Inteligência Artificial

Relatório do 1º trabalho prático

2022/2023

Resolução de problemas como problemas de pesquisa no espaço de estados

Trabalho realizado por:
- Rodrigo Alves, nº48681
- Luís Simões, nº48726

Introdução

O primeiro trabalho prático proposto tem como objetivo utilizarmos as capacidades de resolução de problemas de pesquisa no espaço de estados através dos conhecimentos adquiridos nas aulas.

Para isso, devemos entender todas as pesquisas estudadas nas aulas, tanto pesquisas informadas, como pesquisas não informadas.

O objetivo final é, portanto, encontrar o algoritmo mais eficiente para resolver os problemas através de testes dos mesmos em cada problema.

Resolução dos Exercícios

Exercício 1

Para começar, foram feitas algumas decisões da nossa parte que devem ser explicadas para melhor entendimento da nossa resolução:

- ➔ O nosso tabuleiro começa no canto superior esquerdo;
- ➔ Os nossos estados são representados por e(coluna, linha);
- ➔ Dividimos os algoritmos de pesquisa informada e pesquisa não informada em dois ficheiros diferentes, chamados “*pi.pl*” e “*pni.pl*” respectivamente.

a) Para representar os estados do agente fizemos o seguinte:

```
estado_inicial(e(2,7)).
```

```
estado_final(e(5,1)).
```

Para representar os bloqueios, ou seja, os espaços que o nosso agente não pode passar ou ocupar fizemos algo semelhante ao seguinte:

```
posicao_bloqueios(e(1,2)).
```

Para representar os movimentos (operações) que o nosso agente pode fazer (cima, direita, baixo, esquerda), onde começamos por ver se a posição seguinte (e(X, Y1) - por exemplo) está dentro do tabuleiro, fazemos a operação correspondente ao movimento e verificamos se a posição para onde queremos movimentar está ou não bloqueada:

```
op(e(X,Y),cima,e(X,Y1),1):-
    Y > 1,
    Y1 IS Y-1,
    \+ posicao_bloqueios(e(X,Y1)).

op(e(X,Y),direita,e(X1,Y), 1) :-
   X < 7,
   X1 is X+1,
   \+ posicao_bloqueios(e(X1,Y)).

op(e(X,Y),baixo,e(X,Y1),1) :-
    Y < 7,
    Y1 is Y+1,
    \+ posicao_bloqueios(e(X,Y1)).

op(e(X,Y),esquerda,e(X1,Y),1) :-
    X > 1,
    X1 is X-1,
    \+ posicao_bloqueios(e(X1,Y)).
```

b) O algoritmo de pesquisa não informada mais eficiente a resolver este problema está disponível no ficheiro '*pni.pl*' .

c) Ao executar os algoritmos de pesquisa não informada sobre o problema 1 obtemos os seguintes resultados:

	P	L	PI
Estados Visitados	13	113	22449
Número máximo de nós em memória	13	57	13
Custo	9	9	9
Profundidade			

P - Pesquisa em profundidade;

L - Pesquisa em largura;

PI - Pesquisa em profundidade iterativa;

A partir da observação dos valores dados pela tabela, podemos concluir que o algoritmo que melhor se ajusta ao problema é o algoritmo de pesquisa em profundidade uma vez que este apresenta um valor de estados visitados muito inferior comparado com a dos outros algoritmos; também este possui o menor valor de estados guardados em memória.

d) Para este problema decidimos utilizar uma heurística muito conhecida e que é muitas vezes utilizada em vários problemas, que é a distância de Manhattan. Para além dessa, decidimos utilizar também uma heurística que consiste em calcular a distância considerando apenas o eixo das abscissas.

```
%Heurística das abscissas(colunas)
```

```
h(e(Xi,_),R):-
    estado_final(e(Xf,_)),
    S is abs(Xi-Xf),
    R is S.

%Heurística de Manhattan
h(e(Xi,Yi),R):-
    estado_final(e(Xf,Yf)),
    S is abs(Xi-Xf),
    D is abs(Yi-Yf),
    R is S+D.
```

e) O algoritmo de pesquisa informada mais eficiente a resolver este problema está disponível no ficheiro '*pi.pl*' .

f) Ao executar os algoritmos de pesquisa informada sobre o problema 1 obtemos os seguintes resultados:

	A* com H1	**A* com H2**	**Greedy com H1**	**Greedy com H2**
Estados Visitados	60	81	11	9
Número máximo de nós em memória	43	30	14	12
Custo	9	9	9	9
Profundidade	9	9	9	9

A* - Pesquisa ansiosa;

Greedy - Pesquisa gananciosa;

H1 - Distância de Manhattan;

H2 - Distância das abscissas;

Comparando os valores de cada algoritmo utilizado na pesquisa informada, podemos concluir que o algoritmo mais eficiente para este problema é o algoritmo de pesquisa greedy com a heurística das abscissas, pois esta pesquisa possui o menor valor de estados visitados como também do número máximo de nós em memória.

A partir dos resultados obtidos na pesquisa informada e não informada, apercebemo-nos que para este problema a pesquisa em profundidade teve um rendimento muito parecido relativamente à melhor pesquisa informada, a Greedy com H2.

Exercício 2

a) Para representar os estados do agente e da caixa decidimos pegar no que tínhamos feito antes e adicionar aos estados inicial e final, a posição da caixa ficando representado por *e(coluna do agente, linha do agente, coluna da caixa, linha da caixa)* e, por isso, fizemos o seguinte:

```
estado_inicial(e(2,7,2,6)).
```

```
estado_final(e(_,_,5,1)).
```

Para representar os bloqueios, ou seja, os espaços que o nosso agente não pode passar ou ocupar fizemos o mesmo que no exercício anterior:

```
posicao_bloqueios(e(1,2)).
```

Para representar os movimentos (operações) que o nosso agente pode fazer (cima, direita, baixo, esquerda) bem como a caixa, começamos realizar a operação sobre o agente e posteriormente averiguamos se a posição do mesmo (estado seguinte) é igual à posição da caixa (estado atual), caso isso aconteça é realizada sobre a caixa o mesmo movimento que o agente sofreu. Contudo, se isto não se realizar apenas o agente realiza a operação e o estado seguinte da caixa é igual ao estado atual. Após estes passos, é verificado se a posição do agente ou da caixa não coincide com um bloqueio, se não se verificar são atualizadas as posições caso contrário a operação fica sem efeito.

```
op(e(X,Y,Xc,Yc),cima,e(X,Y1,Xc,Y1c),1) :-
    Y1 is Y-1,
    (X =:= Xc , Y1 =:= Yc),
    Y1c is Yc-1,
    Y1c >=1, Y1c =< 7, Xc >= 1, Xc =< 7,
    \+ posicao_bloqueios(e(Xc,Y1c)).

op(e(X,Y,Xc,Yc),cima,e(X,Y1,Xc,Y1c),1) :-
    Y1 is Y-1,
    \+ (X =:= Xc , Y1 =:= Yc),
    Y1c is Yc,
    Y1 >=1, Y1 =< 7, X >= 1, X =< 7,
    \+ posicao_bloqueios(e(X,Y1)).

op(e(X,Y,Xc,Yc),direita,e(X1,Y,X1c,Yc),1) :-
    X1 is X+1,
    (X1 =:= Xc , Y =:= Yc),
```

```
    X1c is Xc+1,
    X1c >= 1, X1c =< 7, Yc >= 1, Yc =< 7,
    \+ posicao_bloqueios(e(X1c,Yc)).

op(e(X,Y,Xc,Yc),direita,e(X1,Y,X1c,Yc),1) :-
    X1 is X+1,
    \+ (X1 =:= Xc , Y =:= Yc),
    X1c is Xc,
    X1 >= 1, X1 =< 7, Y >= 1, Y =< 7,
    \+ posicao_bloqueios(e(X1,Y)).

op(e(X,Y,Xc,Yc),baixo,e(X,Y1,Xc,Y1c),1) :-
    Y1 is Y+1,
    (X =:= Xc , Y1 =:= Yc),
    Y1c is Yc+1,
    Y1c >= 1, Y1c =< 7, Xc >= 1, Xc =< 7,
    \+ posicao_bloqueios(e(Xc,Y1c)).

op(e(X,Y,Xc,Yc),baixo,e(X,Y1,Xc,Y1c),1) :-
    Y1 is Y+1,
    \+ (X =:= Xc , Y1 =:= Yc),
    Y1c is Yc,
    Y1 >= 1, Y1 =< 7, X >= 1, X =< 7,
    \+ posicao_bloqueios(e(X,Y1)).
```

```
op(e(X,Y,Xc,Yc),esquerda,e(X1,Y,X1c,Yc),1) :-
    X1 is X-1,
    (X1 =:= Xc , Y =:= Yc),
    X1c is Xc-1,
    X1c >= 1, X1c =< 7, Yc >= 1, Yc =< 7,
    \+ posicao_bloqueios(e(X1c,Yc)).

op(e(X,Y,Xc,Yc),esquerda,e(X1,Y,X1c,Yc),1) :-
    X1 is X-1,
    \+ (X1 =:= Xc , Y =:= Yc),
    X1c is Xc,
    X1 >= 1, X1 =< 7, Y >= 1, Y =< 7,
    \+ posicao_bloqueios(e(X1,Y)).
```

b) O algoritmo de pesquisa não informada mais eficiente a resolver este problema está disponível no ficheiro '*pni.pl*' .

c) Ao executar os algoritmos de pesquisa não informada sobre o problema 2 obtemos os seguintes resultados:

<table>
<tr><th></th><th>P</th><th>L</th><th>PI</th></tr>
<tr><td>Estados Visitados</td><td>187</td><td>3775</td><td rowspan="4">Indefinido</td></tr>
<tr><td>Número máximo de nós em memória</td><td>50</td><td>582</td></tr>
<tr><td>Custo</td><td>58</td><td>16</td></tr>
<tr><td>Profundidade</td><td>58</td><td>16</td></tr>
</table>

P - Pesquisa em profundidade;

L - Pesquisa em largura;

PI - Pesquisa em profundidade iterativa;

Ao tentar resolver o problema utilizando o algoritmo de pesquisa em profundidade iterativa não foi possível constatar nenhum valor, uma vez que a pesquisa não terminou e, por isso, não retornou nenhum output no terminal.

A partir da observação dos valores dados pela tabela, podemos concluir que o algoritmo que melhor se ajusta ao problema é o algoritmo de pesquisa em profundidade uma vez que este apresenta um valor de estados visitados muito inferior comparado com a dos outros algoritmos; também este possui o menor valor de estados guardados em memória.

d) Assim como no problema anterior, utilizámos a distância de Manhattan e uma heurística que consiste em calcular a distância considerando apenas o eixo das abscissas.

```
%Heurística das abscissas(colunas)

h(e(_,_,Xi,_),R):-

    estado_final(e(_,_,Xf,_)),
```

```
    S is abs(Xi-Xf),
    R is S.

%Heurística de Manhattan
h(e(_,_,Xi,Yi),R):-
    estado_final(e(_,_,Xf,Yf)),
    S is abs(Xi-Xf),
    D is abs(Yi-Yf),
    R is S+D.
```

e) O algoritmo de pesquisa informada mais eficiente a resolver este problema está disponível no ficheiro '*pi.pl*' .

f) Ao executar os algoritmos de pesquisa informada sobre o problema 2 obtemos os seguintes resultados:

	A* com H1	A* com H2	Greedy com H1	Greedy com H2
Estados Visitados	1166	Indefinido	311	987
Número máximo de nós em memória	436		51	179
Custo	16		16	20
Profundidade	16		16	20

A* - Pesquisa ansiosa;

Greedy - Pesquisa gananciosa;

H1 - Distância de Manhattan;

H2 - Distância das abscissas;

Ao tentar resolver o problema utilizando o algoritmo de pesquisa ansiosa com a heurística das abscissas, não foi possível constatar nenhum valor, uma vez que a pesquisa não terminou e, por isso, não retornou nenhum output no terminal.

Comparando os valores de cada algoritmo utilizado na pesquisa informada, podemos concluir que o algoritmo mais eficiente para este problema é o algoritmo de pesquisa greedy com a heurística de Manhattan, pois esta pesquisa possui o menor valor de estados visitados como também do número máximo de nós em memória.

Conclusão

Através da realização deste trabalho prático, foi nos possível pôr em prática as matérias lecionadas e permitiu-nos consolidar melhor essa mesma matéria.

Uma das formas em que nos beneficiou foi no facto de nos mostrar como realmente funcionam estes dois tipos de pesquisas (informada e não informada), assim como os algoritmos que as representam.

Por fim, concluímos que foram alcançados os objetivos iniciais do trabalho mas que, contudo, poderiam ter sido melhorados alguns aspetos nas implementações dos algoritmos que fariam um resultado mais eficiente, ou seja, matematicamente perfeito.