Universidade do Minho
Escola de Engenharia

Inteligência Artificial

Grupo 2

Marco Soares Gonçalves - a104614
Salvador Duarte Magalhães Barreto - a104520
André Filipe Soares Pereira - a104275
Leonardo Gomes Alves - a104093

3 de janeiro de 2025

Conteúdo

1 Avaliação por pares

A104614 Marco - 0
A104520 Salvador - 0
A104275 André - 0
A104093 Leonardo - 0

2 Descrição do problema

Diante de uma catástrofe natural, é necessário que exista uma distribuição de suprimentos pelas várias zonas afetadas. Esta distribuição pode ser feita por diferentes veículos e deve definir prioridades de localidades. Foi-nos proposto levar em conta diversos aspetos e objetivos como:

- Limitações geográficas
- Condições meteorológicas
- Prioridades das zonas com base em necessidades, gravidade da situação e população
- Limitação de recursos e minimização do desperdício de recursos(peso de carga e combustível)
- Janelas de tempo críticas
- Utilização de diversos veículos de transporte (com suas limitações)
- Maximização cobertura de zonas afetadas

Queremos criar um algoritmo que seja capaz de alcançar soluções minimamente optimizadas ao problema levando todos estes fatores em conta. Devemos basear-nos em algoritmos de procura e , por consequência, recorrer a representações sob a forma de grafo para sustentar o problema e a respetiva solução.

3 Formulação do problema

O problema a tratar é , da forma que o decidimos estruturar, um problema de **contingência**, visto que, a cada movimento, o ator vai recolher informação sobre os nós vizinhos, e só depois decide para qual se deve movimentar. No **grafo que desenvolvemos**, dada a sua baixa complexidade, seria viável procurar uma solução ótima mas, pressupondo a utilização do algoritmo em grafos de grande escala onde seria inviável fazê-lo, não nos demos a esse luxo.

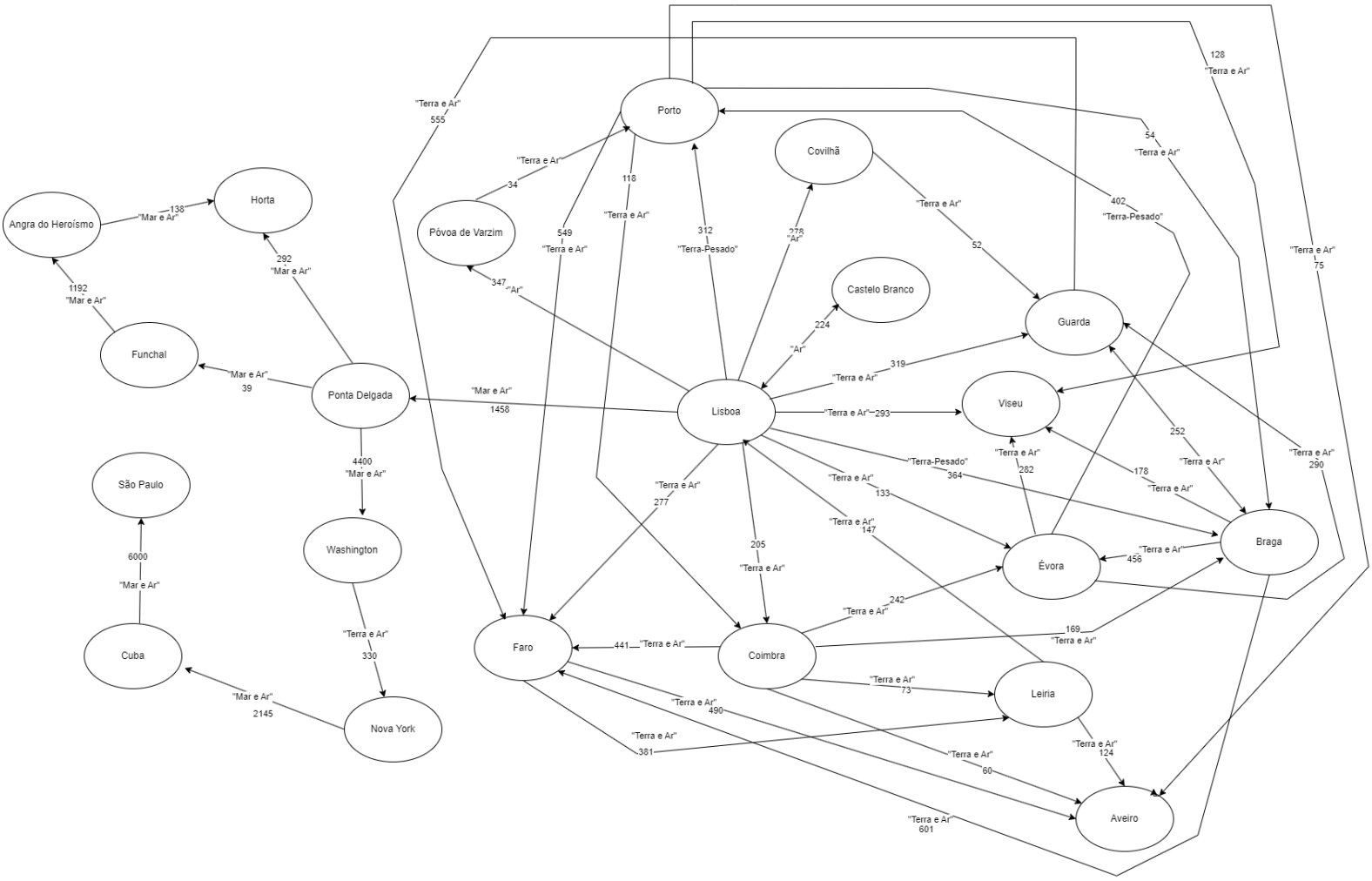
Para representar o estado temos de considerar os diversos **nodos** que criamos e as suas características, cada um deles representa uma região com diversas parâmetros relevantes:

Nome	**Gravidade**	**Supri.**	**População**	**Meteorologia**	**Reabastecer**	**Limite**	**Perecer**
Aveiro	Média	2380	68560	Moderada	Sim	26	Sim
Funchal	Alta	1300	105701	Boa	Sim	40	Não
Ponta Delgada	Alta	1250	67229	Boa	Sim	40	Não
Angra do Heroísmo	Média	1150	7953	Moderada	Não	32	Sim
Horta	Mínima	1100	14356	Má	Não	34	Não
Lisboa	Mínima	0	545142	Boa	Sim	300	Não
Porto	Alta	3120	231800	Moderada	Sim	30	Sim
Coimbra	Máxima	1200	106768	Má	Não	18	Não
Faro	Mínima	1100	67859	Boa	Sim	30	Sim
Braga	Alta	2800	148977	Moderada	Sim	22	Não
Évora	Média	1590	53577	Má	Não	25	Não
Guarda	Mínima	1800	25833	Boa	Não	35	Sim
Viseu	Máxima	1400	110000	Moderada	Sim	16	Não
Leiria	Alta	1220	128616	Boa	Não	28	Não
Aveiro	Média	3120	68560	Moderada	Sim	26	Sim
Castelo Branco	Alta	1200	56000	Má	Não	33	Não
Póvoa de Varzim	Mínima	1120	63408	Boa	Sim	30	Não
Covilhã	Máxima	1180	46457	Moderada	Não	35	Sim
Washington	Máxima	100	678972	Moderada	Não	45	Não
Cuba	Alta	100	11190000	Moderada	Sim	49	Não
Nova York	Média	100	8258000	Boa	Sim	42	Sim
São Paulo	Máxima	100	11450000	Má	Não	56	Não

Cada região possui um **nome** que a identifica, a sua **densidade populacional** e uma **gravidade** que irá ditar o quão grave foi a ocorrência no local. Cada região possui também uma quantidade de **suprimentos**("Supri.") necessária, suprimentos estes que podem ou não ser **perecíveis**, pelo que, cada nó possui uma "flag"que o indica. Para além disso cada local possui as suas **condições meteorológicas** e uma "flag"que indica se um veiculo **pode reabastecer** o seu combustível. Por fim, cada nó possui um **limite**, que representa o período máximo em que ainda é viável prestar assistência ao local; após esse prazo, a ajuda deixa de ser viável.

Para além destes fatores temos de levar em conta as restrições do tipo de veículos que podem viajar por cada caminho entre nodos; estes estão descritos na seguinte representação do estado, para além de todos os arcos entre nodos e as distâncias associadas:

1. **Representação do Estado:**



Temos 6 tipos de qualificações dependentes das restrições geográficas impostas: "**Terra**","**Terra-Pesados**","**Ar**","**Mar**","**Mar e Ar**","**Terra e Ar**"(Nem todas incluídas no estado inicial) que restringem os tipos de veículos que podem usufruir dos arcos do grafo.

2. **Estado Inicial:** Os veículos partem sempre de "**Lisboa**"e existe um número definido de suprimentos para entregar. O estado inicial será o conjunto de "**Lisboa**"com total de suprimentos disponíveis para entregar e os **veículos** disponíveis.

3. **Estado Objetivo:** A ideia por de trás do sistema é a **entrega** rápida e **eficiente** dos suprimentos requisitados. O estado objetivo é a **entrega de todos os suprimentos**, ou seja a satisfação de todas as localidades(**necessidades = 0**).

4. **Operadores: Movimentar** um veículo de uma localização para outra, localização esta que poderá reabastecer ou não o veículo. **Distribuir** suprimentos(acontece sempre que possível).

5. **Custo da Solução:** Para os algoritmos informados podemos dizer q a heurística total representa o custo da solução, este valor terá em conta uma relação custo benefício mas tem sempre como prioridade entregar o máximo de suprimentos. Para o algoritmo DFS será a distancia percorrida.

4 Descrição de todas as tarefas realizadas, bem como de todas as decisões tomadas pelo grupo de trabalho

Para os veículos criamos um tabela que represente toda a informação sobre eles. Cada veiculo possui um **tipo**, uma **capacidade máxima** que indica quantos suprimentos este consegue levar em cada viagem e possui também uma **autonomia**, que representa o número de litros de combustível que o veiculo suporta no seu tanque. Para além disso, cada veículo possui uma **velocidade** média de viagem(quilómetros/hora) e um **consumo** de combustível(litros/quilómetro).

Tipo	**Nome**	**Cap. Max.**	**Autonomia**	**Velocidade**	**Consumo**
Carro	Carro1	1500	60	100	0.07
	Carro2	1500	60	100	0.07
	Carro3	1500	60	100	0.07
	Carro4	1500	60	100	0.07
	Carro5	1500	60	100	0.07
	Carro6	1500	60	100	0.07
Camião	Camiao1	10000	3000	80	0.35
	Camiao2	10000	300	80	0.35
	Camiao3	10000	300	80	0.35
	Camiao4	10000	300	80	0.35
	Camiao5	10000	300	80	0.35
	Camiao6	10000	300	80	0.35
Aéreos	Avioneta1	1000	200	200	0.1
	Avioneta2	1000	200	200	0.1
	Avioneta3	1000	200	200	0.1
	Avioneta4	1000	200	200	0.1
	Avioneta5	1000	200	200	0.1
	Avioneta6	1000	200	200	0.1
	Drone1	50	15	400	0.008
	Drone2	50	15	400	0.008
Barco	Barquinho1	2000	450	55	0.1
	Barquinho2	2000	450	55	0.1
	Barquinho3	2000	450	55	0.1
	Barquinho4	2000	450	55	0.1
	Barquinho5	2000	450	55	0.1

Temos que ressaltar a remodelação mais notável entre os algoritmos que foram desenvolvidos e as inspirações para os mesmos, a **falta de um nodo objetivo**; no contexto do problema que estruturamos , baseado na nossa interpretação, isso está claro e teve grande impacto nos algoritmos desenvolvidos.

Os **algoritmos informados** desenvolvidos são extremamente **ineficientes** pois fazem o cálculo do caminho para todos os veículos **n vezes** onde n é o numero de veículos, ou seja, a cada iteração calcula um caminho para todos os veículos, seleciona o melhor e na iteração seguinte testa todos os que sobram até testar todos os veículos. No entanto, como queremos garantir **otimização máxima da distribuição**, adotamos este método.

Quanto aos **suprimentos** tomamos uma abordagem talvez **excessivamente simplista**. Nos algoritmos de procura informada **não distinguimos** os tipos de suprimentos nem a sua validade; tomamos como garantido que um caminho será decidido antes de enviar o veículo com o algoritmo desenvolvido e que , nesse cenário, o peso que o veículo pode suportar seria **preenchido** com os **suprimentos correspondentes** às regiões que decidiu visitar. Para além disso assumimos que a janela de tempo crítico serviria como temporizador para os suprimentos perecíveis pois , na nossa estrutura, nunca será possível reabastecer suprimentos depois do instante 0 (em Lisboa).

Não incorporamos distinção entre diversos suprimentos nem implementamos o perecer destes expecto no algoritmo **Depth-First-Search** onde , apesar de também não distinguir tipos de suprimentos (água, comida etc.) distingue entre suprimentos perecíveis e não perecíveis e o veículo carrega quantidade diferentes de cada tipo, no caso, **30% dos suprimentos** são considerados **perecíveis**.

No **cálculo de combustível** garantimos que temos combustível para ir e voltar sempre que avançamos para um nó vizinho pois, no final de uma iteração, é suposto o veículo voltar a Lisboa pelo caminho que percorreu. Quando é possível reabastecer no nodo destino só precisamos de combustível para chegar a ele.

4.1 Heurística

Um dos desafio do projeto foi desenvolver uma **heurística** que pudesse , de forma **justa**, levar em conta todos os fatores relevantes. Vale ressaltar que , ao **contrário** dos algoritmos **tradicionais**, a nossa heurística é calculada com um nó de origem e de destino em consideração e não apenas uma relação 1 nodo - 1 heurística independente da origem. Descobrimos ser imperativo desenvolver uma **heurística dinâmica** dependente também do veículo a ser utilizado e das suas características, e de ambos os nós em questão.

A **heurística desenvolvida** avalia em função de: distancias, janelas críticas, velocidades de transporte, suprimentos perecíveis/não perecíveis, consumos de combustível, quantidade de suprimentos e pelo "fator_critico"que resulta da densidade populacional e da gravidade da situação na região.

Decidimos que quanto maior a heurística for menor será a chance de esse caminho ser escolhido, e em caso de heurística negativa o caminho **nunca** será escolhido para além de um **caso de exceção**, uma vez que valores < 0 indicam impossibilidade de mover para o nodo destino (p.e. um carro num arco restrito a "Ar"). Os diferentes fatores são os seguintes:

- Caso não existam necessidades a satisfazer, a heurística retorna -2.
- Caso a janela de tempo seja ultrapassada, não haja acessibilidade ou caso o combustível não seja suficiente para se movimentar de um nó para o outro, então a heurística retorna

-1.

- A densidade populacional é o fator mais influente da heurística, visto que é preferível ajudar um maior número de pessoas do que minimizar os gastos.
- Gravidades baixas e janelas de tempo altas são penalizadas, simbolizando que têm tempo para ser resolvidas mais tarde.
- Alto consumo de combustível é penalizado.
- Nodos objetivo com necessidades maiores ou muito maiores que os suprimentos que temos são também penalizados.
- Locais que necessitem de alimentos perecíveis são priorizados.

4.2 Algoritmos

Para resolver o problema decidimos utilizar **3 algoritmos diferentes**, um de procura cega e dois de procura informada. Todos estes algoritmos são baseados em algoritmos estudados ao longo deste semestre, mas com várias **mudanças** para que se possam adaptar ao nosso caso de estudo. Os algoritmos escolhidos foram:

- **Procura cega:** Depth-First-Search(DFS)
- **Procura informada:** A*(personalizado) e "Greedy"

Nota: Todos os algoritmos desenvolvidos não possuem um nodo objetivo , como mencionado previamente, de modo a refletir um cenário real critico em que o verdadeiro objetivo é simplesmente maximizar o custo beneficio da distribuição.

O **caso de exceção** referido na "**Heurística**"acontece nas procuras informadas quando todos os vizinhos de um nó retornarem heurística -2, nesse caso será obrigatório escolher um nó sem necessidades e isto é feito **aleatoriamente** tendo em conta apenas os nós para os quais é possível nos movermos.

Os algoritmos informados **penalizam baixa eficiência** na entrega (depois de calcular todo o percurso), ou seja, entregar poucos suprimentos relativamente à capacidade máxima do veículo usado, dando **prioridade** a **eficiências maiores** quando escolhem um caminho.

4.2.1 Procura cega

Depth-First-Search: A nossa implementação é relativamente parecida com a implementação normal do algoritmo, com algumas modificações, nomeadamente:

1. Foi necessário introduzir **condições** que verifiquem o **combustível** disponível, os **reabastecimentos** e ambas as **janelas de tempo** (janela crítica e de suprimentos perecíveis).
2. Tal como mencionado anteriormente, não existe um nodo objetivo, mas sim vários **estados objetivo**. O algoritmo apenas acaba caso todos os **suprimentos** estiverem **entregues**, caso já não hajam **vizinhos** a visitar ou caso o veículo **não consiga visitar** nenhum dos vizinhos.

Este algoritmo é chamado para cada um dos veículos disponíveis, sendo que, cada vez que um veículo completa a sua iteração do algoritmo, atualiza as necessidades do grafo, garantindo que o próximo veículo não entrega suprimentos onde não é necessário.

Resultados:

Veiculo	Heuristica	Distancia	Entregas		Horas	Caminho
Carro1	0	430	1500	100.0%	7.63	[Lisboa, Porto, Coimbra]
Carro2	0	430	1500	100.0%	7.63	[Lisboa, Porto, Coimbra]
Carro3	0	672	1170.0	78.0%	13.68	[Lisboa, Porto, Coimbra, Évora]
Carro4	0	672	1050.0	70.0%	13.68	[Lisboa, Porto, Coimbra, Évora]
Carro5	0	672	690.0	46.0%	13.68	[Lisboa, Porto, Coimbra, Évora]
Avioneta1	0	989	1000	100.0%	8.93	[Lisboa, Coimbra, Évora, Guarda, Braga]
Avioneta2	0	989	1000	100.0%	8.93	[Lisboa, Coimbra, Évora, Guarda, Braga]
Avioneta3	0	1590	1000	100.0%	11.93	[Lisboa, Coimbra, Évora, Guarda, Braga, Faro]
Avioneta4	0	1590	1000	100.0%	11.93	[Lisboa, Coimbra, Évora, Guarda, Braga, Faro]
Avioneta5	0	1971	1000	100.0%	13.84	[Lisboa, Coimbra, Évora, Guarda, Braga, Faro, Leiria]
Avioneta6	0	1971	1000	100.0%	13.84	[Lisboa, Coimbra, Évora, Guarda, Braga, Faro, Leiria]
Barquinho1	0	2689	2000	100.0%	59.73	[Lisboa, Ponta Delgada, Funchal, Angra do Heroísmo]
Barquinho2	0	2827	1150.0	57.5%	66.0	[Lisboa, Ponta Delgada, Funchal, Angra do Heroísmo, Horta]
Barquinho3	0	2827	550.0	27.5%	66.0	[Lisboa, Ponta Delgada, Funchal, Angra do Heroísmo, Horta]

- **Suplementos entregues:** 15610/26050
- **Capacidade dos veículos:** 19500(80.05%eficiência)
- **Nodos não visitados:** ['Viseu', 'Covilhã', 'Póvoa de Varzim', 'Castelo Branco', 'Washington', 'Nova York', 'Cuba', 'São Paulo', 'Aveiro']
- **Veiculos nao usados:** ['Carro6', 'Camiao1', 'Camiao2', 'Camiao3', 'Camiao4', 'Camiao5', 'Camiao6', , 'Drone1', 'Drone2', 'Barquinho4', 'Barquinho5']
- **Nodos com necessidades:** [('Viseu', 1400), ('Covilhã', 1180), ('Póvoa de Varzim', 1120), ('Castelo Branco', 1200), ('Washington', 100), ('Nova York', 100), ('Cuba', 100), ('São Paulo', 100), ('Horta', 1100), ('Leiria',920), ('Aveiro', 3120)]

4.2.2 Procura informada:

A*(personalizado):

O algoritmo desenvolvido inicia com a criação de duas listas: **open_list** (contém nós a serem explorados) e **closed_list** (contém nós já explorados). Enquanto existirem nós na **open_list** seleciona-se sempre o próximo nó com base na soma da **distância total percorrida** com a **heurística**, garantindo assim que o nó com maior prioridade seja o preferido.

No caso de **não existirem** vizinhos com **necessidades**, o sistema seleciona um dos vizinhos **aleatoriamente**, garantindo assim que o sistema continuará a calcular a "melhor"rota. Sempre que um movimento é feito de um nó para o outro o impacto da viagem é calculado tendo em conta as **condições meteorológicas, tempo de viagem e o combustível gasto**. O nó visitado é então introduzido na **closed_list**.

Para que o algoritmo termine é necessário que uma das 3 seguintes condições se verifique:

1. Todos os suprimentos foram entregues.
2. Todos os nós foram explorados (open_list vazia).
3. Não haja combustível suficiente para se movimentar para nenhum outro nó.

Este algoritmo é **repetido** por **todos os veículos** de modo a obter a melhor rota a cada iteração. Para isso priorizam-se os veículos que possuam rotas com **menor heurística**. Após cada iteração o grafo é atualizado, o **melhor veículo** e as estatísticas relativas ao seu percurso são **registadas** e repetimos o processo até não termos mais veículos por testar. Por vezes **é possível** não usar certos veículos caso não exista qualquer sitio onde eles possam entregar.

Resultados:

```
Veiculo    Heuristica  Distancia  Entregas       Horas  Caminho

Drone1     0.28        278        50      100.0%  1.04   [Lisboa, Covilhã]
Drone2     0.27        278        50      100.0%  1.04   [Lisboa, Covilhã]
Carro1     1.88        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro2     1.69        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro3     2.04        387        1500    100.0%  5.8    [Lisboa, Porto, Aveiro]
Avioneta1  2.93        278        1000    100.0%  2.08   [Lisboa, Covilhã]
Carro4     7.06        767        1500    100.0%  10.12  [Lisboa, Faro, Aveiro]
Carro5     6.69        265        1500    100.0%  6.03   [Lisboa, Coimbra, Aveiro]
Carro6     7.88        293        1400     93.3%  4.4    [Lisboa, Viseu]
Avioneta2  35.49       278        80        8.0%  2.08   [Lisboa, Covilhã]
Avioneta3  10.16       319        1000    100.0%  1.59   [Lisboa, Guarda]
Avioneta4  12.29       319        800      80.0%  1.59   [Lisboa, Guarda]
Avioneta5  27.56       133        1000    100.0%  1.66   [Lisboa, Évora]
Avioneta6  25.53       133        590      59.0%  1.66   [Lisboa, Évora]
Barquinho1 189.12      1497       2000    100.0%  27.22  [Lisboa, Ponta Delgada, Funchal]
Barquinho2 24.34       1497       550      27.5%  27.22  [Lisboa, Ponta Delgada, Funchal]
Camiao1    330.66      364        2800     28.0%  6.82   [Lisboa, Braga]
Camiao5    933.19      387        1040     10.4%  7.26   [Lisboa, Porto, Aveiro]

Suplementos entregues = 19860/26050 - Capacidade dos veiculos = 39100(50.79% eficiencia)
Nodos nao visitados: ['Póvoa de Varzim', 'Castelo Branco', 'Washington', 'Nova York', 'Cuba', 'São Paulo', 'Angra do Heroísmo', 'Horta', 'Leiria']
Veiculos nao usados: ['Barquinho3', 'Barquinho4', 'Barquinho5', 'Camiao3', 'Camiao4', 'Camiao6', 'Camiao2']
Nodos com necessidades: [('Póvoa de Varzim', 1120), ('Castelo Branco', 1200), ('Washington', 100), ('Nova York', 100), ('Cuba', 100), ('São Paulo', 100), ('Angra do Heroísmo', 115
0), ('Horta', 1100), ('Leiria', 1220)]
```

"Greedy": A nossa implementação do algoritmo "**Greedy**"é muito semelhante à do nosso algoritmo **A***(personalizado), a única diferença é o facto de que a seleção do nosso vizinho deixa de ter em consideração a distância total percorrida, focando-se exclusivamente na **heurística**.

Resultados:

```
Veiculo    Heuristica  Distancia  Entregas       Horas  Caminho

Drone1     0.28        278        50      100.0%  1.04   [Lisboa, Covilhã]
Drone2     0.27        278        50      100.0%  1.04   [Lisboa, Covilhã]
Carro1     1.88        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro2     1.69        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro3     2.04        387        1500    100.0%  5.8    [Lisboa, Porto, Aveiro]
Avioneta1  2.93        278        1000    100.0%  2.08   [Lisboa, Covilhã]
Carro4     7.06        767        1500    100.0%  10.12  [Lisboa, Faro, Aveiro]
Carro5     6.69        265        1500    100.0%  6.03   [Lisboa, Coimbra, Aveiro]
Carro6     7.88        293        1400     93.3%  4.4    [Lisboa, Viseu]
Avioneta2  35.49       278        80        8.0%  2.08   [Lisboa, Covilhã]
Avioneta3  10.16       319        1000    100.0%  1.59   [Lisboa, Guarda]
Avioneta4  12.29       319        800      80.0%  1.59   [Lisboa, Guarda]
Avioneta5  27.56       133        1000    100.0%  1.66   [Lisboa, Évora]
Avioneta6  25.53       133        590      59.0%  1.66   [Lisboa, Évora]
Barquinho1 189.12      1497       2000    100.0%  27.22  [Lisboa, Ponta Delgada, Funchal]
Barquinho2 24.34       1497       550      27.5%  27.22  [Lisboa, Ponta Delgada, Funchal]
Camiao1    330.66      364        2800     28.0%  6.82   [Lisboa, Braga]
Camiao5    933.19      387        1040     10.4%  7.26   [Lisboa, Porto, Aveiro]

Suplementos entregues = 19860/26050 - Capacidade dos veiculos = 39100(50.79% eficiencia)
Nodos nao visitados: ['Póvoa de Varzim', 'Castelo Branco', 'Washington', 'Nova York', 'Cuba', 'São Paulo', 'Angra do Heroísmo', 'Horta', 'Leiria']
Veiculos nao usados: ['Barquinho3', 'Barquinho4', 'Barquinho5', 'Camiao2', 'Camiao3', 'Camiao4', 'Camiao6']
Nodos com necessidades: [('Póvoa de Varzim', 1120), ('Castelo Branco', 1200), ('Washington', 100), ('Nova York', 100), ('Cuba', 100), ('São Paulo', 100), ('Angra do Heroísmo', 115
0), ('Horta', 1100), ('Leiria', 1220)]
```

Os resultados entre a procura "**A***"e a "**Greedy**"foram **iguais** , mais uma vez, a natureza do nosso problema e a falta de nó objetivo que num problema clássico daria vantagem a uma procura **A*** fazem com que o **comportamento** dos dois algoritmos seja **idêntico**; no A* **não temos a capacidade de "voltar atrás"** logo também irá simplesmente escolher a heurística de menor valor.

Nota: Ambos os algoritmos variam na solução mas de forma igual; por vezes a solução é ligeira mente diferente ou o algoritmo escolhe caminhos alternativos para chegar às mesmas regiões.

5 Programa final

O programa final ficou dividido em **7 ficheiros**:

- **Algoritmo_personalizado.py**
- **Depth_First.py**
- **Grafo.py**
- **Greedy.py**
- **Main.py**
- **Node.py**
- **Veiculo.py**

Os ficheiros **Algoritmo_personalizado.py**, **Depth_First.py** e **Greedy.py** contém, respetivamente, as funções associadas ao algoritmo personalizado (A*), ao algoritmo DFS e ao algoritmo Greedy. Os ficheiros **Node.py** e **Veiculo.py** contém as estruturas de dados usadas para registar nodos e veículos e algumas funções associadas. O ficheiro **Grafo.py** inclui varias funções que manipulam o grafo e também uma função de desenho e a função de heurística.

No ficheiro **Main.py** são criados os nodos e os arcos entre eles num grafo "**g**".Neste ficheiro podemos editar os valores **M1** e **M2** para multiplicar, respetivamente, a janela de tempo e as necessidades dos nodos do grafo e , assim, causar alterações na solução; para além destes o valor booleano **R** permite impedir reabastecimentos se for definida como "**False**".

```
g = Graph()
##Variaveis para alterar comportamento##
M1 = 1 ##Janela de Tempo
M2 = 1 ##Necessidades
R = True ## Por em false para que ninguem reabasteça
##############################################
```

Ao executar o ficheiro **Main.py** temos acesso a um **menu** com as seguinte opções:

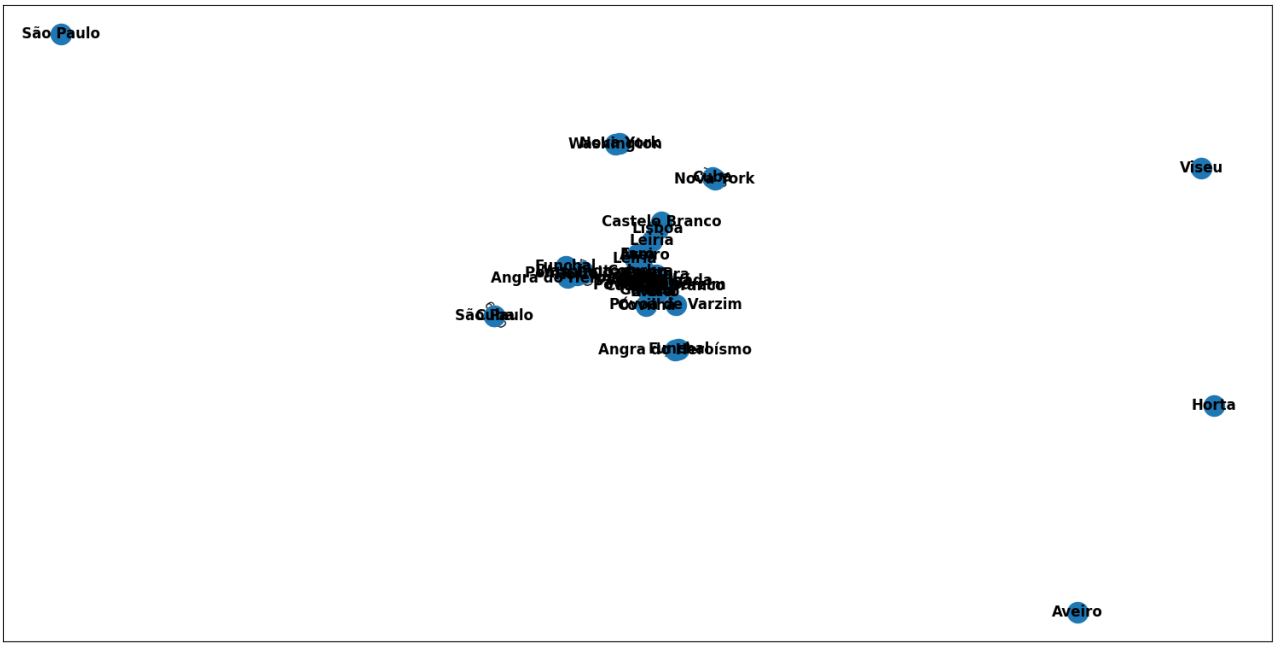
```
1-Imprimir Grafo
2-Desenhar Grafo
3-Imprimir Nomes de Nodos
4-Algoritmo Personalizado
5-Algoritmo Personalizado Greedy (best)
6-Algoritmo DFS
7-Seleciona para alterar  O grafo não será alterado
0-Sair
introduza a sua opcao->
```

Nota: O algoritmo Greedy é denotado como "(best)"pois apresenta os melhores resultados (equivalentes ao Algoritmo Personalizado) mas sem ter que calcular custos totais logo é (muito ligeiramente) mais eficiente.

Na **primeira opção** podemos imprimir os nodos do Grafo e os vizinhos associados.

```
Lisboa: [(Porto, 312, 'TERRA-PESADO'), (Braga, 364, 'TERRA-PESADO'), (Coimbra, 205, 'TERRA E AR'), (Faro,
ilhã, 278, 'AR'), (Póvoa de Varzim, 347, 'AR'), (Castelo Branco, 224, 'AR'), (Ponta Delgada, 1458, 'MAR E
Porto: [(Coimbra, 118, 'TERRA E AR'), (Faro, 549, 'TERRA E AR'), (Braga, 54, 'TERRA E AR'), (Viseu, 128, '
Braga: [(Guarda, 252, 'TERRA E AR'), (Faro, 601, 'TERRA E AR'), (Évora, 456, 'TERRA E AR'), (Viseu, 178, '
Coimbra: [(Évora, 242, 'TERRA E AR'), (Faro, 441, 'TERRA E AR'), (Braga, 169, 'TERRA E AR'), (Leiria, 73,
Faro: [(Leiria, 381, 'TERRA E AR'), (Aveiro, 490, 'TERRA E AR')]
Évora: [(Porto, 402, 'TERRA-PESADO'), (Guarda, 290, 'TERRA E AR'), (Viseu, 282, 'TERRA E AR')]
Guarda: [(Braga, 252, 'TERRA E AR'), (Faro, 555, 'TERRA E AR')]
Viseu: []
Covilhã: [(Guarda, 52, 'TERRA E AR')]
Póvoa de Varzim: [(Porto, 34, 'TERRA E AR')]
Castelo Branco: [(Lisboa, 224, 'TERRA E AR')]
Ponta Delgada: [(Washington, 4400, 'MAR E AR'), (Funchal, 39, 'MAR E AR'), (Horta, 292, 'MAR E AR')]
Washington: [(Nova York, 330, 'TERRA E AR')]
Nova York: [(Cuba, 2145, 'MAR E AR')]
Cuba: [(São Paulo, 6000, 'MAR E AR')]
São Paulo: []
Funchal: [(Angra do Heroísmo, 1192, 'MAR E AR')]
Angra do Heroísmo: [(Horta, 138, 'MAR E AR')]
Horta: []
Leiria: [(Lisboa, 147, 'TERRA E AR'), (Aveiro, 124, 'TERRA E AR')]
Aveiro: []
```

Na **segunda opção** podemos desenhar o grafo recorrendo à livraria "networkx". A representação não é de fácil visualização mas corresponde aos arcos do grafo de forma fiel.



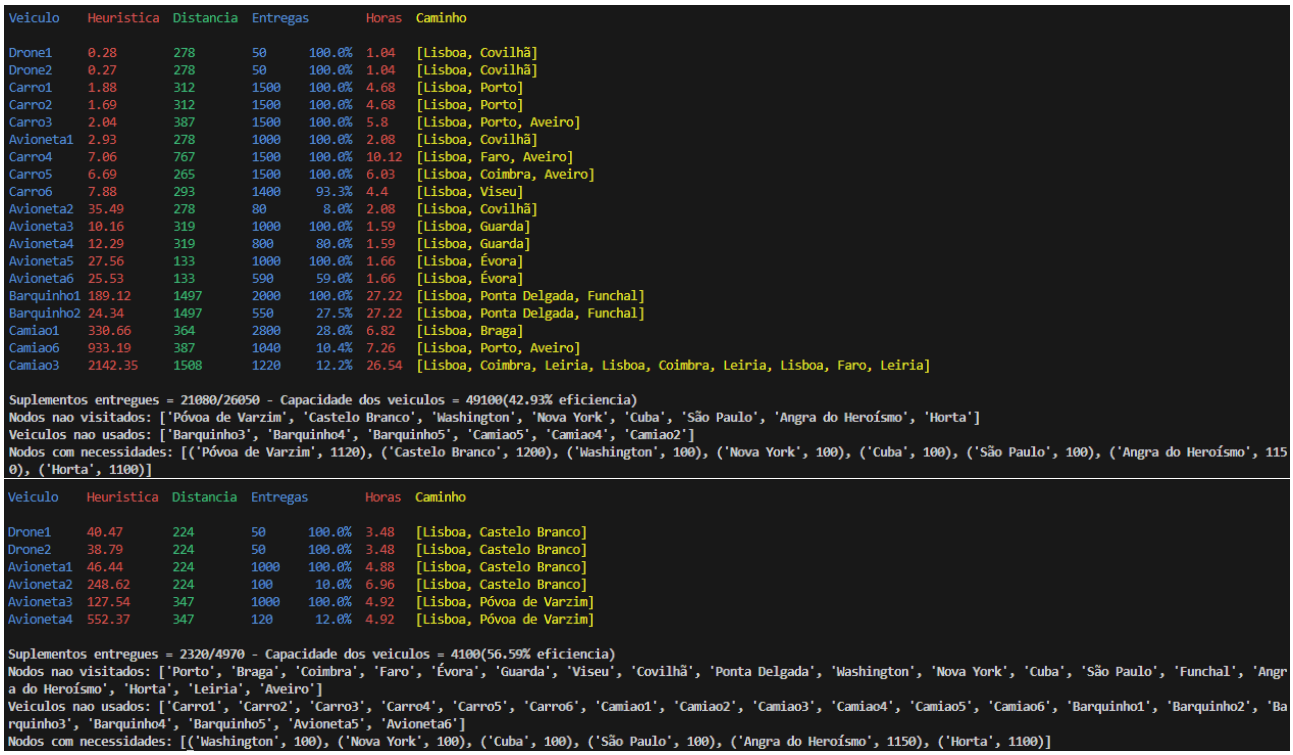
Na **terceira opção** permitimos um print direto dos nomes dos nodos guardados no grafo.

```
introduza a sua opcao-> 3
dict_keys(['Lisboa', 'Porto', 'Braga', 'Coimbra', 'Faro', 'Évora', 'Guarda', 'Viseu', 'Covilhã', 'Póvoa de Varzim', 'Castelo Branco', 'Ponta Delgada', 'Washingt
on', 'Nova York', 'Cuba', 'São Paulo', 'Funchal', 'Angra do Heroísmo', 'Horta', 'Leiria', 'Aveiro'])
```

A **quarta**, **quinta** e **sexta opção** executam a pesquisar correspondente e imprimem para o terminal tal e qual demonstrado no capítulo **4.2**. A **opção 0** simplesmente termina a execução do programa.

Finalmente a **opção 7** permite definir se o grafo **será alterado ou não**, com isto queremos dizer que se, por exemplo, executarmos a **opção 5** sem alterar o grafo o resultado será baseado no mesmo grafo ao longo de várias iterações, mas, **se ativarmos** a alteração do grafo, será calculado como se a primeira iteração tivesse distruibuído suprimentos pelas rotas calculadas e alterado , assim, as necessidades das regiões do grafo.

Exemplificação de alteração do grafo com o algoritmo Greedy:



```
Veiculo    Heurística  Distancia  Entregas       Horas  Caminho

Drone1     0.28        278        50      100.0%  1.04   [Lisboa, Covilhã]
Drone2     0.27        278        50      100.0%  1.04   [Lisboa, Covilhã]
Carro1     1.88        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro2     1.69        312        1500    100.0%  4.68   [Lisboa, Porto]
Carro3     2.04        387        1500    100.0%  5.8    [Lisboa, Porto, Aveiro]
Avioneta1  2.93        278        1000    100.0%  2.08   [Lisboa, Covilhã]
Carro4     7.06        767        1500    100.0%  10.12  [Lisboa, Faro, Aveiro]
Carro5     6.69        265        1500    100.0%  6.03   [Lisboa, Coimbra, Aveiro]
Carro6     7.88        293        1400     93.3%  4.4    [Lisboa, Viseu]
Avioneta2  35.49       278        80        8.0%  2.08   [Lisboa, Covilhã]
Avioneta3  10.16       319        1000    100.0%  1.59   [Lisboa, Guarda]
Avioneta4  12.29       319        800      80.0%  1.59   [Lisboa, Guarda]
Avioneta5  27.56       133        1000    100.0%  1.66   [Lisboa, Évora]
Avioneta6  25.53       133        590      59.0%  1.66   [Lisboa, Évora]
Barquinho1 189.12      1497       2000    100.0%  27.22  [Lisboa, Ponta Delgada, Funchal]
Barquinho2 24.34       1497       550      27.5%  27.22  [Lisboa, Ponta Delgada, Funchal]
Camiao1    330.66      364        2800     28.0%  6.82   [Lisboa, Braga]
Camiao6    933.19      387        1040     10.4%  7.26   [Lisboa, Porto, Aveiro]
Camiao3    2142.35     1508       1220     12.2%  26.54  [Lisboa, Coimbra, Leiria, Lisboa, Coimbra, Leiria, Lisboa, Faro, Leiria]

Suplementos entregues = 21080/26050 - Capacidade dos veiculos = 49100(42.93% eficiencia)
Nodos nao visitados: ['Póvoa de Varzim', 'Castelo Branco', 'Washington', 'Nova York', 'Cuba', 'São Paulo', 'Angra do Heroísmo', 'Horta']
Veiculos nao usados: ['Barquinho3', 'Barquinho4', 'Barquinho5', 'Camiao5', 'Camiao4', 'Camiao2']
Nodos com necessidades: [('Póvoa de Varzim', 1120), ('Castelo Branco', 1200), ('Washington', 100), ('Nova York', 100), ('Cuba', 100), ('São Paulo', 100), ('Angra do Heroísmo', 115
0), ('Horta', 1100)]
Veiculo    Heurística  Distancia  Entregas       Horas  Caminho

Drone1     40.47       224        50      100.0%  3.48   [Lisboa, Castelo Branco]
Drone2     38.79       224        50      100.0%  3.48   [Lisboa, Castelo Branco]
Avioneta1  46.44       224        1000    100.0%  4.88   [Lisboa, Castelo Branco]
Avioneta2  248.62      224        100      10.0%  6.96   [Lisboa, Castelo Branco]
Avioneta3  127.54      347        1000    100.0%  4.92   [Lisboa, Póvoa de Varzim]
Avioneta4  552.37      347        120      12.0%  4.92   [Lisboa, Póvoa de Varzim]

Suplementos entregues = 2320/4970 - Capacidade dos veiculos = 4100(56.59% eficiencia)
Nodos nao visitados: ['Porto', 'Braga', 'Coimbra', 'Faro', 'Évora', 'Guarda', 'Viseu', 'Covilhã', 'Ponta Delgada', 'Washington', 'Nova York', 'Cuba', 'São Paulo', 'Funchal', 'Angr
a do Heroísmo', 'Horta', 'Leiria', 'Aveiro']
Veiculos nao usados: ['Carro1', 'Carro2', 'Carro3', 'Carro4', 'Carro5', 'Carro6', 'Camiao1', 'Camiao2', 'Camiao3', 'Camiao4', 'Camiao5', 'Camiao6', 'Barquinho1', 'Barquinho2', 'Ba
rquinho3', 'Barquinho4', 'Barquinho5', 'Avioneta5', 'Avioneta6']
Nodos com necessidades: [('Washington', 100), ('Nova York', 100), ('Cuba', 100), ('São Paulo', 100), ('Angra do Heroísmo', 1150), ('Horta', 1100)]
```

Podemos ver que na segunda iteração o progresso feito nas entregas da primeira foi registado. Para além disso as "**Horas**"**acumulam** ou seja, por exemplo, o **Drone1** demorou 1.04 horas para ir de Lisboa a Covilhã; na iteração seguinte vemos q demorou 3.48 horas mas este valor representa o tempo 1.04 vezes 2 pois é o tempo de **realizar** a sua **distribuição** e **voltar** a **Lisboa** pelo mesmo percurso, mais o tempo de **Lisboa a Castelo Branco** que foi 1.4 horas, o **Drone1** consegue percurrer esta distância mais rápido mas, dadas as condições meteorológicas más em Castelo Branco, demorou **2.5 vezes mais** tempo. (**Cálculo final = 1.04*2 + (224/400)*2.5**) Na iteração seguinte, se o drone fosse usado, íamos observar o mesmo comportamento.

Nota: 224/400 representa os 224 quilómetros a ser percorridos a 400 quilómetros hora pelo drone.

6 Sumário e discussão dos resultados obtidos

Nos nossos testes conseguimos obter os resultados que pretendíamos:

-Alterar os valores **M1**,**M2** e **R** tem as consequências esperadas nos resultados(por vezes não na primeira iteração).

-Incluir **mais arcos** que permitam transporte por camião ("Terra-Pesado"ou "Terra e Ar"ou "Tudo") ou o equivalente para outros veículos reflete-se nas rotas escolhidas.

-**Alterar valores dos nodos** desde a gravidade às condições meteorológicas, densidade populacional, suprimentos necessários entre outros, muda as prioridades da forma pretendida, influenciando a heurística e a solução final.

Para concluir, consideramos que **alcançamos** uma **solução boa** que permite distribuições de recursos justas. As **limitações** em termos de representação e dificuldade na escolha de valores tornaram complexa a criação de uma heurística, mas, tomamos boas decisões de implementação e conseguimos alcançar **soluções** perfeitamente **viáveis**.
A **aparência** do programa e as **estatísticas** convenientemente obtidas ao executar os algoritmos **facilitaram** não só o nosso trabalho como irão também facilitar profundamente a interpretação dos resultados por parte de usurários do programa e nesse aspeto estamos muito satisfeitos.

Conseguimos atingir boa correção no que toca à lógica do programa; realizamos diversos testes e verificamos os valores das nossas soluções para garantir que batiam certo com o esperado para atestar essa noção.

No que toca a **melhorarias futuras**, destaca-se a implementação de uma lógica mais robusta e detalhada relativa ao suprimentos e, para além disso, realizar uma otimização dos algoritmos que foi deixada de lado em troca de eficiência nos resultados finais.