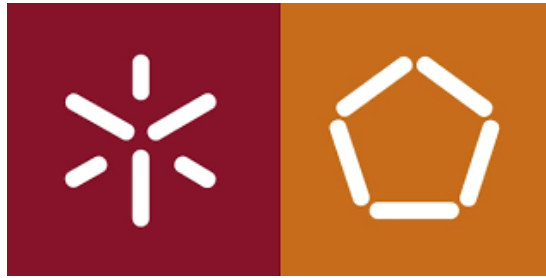


Universidade do Minho



Inteligência Artificial

Trabalho Prático

1ª Fase

Ano letivo 2022/2023

a96106, Miguel Silva Pinto

a97755, Orlando José da Cunha Palmeira

a97613, Pedro Miguel Castilho Martins

1. Introdução	3
2. Formulação do problema	4
3. Criação do circuito	4
4. Estratégias de procura	5
4.1. Procura em profundidade (DFS)	5
4.1.1. Exemplo de uma procura em profundidade (DFS)	6
4.2. Procura em largura (BFS)	6
4.2.1. Exemplo de uma procura em largura (BFS)	7
5. Conclusão	7

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Inteligência Artificial da Licenciatura em Engenharia Informática da Universidade do Minho, e teve como objetivo a resolução de problemas através da conceção e implementação de algoritmos de procura. Neste trabalho foram implementadas estratégias para a resolução de problemas com o uso de algoritmos de procura, partindo da formulação do problema em questão. Ao longo deste documento vai ser explicado todo o processo de desenvolvimento dos mecanismos de procura implementadas nesta primeira fase.

2. Formulação do problema

Uma vez que o problema se encontra num ambiente determinístico e completamente observável, em que o agente “sabe” exatamente o estado em que estará e as soluções pretendidas são sequências, podemos afirmar que este é um problema de estado único. Além disso, a sua formulação pode ser efetuada da seguinte forma:

- **Representação do estado:** Grafo não orientado, em que cada nodo representa uma posição possível no circuito e cada aresta representa o trajeto entre posições.
- **Estado inicial:** A posição inicial representa o local de início da corrida de onde o carro parte.
- **Estado/teste objetivo:** Alcançar uma das posições finais da meta.
- **Operadores:** Deslocação entre posições adjacentes incluindo as diagonais.
- **Solução:** Um caminho válido (sequência de posições percorridas) que comece na posição inicial e termine numa das posições finais.
- **Custo da solução:** Distância total do percurso feito pelo agente.

3. Criação do circuito

O circuito é criado de acordo com um determinado ficheiro que representa esse circuito, com um formato semelhante ao seguinte:

```
XXXXXXXXXX
XXX--X--XX
XP---X---F
X---XX---F
X-----X
XXX--X-XXX
XXXXXXXXXX
```

Figura 1: Exemplo de ficheiro de circuito.

O circuito mostrado na Figura 1 é um circuito com uma posição inicial para o carro (representada pela letra “P”), com duas posições finais possíveis (representadas pela letra “F”) e com as delimitações do circuito representadas pela letra “X”, representando obstáculos que não podem ser transpostos. Existem diversos circuitos possíveis que poderiam ser analisados através do nosso programa, mas iremos utilizar este circuito para demonstrar as estratégias de procura que implementamos.

4. Estratégias de procura

Nesta primeira fase implementamos as seguintes estratégias de procura não informada:

4.1. Procura em profundidade (DFS)

Este tipo de procura usa como estratégia, expandir sempre um dos nodos mais profundos do grafo.

As **vantagens** deste tipo de procura são:

- Pouco uso de memória.
- Bom para problemas com múltiplas soluções, pois a probabilidade de estar a procurar por um caminho possível aumenta.

A **desvantagem** deste tipo de procura são:

- Pouca eficiência em grafos com uma profundidade elevada e poucas soluções.
- A solução obtida pode não ser a solução ótima.

No entanto, no nosso problema o circuito terá vários caminhos possíveis para chegar ao destino final, logo esta estratégia irá conseguir obter um percurso válido e de forma relativamente eficiente.

A nossa implementação do algoritmo de procura em profundidade impede que os nodos que já tenham sido visitados sejam explorados pelo algoritmo de maneira a evitar a ocorrência de loops no processo de pesquisa.

Esta estratégia de procura apresenta as seguintes propriedades:

- Tempo: $O(b^m)$
- Espaço: $O(bm)$

Onde **b** é o número máximo de sucessores de um nó da árvore de procura e **m** a máxima profundidade do espaço de estados.

4.1.1. Exemplo de uma procura em profundidade (DFS)

```
Fez o seguinte caminho:  
XXXXXXXXXX  
XXXcdXprXX  
Xabe-Xoqsx  
XgfjXXntwF  
Xhiklmuv-X  
XXX--X-XXX  
XXXXXXXXXX  
Prima enter para continuar.
```

Com o custo: 23

Figura 2: Exemplo de uma procura em profundidade.

O programa imprime o circuito com o percurso encontrado pelo algoritmo de procura. O caminho está representado por letras ordenadas por ordem alfabética e pelas seguintes cores: Vermelho(Nodo inicial), Verde(Caminho percorrido) e Azul(Nodo final/destino).

Como podemos ver na figura acima, o algoritmo de procura em profundidade obteve um caminho não ideal devido à sua estratégia de expandir sempre para nodos mais profundos do grafo, não sendo a melhor opção de escolha para este tipo de problemas uma vez que existem ramificações bastante longas que não vão ao encontro de uma solução ótima.

4.2. Procura em largura (BFS)

Este tipo de procura expande primeiro os nós de menor profundidade do grafo.

As **vantagens** deste tipo de procura são:

- Solução ótima quando todas as arestas têm custo 1.
- Procura muito sistemática.

As **desvantagens** deste tipo de procura são:

- Tempo de pesquisa elevado visto que tende a percorrer muitos mais nodos do que os necessários para criar o caminho válido.
- Ocupa muito espaço em memória.

No nosso problema o tamanho do grafo irá depender do tamanho do circuito logo este tipo de procura irá tornar-se bastante lento para circuitos de grandes dimensões.

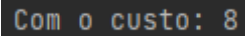
Esta estratégia de procura apresenta as seguintes propriedades:

- Tempo: $O(b^d)$
- Espaço: $O(b^d)$

Onde **b** é o número máximo de sucessores de um nó da árvore de procura e **d** a profundidade da melhor solução.

4.2.1. Exemplo de uma procura em largura (BFS)

```
Fez o seguinte caminho:
XXXXXXXXXX
XXX--X--XX
Xab--X-ghi
X--cXXf--F
X---de---X
XXX--X-XXX
XXXXXXXXXX
Prima enter para continuar. |
```



Com o custo: 8

Figura 3: Exemplo de uma procura em largura.

Como podemos ver na figura acima, o algoritmo de procura em largura obteve um caminho ótimo uma vez que no nosso problema todas as arestas têm um custo igual a 1, o que é a melhor circunstância possível para a utilização deste algoritmo, visto que um grafo com esta característica permite ao algoritmo BFS encontrar sempre o caminho mais curto devido à estratégia de expansão dos nodos de menor profundidade em primeiro.

5. Conclusão

Em suma, nesta primeira fase o nosso grupo desenvolveu um programa que dado um circuito fornecido em formato de ficheiro, é capaz de criar o grafo de posições possíveis, possibilitando a visualização desse grafo de forma gráfica, mas também mostrar os seus nodos e as respetivas arestas. Mais importante, permite utilizar algoritmos de procura para explorar caminhos possíveis e atingir uma solução, tendo sido descritas todas as vantagens e desvantagens de cada estratégia utilizada.