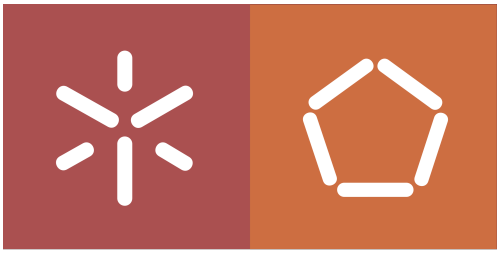

Universidade do Minho
Escola de Engenharia

Inteligência Artifical

Trabalho Prático

Grupo 38

Gonçalo Alves	**João Cunha**	**João Sá**
a104079	a104611	a104612

Indíce

Indíce de Figuras

1. Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Inteligência Artificial da Licenciatura em Engenharia Informática da Universidade do Minho e teve como objetivo o desenvolvimento e teste de algoritmos de procura eficientes. O problema abordado está relacionado com a distribuição de alimentos em regiões afetadas por catástrofes naturais, um desafio com grande importância prática e impacto social.

Neste cenário, pretendeu-se criar soluções computacionais que otimizassem a alocação de recursos, assegurando uma distribuição eficaz e equitativa. Para tal, foi necessário considerar múltiplos fatores, como a prioridade de entrega a áreas mais necessitadas, as limitações de capacidade dos veículos, as condições climáticas e os bloqueios de rotas. O objetivo principal foi encontrar soluções que equilibrassem a eficiência das operações com a urgência imposta pelo contexto de emergência.

O trabalho envolveu a formalização do problema como um problema de procura, permitindo a aplicação de algoritmos de procura específicos. Estas abordagens foram implementadas e testadas em diferentes cenários, considerando variáveis como o número de áreas a atender, as restrições dos veículos e as condições das rotas. As soluções desenvolvidas foram avaliadas em termos de desempenho e adaptabilidade às exigências do problema.

Este relatório apresenta os detalhes mais importantes do processo de desenvolvimento e os resultados obtidos, destacando o potencial dos algoritmos de procura na resolução de problemas complexos e relevantes. Acredita-se que as soluções propostas oferecem uma base sólida para futuras melhorias e aplicações em cenários reais.

2. Formulação do Problema

Estamos perante um ambiente parcialmente acessível, logo o agente não consegue obter informações completas sobre o estado do ambiente ou sobre os resultados das suas ações em todos os momentos, o que pode ser descrito também como ambiente não determinístico. Esta falta de informações deve-se a mudanças que podem aparecer nos caminhos, como por exemplo o bloqueio de estradas. Por estas razões, podemos afirmar que o problema é um problema de contingência, onde as perceções fornecem novas informações sobre o estado atual e onde é frequentemente intercalada a procura e a execução. A sua formulação pode ser efetuada da seguinte forma:

- **Estado Inicial:** Frota de veículos, totalmente carregados com alimentos e suprimentos, na base da empresa *PathFinder* em Pedome. Mapeamento das áreas afetadas com as suas necessidades e gravidade. Condições geográficas e meteorológicas iniciais

- **Objetivo:** Alcançar uma distribuição de suprimentos que minimiza o tempo de entrega e maximiza o número de zona atendidas, dentro do tempo crítico de cada uma. Priorizar zonas com maior densidade populacional e maior gravidade (maior prioridade) e minimizar o desperdício de recursos.

-**Ações Disponíveis:** As ações representam os movimentos ou decisões que cada veículo pode tomar:
  1- Mover para outra zona (o veículo viaja de uma zona para outra, com custo de tempo)
  2- Distribuir suprimentos (O veículo deixa parte de sua carga na zona onde se encontra)
  3- Reabastecer *(Se aplicável)*

- **Custo da Solução:** Tempo (para uma solução ótima, este valor deve ser o mais baixo possível)

2.1. Mapa

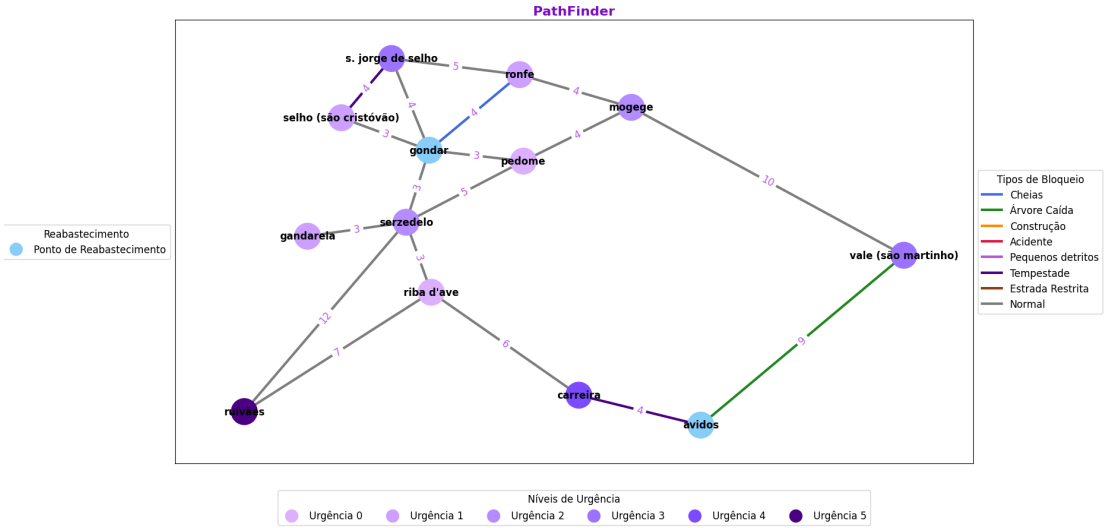

Figura 1: Mapa

Na imagem acima estão representados os nodos que decidimos utilizar para o trabalho. Cada nodo representa uma freguesia, nos arredores de Guimarães. As diferentes cores representam níveis de urgência e postos de reabastecimento. Os nodos estão interligados por ruas (arestas), que podem ou não estar bloqueadas (existem diferentes tipos de bloqueio, tal como representado na legenda).

2.2. PathFinder

A *PathFinder* foi uma empresa que decidimos criar, como o cérebro, por detrás desta alocação de recursos. É uma empresa sem fins lucrativos, cujo único objetivo é distribuir recursos recorrendo a veículos. Na sua sede, a empresa conta com os seguintes veículos:

1. Camião
2. Carrinha
3. Moto
4. Drone M
5. Drone S

Cada veículo tem uma **capacidade de carga**, **velocidade média** e um **id**.

3. Funcionalidades Implementadas

Para realizar esta fase do trabalho prático, implementámos tanto algoritmos de procura não informada (BFS, DFS, Custo Uniforme) como informada (A*, Greedy, Hill Climbing) para obter um caminho entre um local e o destino e cuja formulação já foi mencionada.

No entanto, como queríamos entregar a vários nodos, decidimos fazer uma função auxiliar que executava os algoritmos para cada nodo da lista. Essa lista consiste nos nodos, ordenados crescentemente por urgência. Ou seja, os nodos com maior urgência serão atendidos em primeiro lugar. Decidimos apenas entregar ao nodo de destino (esta decisão restringe as entregas e implica tempos maiores)

Para cada destino, decidimos encontrar o melhor caminho (em relação ao tempo de cada veículo). Para isso, executamos o algoritmo para o destino com todos os veículos e selecionamos apenas o melhor. Este vai ser diferente em várias execuções devido às propriedades das estradas.

Para definir se uma estrada estava (ou não) bloqueada, definimos uma probabilidade (10%) para esta estar bloqueada. Cada estrada bloqueada tem um tipo de bloqueio, sendo que cada um, permite a passagem a certos veículos. Os tipos de bloqueio são:

- Cheias -> [Drone S, Drone M]
- Árvore Caída -> [Drone S, Drone M]
- Construção -> [Moto, Drone S, Drone M]
- Acidente -> [Drone M, Drone S]
- Pequenos detritos -> [Drone S, Drone M, Carrinha, Camião]
- Tempestade -> [Camião, Carrinha, Moto]
- Estrada Protegida -> [Camião]

Ao fim de cada entrega, existe uma probabilidade (2%) de alterar o estado de cada estrada. Notámos que se esse valor fosse maior, no final das iterações iam existar demasiadas estradas bloqueadas.

Para além disso, foram definidos 2 pontos de reabastecimento. Quando um veículo fica sem carga, este dirige-se ao ponto de reabastecimento que o algoritmo, a executar, encontrar.

Quando o destino é alcançado, o sistema simula a entrega da carga disponível no veículo, verificando se a quantidade de suprimentos entregue foi suficiente para atender totalmente as necessidades do destino.

Caso a entrega realizada não seja suficiente para atender às necessidades da freguesia, o veiculo dirige-se ao ponto de reabastecimento mais próximo e reabastece.

Após reabastecer no ponto mais próximo, o veículo retorna ao destino e realiza uma nova entrega. Esse processo repete-se até que todas as necessidades do destino sejam atendidas.

Foi implementada, também, a noção de tempo de cada nodo. À medida que um veículo se move, o tempo desse movimento é retirado ao tempo limite de cada nodo. Na eventualidade desse tempo limite atingir 0, o nodo fica inacessível e é exibida essa informação ao utilizador.

Permitimos, também, ao utilizador desenhar o gráfico (onde contém os nodos e o estado das estradas), listar os nodos (com a informação correspondente), as ruas (informação correspondente a cada aresta) e também os veículos disponíveis.

3.1. Algoritmos Implementados

3.1.1. Pesquisa não informada

Tal como referido acima, implementámos 3 algoritmos (DFS, BFS e Custo Uniforme). Estes algoritmos tentam alcançar um nodo objetivo sem ter em conta informações específicas.

3.1.1.1. DFS (Depth-First search)

Este algoritmo realiza uma procura não informada que avança para através da expansão do primeiro nodo do grafo, pesquisando o nodo objetivo num nível de profundidade cada vez maior, até que o alvo da pesquisa seja encontrado ou até encontrar um nodo que não possui mais nodos adjacentes por visitar. É ideal para cenários com múltiplas soluções e é eficiente em termos de memória, mas pode não encontrar a solução ótima. Por outro lado, é menos eficiente em grafos com grande profundidade. Para a implementação deste algoritmo, fazemos a procura entre dois nodos, considerando tanto bloqueios nas estradas como o tipo do veículo. A mesma também simula entregas no nodo destino (tal como referido anteriormente) e realiza reabastecimentos dinâmicos. Com verificações que permitem evitar ciclos e adptação às nossas condições, o algoritmo apresenta alguma flexibilidade e funcionalidade para resolver o problema. No entanto, existem vezes em que o algoritmo acaba por se mostrar extremamente ineficiente.

3.1.1.2. BFS (Breadth-First Search)

A procura em largura explora o grafo nível a nível, recorrendo a uma fila para armazenar os nodos em fila de espera e a uma lista para manter o registo de nodos já explorados (evitando ciclos).

Tal como o algoritmo anterior, este considera as restrições de estradas e veículos, reabastecimento/ entrega adicional e também é ineficiente em vários casos.

3.1.1.3. UCS (Uniform-cost Search)

Este algoritmo, em contrário aos outros 2, expande nodos com o menor custo acumulado. Para esse efeito, o algoritmo guarda o custo total do caminho do nodo inicial até ao nodo onde se encontra. Em certos casos, pode até se comportar como o bfs (caso as arestas tiverem o mesmo custo). Recorre a uma *priority queue*, ordenada pelo custo total para alcancar o nodo (a partir do inicial). Esta estrutura é pesada, logo o algoritmo acaba por ocupar bastante espaço. Tal como todos os outros, tem em conta as especificações necessários no nosso contexto. Este algoritmo, mostrou-se num dos melhores para o nosso contexto.

3.1.2. Pesquisa Informada

Tal como referido acima, implementámos algoritmos de procura informada, como o A*, o Greedy e o Hill Climbing. Estes algoritmos tentam alcançar um nodo objetivo utilizando informações específicas (heurística), que ajudam a guiar a busca de uma forma mais eficiente. Ao incorporar dados adicionais sobre o problema, estes métodos conseguem, em muitos casos, reduzir o espaço de busca e encontrar soluções mais rapidamente, especialmente em problemas onde a estimativa heurística é precisa e consistente. Para a nossa heurística, optámos pela distância em linha reta ao objetivo.

3.1.2.1. A* Search

O algoritmo A* é um algoritmo de procura informada que recorre ao custo acumulado desde o nodo inicial com uma estimativa da heurística até ao nodo objetivo. Para isso, recorre a uma lista onde estão os nodos por explorar e a outra que armazena os nodos visitados. De modo a reconstruir o caminho no final, é mantido o registo dos custos acumulados.

Tal como os outros algoritmos abordados, este algoritmo tem em conta as restrições impostas e nuances de reabastecimento e entregas adicionais.

Apesar do A* ser mais pesado computacionalmente, para o nosso problema, demonstrou ser o melhor algoritmo. Este facto permaneceu constante na maior parte dos testes. No entanto, julgamos que se a complexidade do trabalho aumentasse (mais nodos e mais restrições), o algoritmo iria ter algumas dificuldades devido à necessidade de efetuar diversos cálculos.

3.1.2.2. Greedy Search

O algoritmo *Greedy* tem uma funcionalidade parecida ao UCS, porém usa como guia a heurística. Para saber que nodo escolher, avalia as heurísticas (dos nodos de uma lista) e escolhe o nodo com o menor valor. O resultado, muitas das vezes, não é o mais eficiente.

O algoritmo considera as restrições relacionadas com bloqueios nas estradas e a compatibilidade com o tipo de veículo. Além disso, inclui a capacidade de lidar com reabastecimentos e entregas adicionais.

Esta estratégia costuma ser mais rápida mas como depende única e exclusivamente da heurística, apresentou resultados inconsistentes. O algoritmo A* utiliza a nuances deste algoritmo

3.1.2.3. Hill Climbing

O algoritmo *Hill Climbing* foi o último algoritmo de procura informada que implementámos. O algoritmo avalia os nodos adjacentes (ignorando os que já foram visitados), selecionando o nodo como menor heurística. Este algoritmo não guarda alternativas, ou seja, avança de uma forma local.

Tal como todos os outros algoritmos implementados, existe a verificação do veículo, das estradas e rebastecimentos/entregas.

Teoricamente, o *Hill Climbing* é eficiente em cenários simples, mas para o nosso contexto, nunca apresentou resultados consistentes. Como a heurística escolhida é bastante semelhante para todos os nós, o algoritmo provavelmente ficou maior parte das vezes preso em planaltos. Portanto, o algoritmo quase nunca apresentou uma solução para o problema.

4. Interface

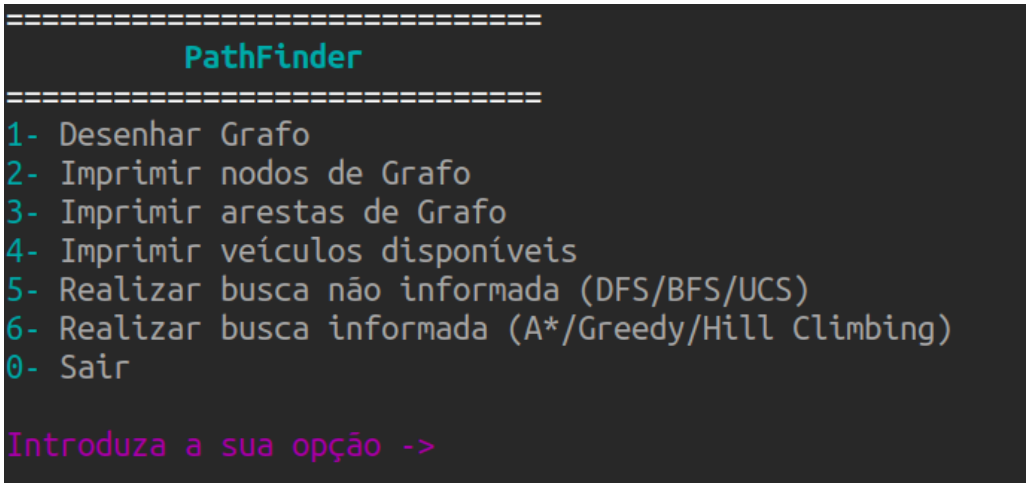

Figura 2: Menu inicial

A imagem acima representa o menu exibido ao executar o programa "PathFinder". Este menu contém 7 opções principais:

1. **Desenhar Grafo:** Gera uma representação gráfica semelhante ao grafo apresentado na seção 2.1.
2. **Imprimir Nós do Grafo:** Mostra as freguesias, suas necessidades e níveis de urgência.
3. **Imprimir Arestas do Grafo:** Exibe as estradas, seus estados de conservação e os veículos que as utilizam.
4. **Imprimir Veículos Disponíveis:** Lista o tipo de veículo, sua capacidade, autonomia e velocidade máxima.
5. **Realizar Busca Não Informada (DFS/BFS/UCS):** Permite executar uma das três estratégias de busca não informada disponíveis.
6. **Realizar Busca Informada (A/Greedy/Hill Climbing)**: Permite executar uma das três estratégias de busca informada disponíveis.
7. **Sair (0):** Encerra o programa.

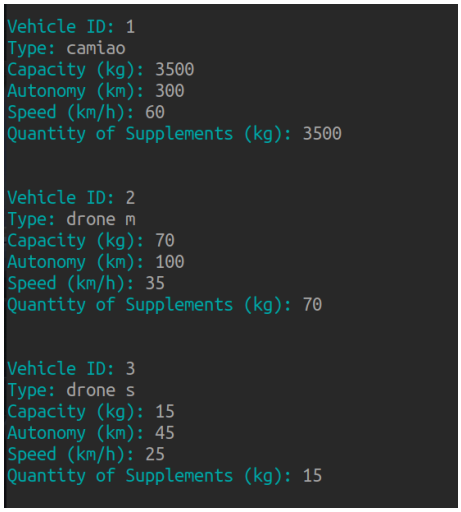

Figura 3: Listar veículos

5. Conclusões e resultados obtidos

No contexto do problema abordado, o trabalho evidenciou a eficácia dos algoritmos de procura na otimização da entrega de recursos em situações de emergência. Dentre as abordagens implementadas, os algoritmos **A*** e **UCS** destacaram-se como os mais consistentes, principalmente devido à simplicidade da heurística utilizada.

Como potenciais melhorias, destacam-se a possibilidade de desenvolver heurísticas mais robustas para melhorar a performance dos algoritmos e a implementação de estratégias para otimização do consumo de combustível, garantindo uma operação ainda mais eficiente.

O trabalho proporcionou uma visão prática e detalhada do impacto dos algoritmos de procura em problemas complexos, servindo como uma base sólida para o desenvolvimento de possíveis aplicações, como a otimização da logística humanitária ou a gestão de recursos em situações de crise.

6. Avaliação por pares

Gonçalo da Silva Alves (a104079) -> 0
João Manuel Machado da Cunha (a104611) -> 0
João Pedro Ribeiro de Sá (a104612) -> 0