



**Universidade do Minho**  
Escola de Engenharia  
Licenciatura em Engenharia Informática

## **Unidade Curricular de Inteligência Artificial**

Ano Letivo de 2024/2025

### **Trabalho Prático**

**Eduardo Faria**  
a104353

**Hélder Gomes**  
a104100

**Nuno Silva**  
a104089

**Pedro Pereira**  
a104082

3 de janeiro de 2025

Data da Receção	
Responsável	
Avaliação	
Observações	

## Trabalho Prático

**Eduardo Faria**  
a104353

**Hélder Gomes**  
a104100

**Nuno Silva**  
a104089

**Pedro Pereira**  
a104082

3 de janeiro de 2025

## **Avaliação pelos pares**

a104353 Eduardo Faria DELTA = 0

a104100 Hélder Gomes DELTA = 0

a104089 Nuno Silva DELTA = 0

a104082 Pedro Pereira DELTA = 0

## Resumo

Este trabalho explora o desenvolvimento de algoritmos de procura para a otimização na distribuição de recursos em zonas afetadas por catástrofes naturais. Com base numa representação em grafo das zonas e caminhos, foram implementadas soluções que priorizam áreas mais necessitadas, considerando restrições de capacidade dos veículos, tempos de trajetos e condições dinâmicas. As simulações realizadas mostram uma distribuição eficiente de recursos, minimizando desperdícios e maximizando a assistência prestada no menor tempo possível.

**Área de Aplicação:** Utilização da Inteligência Artificial na otimização de atividades de contexto real.

**Palavras-Chave:** Algoritmos de Procura, Otimização de Recursos, Catástrofes Naturais, Planeamento de Rotas, Inteligência Artificial.

# Índice

<b>1. Descrição do problema</b>	<b>1</b>
<b>2. Formulação do problema</b>	<b>3</b>
2.1. Tipo de Problema	3
2.2. Representação do Estado	3
2.3. Operadores (Ações Disponíveis)	3
2.4. Tipos de Transições	3
2.5. Função de Custo	3
2.6. Natureza do Problema	4
<b>3. Descrição das tarefas e decisões realizadas</b>	<b>5</b>
3.1. Estruturação	5
3.2. Algoritmia	7
3.2.1. Procura não informada	7
3.2.1.1. BFS (Busca em Largura)	7
3.2.1.2. DFS (Busca em Profundidade)	7
3.2.2. Procura informada	8
3.2.2.1. <i>Greedy</i>	8
3.2.2.2. A*	8
3.2.2.3. A* Dinâmico	8
3.3. Métricas de desempenho	8
3.3.1. Estruturação	8
3.3.2. Parametrização	9
3.3.2.1. Estratégia de Parametrização	9
3.3.2.2. Parâmetros Monitorizados	9
3.3.3. Resultados obtidos	9
3.4. Extras	13
3.4.1. Mapa de Portugal	13
<b>4. Sumário e discussão dos resultados obtidos</b>	<b>15</b>
4.1. Resumo dos resultados	15
4.2. Análise Crítica	15
4.3. Conclusão	16
<b>Referências</b>	<b>17</b>
<b>Lista de Siglas e Acrónimos</b>	<b>18</b>
<b>Anexos</b>	<b>19</b>
Anexo 1: Logo da Universidade do Minho	19
Anexo 2: Ilustração representativa de um exemplo de visualização do problema.	20
Anexo 3: Definição da entidade Zona através de uma classe.	20
Anexo 4: Definição da entidade Estrada através de uma classe.	21
Anexo 5: Definição da entidade Veículo através de uma classe.	21
Anexo 6: Grafo dos municípios de Portugal.	22
Anexo 7: Ponderação da métrica de tempo de execução.	23

Anexo 8: Ponderação da métrica de utilização de memória.	23
Anexo 9: Ponderação da métrica de visita a zonas.	24
Anexo 10: Ponderação da métrica de profundidade da solução.	24
Anexo 11: Ponderação da métrica de custo da solução.	25
Anexo 12: Ponderação da métrica de média de custo por conexão.	25
Anexo 13: Ponderação da métrica de percentagem de zonas visitadas.	26

## Lista de Figuras

Figura 1: Ilustração representativa de um exemplo de visualização do problema.	2
Figura 2: Definição da entidade Zona através de uma classe.	5
Figura 3: Definição da entidade Estrada através de uma classe.	6
Figura 4: Definição da entidade Veículo através de uma classe.	6
Figura 5: Partição da definição do Grafo através de uma classe.	7
Figura 6: Ponderação da métrica de tempo de execução.	10
Figura 7: Ponderação da métrica de utilização de memória.	10
Figura 8: Ponderação da métrica de visita a zonas.	11
Figura 9: Ponderação da métrica de profundidade da solução.	11
Figura 10: Ponderação da métrica de custo da solução.	12
Figura 11: Ponderação da métrica de média de custo por conexão.	12
Figura 12: Ponderação da métrica de percentagem de zonas visitadas.	13
Figura 13: Grafo dos municípios de Portugal.	14

# 1. Descrição do problema

O problema de distribuição de suprimentos essenciais em zonas afetadas por catástrofes naturais é caracterizado por uma série de desafios complexos. Cada zona apresenta necessidades distintas, variando desde locais densamente povoados até regiões isoladas e de difícil acesso. Estes fatores tornam essencial a criação de um sistema que maximize a eficiência da distribuição de recursos, atendendo às áreas mais críticas no menor tempo possível e com o mínimo desperdício.

Entre os principais desafios estão:

- **Prioridades de Distribuição:** É necessário priorizar as zonas com maior severidade e maior densidade populacional, garantindo a entrega rápida em áreas onde a situação pode deteriorar-se rapidamente.
- **Condições Dinâmicas:** Fatores como mudanças meteorológicas e bloqueios de rotas exigem uma abordagem adaptativa que possa reagir rapidamente a alterações no cenário.
- **Limitações Operacionais:** Cada veículo possui restrições de carga, autonomia e acesso, que influenciam diretamente a estratégia de distribuição.
- **Geografia Variável:** O terreno e a acessibilidade diferem entre as zonas, com condições associadas às estradas.

Diante destes desafios, o problema foi formulado como uma tarefa de otimização representada por um grafo, onde os nós correspondem às zonas de entrega e as arestas representam as rotas possíveis. A solução exige o desenvolvimento de algoritmos de procura que equilibrem os custos operacionais com a eficácia na entrega, otimizando o uso de recursos limitados e garantindo uma resposta eficaz em contextos de emergência.



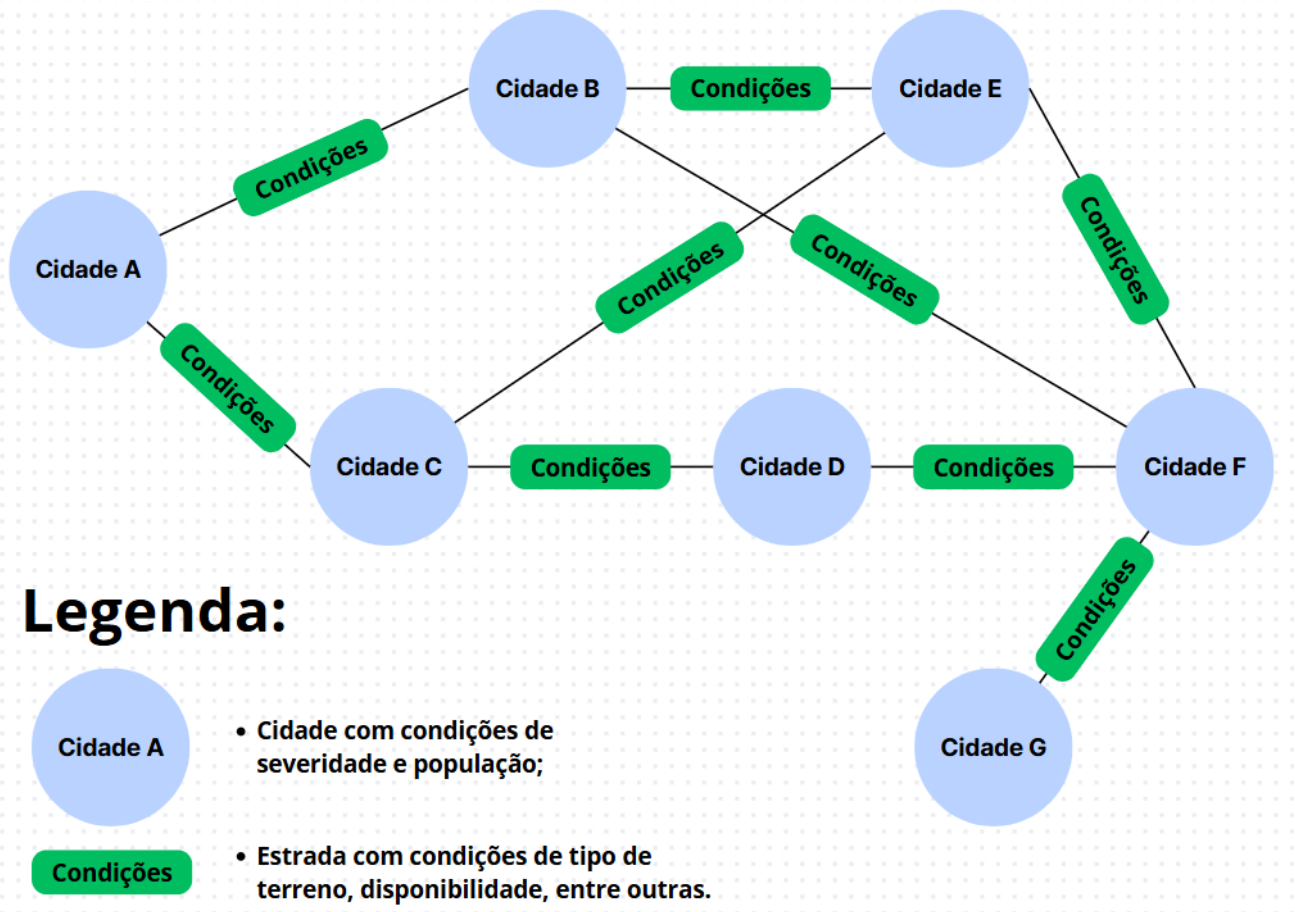


Figura 1: Ilustração representativa de um exemplo de visualização do problema.

## 2. Formulação do problema

Esta secção trata da caracterização do problema através de tópicos que facilitam a sua formulação que, por sua vez, capacita o grupo de trabalho de uma visão mais clarificada e concreta do problema a solucionar.

### 2.1. Tipo de Problema

- **Estado Único com Contingência:** Embora as ações possam ser planeadas com base nas condições conhecidas, o sistema deve ser capaz de adaptar-se a mudanças inesperadas, como bloqueios de rotas ou alterações meteorológicas.
- **Problema Semi-Determinístico:** Algumas ações e transições são previsíveis em condições fixas, mas há elementos dinâmicos que podem alterar o curso do planeamento inicial.

### 2.2. Representação do Estado

- **Estado Inicial:** Representa todos os suprimentos e o veículo que os transportará localizados na base principal.
- **Estado Final (Objetivo):** Atenderam-se o máximo de cidades possíveis até à chegada à cidade objetivo, com a entrega de recursos concluída.
- **Ambiente:** Conjunto de todas as possíveis configurações de distribuição de recursos e localização dos veículos, incluindo ajustes dinâmicos.

### 2.3. Operadores (Ações Disponíveis)

- **Deslocar Veículo:** Move um veículo de uma zona para outra ao longo de uma rota definida no grafo.
- **Entregar Recursos:** Reduz a carga do veículo e atualiza o estado da zona com os recursos recebidos.

### 2.4. Tipos de Transições

- **Transições Determinísticas:** Ações que levam diretamente de um estado a outro em condições estáveis.
- **Transições Contingentes:** Ações que requerem replaneamento devido a mudanças inesperadas, como rotas bloqueadas ou aumento da severidade numa zona específica.

### 2.5. Função de Custo

- **Custo Total:** Minimizar o gasto (custo) das operações.

- **Custo por Zona:** Minimizar desperdício de recursos em zonas menos prioritárias.
- **Custo por Estrada:** Minimizar más escolhas de ligação para escolha da próxima zona de acordo com as condições disponíveis.

## 2.6. Natureza do Problema

- **Exploração Completa e Adaptativa:** O algoritmo deve considerar todas as soluções iniciais possíveis, mas também ser capaz de reagir dinamicamente às mudanças no ambiente.
- **Planeamento Estático e Dinâmico:** A solução inicial é planeada com base nas condições conhecidas, mas ajustes são realizados em resposta a mudanças nas condições, garantindo robustez e flexibilidade.

## 3. Descrição das tarefas e decisões realizadas

Nesta secção são abordadas as decisões de implementação relativas à formulação do problema evidenciada no tópico anterior.

### 3.1. Estruturação

Inicialmente, foram determinadas quais as entidades do sistema e respetivas condições associadas, pelo que surgiu a seguinte configuração:

- **Zona:**
  - **Nome:** O nome da zona.
  - **Coordenada:** A coordenada geográfica da zona.
  - **Severidade:** A prioridade baseada na gravidade da situação.
  - **População:** O tamanho da população dentro da zona.
  - **Distância:** A distância em linha reta até à zona de destino.
  - **Heurística:** O valor de heurística da zona em específico.
  - **Suprimentos:** A quantidade de suprimentos entregues à zona.

```
1 class Zone:
2     """
3     Class representing a disaster-affected zone.
4
5     Attributes:
6         name (string): The name of the zone.
7         coordinate (Coordinate): The geographical coordinate of the zone.
8         severity (Severity): Priority based on severity of the situation.
9         population (int): The population size within the zone.
10        distanceToGoal (float): The distance to the goal zone.
11        heuristic (float): The heuristic value of the zone.
12        supplies (int): The amount of supplies available in the zone.
13    """
14
15    def __init__(self, name: string="", coordinate: Coordinate=Coordinate(), severity: Severity=3, population: int=0):
16        self.name = name
17        self.coordinate = coordinate
18        self.severity = severity
19        self.population = population
20        self.distanceToGoal = -1
21        self.heuristic = -1
22        self.supplies = 0
```

Figura 2: Definição da entidade Zona através de uma classe.

- **Estrada:**
  - **Custo:** O custo de percorrer a estrada.
  - **Condições:** As condições meteorológicas da estrada.

- **Geografia:** As características geográficas da estrada.
- **Infraestrutura:** A infraestrutura da estrada.
- **Disponibilidade:** A disponibilidade da estrada.

```

1 class Road:
2     """
3     Class to represent a road in the simulation.
4
5     Attributes:
6         cost (float): The cost of traveling on the road.
7         conditions (Conditions): The weather conditions on the road.
8         geography (Geography): The geographical features of the road.
9         infrastructure (Infrastructure): The infrastructure of the road.
10        availability (bool): The availability of the road.
11    """
12
13    def __init__(self, cost=0, conditions=None, geography=None, infrastructure=None, availability=True):
14        self.cost = cost
15        self.conditions = conditions
16        self.geography = geography
17        self.infrastructure = infrastructure
18        self.availability = availability

```

Figura 3: Definição da entidade Estrada através de uma classe.

- **Veículo:**
  - **Tipo:** O tipo de veículo, como drone, carro ou caminhão.
  - **Autonomia:** A autonomia do veículo em quilómetros.
  - **Capacidade:** A capacidade de carga do veículo em quilogramas.
  - **Carregamento:** A quantidade de carga que o veículo está a transportar, num dado momento.

```

1 class Vehicle:
2     """
3     Class representing a vehicle used in resource distribution during disaster relief.
4
5     Attributes:
6         type (VehicleType): The type of the vehicle, such as drone, car, or truck.
7         autonomy (float): The autonomy of the vehicle in kilometers.
8         capacity (float): The load capacity of the vehicle in kilograms.
9         load (float): The current load of the vehicle in kilograms
10    """
11
12    def __init__(self, vehicle_type: VehicleType, autonomy: float, capacity: float):
13        self.type = vehicle_type
14        self.autonomy = autonomy
15        self.capacity = capacity
16        self.load = capacity

```

Figura 4: Definição da entidade Veículo através de uma classe.

De seguida, o grupo tratou da concretização do seu recurso principal de trabalho. Tal como evidenciado na [Descrição do Problema](#), procedeu-se à implementação de um grafo para servir como base das operações e representação dos estados.

```
1 class Graph:
2     """
3     Class representing a basic undirected graph of zones with edge costs.
4     """
5
6     def __init__(self):
7         self.graph = {}
8
9     def add_connection(self, zone1: Zone, zone2: Zone, road: Road):
10        """
11        Create a bidirectional connection between two zones with a cost, weather conditions, geography, infrastructure, and availability.
12        """
13        Args:
14            zone1 (Zone): First zone.
15            zone2 (Zone): Second zone.
16            road (Road): Road object representing the connection between the two zones
17        """
18        self.add_zone(zone1)
19        self.add_zone(zone2)
20
21        self.graph[zone1].append((zone2, road))
22        self.graph[zone2].append((zone1, road))
```

Figura 5: Partição da definição do Grafo através de uma classe.

Após finalizar a consubstancialização dos detalhes fundamentais à transcrição do problema de linguagem natural para código, surgiu a necessidade de capacitar o projeto para lidar com a estrutura definida de forma a facilitar o desenvolvimento com versatilidade e legibilidade. Para tal, foram produzidas funções de apoio à criação e visualização de grafos, tais como: `generate_random_graph(num_nodes: int) -> Graph`, `print_graph(graph: Graph)` e `visualize_graph(graph: Graph)`, sendo a primeira responsável pela criação de diferentes grafos base a serem completados com as suas condições dinâmicas e as duas seguintes para imprimir o grafo de trabalho no terminal e visualizá-lo graficamente, respetivamente.

## 3.2. Algoritmia

### 3.2.1. Procura não informada

Os algoritmos de procura não informada têm a vantagem de não necessitarem de conhecimento prévio sobre o problema. Além disso, são relativamente simples de implementar e garantem encontrar uma solução, caso exista (no caso do BFS, a solução encontrada é o caminho mais curto). No entanto, apresentam desvantagens, como o consumo elevado de memória (BFS) ou a possibilidade de ficar preso em ciclos ou soluções não ótimas (DFS).

#### 3.2.1.1. BFS (Busca em Largura)

O algoritmo BFS explora o grafo camada por camada, garantindo encontrar o caminho mais curto em grafos não ponderados. É ideal para situações onde a prioridade é descobrir uma solução eficiente sem necessidade de heurísticas. Contudo, apresenta alta demanda de memória, especialmente em grafos densos ou com muitos nós.

#### 3.2.1.2. DFS (Busca em Profundidade)

O algoritmo DFS explora o grafo seguindo um caminho até ao fim antes de retroceder. É eficiente em termos de memória e é útil para explorar caminhos profundos rapidamente. Entretanto, não garante o menor caminho e pode ser ineficiente ao ficar preso em ciclos ou soluções subótimas.

### 3.2.2. Procura informada

No contexto do trabalho, os algoritmos de procura informada destacam-se pela utilização de heurísticas para orientar a pesquisa de forma mais eficiente, quando aplicados à tarefa de encontrar o melhor caminho entre duas cidades.

#### 3.2.2.1. Greedy

O algoritmo *Greedy* utiliza uma abordagem simples e rápida, baseando-se exclusivamente na heurística para decidir o próximo nó a ser explorado. No contexto do problema, isso significa priorizar a zona que aparenta estar mais próxima da zona objetivo com base na distância em linha reta até à mesma. Embora seja eficiente em termos de tempo, o *Greedy* não leva em conta o custo acumulado ( $g$ ), o que pode resultar em caminhos subótimos.

#### 3.2.2.2. A\*

O algoritmo A\* vai além da abordagem *Greedy* ao combinar o custo acumulado ( $g$ ) com a heurística ( $h$ ), formando a função de avaliação  $f(n) = g(n) + h(n)$ . Essa integração permite ao A\* **garantir sempre** o caminho mais curto, desde que a heurística seja admissível, ou seja, não sobrestime o custo real. No problema em questão, essa abordagem é particularmente vantajosa, pois considera tanto o progresso feito quanto a estimativa do custo restante, tornando-o mais robusto e confiável para encontrar soluções ótimas.

#### 3.2.2.3. A\* Dinâmico

O A\* Dinâmico, **concebido especificamente para este trabalho**, é uma extensão do A\* que opera em mapas dinâmicos, onde as condições mudam aleatoriamente a cada certo número de iterações. Este algoritmo utiliza uma heurística mais sofisticada, considerando fatores como condições meteorológicas, geografia e infraestrutura das estradas, ajustando o cálculo conforme a dinâmica do ambiente.

O A\* Dinâmico não só encontra o caminho mais curto, mas também adapta a rota em tempo real a partir de alterações nas ligações. Além disso, incorpora restrições como autonomia do veículo e a gestão de suprimentos, tornando-o uma solução robusta para problemas onde mudanças constantes no ambiente impactam diretamente a viabilidade do caminho. Esta abordagem foi fundamental para lidar com a natureza imprevisível dos mapas no problema, garantindo maior flexibilidade e eficácia no contexto de cenários reais.

## 3.3. Métricas de desempenho

Nesta secção é abordado um dos tópicos mais importantes para o entendimento da solução apresentada, a realização de métricas que permitem a visualização do desempenho e eficiência da proposta final de algoritmo.

### 3.3.1. Estruturação

As métricas foram subdivididas em dois tipos:

- **Benchmark Testing:** Nesta primeira versão é viável para o utilizador controlar manualmente o grafo sobre o qual quer executar os vários algoritmos em simultâneo e, desta forma, analisar os parâmetros segundo a sua intenção de uso.
- **Bulk Benchmark Testing:** Numa segunda iteração foi desenvolvida uma funcionalidade que permite medir métricas em massa através da aplicação de diferentes grafos aos algoritmos disponíveis na aplicação. Esta abordagem induz facilidade na produção de múltiplos dados credíveis e minimização das divergências na execução dos algoritmos.

Após implementação e aprimoramento das funções de medição de métricas, surgiu a necessidade de conceção de um sistema de visualização e comparação dos dados retirados com priorização à sua representação gráfica.

### 3.3.2. Parametrização

A parametrização é um elemento essencial para a coleta e análise eficaz das métricas de desempenho descritas no tópico anterior. Ela permite que o sistema seja configurado de forma flexível e adaptada às necessidades específicas dos utilizadores, garantindo a reprodutibilidade dos testes e a comparabilidade dos resultados obtidos.

#### 3.3.2.1. Estratégia de Parametrização

A estrutura de parametrização foi desenvolvida com base nos seguintes pilares:

**Configurações de Algoritmos:** Os algoritmos disponíveis na solução permitem a definição de certos parâmetros específicos como a zona de início, utilização de heurística ou escolha de veículo, entre outros. Isso assegura que diferentes cenários possam ser simulados de forma granular e controlada.

**Características do Grafo:** Na criação de grafos é possível indicar a quantidade de zonas a aplicar. Automaticamente são criadas todas as outras características pelo sistema, como a sua população e severidade, as ligações a outras zonas através de estradas e até a aleatoriedade imposta de disponibilidade das estradas.

**Condições de Execução:** Para execução em massa (Bulk Benchmark Testing), é possível configurar lotes de testes com variações automáticas dos parâmetros acima. Esta funcionalidade é útil para identificar padrões de desempenho em diferentes cenários.

#### 3.3.2.2. Parâmetros Monitorizados

Durante a execução das simulações, as seguintes métricas são coletadas:

- **execution\_time:** Tempo total necessário para a execução do algoritmo, medido em milissegundos.
- **peak\_memory\_usage:** Quantidade máxima de memória utilizada durante a execução.
- **zones\_visited:** Número total de zonas exploradas pelo algoritmo.
- **solution\_depth:** Profundidade da solução encontrada, indicando o número de passos necessários.
- **solution\_cost:** Custo total da solução, considerando os pesos das arestas atravessadas.
- **avg\_cost\_per\_edge:** Custo médio por aresta na solução final.
- **zones\_visited\_percentage:** Percentagem de zonas exploradas em relação ao total disponível no grafo.

### 3.3.3. Resultados obtidos

A equipa de trabalho executou dez *bulk operations* e estabeleceu a média destes resultados para os mesmos parâmetros de teste no tamanho da criação dos gráficos a submeter nos exercícios de medição, sendo eles: 5, 10, 20, 50, 75, 100, 200, 350, 500. Após ponderação dos dez apuramentos, foi produzido um gráfico associado à sua relação com o número de zonas impostas pelo instrumento de teste, para cada uma, procedeu-se à especulação dos resultados finais obtidos.

O primeiro gráfico produzido é relativo ao tempo de execução, no qual é possível dividir os resultados em três níveis:

- O DFS, BFS, Greedy e A\* destacam a sua prestação devida à sua essência de atuação num grafo estático.
- O A\* dinâmico demonstra uma alteração significativa relativamente aos algoritmos anteriores, dada a necessidade de tomadas de decisão mais complexas e avaliação da heurística, principalmente acentuado num ambiente com muitas zonas e, conseqüentemente, muitas variáveis a cada momento.



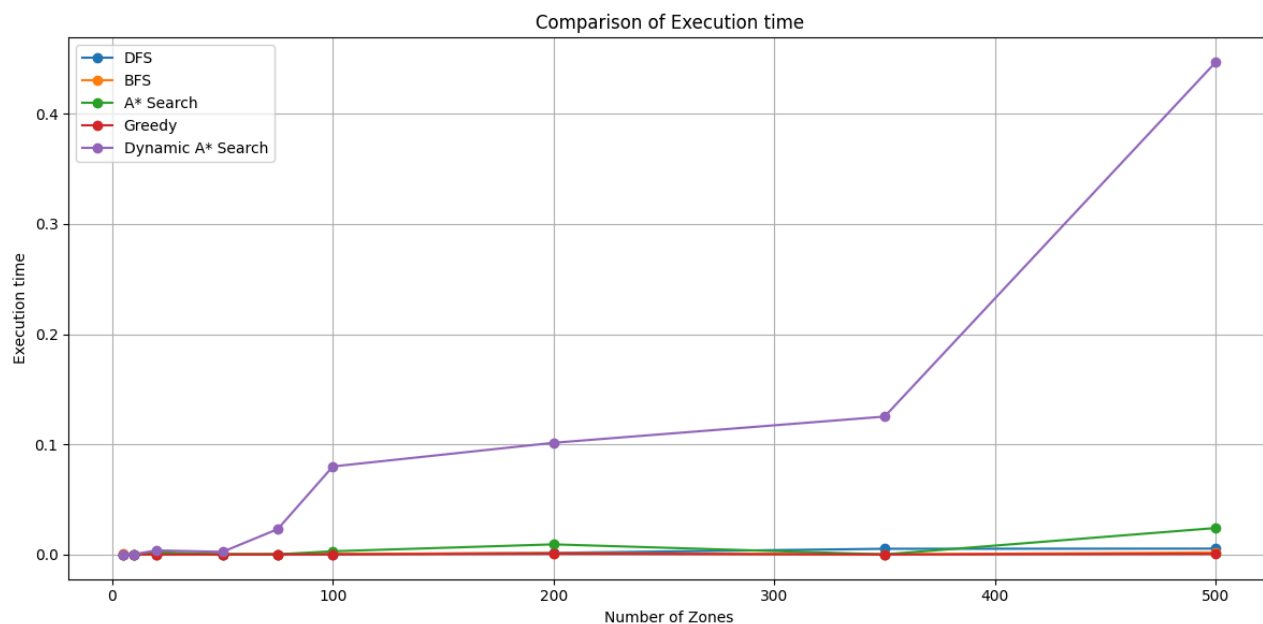


Figura 6: Ponderação da métrica de tempo de execução.

Seguiu-se a conceção do gráfico que descreve a variação de valores para a utilização memória, no qual, é possível observar, um resultado totalmente previsto teoricamente para os algoritmos modelo (DFS, BFS, Greedy e A\*), destacando-se a boa prestação do A\* dinâmico que, ao usar um modelo de decisão idêntico ao A\* revela bons indicadores de uso de memória.

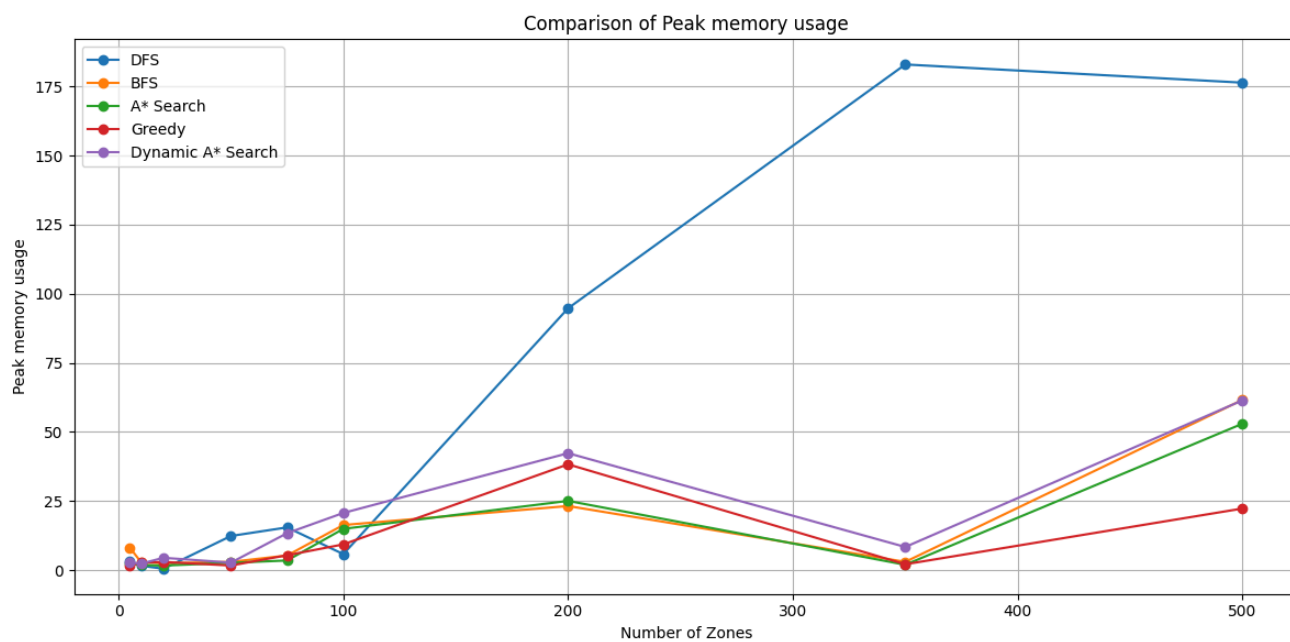


Figura 7: Ponderação da métrica de utilização de memória.

Avançou-se para a avaliação do número de zonas visitadas, no qual se denota uma diferença acentuada entre algoritmos de procura não informada e informada, a ponderação de uma boa heurística para utilização nos algoritmos de procura informada permite tomadas de decisão mais assertivas e, por conseguinte, mais eficazes.

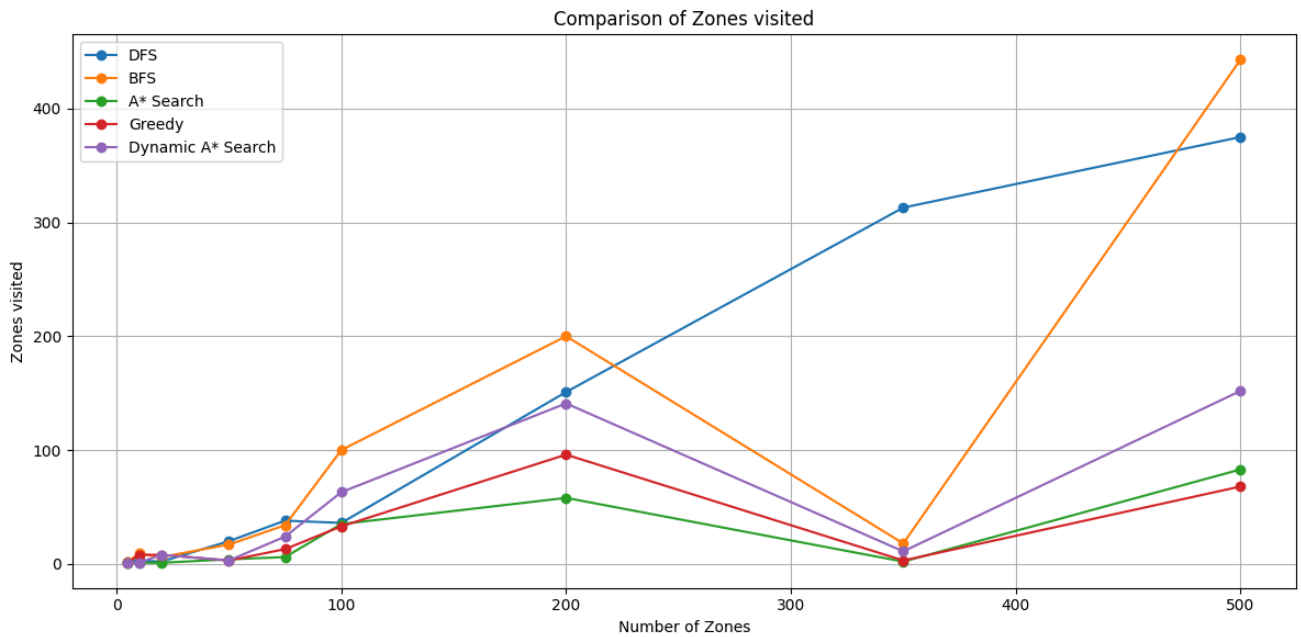


Figura 8: Ponderação da métrica de visita a zonas.

No gráfico seguinte é imediatamente focada a atenção nos resultados apresentados pelo DFS, que mesmo sendo totalmente expectáveis pelo conhecimento teórico prévio, comprovam mais uma vez a sua ineficiência em grafos grandes e dispersos. No entanto, é do interesse da equipa de trabalho a especial atenção no algoritmo principal, o A\* dinâmico, que apesar das condições variáveis e possibilidade de ligações cortadas (com obrigação de alteração de plano), consegue manter a sua prestação muito próxima do seu semelhante, mesmo este atuando em condições estáticas.

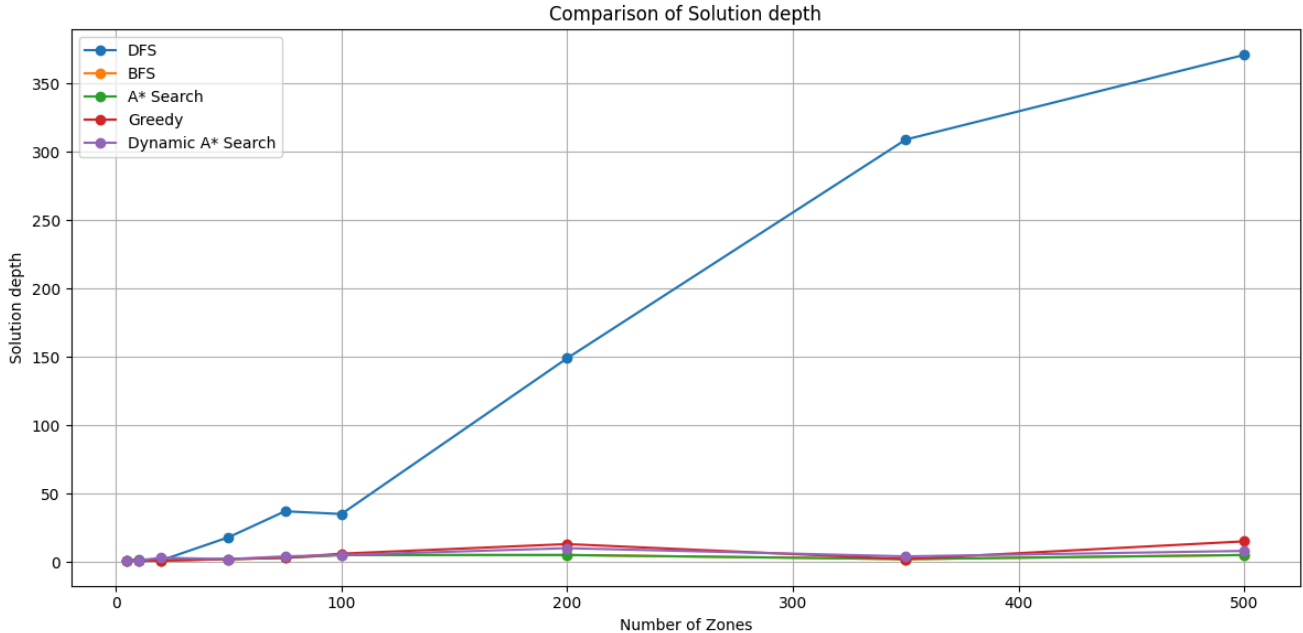


Figura 9: Ponderação da métrica de profundidade da solução.

No seguimento dos resultados exibidos, apresenta-se o gráfico relativo ao custo de solução que garantem suporte à previsão teórica da atuação dos diferentes algoritmos e salientam a incorporação do A\* no bloco de operações com melhores minimização de custo.

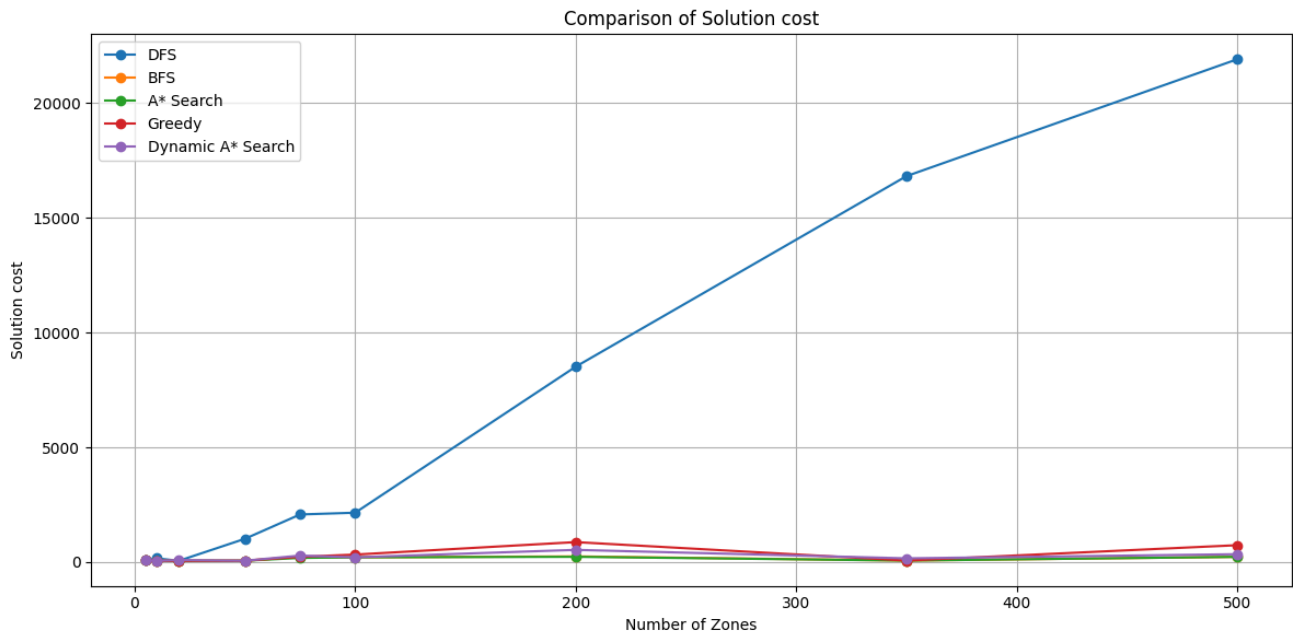


Figura 10: Ponderação da métrica de custo da solução.

Pode observar-se em baixo o gráfico que representa a métrica do custo médio de cada conexão do instrumento de estudo. Para a equipa de trabalho, esta métrica apoia o entendimento das restantes e complementa a compreensão dos resultados, uma vez que possibilita o ganho de noção sobre a dispersão do grafo em utilização.

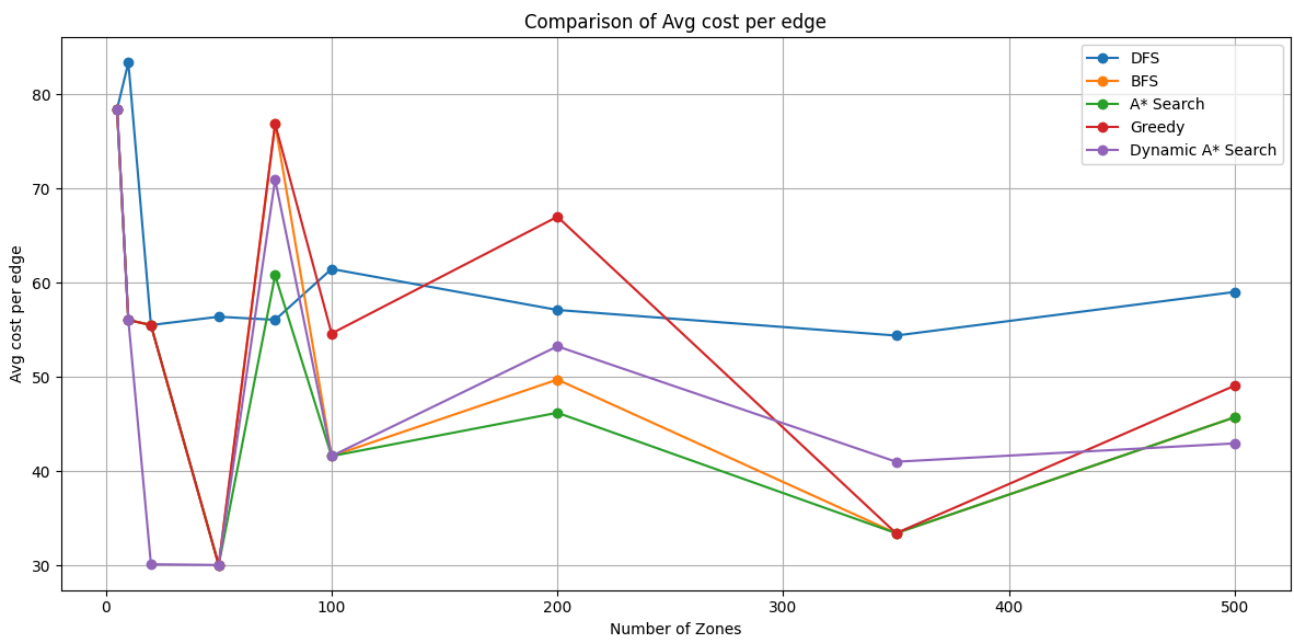


Figura 11: Ponderação da métrica de média de custo por conexão.

Finalmente, não suficientemente satisfeitos com o resultado obtido no gráfico da métrica de visita de zonas, devido à escala imposta pela configuração dos tamanhos dos grafos. Como tal, reproduziu percentualmente os valores obtidos. Concluindo-se uma grande variação das soluções implementação de acordo com o grafo proposto e a sua definição.

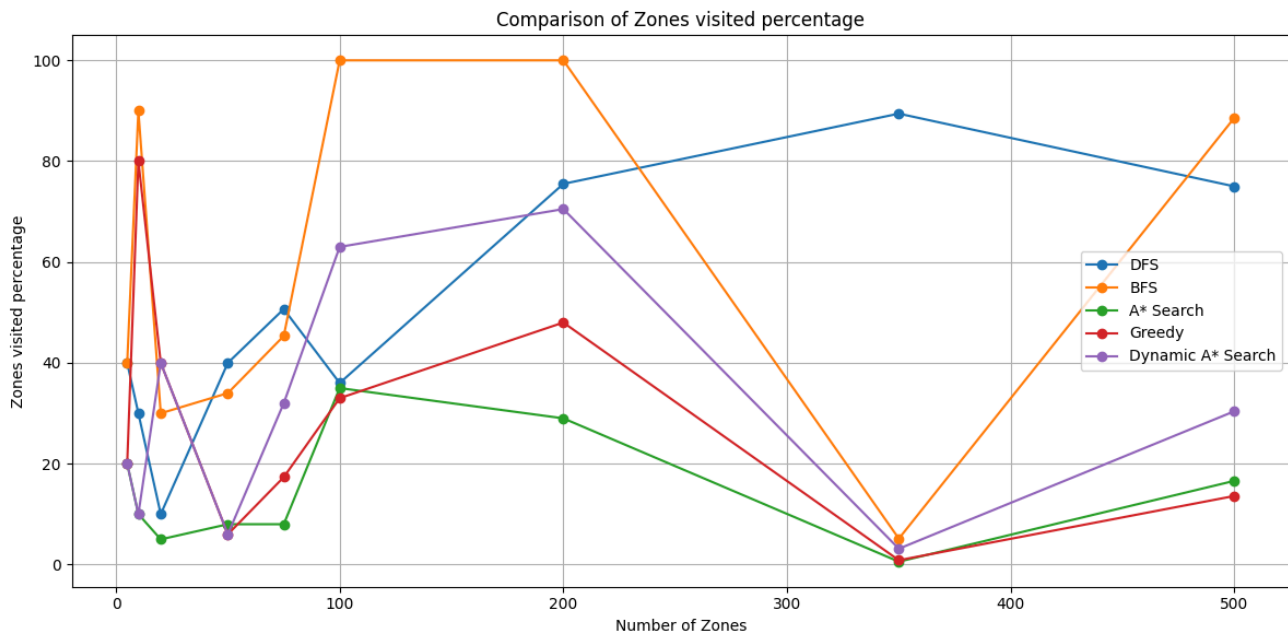


Figura 12: Ponderação da métrica de percentagem de zonas visitadas.

## 3.4. Extras

### 3.4.1. Mapa de Portugal

O projeto conta com a integração de um mapa que permite a visualização dos municípios de Portugal utilizando uma estrutura de grafo com ajuda das bibliotecas [NetworkX](#) e [Matplotlib](#). Os dados foram baseados na [Carta Administrativa Oficial de Portugal, versão de 2023 - CAOP2023](#) e na [Lista de municípios de Portugal](#).

Cada nó representa um concelho, e as arestas representam conexões baseadas em distâncias geográficas, podendo as conexões ser alvo de eventualidades que condicionem o seu acesso. Adicionalmente, os tamanhos dos nós são proporcionais à população de cada município.

Diferentes modos de visualização do grafo são permitidas na execução do programa, permitindo ao utilizador ver dados extra relativos aos nodos, como o nome dos municípios, e conexões, como as condições, geografia, infraestrutura e custo das estradas.

Segue-se uma imagem representativa da estrutura do grafo num dado estado do programa, onde as arestas verdes representam as estradas acessíveis, as arestas vermelhas (a tracejado) representam as estradas inacessíveis (estradas cortadas) e os círculos azuis representam as cidades.

Portugal Municipalities Graph (Continental Only)

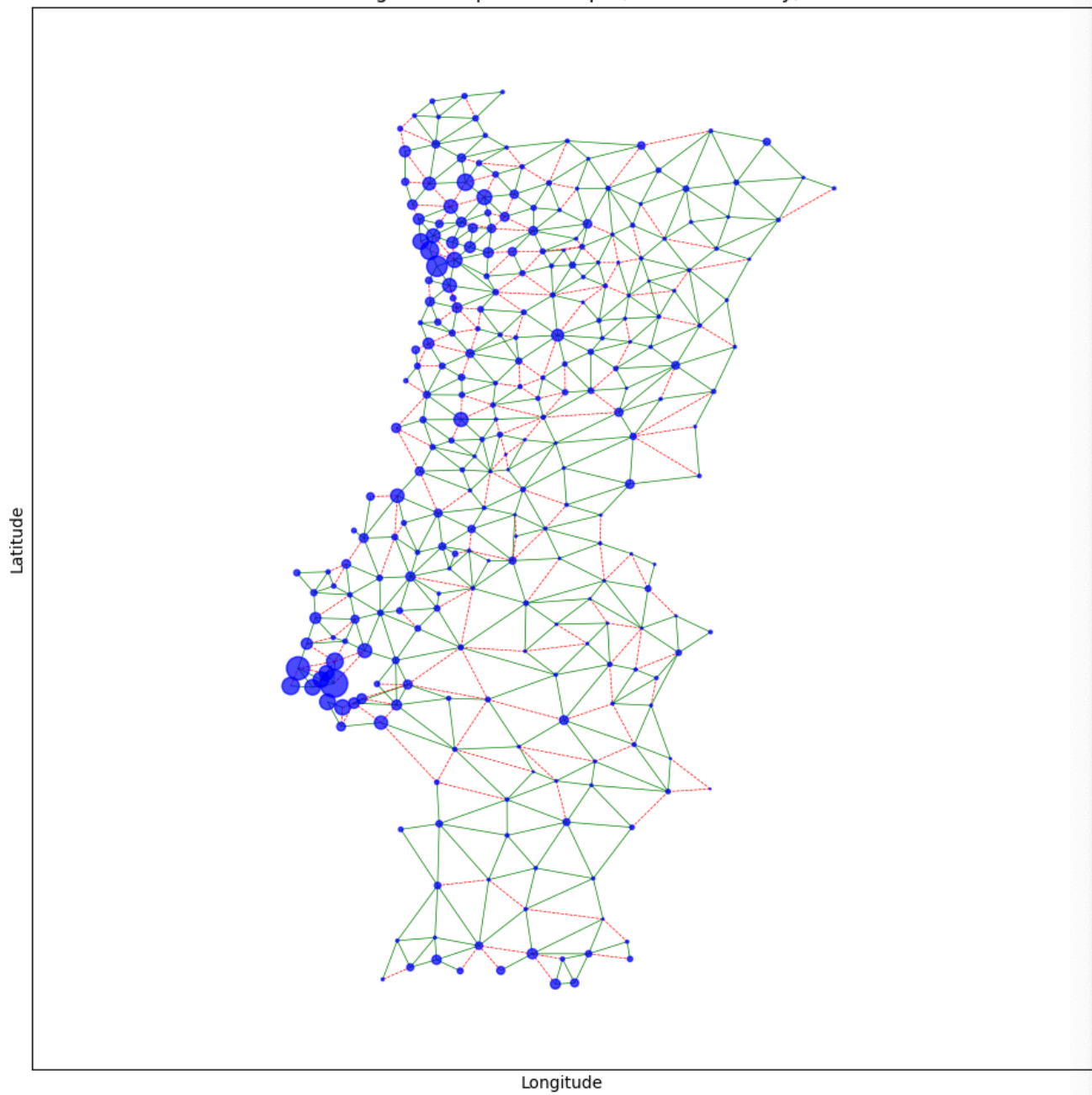


Figura 13: Grafo dos municípios de Portugal.

## 4. Sumário e discussão dos resultados obtidos

### 4.1. Resumo dos resultados

Os resultados obtidos através da implementação dos diferentes algoritmos de procura e das simulações realizadas evidenciam variações significativas no desempenho, tanto em cenários estáticos como dinâmicos. Os algoritmos de procura informada, nomeadamente o A\*, apresentaram resultados consistentes em termos de eficiência e custo total da solução. Por outro lado, o A\* dinâmico destacou-se por sua capacidade de adaptação às mudanças no ambiente, um requisito crítico no contexto de distribuições em zonas afetadas por catástrofes naturais.

As métricas de desempenho analisadas confirmaram as expectativas teóricas em relação ao comportamento dos algoritmos implementados. As principais conclusões derivadas das métricas incluem:

- **Tempo de Execução:** O A\* dinâmico apresenta um tempo de execução superior em comparação aos restantes algoritmos, devido à necessidade de ajustes constantes em mapas dinâmicos. No entanto, esta diferença é compensada pela sua eficácia em encontrar soluções robustas em ambientes incertos.
- **Utilização de Memória:** Os algoritmos de procura não informada, como o BFS e o DFS, exibiram maior consumo de memória em grafos extensos, enquanto o A\* e o A\* dinâmico demonstraram um uso de memória mais equilibrado, mesmo em cenários de elevada complexidade.
- **Zonas Visitadas:** Os algoritmos de procura informada exploraram significativamente menos zonas em comparação com os não informados, destacando a importância de heurísticas eficazes na redução de esforço computacional.
- **Profundidade da Solução e Custo:** O A\* dinâmico manteve-se próximo do A\* tradicional em termos de profundidade da solução e custo total, mesmo sob condições variáveis, reforçando a sua adequação para cenários dinâmicos.

### 4.2. Análise Crítica

O A\* dinâmico destacou-se pela sua capacidade de lidar com alterações no ambiente de forma eficiente e adaptativa. A utilização de heurísticas sofisticadas, que consideram fatores como condições meteorológicas e restrições de veículos, permitiu alcançar soluções que minimizam custos e maximizam a cobertura das zonas críticas. Esta abordagem é particularmente útil em contextos de emergência, onde a imprevisibilidade é uma constante.

Apesar do sucesso geral, foram identificadas algumas limitações:

- **Complexidade Computacional:** O A\* dinâmico requer maior capacidade computacional, o que pode limitar a sua aplicação em dispositivos com recursos restritos.
- **Dependência de Heurísticas:** O desempenho do A\* dinâmico é fortemente influenciado pela qualidade das heurísticas utilizadas. Em cenários onde estas não são suficientemente precisas, o algoritmo pode apresentar resultados subótimos.
- **Escalabilidade:** À medida que o número de zonas e variáveis aumenta, o tempo de execução pode crescer exponencialmente, mesmo com otimizações implementadas.

### 4.3. Conclusão

A análise dos resultados reafirma a relevância dos algoritmos de procura informada, em especial o A\* dinâmico, para aplicações em contextos reais de distribuição de recursos em cenários de catástrofes naturais. A capacidade de adaptação e a robustez demonstradas tornam-no uma ferramenta promissora, apesar das suas limitações.

No futuro, melhorias podem ser direccionadas para a otimização de heurísticas e para a redução da complexidade computacional, visando maior escalabilidade e aplicabilidade em sistemas de resposta em tempo real. Além disso, a integração de dados em tempo real, como informações meteorológicas ou tráfego, pode potencializar ainda mais a eficácia das soluções propostas.

## Referências

Russell and Norvig (2009). Artificial Intelligence - A Modern Approach, 3rd edition, ISBN-13: 9780136042594



## **Lista de Siglas e Acrónimos**

**A\*** Algorithm A Star

**BFS** Breadth-First Search

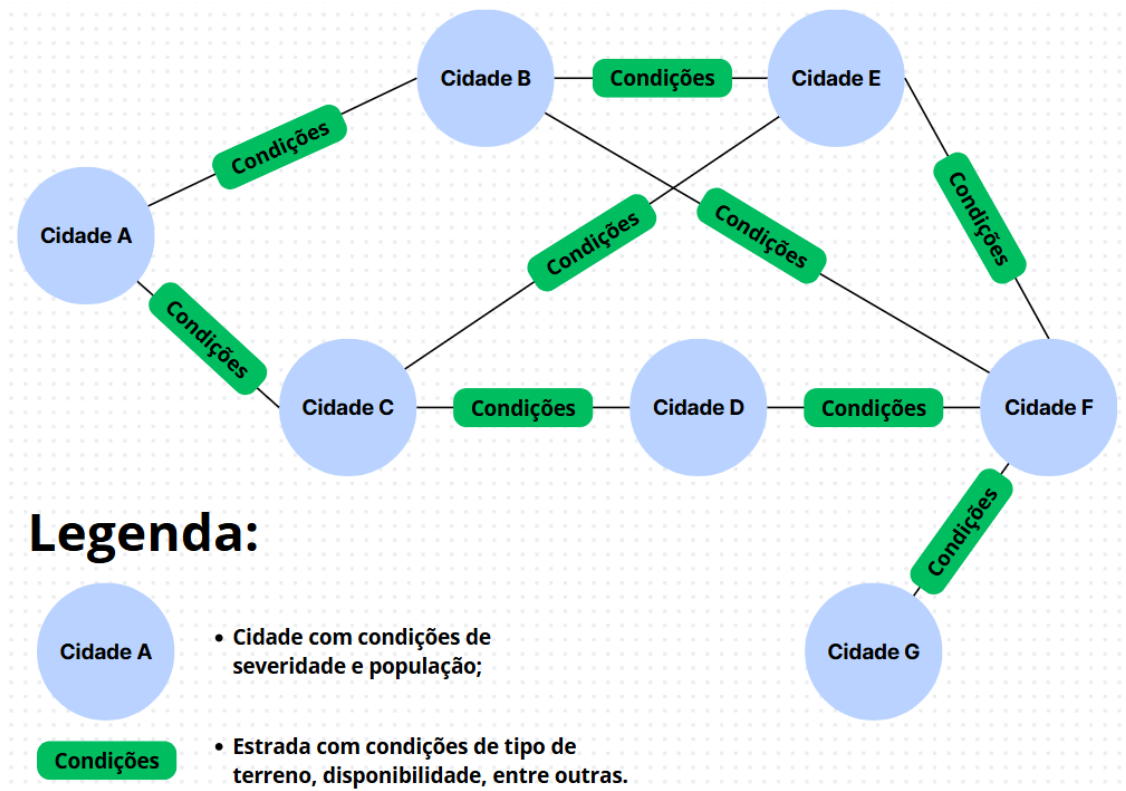
**DFS** Depth-First Search

## Anexos

### Anexo 1: Logo da Universidade do Minho



## Anexo 2: Ilustração representativa de um exemplo de visualização do problema.



## Anexo 3: Definição da entidade Zona através de uma classe.

```
1 class Zone:
2     """
3     Class representing a disaster-affected zone.
4
5     Attributes:
6         name (string): The name of the zone.
7         coordinate (Coordinate): The geographical coordinate of the zone.
8         severity (Severity): Priority based on severity of the situation.
9         population (int): The population size within the zone.
10        distanceToGoal (float): The distance to the goal zone.
11        heuristic (float): The heuristic value of the zone.
12        supplies (int): The amount of supplies available in the zone.
13    """
14
15    def __init__(self, name: string="", coordinate: Coordinate=Coordinate(), severity: Severity=3, population: int=0):
16        self.name = name
17        self.coordinate = coordinate
18        self.severity = severity
19        self.population = population
20        self.distanceToGoal = -1
21        self.heuristic = -1
22        self.supplies = 0
```

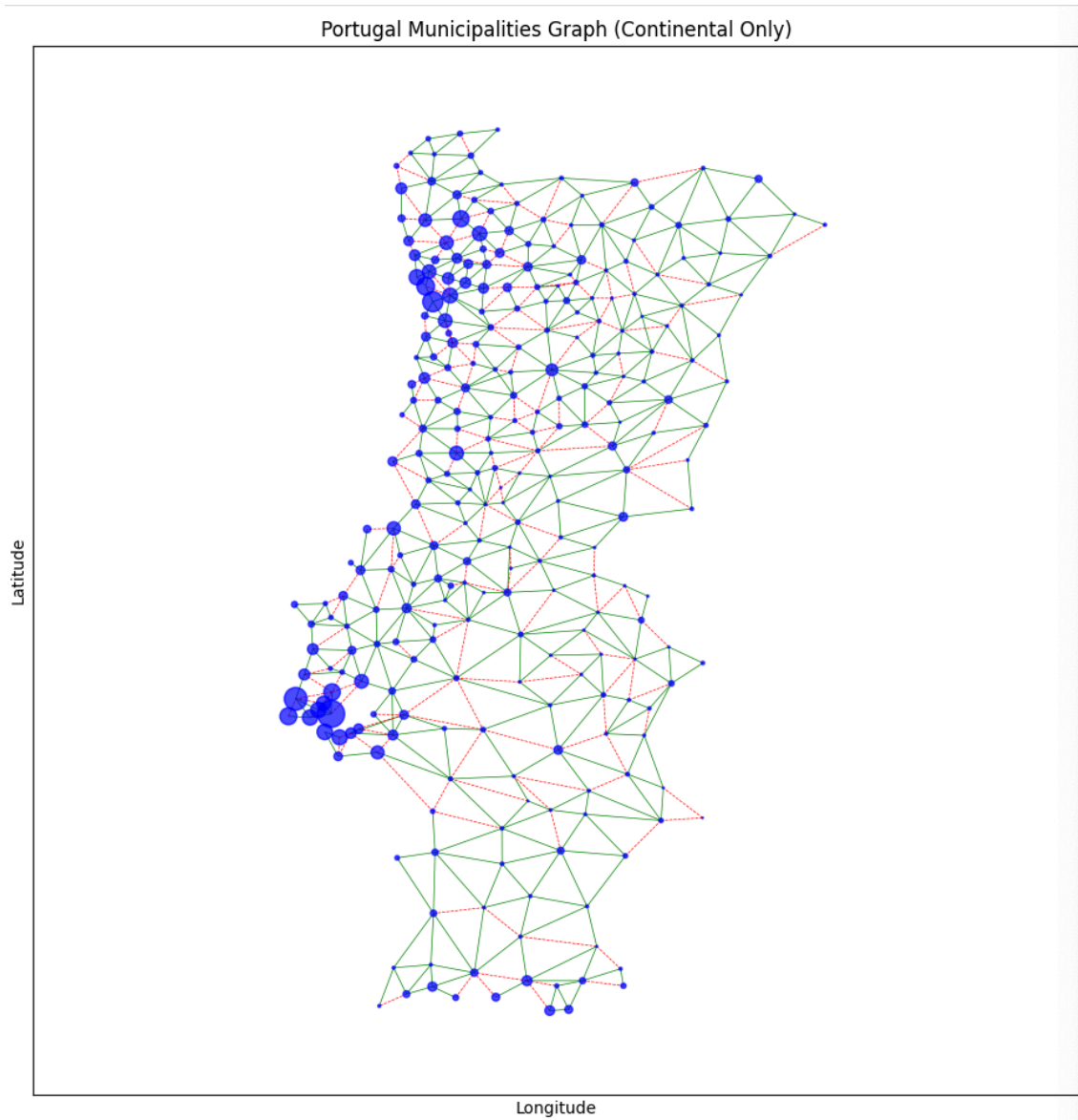
## Anexo 4: Definição da entidade Estrada através de uma classe.

```
1 class Road:
2     """
3     Class to represent a road in the simulation.
4
5     Attributes:
6         cost (float): The cost of traveling on the road.
7         conditions (Conditions): The weather conditions on the road.
8         geography (Geography): The geographical features of the road.
9         infrastructure (Infrastructure): The infrastructure of the road.
10        availability (bool): The availability of the road.
11    """
12
13    def __init__(self, cost=0, conditions=None, geography=None, infrastructure=None, availability=True):
14        self.cost = cost
15        self.conditions = conditions
16        self.geography = geography
17        self.infrastructure = infrastructure
18        self.availability = availability
```

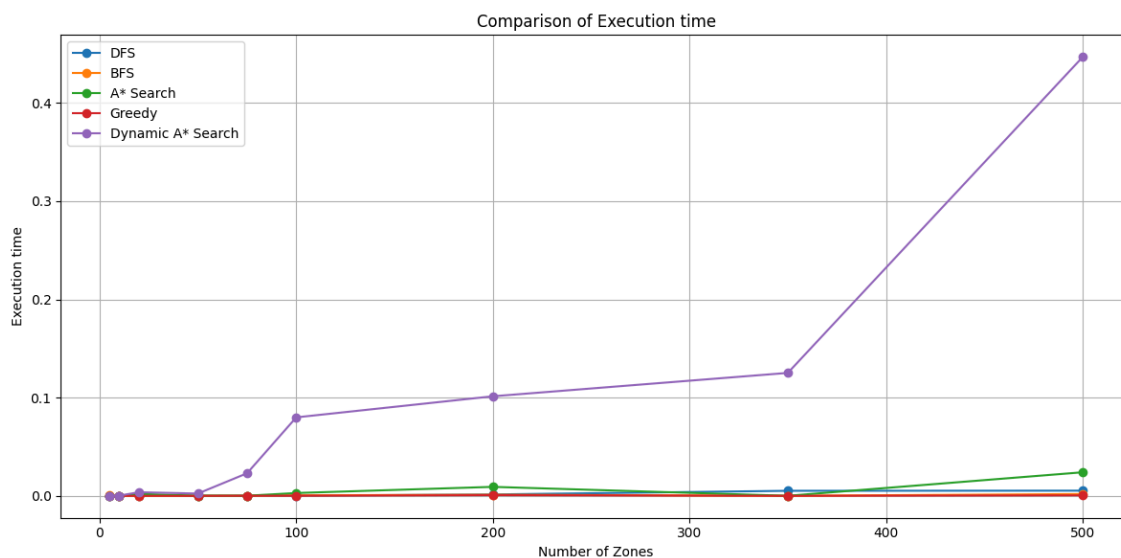
## Anexo 5: Definição da entidade Veículo através de uma classe.

```
1 class Vehicle:
2     """
3     Class representing a vehicle used in resource distribution during disaster relief.
4
5     Attributes:
6         type (VehicleType): The type of the vehicle, such as drone, car, or truck.
7         autonomy (float): The autonomy of the vehicle in kilometers.
8         capacity (float): The load capacity of the vehicle in kilograms.
9         load (float): The current load of the vehicle in kilograms
10    """
11
12    def __init__(self, vehicle_type: VehicleType, autonomy: float, capacity: float):
13        self.type = vehicle_type
14        self.autonomy = autonomy
15        self.capacity = capacity
16        self.load = capacity
```

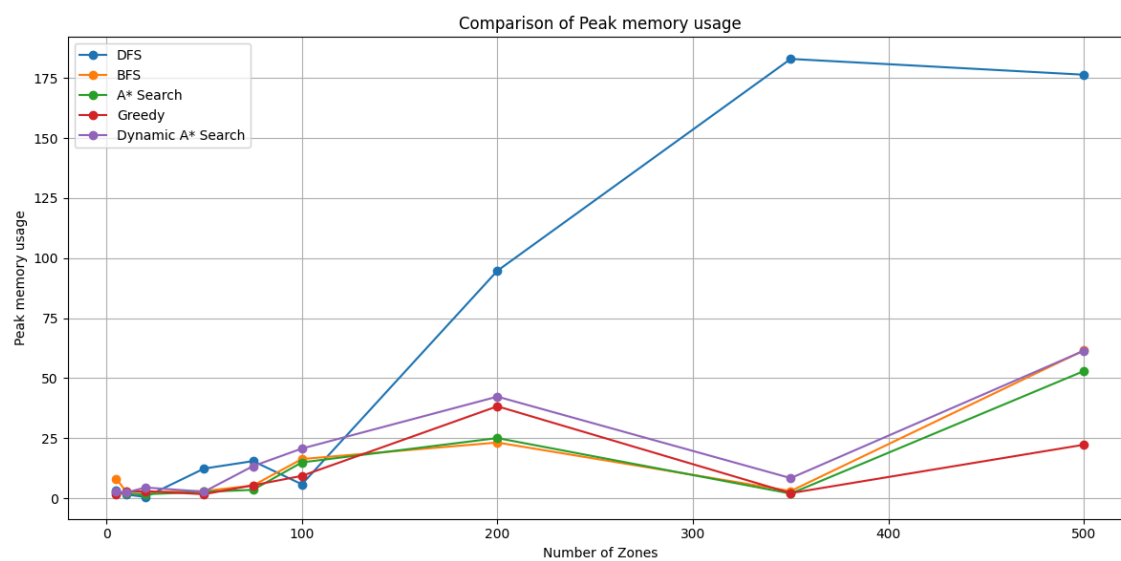
## Anexo 6: Grafo dos municípios de Portugal.



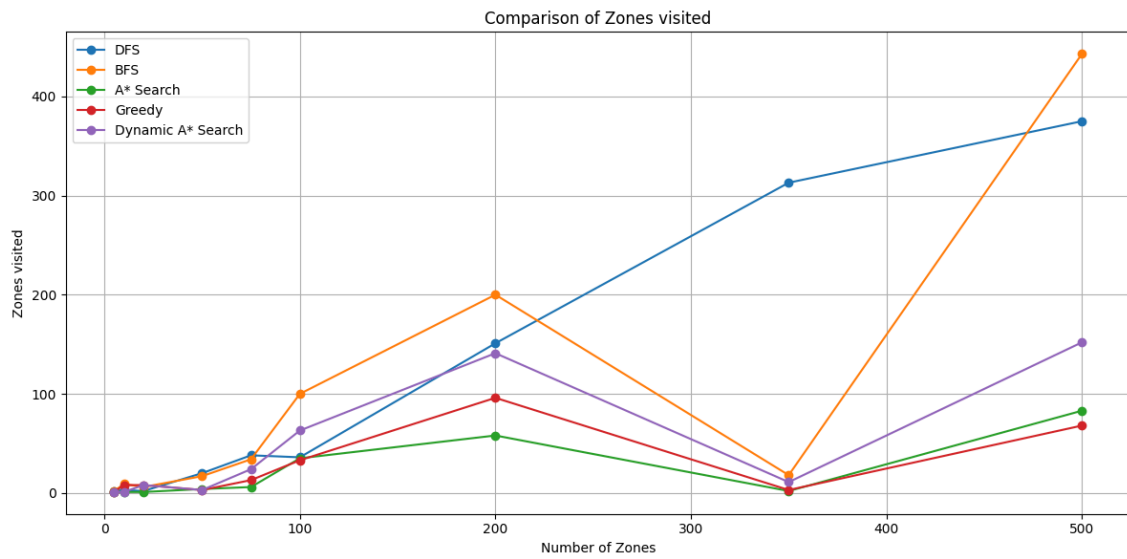
## Anexo 7: Ponderação da métrica de tempo de execução.



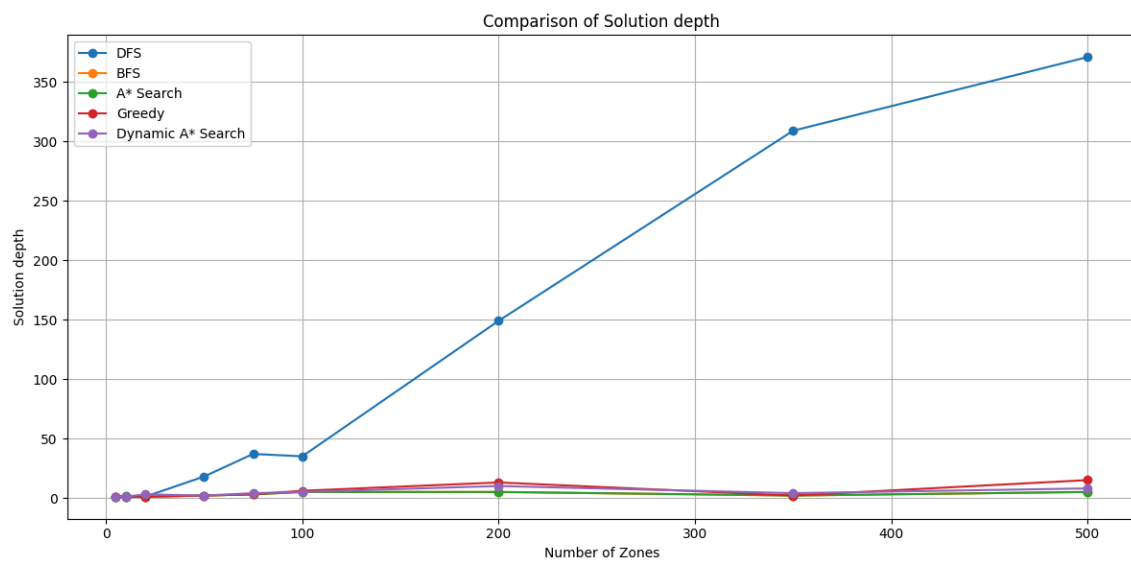
## Anexo 8: Ponderação da métrica de utilização de memória.



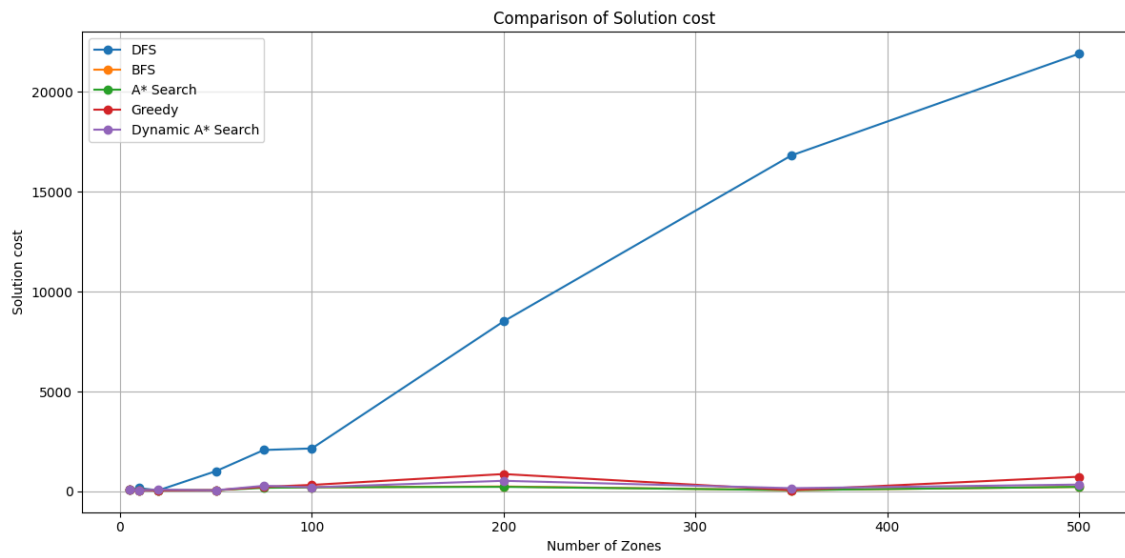
## Anexo 9: Ponderação da métrica de visita a zonas.



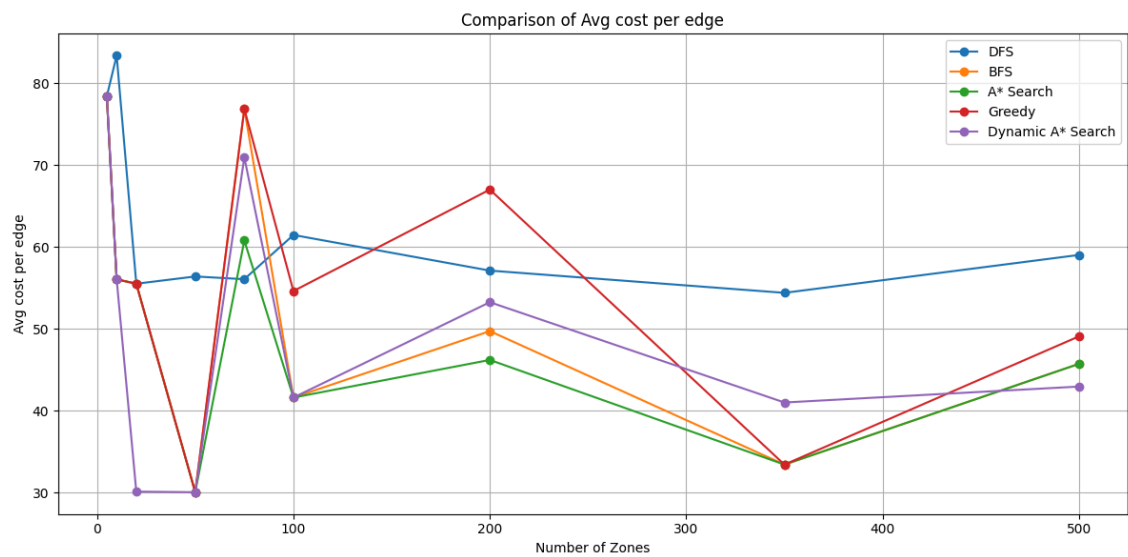
## Anexo 10: Ponderação da métrica de profundidade da solução.



## Anexo 11: Ponderação da métrica de custo da solução.



## Anexo 12: Ponderação da métrica de média de custo por conexão.





Anexo 13: Ponderação da métrica de percentagem de zonas visitadas.

