

Inteligência Artificial
Primeiro Trabalho – Algoritmos de Busca

Profa. Heloisa 2019/2

O problema do Aspirador de Pó foi implementado levando-se em consideração o uso de vetores, assim como o exemplo dos Canibais.

A matriz 2x2 foi representada da seguinte forma:

Q1	Q2
Q3	Q4

- Representação do estado

Levando-se em consideração a matriz acima, os estados foram representados no vetor da seguinte forma:

(Q1 Q2 Q3 Q4 PosiçãoAspirador)

Os quadrados Q1, Q2, Q3 e Q4 podem assumir valores 0 ou 1, em que 0 indica que o quadrado está limpo e 1 indica que o quadrado está sujo. Além disso, a variável PosiçãoAspirador indica em qual quadrado o aspirador se encontra e pode assumir valores de 1 a 4.

Os estados representados levando-se em conta em que quadrado o aspirador se encontra foram:

Aspirador no Quadrado 1	
Situação	Ação
Limpo	Direita
Limpo	Baixo

Sujo	Aspirar
------	---------

Aspirador no Quadrado 2	
Situação	Ação
Limpo	Baixo
Limpo	Esquerda
Sujo	Aspirar

Aspirador no Quadrado 3	
Situação	Ação
Limpo	Direita
Limpo	Cima
Sujo	Aspirar

Aspirador no Quadrado 4	
Situação	Ação
Limpo	Esquerda
Limpo	Cima
Sujo	Aspirar

- Operadores aplicáveis aos estados (ações)

O enunciado do problema indica que há 5 ações possíveis aplicáveis aos estados:

- 1) Aspirar;
- 2) Mover para a direita;
- 3) Mover para baixo;
- 4) Mover para a esquerda;
- 5) Mover para cima;

Seguindo essa proposta, foram aplicadas as ações para os quadrados dependendo da posição do aspirador de pó e se o quadrado está sujo ou não. Supondo que o aspirador de pó esteja no quadrado 1, é possível aplicar o operador 1 (Aspirar), 2 (Mover para a direita) ou 3 (Mover para a esquerda). Assim, essa funcionalidade foi implementada da seguinte forma:

```
if(qAtual == 1){  
  operadores <- list(c(-1,0,0,0,0), c(0,0,0,0,1), c(0,0,0,0,2))  
  filhosDesc <- lapply(operadores, function(op) desc+op)  
}
```

Os demais quadrados seguem a mesma linha de raciocínio e implementam somente as ações que conseguem realmente realizar.

- Os custos dos operadores

A função de custo foi implementada da seguintes forma: inicialmente é verificado se subtraindo a posição 1 da lista de operadores (indica que houve a ação de aspirar) do estado atual, o estado se torna igual ao estado pai, ou seja, houve a operação de aspirar. Caso seja verdade, retorna o custo do pai + 2:

```
if(all(atual$desc - ops[[1]] == atual$pai$desc)){  
  return(atual$pai$g + 2)
```

Da mesma forma, posteriormente é verificado se subtraindo a posição 2 da lista de operadores (indica que houve movimento a direita ou a esquerda) do estado atual, o estado se torna igual ao estado pai, ou seja, houve movimento ou a direita ou a esquerda. Caso seja verdade, retorna o custo do pai + 1:

```
if(all(atual$desc - ops[[2]] == atual$pai$desc)){  
  return(atual$pai$g + 1)
```

Caso nenhuma das opções acima aconteça, quer dizer que houve movimento ou para baixo ou para cima, assim é retornado o custo do pai + 3:

```
return(atual$pai$g + 3)
```

- Função heurística

A função heurística foi implementada da seguinte forma: inicialmente foi verificado se o aspirador está em um quadrado sem sujeira e, caso for verdade, é retornado duas vezes o número de quadrados sujos +1:

```
if(is.null(atual$desc[atual$desc[5]]))  
  return((2 * sum(atual$desc[-5])) + 1)
```

Caso contrário, é retornado duas vezes o número de quadrados sujos:

```
return(2 * sum(atual$desc[-5]))
```

- Estado inicial

O estado inicial especificado pelo problema deveria ter pelo menos dois quadrados sujos e o aspirador começando por um quadrado limpo. Dessa forma foi definido o seguinte estado inicial:

$$(Q1 = 1, Q2 = 0, Q3 = 1, Q4 = 1, A = 2)$$

- Estado objetivo

O estado objetivo definido pelo problema deveria ter todos os quadrados limpos. Dessa forma foi definido o seguinte estado objetivo:

$$(Q1 = 0, Q2 = 0, Q3 = 0, Q4 = 0, A = 1)$$