

Magic Cube 2x2 - MDP e Q-Learning

Trabalho realizado para a disciplina de Inteligência Artificial

Projeto: Resolução de um cubo mágico 2x2 utilizando MDP e Q-Learning

Autor: Gilmar Correia Jeronimo – 11014515

Sobre o projeto:

A ideia de projeto vem em resolver um cubo mágico 2x2 orientado, ou seja, conhecendo a posição de cada uma das faces do cubo. Por exemplo, a face 1 é a face orientada pela "centro virtual" verde, a face 2 pelo "centro" laranja e assim por diante.

Os estados são representados pelas posições de cada face numa tupla:

```
[=====]  
[==|G1,G2| |O1,O2| |Y1,Y2| |W1,W2| |B1,B2| |R1,R2|==]  
[==|G3,G4| , |O3,O4| , |Y3,Y4| , |W3,W4| , |B3,B4| , |R3,R4|==]  
[=====]
```



Sendo a face verde (G) a primeira posição da tupla, a laranja (O) a segunda posição da tupla, e assim por diante. Modelando o problema de modo que:

A face G é orientada de forma que Y está na direita, O em baixo, W na esquerda e R em cima, e B atrás.

Função ResultadoDaAção

A função ResultadoDaAcao recebe um estado atual, e uma ação executada neste estado, retornando assim o estado alterado pela ação. A modelagem do problema conta com os movimentos:

- > 'R','CW' = mover à direita no sentido horário.
- > 'L','CW' = mover à esquerda no sentido horário.
- > 'F','CW' = mover a face no sentido horário.
- > 'T','CW' = mover o topo no sentido horário.
- > 'B','CW' = mover a base no sentido horário.
- > 'R','CCW' = mover à direita no sentido anti-horário.
- > 'L','CCW' = mover à esquerda no sentido anti-horário.
- > 'F','CCW' = mover a face no sentido anti-horário.
- > 'T','CCW' = mover o topo no sentido anti-horário.
- > 'B','CCW' = mover a base no sentido anti-horário.

Resultado do MDP

Para emular o MDP foi criado um vetor de estados S com alguns elementos amostrados.

Como S não possui todos os estados, a função de utilidade se aproxima bem para estados próximos da solução, com até dois movimentos finais. Com mais movimentos executados, o algoritmo passa a transitar aleatoriamente entre os estados vizinhos até chegar em um mais próximo da solução.

Ex 1:

Embaralhar: (2, 'L', 'CCW'), (3, 'T', 'CCW'), (3, 'R', 'CW'), (6, 'F', 'CCW'), (2, 'L', 'CCW')

Solução: (6, 'L', 'CW'), (4, 'T', 'CW'), (5, 'L', 'CW')

Ex 2:

Embaralhar: (6, 'R', 'CCW'), (1, 'B', 'CW'), (6, 'L', 'CCW'), (3, 'T', 'CCW'), (3, 'B', 'CCW')

Solução: Não achou solução

Resultado do Q-Learning

Com o Q-Learning foi possível chegar em soluções mais factíveis dependendo do estados iniciais do problema e da quantidade de iterações do algoritmo para atualizar a função Q.

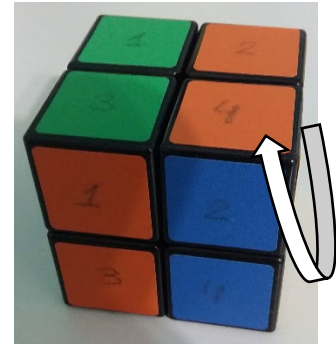
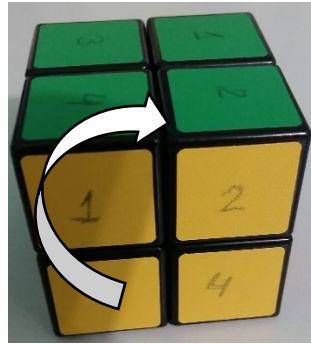
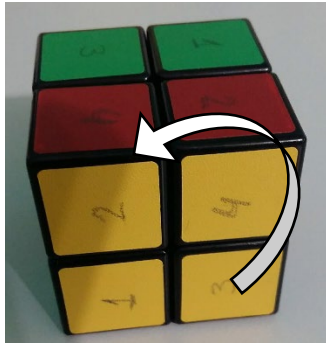
Ex 1:

Ao embaralhar o cubo realizando as respectivas ações aleatoriamente:

(3, 'F', 'CCW') -> Rodou a face amarela no sentido anti-horário

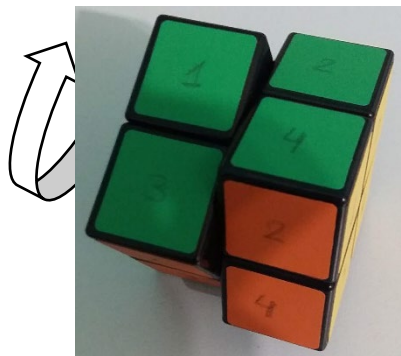
(3, 'F', 'CW') -> Rodou a face amarela no sentido horário

(5, 'L', 'CW') -> Rodou a parte esquerda da face azul no sentido horário



O algoritmo retorna as melhores ações partindo do estado embaralhado:

(1, 'L', 'CCW') -> Rodou a parte esquerda da face verde no sentido anti-horário



Chegando assim no estado final.

Ex 2:

Embaralhando: (5, 'B', 'CCW'), (3, 'R', 'CCW'), (3, 'L', 'CCW'), (5, 'B', 'CW'), (1, 'R', 'CCW')

Resposta: (2, 'R', 'CW'), (3, 'R', 'CW'), (3, 'L', 'CW')

Ex 3:

Embaralhando: (2, 'L', 'CW'), (3, 'T', 'CCW'), (2, 'B', 'CCW'), (4, 'B', 'CCW'), (6, 'B', 'CW')

Resposta: Não resolveu