



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Inteligência Artificial

Resolução de Problemas com Algoritmos de Procura

Ano Letivo de 2024/2025

Otimização de Distribuição Alimentar numa Catástrofe

Flávia Alexandra da Silva Araújo (A96587)
Joshua David Amaral Moreira (A105684)
Miguel Fernandes Barbosa (A104451)
Miguel Torres Carvalho (A95485)

4 de janeiro de 2025



Resumo

As catástrofes naturais, como furacões, terremotos e inundações, frequentemente deixam um rasto de destruição, tornando fundamental a gestão eficiente de recursos para mitigar o impacto humanitário. Em situações de emergência, otimizar a distribuição de suprimentos essenciais, como alimentos, água e medicamentos, pode ser a diferença entre salvar ou perder vidas.

Este trabalho propõe uma abordagem baseada em inteligência artificial, especificamente através de algoritmos de procura, para resolver o problema de logística em cenários de crise. A formulação do problema considera fatores como a perecibilidade dos alimentos, as limitações operacionais dos veículos, e a urgência das necessidades das áreas afetadas. O objetivo principal é minimizar o tempo de resposta, reduzindo desperdícios e maximizando a eficiência operacional.

Além disso, este estudo explora os diferentes resultados obtidos por diversos algoritmos de procura, bem como as diferentes definições de heurísticas influenciam os mesmos. Com uma análise focada na eficiência logística, tanto a nível de recursos utilizados como tempo de resposta.

Área de Aplicação: Inteligência Artificial, Distribuição de Recursos.

Palavras-Chave: Algoritmos de Procura, Heurísticas, A*, Gulosa, BFS, DFS, Procura Uniforme.

Índice

1	Descrição do Problema	1
2	Formulação do Problema	2
3	Usagem	3
4	Tarefas Realizadas e Decisões Tomadas	6
4.1	Grafos Criados	6
4.2	Implementação do Problema	8
4.2.1	Classes Implementadas	8
4.2.2	Operadores Implementados	9
4.2.3	Implementação de Restrições	10
4.3	Algoritmos de Procura Utilizados	10
4.4	Heurísticas Desenvolvidas	11
4.5	Estratégias para a Resolução do Problema	12
5	Discussão dos Resultados Obtidos	13
5.1	Resultados com BFS	13
5.2	Resultados com DFS	15
5.3	Resultados com Uniforme	17
5.4	Resultados com Gulosa	19
5.5	Resultados com A*	21
5.6	Análise dos Resultados	23
6	Conclusão e Trabalho Futuro	24
6.1	Conclusão	24
6.2	Trabalho Futuro	24
	Referências	25

Índice de Figuras

3.1	Menu Principal	3
3.2	Submenu de Simulações	3
3.3	Submenu Gráfico	4
3.4	Submenu de Algoritmos de Procura	5
3.5	Submenu de Heurísticas	5
4.1	Grafo de Simulação de uma Cidade Média	6
4.2	Grafo de Simulação do Arquipélago dos Açores	7
4.3	Grafo Teste da Simulação	7

Índice de Tabelas

1.1 Características dos Veículos	1
--	---

1 Descrição do Problema

Este problema trata-se de um cenário de gestão de recursos em situações de emergência, onde é necessário coordenar a distribuição de suprimentos em zonas afetadas por desastres naturais. O objetivo é otimizar a distribuição de recursos, minimizando o tempo de resposta e garantindo a satisfação das necessidades das zonas afetadas.

O problema é composto por um conjunto de localizações, algumas das quais são zonas afetadas (e, de entre estas, algumas com maior urgência), com diferentes níveis de acesso. Para chegar a estas zonas, é necessário utilizar veículos que partem de diferentes localizações. Existem diferentes tipos de veículos: mota, carro, camião, *drone*, helicóptero, avião, e barco. Cada veículo tem uma capacidade de carga, velocidade, capacidade de combustível e gasto de combustível diferente. Como tal, deve ser feita uma gestão eficiente dos mesmos, de forma a que sejam utilizados de maneira a otimizar as suas capacidades de transporte.

Veículo	Tipo	Capacidade (kg)	Velocidade (km/h)	Cap. Comb. (L)	Cons. Comb. (L/100km)
Mota	Terrestre	10	120	15	4
Carro	Terrestre	500	160	50	8
Camião	Terrestre	2000	100	300	30
Drone	Aéreo	5	100	5	2
Helicóptero	Aéreo	700	250	300	70
Avião	Aéreo	800	250	200	40
Barco	Marítimo	5000	50	1000	50

Tabela 1.1: Características dos Veículos

Os suprimentos a distribuir são de diferentes tipos: alimentos, água e *kits* de primeiros socorros. Cada zona afetada tem necessidades específicas, que devem ser satisfeitas com a distribuição dos suprimentos. A distribuição dos suprimentos deve ser feita de forma a minimizar o tempo de resposta e o uso de combustível, garantindo que as zonas com maior urgência são atendidas primeiro, bem como minimizar o desperdício de suprimentos perecíveis.

2 Formulação do Problema

- **Tipo:** Problema de Procura de Múltiplos Estados
- **Estado Inicial:** Conjunto de nodos com suprimentos para distribuir. Zonas afetadas (catástrofes) com respetivas necessidades, e veículo(s) inicialmente estacionados, por um ou mais nodos, com combustível cheio e carga vazia.
- **Estado Objetivo:** Todas as zonas afetadas atendidas, com suprimentos distribuídos de forma eficiente, minimizando desperdícios de suprimentos perecíveis e combustível, assim como o tempo de resposta.
- **Operadores (Ações Disponíveis):**
 - *Mover Veículo:* Deslocar um veículo de uma zona para outra, respeitando as limitações de combustível e condições da rota.
 - *Carregar Suprimentos:* Carregar suprimentos no veículo, respeitando a capacidade de carga do veículo.
 - *Descarregar Suprimentos:* Descarregar suprimentos existentes do veículo.
 - *Reabastecer Combustível:* Realizar paragens no percurso para reabastecimento dos veículos, tendo em conta o tempo necessário e as capacidades de combustível, tanto no nodo, como no veículo.
- **Custo da Solução:** O custo é calculado com base no tempo de resposta, combustível utilizado, e suprimentos distribuídos.

3 Usagem

O programa é executado através de `src/main.py`. O programa aceita o argumento `--verbose` para mostrar informação detalhada dos algoritmos de pesquisa, o que pode ser desativado ou ativado no menu principal.

Ao executar o programa, é apresentado um menu principal, onde o utilizador pode escolher entre diferentes opções:

```
Main menu:
1 -> Change simulation
2 -> View graph menu
3 -> View catastrophes, fleet and supplies
4 -> Search menu
9 -> Enable verbose mode
0 -> Exit
Enter the option: █
```

Figura 3.1: Menu Principal

Caso o utilizador escolha a opção 1, é apresentado um menu para escolher a simulação a utilizar:

```
Currently selected: 1
Change to:
1 -> Medium city with 10 nodes
2 -> Azores archipelago with 9 nodes
3 -> Simulation test
0 -> Back
Enter the option: █
```

Figura 3.2: Submenu de Simulações

Caso o utilizador escolha a opção 2, é apresentado o menu de visualização do grafo. Neste, é possível imprimir o grafo, assim como nodos, arestas e valores heurísticos, de forma textual, bem como é dada a opção de visualizar o grafo em formato de imagem, recorrendo à biblioteca `graphviz`, ou `matplotlib`.

```
View menu:
1 -> Draw graph with graphviz
2 -> Draw graph with matplotlib
3 -> Print graph
4 -> Print graph nodes
5 -> Print graph edges
6 -> Print heuristic values
0 -> Back
Enter the option: █
```

Figura 3.3: Submenu Gráfico

Caso o utilizador escolha a opção 3, é apresentado em formato *JSON* a lista de catástrofes, frota de veículos, e suprimentos, com os detalhes respetivos de cada um.

Caso o utilizador escolha a opção 4, é apresentado o menu de algoritmos de procura, onde é possível executar os algoritmos disponíveis: BFS, DFS, UCS, A* e Gulosa. Também é possível escolher a heurística a utilizar para algoritmos de procura informada.

```
Search menu:  
1 -> Breadth-first search  
2 -> Depth-first search  
3 -> Uniform-cost search  
4 -> Greedy search  
5 -> A* search  
9 -> Change heuristic  
0 -> Back  
Enter the option: █
```

Figura 3.4: Submenu de Algoritmos de Procura

```
Currently selected: 1  
Change heuristic function:  
1 -> Heuristic 1  
2 -> Heuristic 2  
3 -> Heuristic 3  
0 -> Back  
Enter the option: █
```

Figura 3.5: Submenu de Heurísticas

Caso o utilizador escolha a opção 9, alterna o modo verboso, que, como supramencionado, informação detalhada dos algoritmos de pesquisa, como o conjunto de veículos que conseguem aceder a uma catástrofe a tempo, bem como os veículos designados para cada catástrofe.

A opção 0 permite terminar a execução do programa.

4 Tarefas Realizadas e Decisões Tomadas

4.1 Grafos Criados

O grafo correspondente ao problema é composto por um conjunto de vértices, que representam as localizações, e um conjunto de arestas, que representam as ligações entre essas. Cada aresta tem a respetiva distância, o tipo desta ligação, o nível de acessibilidade, e uma percentagem de multiplicador de velocidade. Na representação gráfica com a biblioteca `graphviz`, as arestas podem ter diferentes cores - azul para ligações marítimas, verde para ligações terrestres, e vermelho para ligações aéreas -, bem como diferentes espessuras, que representam o nível de acessibilidade entre as localizações - quanto mais espessa a aresta, mais tipos de veículos podem circular por ela. Por exemplo, uma aresta fina só pode ser percorrida por veículos ligeiros, como motas, *drones*, ou barcos pequenos, dependendo do método de viagem dessa aresta.

Foram efetuados os seguintes grafos para representar o problema:

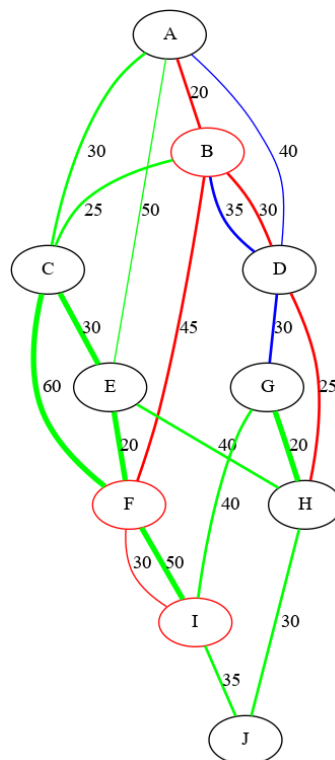


Figura 4.1: Grafo de Simulação de uma Cidade Média

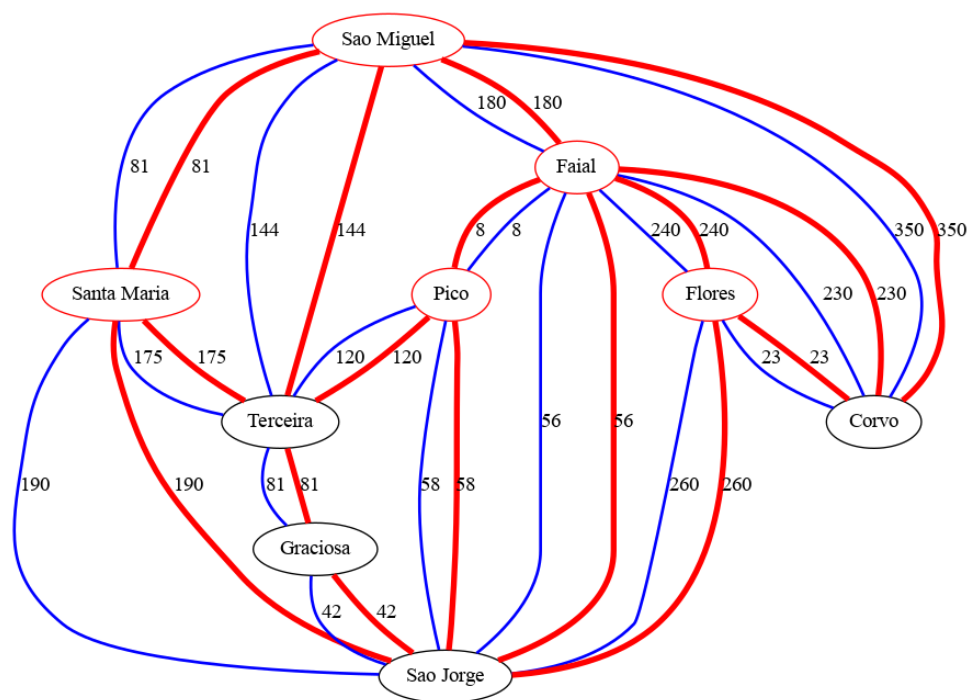


Figura 4.2: Grafo de Simulação do Arquipélago dos Açores

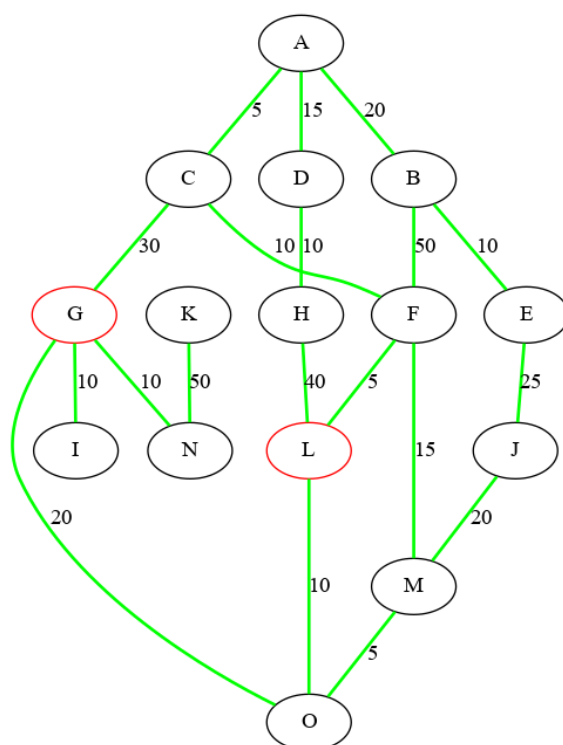


Figura 4.3: Grafo Teste da Simulação

4.2 Implementação do Problema

4.2.1 Classes Implementadas

Para a implementação do problema, foram criadas os seguintes módulos:

- `algorithms/`
 - `bfs`: Implementa o algoritmo de procura em largura;
 - `dfs`: Implementa o algoritmo de procura em profundidade;
 - `ucs`: Implementa o algoritmo de procura de custo uniforme;
 - `astar`: Implementa o algoritmo A*;
 - `greedy`: Implementa o algoritmo Gulosa.
- `graph/`
 - `graph`: Representa um grafo com atributos como nós, arestas, custos e valores heurísticos;
 - `node`: Representa um nó no grafo, e os seus dados, como nome, combustível, catástrofe, veículos e suprimentos;
 - `heuristics`: Implementa três diferentes heurísticas para os algoritmos de procura A* e Gulosa, que são usadas para calcular os valores heurísticos dos nodos;
- `catastrophe`: Armazena informações sobre uma catástrofe, como tempo de resposta e demanda de suprimentos;
- `vehicle`: Modela veículos com atributos como nome, categoria, velocidade, capacidade de carga e métodos para gerenciar combustível e carga;
- `supply`: Armazena informações sobre um suprimento, como tipo, quantidade e tempo de perecibilidade;
- `mission_planner`: É responsável por planejar a missão. Chama os algoritmos de procura e retorna o melhor caminho para cada veículo, considerando os suprimentos e catástrofes no ambiente. Armazena o grafo, a frota de veículos, as catástrofes e os suprimentos;
- `operation`: Representa o conjunto de operações *start*, *move*, *load*, *drop* e *refuel* dos veículos, e onde acontecem;

- `simulation_data`: Inicializa a simulação criando catástrofes e veículos, e inciliando o `Mission Planner`. É usada para definir os dados iniciais dos quais, grafo, veículos, suprimentos e nodos de catástrofes, bem como a definição de função de heurística a ser utilizada e nodos e arestas destrutíveis;
- `main`: Módulo principal que gere a execução do programa, e interage com o utilizador através de menus.

4.2.2 Operadores Implementados

Foram implementados os seguintes operadores para simular as ações possíveis:

- `Start`: Indica o nodo onde o veículo começa a sua missão (somente usado para aprimoração do *output*);
- `Move`: Desloca o veículo entre localizações, considerando consumo de combustível;
- `Load`: Carrega suprimentos no veículo, respeitando a capacidade de carga;
- `Drop`: Descarrega suprimentos do veículo;
- `Refuel`: Simula o tempo de reabastecimento (6 segundos por litro).

4.2.3 Implementação de Restrições

Os veículos começam a sua missão com o combustível cheio (de acordo com as capacidades correspondentes), e com a carga de suprimentos vazia. Logo no primeiro nodo, são carregados suprimentos de acordo com a capacidade do veículo.

A cada movimentação, é diminuído o combustível disponível, até que o veículo necessite de reabastecimento, cujo tempo é calculado com base na capacidade de combustível e no tempo de abastecimento de um litro: 6 segundos.

Os suprimentos também vão diminuindo à medida que são descarregados, e são reabastecidos nos locais de abastecimento nos nodos (para conveniência, existe uma quantidade infinita de suprimentos, bem como combustível).

Também foram implementadas destruições de nodos, bem como de arestas, que representam o bloqueio de certos caminhos devido a desastres naturais. Quando um nodo do grafo é destruído, o veículo que lá se encontra é dado como perdido e todas as arestas que o ligam a outros nodos são removidas. Como supramencionado, a destruição de nodos e/ou arestas são definidas no ficheiro `simulation_data.py`. Os algoritmos de procura não têm conhecimento atempado dessas condições de destruição, sendo assim, o programa responsável por se adaptar dinamicamente a essas condições.

O grau de urgência de uma catástrofe é dado pelo tempo de chegada e o seu tempo de resposta máximo.

4.3 Algoritmos de Procura Utilizados

Para a resolução do problema, foram utilizados os seguintes algoritmos de procura:

- **Breadth-First Search (BFS):** Procura em Largura, que expande todos os nós de um nível antes de passar para o próximo.
- **Depth-First Search (DFS):** Procura em Profundidade, que expande um nó até ao limite antes de retroceder.
- **Uniform-Cost Search (UCS):** Procura de Custo Uniforme, que expande o nó com menor custo de caminho.
- **A* Search:** Procura A*, que combina a procura gulosa com a procura de custo uniforme, utilizando o valor heurístico.
- **Greedy:** Procura Gulosa, que expande o nó que parece mais promissor, com no valor heurístico.

4.4 Heurísticas Desenvolvidas

Para os algoritmos A* e Gulosa, foram criadas as seguintes heurísticas:

- **Distância:** Distâncias para cada catástrofe por tipo de veículo e nível de acesso;
- **Tempo, Distância e Combustível:** Prioridades para cada catástrofe por tipo de veículo, nível de acesso, tempo de chegada do veículo - tempo de resposta na catástrofe e combustível: $\text{distance} + \text{time_arrival_vehicle} - \text{catastrophe_response_time} + \text{fuel}$;
- **Tempo, Distância, Combustível e Carga:** Prioridades para cada catástrofe por tipo de veículo, nível de acesso, tempo de chegada do veículo - tempo de resposta na catástrofe, carga do veículo/demanda da catástrofe e combustível: $\text{distance} + \text{time_arrival_vehicle} - \text{catastrophe_response_time} + \text{fuel} + \text{carga}$.

A distância é calculada com recurso ao algoritmo de Dijkstra (Custo Uniforme), que calcula o caminho mais curto entre dois nodos, para cada tipo de veículo.

Esta decisão permitiu a utilização de uma heurística mais precisa, e consequentemente com melhores resultados, sendo esta mais precisa que por exemplo a distância Euclidiana ou a distância de Manhattan.

O cálculo dos valores heurísticos é realizado na inicialização do Mission Planner, ou quando o utilizador escolhe outra heurística a utilizar, em vez de iterativamente nos algoritmos de pesquisa, devido ao cálculo da distância ser custoso computacionalmente.

4.5 Estratégias para a Resolução do Problema

Um dos maiores desafios deste projeto foi garantir que os veículos são utilizados de forma conjunta, de forma a otimizar a distribuição dos suprimentos. Para tal, começou-se por definir quais dos veículos conseguem aceder as diferentes catástrofes do grafo a tempo, utilizando o algoritmo de pesquisa escolhido pelo utilizador.

Com estes dados, foi desenvolvida uma função que permite, para cada catástrofe, a escolha do veículo mais adequado para a sua resolução, baseado no número de veículos que conseguem aceder a essa catástrofe, o consumo de combustível e a capacidade de carga do veículo.

Na primeira estrutura de dados, é também guardado o conjunto de operações calculadas pelo algoritmo para o veículo no auxílio da catástrofe, de forma a não ser necessário recalculá-las essas operações.

Quando o nodo ou uma aresta são destruídos, a estrutura de dados é recriada, assim como os veículos são revinculados às catástrofes que conseguem aceder.

5 Discussão dos Resultados Obtidos

5.1 Resultados com BFS

Utilizando o algoritmo *Breadth-First Search*, foi possível obter os seguintes resultados:

Tempo de Resposta:

O tempo de resposta foi de 575 segundos para a simulação de uma cidade média, 288 segundos para a simulação do arquipélago dos Açores, e 38 segundos para a simulação de teste.

Combustível Utilizado:

O combustível utilizado foi de 36,6 litros para a simulação de uma cidade média, 382,61 litros para a simulação do arquipélago dos Açores, e 9 litros para a simulação de teste.

Caminho Percorrido:

O caminho percorrido foi o seguinte:

- **Cidade Média:**

- Mota: $A \rightarrow C \rightarrow F \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I$
- Carro: $A \rightarrow C \rightarrow F \rightarrow E \rightarrow F$
- Drone: $A \rightarrow B \rightarrow A$ (várias vezes)

- **Açores:**

- Avião: $Graciosa \rightarrow SaoJorge \rightarrow Pico \rightarrow Faial$
- Helicóptero: $Graciosa \rightarrow Terceira \rightarrow Pico$
- Barco: $Terceira \rightarrow SantaMaria$
- Barco2: $Terceira \rightarrow SaoMiguel$

- **Teste:**

- Mota: $A \rightarrow C \rightarrow G \rightarrow N \rightarrow G$
- Carro: $A \rightarrow B \rightarrow F \rightarrow L$

5.2 Resultados com DFS

Utilizando o algoritmo *Depth-First Search*, foi possível obter os seguintes resultados:

Tempo de Resposta:

O tempo de resposta foi de 575 segundos para a simulação de uma cidade média, 288 segundos para a simulação do arquipélago dos Açores, e 68 segundos para a simulação de teste.

Combustível Utilizado:

O combustível utilizado foi de 37.2 litros para a simulação de uma cidade média, 311.91 litros para a simulação do arquipélago dos Açores, e 10.61 litros para a simulação de teste.

Caminho Percorrido:

O caminho percorrido foi o seguinte:

- **Cidade Média:**

- Mota: $A \rightarrow E \rightarrow H \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I$
- Carro: $A \rightarrow C \rightarrow F \rightarrow E \rightarrow F$
- Drone: $A \rightarrow B \rightarrow A$ (várias vezes)

- **Açores:**

- Avião: $Graciosa \rightarrow SaoJorge \rightarrow Pico \rightarrow Faial$
- Helicóptero: $Graciosa \rightarrow SaoJorge \rightarrow Pico$
- Barco: $Terceira \rightarrow SantaMaria$
- Barco2: $Terceira \rightarrow SaoMiguel$

- **Teste:**

- Mota: $A \rightarrow D \rightarrow H \rightarrow L \rightarrow O \rightarrow G \rightarrow N \rightarrow G \rightarrow N \rightarrow G$
- Carro: $A \rightarrow D \rightarrow H \rightarrow L$

5.3 Resultados com Uniforme

Utilizando o algoritmo *Uniform-Cost Search*, foi possível obter os seguintes resultados:

Tempo de Resposta:

O tempo de resposta foi de 575 segundos para a simulação de uma cidade média, 288 segundos para a simulação do arquipélago dos Açores, e 38 segundos para a simulação de teste.

Combustível Utilizado:

O combustível utilizado foi de 36.6 litros para a simulação de uma cidade média, 382.61 litros para a simulação do arquipélago dos Açores, e 8.2 litros para a simulação de teste.

Caminho Percorrido:

O caminho percorrido foi o seguinte:

- **Cidade Média:**

- Mota: $A \rightarrow C \rightarrow F \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I$
- Carro: $A \rightarrow C \rightarrow F \rightarrow E \rightarrow F$
- Drone: $A \rightarrow B \rightarrow A$ (várias vezes)

- **Açores:**

- Avião: $Graciosa \rightarrow SaoJorge \rightarrow Faial$
- Helicóptero: $Graciosa \rightarrow Terceira \rightarrow Pico$
- Barco: $Terceira \rightarrow SantaMaria$
- Barco2: $Terceira \rightarrow SaoMiguel$

- **Teste:**

- Mota: $A \rightarrow C \rightarrow G \rightarrow N \rightarrow G \rightarrow N \rightarrow G$
- Carro: $A \rightarrow D \rightarrow H \rightarrow L$

5.4 Resultados com Gulosa

Utilizando o algoritmo da Gulosa com a primeira heurística, foi possível obter os seguintes resultados:

Tempo de Resposta:

O tempo de resposta foi de 575 segundos para a simulação de uma cidade média, 288 segundos para a simulação do arquipélago dos Açores, e 38 segundos para a simulação de teste.

Combustível Utilizado:

O combustível utilizado foi de 35,8 litros para a simulação de uma cidade média, 311,91 litros para a simulação do arquipélago dos Açores, e 4,6 litros para a simulação de teste.

Caminho Percorrido:

O caminho percorrido foi o seguinte:

- **Cidade Média:**

- Mota: $A \rightarrow E \rightarrow F \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I$
- Carro: $A \rightarrow C \rightarrow F \rightarrow E \rightarrow F$
- Drone: $A \rightarrow B \rightarrow A$ (várias vezes)

- **Açores:**

- Avião: $Graciosa \rightarrow SaoJorge \rightarrow Faial$
- Helicóptero: $Graciosa \rightarrow SaoJorge \rightarrow Pico$
- Barco: $Terceira \rightarrow SantaMaria$
- Barco2: $Terceira \rightarrow SaoMiguel$

- **Teste:**

- Mota: $A \rightarrow C \rightarrow G \rightarrow N \rightarrow G \rightarrow N \rightarrow G$
- Carro: $A \rightarrow C \rightarrow F \rightarrow L$

5.5 Resultados com A*

Utilizando o algoritmo A* com a primeira heurística, foi possível obter os seguintes resultados:

Tempo de Resposta:

O tempo de resposta foi de 575 segundos para a simulação de uma cidade média, 288 segundos para a simulação do arquipélago dos Açores, e 38 segundos para a simulação de teste.

Combustível Utilizado:

O combustível utilizado foi de 36,4 litros para a simulação de uma cidade média, 311,91 litros para a simulação do arquipélago dos Açores, e 4,6 litros para a simulação de teste.

Caminho Percorrido:

O caminho percorrido foi o seguinte:

- **Cidade Média:**

- Mota: $A \rightarrow E \rightarrow F \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I \rightarrow J \rightarrow I$
- Carro: $A \rightarrow C \rightarrow F \rightarrow E \rightarrow F$
- Drone: $A \rightarrow B \rightarrow A$ (várias vezes)

- **Açores:**

- Avião: $Graciosa \rightarrow SaoJorge \rightarrow Faial$
- Helicóptero: $Graciosa \rightarrow SaoJorge \rightarrow Pico$
- Barco: $Terceira \rightarrow SantaMaria$
- Barco2: $Terceira \rightarrow SaoMiguel$

- **Teste:**

- Mota: $A \rightarrow C \rightarrow G \rightarrow N \rightarrow G \rightarrow N \rightarrow G$
- Carro: $A \rightarrow C \rightarrow F \rightarrow L$

5.6 Análise dos Resultados

Os resultados mostram que, entre os algoritmos de procura informada, A* e Gulosa apresentaram desempenhos muito semelhantes no consumo de combustível, graças ao uso de uma heurística completa. Esta característica permitiu que ambos orientassem eficientemente a procura, resultando em trajetos otimizados e diferenças mínimas entre os dois métodos.

Nos algoritmos de procura cega, o UCS destacou-se por obter trajetos mais eficientes em termos de consumo de combustível quando comparado com BFS e DFS. Este desempenho superior deve-se ao facto de o UCS considerar os custos acumulados ao longo do caminho, enquanto BFS e DFS apenas seguem critérios de ordem ou profundidade, sem avaliar os custos das ações.

Em termos de tempos de resposta, não foram observadas diferenças significativas entre os algoritmos, sugerindo que a complexidade intrínseca dos cenários testados foi o principal fator de variação.

Conclui-se que, enquanto os métodos informados beneficiaram diretamente da heurística completa, o UCS revelou-se a opção mais robusta entre os algoritmos cegos, destacando-se pelo equilíbrio entre simplicidade e eficiência no consumo de recursos.

6 Conclusão e Trabalho Futuro

6.1 Conclusão

O trabalho realizado demonstra de forma clara a eficácia dos algoritmos de procura em resolver problemas logísticos complexos em cenários de emergência. Destaca-se a superioridade do algoritmo A* e do UCS, em diferentes contextos, no equilíbrio entre eficiência de recursos e tempo de resposta. Além disso, a utilização de heurísticas completas evidenciou a importância de critérios bem definidos na melhoria de resultados.

6.2 Trabalho Futuro

A nível de trabalho futuro, destacam-se as seguintes áreas:

- **Perecibilidade dos Alimentos:** Implementar um modelo mais realista para a perecibilidade dos alimentos, considerando que, a partir do momento em que os suprimentos são abastecidos no veículo, o tempo de deterioração começa a contar. Isto seria efetuado através de um contador de tempo que simula a deterioração dos alimentos, e que seria considerado na escolha de rotas e prioridades.
- **Reutilização de Veículos noutras Operações:** Implementar a capacidade do veículo, após ter completado as suas operações, procurar outras zonas afetadas em que o veículo que está a ajudar ainda não tenha, nem vá completar, ou zonas afetadas sem quaisquer veículos a ajudar. Deste modo, este veículo poderia ser reutilizado nessa catástrofe, completando-a mais rapidamente. Tal seria implementado pelo registo dos veículos atribuídos a uma catástrofe, e, no fim de um veículo completar as suas operações, procurar primeiramente, por catástrofes sem veículos atribuídos, ou catástrofes com veículos atribuídos, mas que não vão completar a tempo.

Referências

- [1] Russell, S., & Norvig, P. (2009). *Artificial Intelligence - A Modern Approach* (3rd ed.). Pearson Education. ISBN-13: 9780136042594.
- [2] Costa, E., & Simões, A. (2008). *Inteligência Artificial - Fundamentos e Aplicações*. FCA. ISBN: 978-972-722-34.