Emissão CO2 (g): 1075784 Nodos expandidos: 5937 Nodos solução: 5937
BFS	Distância(Km): 117 Tempo: 18.09 Emissão CO2 (g): 0.0 Nodos expandidos: 9365 Nodos solução: 118	Distância(Km): 29 Tempo: 3.26 Emissão CO2 (g): 2570.6 Nodos expandidos: 1368 Nodos solução: 30	Distância(Km): 53 Tempo: 1.74 Emissão CO2 (g): 9354 Nodos expandidos: 4565 Nodos solução: 54	Distância(Km): 24 Tempo: 4.88 Emissão CO2 (g): 0.0 Nodos expandidos: 861 Nodos solução: 25	Distância(Km): 56 Tempo: 5.74 Emissão CO2 (g): 5037 Nodos expandidos: 5269 Nodos solução: 57	Distância(Km): 64 Tempo: 2.23 Emissão CO2 (g): 11353 Nodos expandidos: 6185 Nodos solução: 65
A*	Distância(Km): 117 Tempo: 17.82 Emissão CO2 (g): 0.0 Nodos expandidos: 118 Nodos solução: 118	Distância(Km): 29 Tempo: 3.25 Emissão CO2 (g): 2565.1 Nodos expandidos: 34 Nodos solução: 30	Distância(Km): 53 Tempo: 1.68 Emissão CO2 (g): 9040 Nodos expandidos: 54 Nodos solução: 54	Distância(Km): 24 Tempo: 4.52 Emissão CO2 (g): 0.0 Nodos expandidos: 25 Nodos solução: 25	Distância(Km): 56 Tempo: 5.64 Emissão CO2 (g): 4955 Nodos expandidos: 57 Nodos solução: 57	Distância(Km): 64 Tempo: 2.22 Emissão CO2 (g): 11319 Nodos expandidos: 65 Nodos solução: 65
Gulosa	Distância(Km): 117 Tempo: 18.06 Emissão CO2 (g): 0.0 Nodos expandidos: 118 Nodos solução: 118	Distância(Km): 29 Tempo: 3.36 Emissão CO2 (g): 2652.5 Nodos expandidos: 30 Nodos solução: 30	Distância(Km): 53 Tempo: 1.77 Emissão CO2 (g): 9531 Nodos expandidos: 54 Nodos solução: 54	Distância(Km): 24 Tempo: 4.61 Emissão CO2 (g): 0.0 Nodos expandidos: 25 Nodos solução: 25	Distância(Km): 56 Tempo: 5.71 Emissão CO2 (g): 5016 Nodos expandidos: 57 Nodos solução: 57	Distância(Km): 64 Tempo: 2.21 Emissão CO2 (g): 11256 Nodos expandidos: 65 Nodos solução: 65
Custo Uniforme	Distância(Km): 119 Tempo: 17.23 Emissão CO2 (g): 0.0 Nodos expandidos: 9400 Nodos solução: 120	Distância(Km): 31 Tempo: 3.29 Emissão CO2 (g): 2594.31 Nodos expandidos: 1439 Nodos solução: 32	Distância(Km): 53 Tempo: 1.67 Emissão CO2 (g): 8943 Nodos expandidos: 4347 Nodos solução: 54	Distância(Km): 24 Tempo: 4.27 Emissão CO2 (g): 0.0 Nodos expandidos: 888 Nodos solução: 25	Distância(Km): 56 Tempo: 5.32 Emissão CO2 (g): 4670 Nodos expandidos: 5567 Nodos solução: 57	Distância(Km): 64 Tempo: 2.17 Emissão CO2 (g): 11040 Nodos expandidos: 5890 Nodos solução: 65
Iterativa	Distância(Km): 121 Tempo: 18.65 Emissão CO2 (g): 0.0 Nodos expandidos: 122 Nodos solução: 122	Distância(Km): 113 Tempo: 12.53 Emissão CO2 (g): 9889.2 Nodos expandidos: 114 Nodos solução: 114	Distância(Km): 719 Tempo: 24.11 Emissão CO2 (g): 129520 Nodos expandidos: 720 Nodos solução: 720	Distância(Km): 226 Tempo: 43.45 Emissão CO2 (g): 0.0 Nodos expandidos: 227 Nodos solução: 227	Distância(Km): 94 Tempo: 9.46 Emissão CO2 (g): 8303 Nodos expandidos: 95 Nodos solução: 95	Distância(Km): 80 Tempo: 2.92 Emissão CO2 (g): 14871 Nodos expandidos: 81 Nodos solução: 81

Figura 10 - Testes aleatórios em grafo grande em grelha

Algoritmo / Teste	Origem: 3350496559 Destino: 6135208350 Veículo: Bicicleta Peso: 3Kg	Origem: 11444278229 Destino: 3431901560 Veículo: Moto Peso: 14Kg	Origem: 2792157078 Destino: 2181285600 Veículo: Carro Peso: 68Kg	Origem: 1760708443 Destino: 1591323272 Veículo: Bicicleta Peso: 4Kg	Origem: 1251308177 Destino: 2232799349 Veículo: Moto Peso: 11Kg	Origem: 2438984779 Destino: 2261684624 Veículo: Carro Peso: 93Kg
DFS	Distância(Km): 175.55 Tempo: 31.15 Emissão CO2 (g): 0.0 Nodos expandidos: 1422 Nodos solução: 1422	Distância(Km): 231.22 Tempo: 25.81 Emissão CO2 (g): 20364 Nodos expandidos: 1810 Nodos solução: 1810	Distância(Km): 16.13 Tempo: 0.54 Emissão CO2 (g): 2902 Nodos expandidos: 113 Nodos solução: 113	Distância(Km): 231.21 Tempo: 44.08 Emissão CO2 (g): 0.0 Nodos expandidos: 1826 Nodos solução: 1826	Distância(Km): 73.3 Tempo: 7.37 Emissão CO2 (g): 6471 Nodos expandidos: 645 Nodos solução: 645	Distância(Km): 53 Tempo: 1.93 Emissão CO2 (g): 9828 Nodos expandidos: 377 Nodos solução: 377
BFS	Distância(Km): 6.53 Tempo: 1.14 Emissão CO2 (g): 0.0 Nodos expandidos: 4382 Nodos solução: 49	Distância(Km): 2.09 Tempo: 0.24 Emissão CO2 (g): 190 Nodos expandidos: 745 Nodos solução: 25	Distância(Km): 2.25 Tempo: 0.08 Emissão CO2 (g): 410 Nodos expandidos: 321 Nodos solução: 16	Distância(Km): 12.92 Tempo: 2.54 Emissão CO2 (g): 0.0 Nodos expandidos: 5662 Nodos solução: 66	Distância(Km): 5.58 Tempo: 0.57 Emissão CO2 (g): 497 Nodos expandidos: 5499 Nodos solução: 46	Distância(Km): 5.99 Tempo: 0.22 Emissão CO2 (g): 1125 Nodos expandidos: 5350 Nodos solução: 53
A*	Distância(Km): 4.65 Tempo: 0.8 Emissão CO2 (g): 0.0 Nodos expandidos: 2486 Nodos solução: 49	Distância(Km): 2.04 Tempo: 0.24 Emissão CO2 (g): 186 Nodos expandidos: 699 Nodos solução: 25	Distância(Km): 2.13 Tempo: 0.07 Emissão CO2 (g): 378 Nodos expandidos: 302 Nodos solução: 18	Distância(Km): 9.97 Tempo: 1.9 Emissão CO2 (g): 0.0 Nodos expandidos: 4143 Nodos solução: 81	Distância(Km): 5.13 Tempo: 0.48 Emissão CO2 (g): 423 Nodos expandidos: 2744 Nodos solução: 70	Distância(Km): 6.00 Tempo: 0.22 Emissão CO2 (g): 1133 Nodos expandidos: 3335 Nodos solução: 61
Gulosa	Distância(Km): 6.11 Tempo: 1.06 Emissão CO2 (g): 0.0 Nodos expandidos: 82 Nodos solução: 66	Distância(Km): 3.32 Tempo: 0.32 Emissão CO2 (g): 256 Nodos expandidos: 49 Nodos solução: 35	Distância(Km): 2.85 Tempo: 0.1 Emissão CO2 (g): 513 Nodos expandidos: 22 Nodos solução: 20	Distância(Km): 13.16 Tempo: 2.48 Emissão CO2 (g): 0.0 Nodos expandidos: 121 Nodos solução: 97	Distância(Km): 5.35 Tempo: 0.51 Emissão CO2 (g): 446 Nodos expandidos: 102 Nodos solução: 69	Distância(Km): 6.94 Tempo: 0.26 Emissão CO2 (g): 1312 Nodos expandidos: 63 Nodos solução: 62
Custo Uniforme	Distância(Km): 4.65 Tempo: 0.8 Emissão CO2 (g): 0.0 Nodos expandidos: 2654 Nodos solução: 49	Distância(Km): 2.04 Tempo: 0.24 Emissão CO2 (g): 186 Nodos expandidos: 756 Nodos solução: 25	Distância(Km): 2.13 Tempo: 0.07 Emissão CO2 (g): 378 Nodos expandidos: 334 Nodos solução: 18	Distância(Km): 9.97 Tempo: 1.9 Emissão CO2 (g): 0.0 Nodos expandidos: 4336 Nodos solução: 81	Distância(Km): 5.13 Tempo: 0.48 Emissão CO2 (g): 423 Nodos expandidos: 2986 Nodos solução: 70	Distância(Km): 6.00 Tempo: 0.22 Emissão CO2 (g): 1133 Nodos expandidos: 3575 Nodos solução: 61
Iterativa	Distância(Km): 6.25 Tempo: 1.14 Emissão CO2 (g): 0.0 Nodos expandidos: 67 Nodos solução: 67	Distância(Km): 4.07 Tempo: 0.47 Emissão CO2 (g): 369 Nodos expandidos: 43 Nodos solução: 43	Distância(Km): 3.36 Tempo: 0.12 Emissão CO2 (g): 628 Nodos expandidos: 27 Nodos solução: 27	Distância(Km): 34.07 Tempo: 6.5 Emissão CO2 (g): 0.0 Nodos expandidos: 227 Nodos solução: 227	Distância(Km): 7.55 Tempo: 0.77 Emissão CO2 (g): 671 Nodos expandidos: 72 Nodos solução: 72	Distância(Km): 17.10 Tempo: 0.62 Emissão CO2 (g): 3154 Nodos expandidos: 111 Nodos solução: 111

Figura 11 - Testes aleatórios em grafo realista (Braga)

Algoritmo / Teste	Origem: Node_120 Destino: Node_144 Veículo: Bicicleta Peso: 2Kg	Origem: Node_197 Destino: Node_45 Veículo: Moto Peso: 4Kg	Origem: Node_77 Destino: Node_73 Veículo: Carro Peso: 72Kg	Origem: Node_164 Destino: Node_61 Veículo: Bicicleta Peso: 5Kg	Origem: Node_182 Destino: Node_190 Veículo: Moto Peso: 17Kg	Origem: Node_142 Destino: Node_55 Veículo: Carro Peso: 76Kg
DFS	Distância(Km): 104.69 Tempo: 17.48 Emissão CO2 (g): 0.0 Nodos expandidos: 23 Nodos solução: 23	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 31 Nodos solução: 31	Distância(Km): 131.1 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 27 Nodos solução: 27	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 10 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 4 Nodos solução: 4	Distância(Km): 103.1 Tempo: 3.51 Emissão CO2 (g): 18565 Nodos expandidos: 23 Nodos solução: 23
BFS	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 129 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 210 Nodos solução: 29	Distância(Km): 131.10 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 155 Nodos solução: 26	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 103 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 20 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10459 Nodos expandidos: 122 Nodos solução: 15
A*	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 144 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 210 Nodos solução: 29	Distância(Km): 131.1 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 118 Nodos solução: 27	Distância(Km): 55.69 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 103 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 5 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10459 Nodos expandidos: 129 Nodos solução: 14
Gulosa	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 210 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 210 Nodos solução: 29	Distância(Km): 131.1 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 118 Nodos solução: 27	Distância(Km): 63.29 Tempo: 13.67 Emissão CO2 (g): 0.0 Nodos expandidos: 75 Nodos solução: 13	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 6 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10459 Nodos expandidos: 141 Nodos solução: 14
Custo Uniforme	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 125 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 210 Nodos solução: 29	Distância(Km): 131.1 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 131 Nodos solução: 27	Distância(Km): 55.29 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 121 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 43 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10459 Nodos expandidos: 95 Nodos solução: 14
Iterativa	Distância(Km): 58.5 Tempo: 9.69 Emissão CO2 (g): 0.0 Nodos expandidos: 14 Nodos solução: 14	Distância(Km): 152.79 Tempo: 12.39 Emissão CO2 (g): 13427 Nodos expandidos: 29 Nodos solução: 29	Distância(Km): 131.1 Tempo: 4.37 Emissão CO2 (g): 23271 Nodos expandidos: 27 Nodos solução: 27	Distância(Km): 55.29 Tempo: 12.12 Emissão CO2 (g): 0.0 Nodos expandidos: 10 Nodos solução: 10	Distância(Km): 22.7 Tempo: 3.16 Emissão CO2 (g): 2208 Nodos expandidos: 4 Nodos solução: 4	Distância(Km): 59.7 Tempo: 1.98 Emissão CO2 (g): 10459 Nodos expandidos: 15 Nodos solução: 14

Figura 12 - Testes em grafo pequeno aleatório com nova heurística.

A partir dos testes apresentados e, conseqüentemente, dos resultados gerados pelos mesmos, foi-nos possível construir os seguintes gráficos estatísticos, sobre os quais nos debruçamos para tirar grande parte das conclusões relativas aos algoritmos de procura, quer informada, quer não informada:

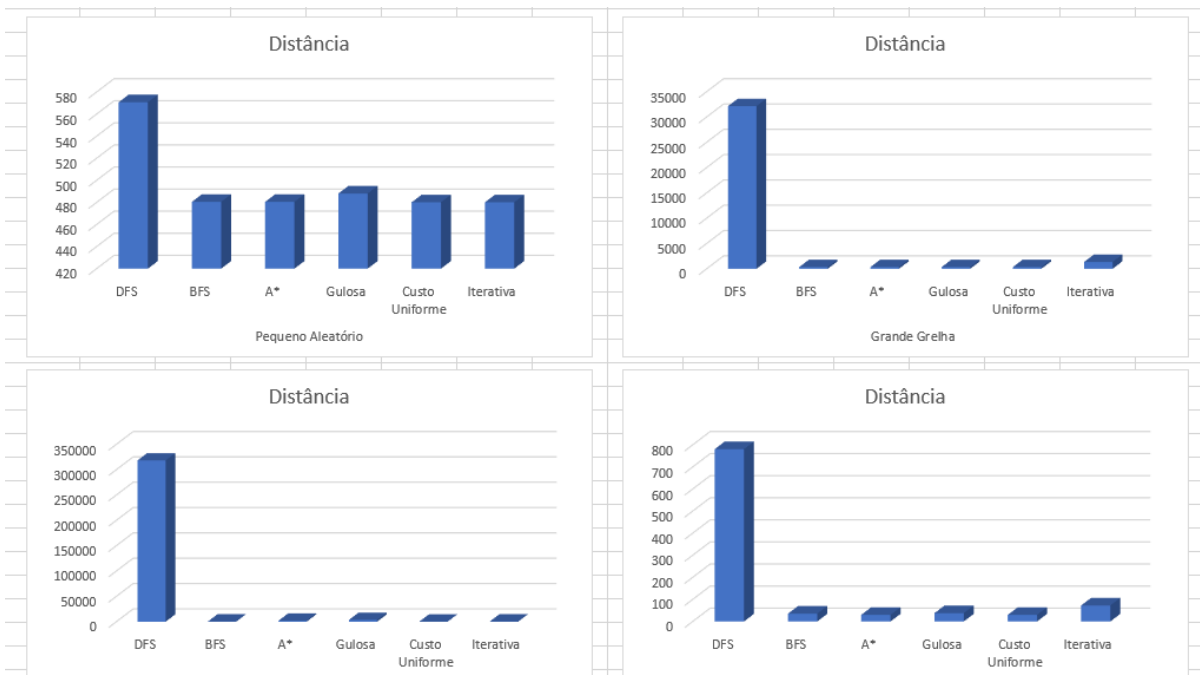


Figura 13 - Gráficos - Distância percorrida por algoritmo, por grafo

A escolha do algoritmo depende das características específicas do grafo. Ainda assim, A* e Custo Uniforme demonstraram ser escolhas geralmente boas em diferentes cenários, no que toca a distância total percorrida pelos testes efetuados, sendo que DFS ocupa a pior posição, com a maior distância percorrida. Contudo, é importante ter em conta que o melhor algoritmo pode variar dependendo do tipo e da estrutura do grafo.

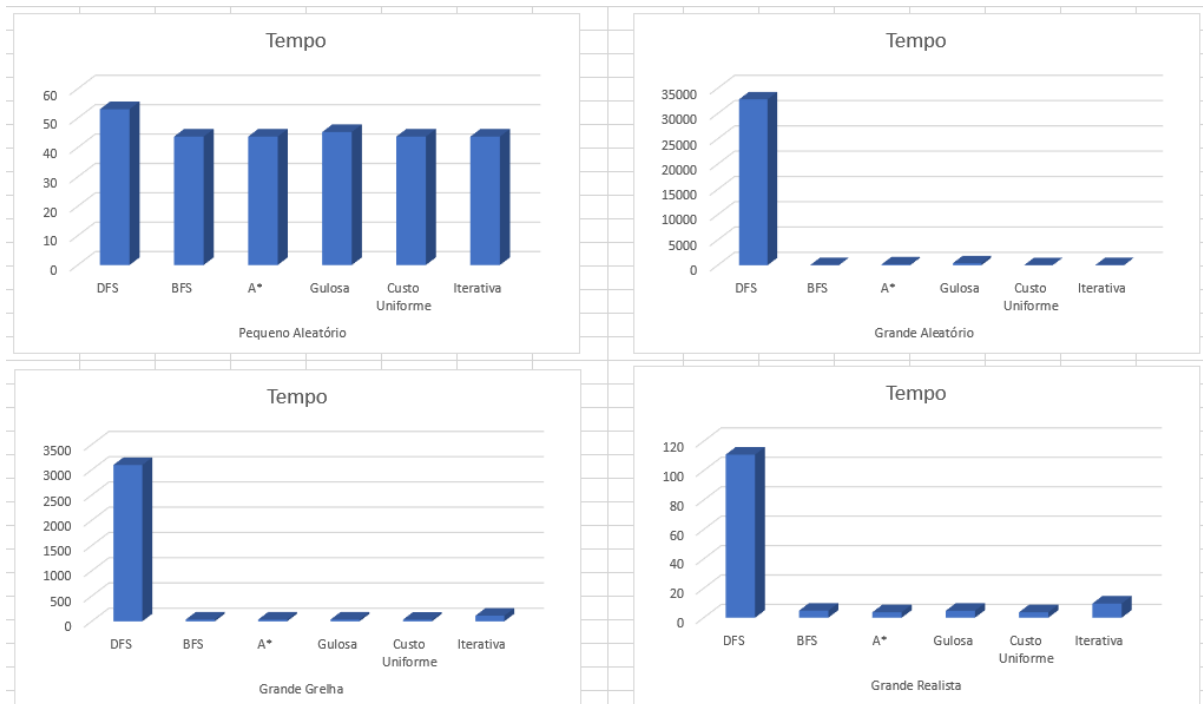


Figura 14 - Gráficos - Tempo de viagem por algoritmo, por grafo

Os tempos de entrega variam significativamente entre os algoritmos e a escolha do algoritmo pode depender tanto da eficiência em termos de tempo quanto da distância percorrida, conforme analisado anteriormente. DFS demonstra-se novamente o pior algoritmo, obtendo tempos de entrega extremamente elevados, proporcionais à distância percorrida. No que toca aos restantes algoritmos, a variância possui um valor relativamente baixo.

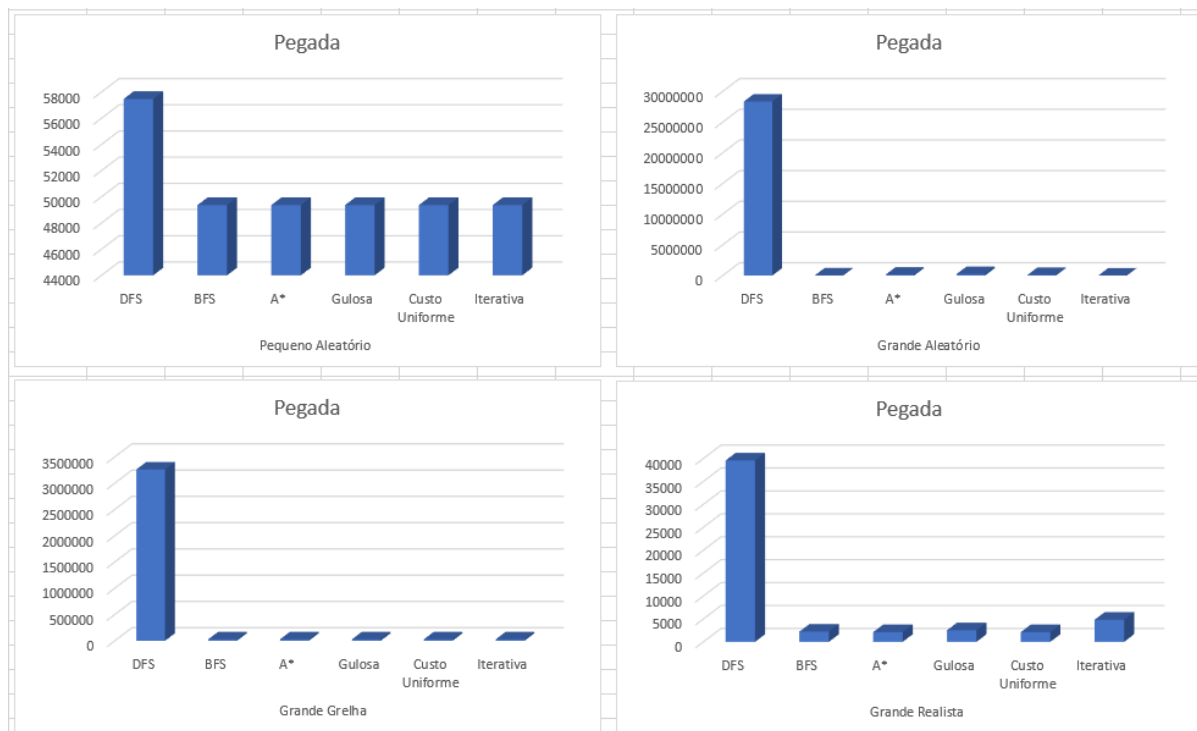


Figura 15 - Gráficos - Pegada por algoritmo, por grafo

A pegada de carbono associada às soluções dos diferentes algoritmos varia entre os grafos, mantendo a mesma proporção entre os algoritmos testados. A BFS geralmente mostra uma exploração mais eficiente em termos de pegada, enquanto a DFS associamos uma pegada muito alta, claramente proporcional à distância percorrida.

Pequeno Aleatório:

- Todos os algoritmos (à exceção de DFS) apresentam desempenho semelhante em termos de distância e pegada.
- A escolha entre eles pode depender de outros fatores, como tempo de execução.

Grande Aleatório:

- Custo Uniforme destaca-se como eficiente tanto em termos de distância como de tempo de entrega.
- BFS também é eficaz, enquanto DFS e Iterativa têm tempos e pegadas mais elevados.

Grande Grelha:

- A BFS, A*, Gulosa e Custo Uniforme apresentam desempenho semelhante em termos de pegada.
- DFS e Iterativa têm pegadas mais altas.

Grande Realista:

- A* e Custo Uniforme destacam-se em termos de distância e tempo de execução.
- BFS também é eficiente. DFS e Iterativa têm pegadas mais altas.

Conclusão Geral:

- Distância: A* e Custo Uniforme oferecem frequentemente os melhores resultados.
- Tempo de Execução: Custo Uniforme e BFS são geralmente eficientes, enquanto DFS e Iterativa tendem a ter tempos mais altos.
- Pegada: BFS, Custo Uniforme e A* mostra uma exploração mais eficiente em termos de pegada, enquanto DFS apresenta uma pegada muito alta.

A escolha do algoritmo ideal depende das prioridades específicas do problema em questão. Se a distância é a principal consideração, A* e Custo Uniforme são boas escolhas. Para eficiência na exploração, especialmente em termos de pegada, podemos considerar também a BFS juntamente com as anteriores. Cada algoritmo tem os seus pontos fortes e fracos e a decisão final deve levar em conta as características específicas do grafo e os objetivos do programa.

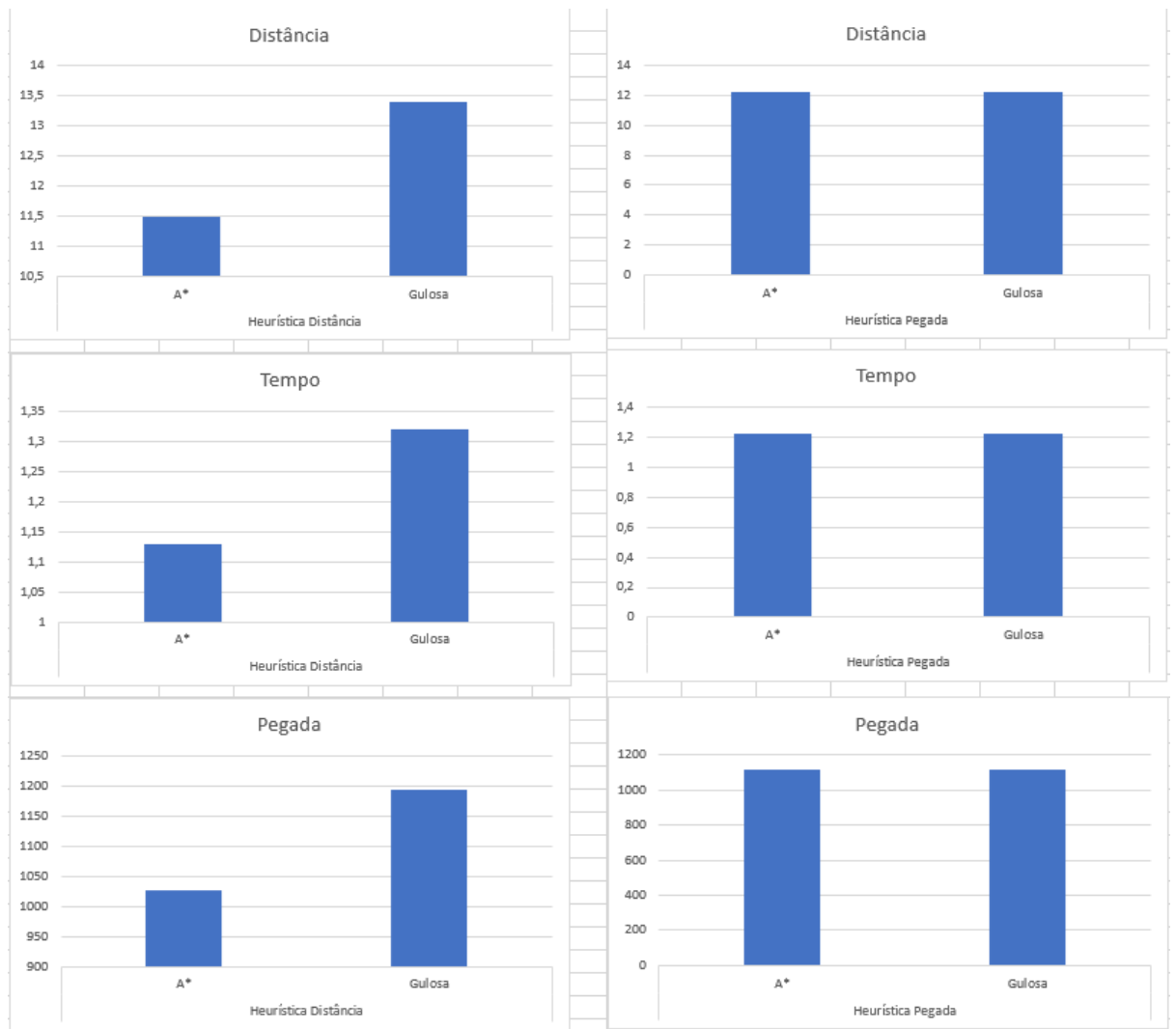


Figura 16 Comparação de gráficos estatísticos do uso de heurísticas diferentes para os mesmos testes, no mesmo grafo

Relativamente à procura informada, achamos relevante testar mais do que uma heurística, dado que a sua influência na eficiência do algoritmo é bastante elevada. Para tal, criamos uma heurística baseada na distância estimada até ao nó objetivo e outra baseada na pegada estimada até ao mesmo.

Neste sentido, deparamo-nos com resultados bastante semelhantes, o que nos chocou ao início, sendo que as funções nada tinham a ver uma com a outra. Contudo, após ponderar os resultados calmamente em grupo, rapidamente nos conformamos de que o resultado era o espetável, sendo que a pegada associada a um percurso é e será sempre diretamente proporcional à distância percorrida.

Procura com ações não determinísticas

Até agora consideramos sempre que o ambiente era determinístico, ou seja, todas as vias estariam sempre disponíveis e o nosso ator consegue sempre transitar entre nós. Contudo, a realidade é bastante diferente, pois, tanto as vias como o condutor podem não conseguir seguir o plano. Um outro aspeto que não é mencionado no enunciado é a questão do ambiente estático contra o ambiente dinâmico. Tudo feito até a este ponto considerou um ambiente estático, mas é muito improvável que as ruas de uma cidade estejam consistentemente com ou sem trânsito. Com isto, vamos apresentar duas procuras, ambas em ambientes não determinísticos, mas considerando um ambiente estático e depois um dinâmico. É importante referir que isto vão ser simulações e que ambicionam alcançar algum realismo, sendo mais significativo perceber os métodos para ultrapassar os problemas e não os métodos que os causam (apesar de serem devidamente explicados).

Ambiente Estático: Quando estamos num ambiente não determinístico, acessível num contexto de procura e estático, acabamos por ter o conhecido GPS que estamos familiarizados.

O primeiro passo é perceber o que pode correr mal na ação do agente. Os motivos podem variar desde obras na estrada, como até a um erro do próprio ator que o leve a um caminho errado. Desta forma, a solução é bastante simples: basta recalcular o melhor caminho desde a sua posição atual e, caso não tenha sido um engano do estafeta, considerar a via inicialmente calculada como “bloqueada”. Esta lógica repete-se até atingir o destino.

Avançando para a implementação, utilizamos a pontuação do estafeta juntamente com a congestão atual da via para “inventar” um problema e, até voltar a conseguir avançar, recalculamos o novo caminho. Ainda, dependendo do problema, é utilizada uma estrutura auxiliar no grafo que guarda arestas bloqueadas. Caso não exista problema nenhum, o ator segue o caminho planeado até ao momento.

Ambiente Dinâmico: Quando estamos num ambiente não determinístico, acessível num contexto de procura e dinâmico, acabamos por ter algo semelhante à conhecida aplicação “Waze”, que se distingue de outros serviços de GPS pela capacidade de considerar a congestão das vias.

Ao contrário do ambiente estático, não há uma garantia que o caminho atual é o melhor possível. Neste cenário temos duas opções: ou mantemos o plano inicial ou somos gananciosos e tentamos avançar sempre pelo melhor caminho na altura do movimento. Nós optamos pela segunda opção porque achamos muito interessante.

Para a simulação criamos uma *thread* que simula novas informações a chegar e mudar o nosso grafo atual. Isto engloba não só a taxa de congestão, mas também a taxa de pegada. Com isto, obtemos a cada novo nó que avançamos um melhor caminho que pode ou não ser possível avançar, algo semelhante ao ambiente estático. Consideramos sempre o cálculo utilizando o destino final, isto porque, caso não o fizéssemos, tornaríamos todos os algoritmos que usássemos em algo parecido à procura gulosa. Na nossa implementação, incluímos também a geração de imagens para observar as variações dos caminhos calculados até ao final.

Um fator comum a ambos os ambientes e a ambas as soluções é a necessidade de recalculamos o caminho até ao destino, logo, sempre que é preciso calcular um caminho utilizamos uma procura e, se queremos ter bons resultados, precisamos de adequar à melhor procura. Na nossa implementação permitimos ao utilizador escolher a pesquisa que quer utilizar.

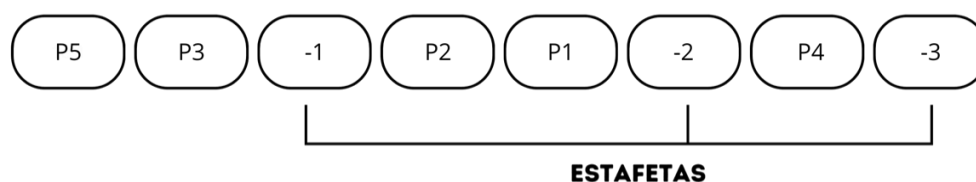
Os resultados foram de acordo com aquilo que observámos anteriormente, onde as melhores performances e soluções provêm das perguntas informadas e da procura custo uniforme. Um problema com a pesquisa informada é que, dependendo da heurística, pode ser preciso recalcular, o que pode prejudicar bastante a performance. Se utilizarmos uma heurística que se baseie em algo estático (como a distância), obtemos excelentes resultados com as procuras informadas.

Extra: O mTSP: Como se liga ao problema de procura e possível solução

O problema desenvolvido tem um problema complementar que é também indispensável para uma empresa que quer ser eficiente e obter bom desempenho dos trabalhadores. Esse problema é uma variante do famoso “multiple Traveling Salesman Problem”, que aparece dentro do grupo de problema “NP-difícil”. Este problema consiste em encontrar a melhor distribuição de cidades que diversos “Salesmans”, partindo de um mesmo ponto inicial, cobram todas as cidades sem repetir nenhuma. Facilmente conseguimos perceber que o funcionamento de uma distribuidora é bastante parecido ao “mTSP”. Contudo, há algumas considerações extras que não aparecem no problema original, nomeadamente: O tempo, todas as soluções precisam que as entregas sejam feitas a tempo; O peso, há um limite para cada estafeta (dependendo do veículo).

Para problemas deste género, como lecionado na disciplina, é apropriado o uso de algoritmos “population based”. De modo geral, não é esperado que estes algoritmos encontrem a melhor solução para o problema, mas podemos obter soluções boas que satisfazem as nossas necessidades, dependendo claro do poder computacional que temos na nossa posse. No caso do nosso grupo, apenas consideramos algoritmos que lecionamos na Unidade Curricular. Este conjunto inclui algoritmos como o “Particle Swarn Optimization”, o “Ant Colony Optimization” e o “Algoritmo Genético”. Por questões de tempo, apenas tivemos tempo de explorar o “Algoritmo Genético”, avançando agora para a forma como foi implementado.

De forma a manter o projeto de acordo com a UC, optamos por basear numa solução conhecida que usa um só cromossoma por solução, existindo diversas formas de o representar claro. Assim, cada cromossoma seguirá a seguinte estrutura:



Na figura em cima encontramos um cromossoma de tamanho $n + m$, onde “n” é o número de pacotes, e m é o número de estafetas. Os pacotes atribuídos a cada estafeta são os que se encontram à esquerda do seu “identificador”. A ordem na qual os pacotes está é importante, ou seja, por exemplo, o estafeta “1” entregaria o pacote na porta número 5 antes do pacote da porta número 3 (o que pode não ser a melhor opção).

Para acompanhar esta estrutura, devemos ainda ter numa estrutura à parte os pesos de cada encomenda assim como o tempo em horas disponível para cada entrega.

Resta ainda cobrir as mutações, os “crossovers”, o mecanismo de seleção de indivíduos e claro a função de *fitness*.

Mutações: Com base no coeficiente escolhido, há chance de ocorrer uma troca de posições entre duas entregas. Isto pode envolver estafetas diferentes ou até mesmo dentro do conjunto do próprio estafeta.

Crossovers: Aqui alteramos os separadores de posição, e ainda trocamos algumas cidades ao calhas, o que forma uma nova geração significativamente diferente da anterior.

Fitness: A parte mais difícil de computar é sem dúvida o fitness. Por cada cromossoma, é feita $m \times 3$ vezes, isto porque tentamos utilizar sempre o melhor veículo possível por cada estafeta tendo em conta os seus fatores limitantes. Caso uma solução ultrapasse o peso ou o tempo, adquire o valor de “10 000 000”, um valor muito grande que nunca se obteria de outra forma. O fitness de um cromossoma é a soma de todas as m melhores soluções (menor pegada e tempo gasto).

Torneio: Existem várias formas de avançar com a população, sendo que nós optamos pelo torneio pela sua simplicidade e convergência rápida, pois é muito improvável encontrar a solução ótima. Como objetivo é encontrar o melhor fitness, escolhemos os x cromossomas com o fitness mais baixo.

É importante destacar que, para conseguir boas soluções, pode ser requisitado populações grandes e várias gerações, o que tem um custo computacional elevado. Daí, ser bastante importante utilizar mecanismos de otimização e procuras apropriadas. Apesar de não implementamos, pensamos em, ou reduzir a área coberta por cada posto, ou tentar ter em memória apenas a informação necessária, o que pode ser bastante complexo.

Este extra serviu para o grupo poder compreender como é que as procuras e a abstração inicial podem ser aplicadas para o funcionamento de uma distribuidora. Dependendo da distribuidora em questão, pode ser preciso o uso diário deste tipo de programas, ou com menos frequência, sendo preciso adaptar a solução a cada problema inicial. Com as pesquisas, conseguimos perceber também que, para casos mais complexos, a vantagem que um algoritmo adequado tem face à clássica “força bruta”.

Conclusões

A adequação de rotas é um desafio bastante complexo. Neste sentido, existem vários fatores a ter em consideração, como a distância, o tempo e as emissões carbónicas e é difícil estar sempre a seguir o caminho ótimo. Além disso, questões como o custo computacional são muito esquecidas no que toca a desenvolver estes modelos, o que afeta drasticamente o uso das ferramentas desenvolvidas.

De modo geral, achamos que o trabalho produzido foi um sucesso, pois conseguimos representar com imenso detalhe uma determinada freguesia ou cidade, e ainda produzir rotas com diferentes algoritmos sobre estas representações.

Os nossos resultados cobrem diversas circunstâncias e permitem um uso eficiente da ferramenta desenvolvida que pode ser adequadamente utilizada. Assim, em qualquer região do globo, desde que seja traduzida para a nossa forma de representação, podem ser feitas pesquisas eficazes que originam boas soluções.

Encontramos também ferramentas externas pertinentes para o projeto, como por exemplo, a biblioteca *osmnx* que utiliza uma base de dados global de mapas que seguem uma representação semelhante à nossa, o que torna ainda mais fácil o uso das ferramentas desenvolvidas.

Ainda fomos capazes de atribuir realismo às procuras, onde simulamos, primeiramente, num ambiente não determinístico estático, que conseguia ser representativo de diversos problemas que levam à indisponibilidade de uma via ou a erros do próprio estafeta. De seguida, quisemos ir mais longe e criar uma estrutura dinâmica, onde o tráfego é variante com o tempo e exige repensar muitas das vezes o nosso caminho, pois o melhor caminho não se mantém igual ao calculado inicialmente.

Por fim, decidimos incluir um extra útil para uma transportadora que se serve da ferramenta desenvolvida e permite o uso eficiente dos recursos da transportadora. Esta questão surge sob uma espécie de *mTSP* que envolve encontrar uma rota boa para todos os estafetas que consiga garantir que todas as encomendas sejam entregues a tempo e que use relativamente bem os recursos da empresa.

Bibliografia

1. Singh, Dharm & Singh, Manoj Kumar & Singh, Tarkeshwar & Prasad, Rajkishore. (2018). Genetic Algorithm for Solving Multiple Traveling Salesmen Problem using a New Crossover and Population Generation. *Computación y Sistemas*. 22. 10.13053/cys-22-2-2956.