



UNIVERSIDADE DE ÉVORA

Relatório Inteligência Artificial

Rúben Peixoto & Vanessa Santos
37514 & 34191

23 de Março 2020

1 Exercício 1

```
Solução:
node([4,4],node([4,3],node([3,3],node([2,3],node([2,2],node([3,2],node([3,1],
node([2,1],node([1,1],[],[],0,0),right,1,1),right,2,2),down,3,3),left,4,4),do
wn,5,5),right,6,6),right,7,7),down,8,8)

0 número total de estados visitados foram: 471
0 número máximo de estados que estiveram simultaneamente em memória foram: 13
```

Figure 1: Resolução do exercício usando a pesquisa iterativa

Para o exercício 1 o algoritmo de pesquisa não informada que o grupo achou mais indicado para a sua resolução foi o algoritmo de **pesquisa iterativa**.

Este algoritmo é o indicado porque: é **completo**, isto é, se tem solução, encontra-o e é **óptimo**, ou seja, a solução encontrada é a mais directa. Quanto à complexidade temporal é $O(b^d)$ em que "b" é o número de nós expandidos por cada n, que no nosso caso iram ser quatro e "d" é a profundidade em que se encontra a solução. O valor da ramificação é no máximo quatro porque as operações que o problema dispõe são: ir para a esquerda, direita, para cima, para baixo.

No nosso caso a sequência de acções foram:

direita > direita > baixo > esquerda > baixo > direita > direita > baixo

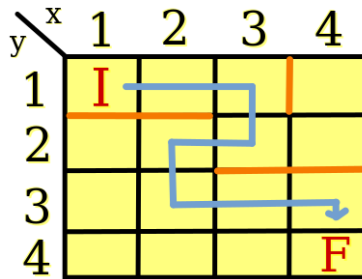


Figure 2: Esquema em que o I indica a posição inicial e F a posição final. As linhas a laranja indicam as passagens bloqueadas e a linha azul indica o caminho escolhido pelo algoritmo.

Para testar este exercício os ficheiros a usar são: p_limitada.pl e problema.pl. Quando importar os ficheiros para o interpretador SWIProlog o predicado para iniciar a pesquisa é "search .".

1.1 Número total de estados visitados

Tal como mostra o print a cima o número de nós visitados foram 471.

1.2 Número máximo de estados que estiveram simultaneamente em memória

O número máximo de estados que estiveram simultaneamente em memória foram 13.

Exercício 2

```
Solução:
node([4,4],node([4,3],node([3,3],node([2,3],node([2,2],node([3,2],node([3,1],
node([2,1],node([1,1],[],[],6,0,0),right,6,1,1),right,6,2,2),down,6,3,3),left
,8,4,4),down,8,5,5),right,8,6,6),right,8,7,7),down,8,8,8)

0 número total de estados visitados foram: 18
0 número máximo de estados que estiveram simultaneamente em memória foram: 23
```

Figure 3: Resolução do exercício usando o algoritmo A*

Para o exercício 2 o algoritmo de pesquisa informada escolhido foi a pesquisa **A***.

O algoritmo A*, tal como o algoritmo de pesquisa iterativa, é **completo** e **óptimo**. A razão pela qual escolhemos este algoritmo foi pelo facto da heurística ser mais específica do que a do algoritmo de pesquisa **Greedy**.

A heurística utilizada consistia na soma do custo do nó expandido com a distância desse nó até ao estado final.

Verificamos que enquanto o algoritmo tentava pesquisar pelo melhor caminho, este expandia acedeu varias vezes a nós que não levavam a resultado algum, por exemplo, este algoritmo foi da posição (3,2) para (4,2) e vice versa, várias vezes. Possivelmente, se afina-se-mos a heurística o algoritmo não iria para as posições erradas com tanta frequência.

Tal como o algoritmo de pesquisa iterativa a sequência de passos que este retornou foram:

direita > direita > baixo > esquerda > baixo > direita > direita > baixo

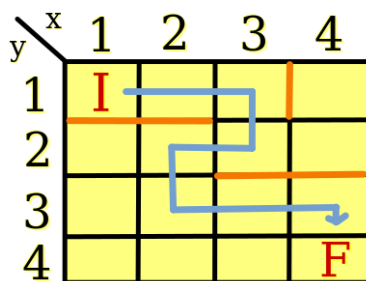


Figure 4: Esquema em que o I indica a posição inicial e F a posição final. As linhas a laranja indicam as passagens bloqueadas e a linha azul indica o caminho escolhido pelo algoritmo.

Para testar este exercício os ficheiros a usar são: `a_star.pl` e `problema.pl`. Quando importar os ficheiros para o interpretador SWIProlog o predicado para iniciar a pesquisa é `search .`.

1.3 Número total de estados visitados

O número de nós visitados foram 18.

1.4 Número máximo de estados que estiveram simultaneamente em memória

O número máximo de estados que estiveram simultaneamente em memória foram 22.