

DCC



DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO



UFMG

RELATÓRIO DO TP

Aluno:
Gabriel Lopes Machado

Disciplina: Inteligência Artificial
Professora: Prof. Luiz Chaimowicz

Modelagem:

Estado

O estado de referência foi uma classe python que possui a seguinte estrutura básica:

```
class EightPuzzle(object):  
  
    def __init__(self, state, index=[], cor=0):  
  
    def indexes(self, lista, parent, n):  
  
    def movements(self):
```

Sendo que o método indexes é usado para criar as arestas da minha estrutura em GRAFO e o método movimento usado pelo método indexes para retornar quais os movimentos são válidos em um determinado estado.

Função Sucessora

Diz respeito de como os estados serão inseridos na fila de expansão de nós

```
if (lista[child[2]].cor == 0) and (node.deep < deep):  
    lista[child[2]].move = child  
    lista[child[2]].deep = node.deep + 1  
    if not test_goal(lista[child[2]]):  
  
        lista[child[2]].cor = 1  
        bisect.insort(frontier, lista[child[2]])  
  
    else:  
        goal_state = lista[child[2]]  
        return goal_state, frontier
```

Basicamente é feita uma simples validação se o nó expandido já está na fronteira conferindo sua cor, sendo que todos os nós são testados no momento em que são expandidos e, obviamente, são tratados os casos de nós repetidos.

Diferença entre os algoritmos

De forma objetiva, o que diferencia os diferentes algoritmos é a forma com que ele organiza a ordem de expansão e visitação de cada nó, sendo uma breve descrição:

1. BFS - Coloca a lista de expansão como sendo os nós pertencentes ao nó mais superficiais antes dos nós mais profundos.
2. IDS - Coloca o nó mais profundidade primeiro na lista de expansão, além de dar um limite incremental na profundidade máxima
3. UCS - Organiza a lista de forma que os nós de menores custo são visitados primeiro.

4. A* - Idem ao UCS, só que o custo é medido como o custo real do nó mais uma heurística que tenta mensurar a distância real até o bjetivo
5. Greedy - Olha apenas para a heurística e monta uma fila baseado apenas na heurística desse estado
6. Hill Climbing - Uma forma de busca que se baseia na heurística e que não apresenta uma fila, apenas tem como objetivo minimizar o próximo passo

Heurísticas Utilizadas

As heurísticas foram a de quadrados fora de posição e a distância de Manhattan, algo importante sobre elas é que ambas são uma simplificação do problemas total, ou seja, elas "subestimam" o problema. Que pelo teorema de otimidade do algoritmo do A* faz com que elas sejam admissíveis e ambas achem uma solução ótima.

Algumas soluções

Vejam alguns tipos de soluções encontrados por dois algrítmos definidos, sendo um ótimo e o outro não

- BFS para a solução 31:
['Cima', 'Cima', 'Direita', 'Baixo', 'Esquerda', 'Esquerda', 'Baixo', 'Direita', 'Direita', 'Cima', 'Esquerda', 'Esquerda', 'Cima', 'Direita', 'Direita', 'Baixo', 'Esquerda', 'Baixo', 'Esquerda', 'Cima', 'Cima', 'Direita', 'Baixo', 'Baixo', 'Esquerda', 'Cima', 'Cima', 'Direita', 'Direita', 'Baixo', 'Baixo']
- Greede para a solução 31:
['Cima', 'Direita', 'Cima', 'Esquerda', 'Esquerda', 'Baixo', 'Baixo', 'Direita', 'Direita', 'Cima', 'Cima', 'Esquerda', 'Baixo', 'Baixo', 'Direita', 'Cima', 'Cima', 'Esquerda', 'Baixo', 'Direita', 'Cima', 'Esquerda', 'Esquerda', 'Baixo', 'Baixo', 'Direita', 'Cima', 'Esquerda', 'Cima', 'Direita', 'Direita', 'Baixo', 'Esquerda', 'Cima', 'Esquerda', 'Baixo', 'Baixo', 'Direita', 'Cima', 'Cima', 'Direita', 'Baixo', 'Esquerda', 'Esquerda', 'Cima', 'Direita', 'Baixo', 'Direita', 'Cima', 'Esquerda', 'Baixo', 'Esquerda', 'Cima', 'Direita', 'Baixo', 'Direita', 'Cima', 'Esquerda', 'Esquerda', 'Baixo', 'Baixo', 'Direita', 'Direita']

TABELA DE COMPARAÇÃO DA SOLUÇÃO 23
(8, 4, 7, 5, 6, 2, 1, 0, 3)

Algoritmo	Estados Expandidos	Passos para a solução	Tempo de execução (s)
BFS	99210	23	0.25819897651672363
IDS	54107	23	1.0552241802215576
UCS	99210	23	1.084855318069458
A* (heurística 1)	128919	23	1.8559460639953613
A* (heurística 2)	41965	23	0.3910808563232422
Greedy(heurística 1)	58579	71	0.6362886428833008

Greedy(heurística 2)	18929	139	0.13306164741516113
Hill Climbing	-----	-----	-----

Obs: Não obtive sucesso na implementação do Hill Climbing

*

Discussão dos resultados obtidos

Pelos resultados obtidos, observo que ainda seria cabível várias otimizações em vários algoritmos, porém, diferentes estratégias, usadas de diversas maneiras, fazem com que seja possível "quebrar" um dado problema usando-se da modelagem de estados bem determinados e determinístico. Os resultados mostram que diferentes implementações e suas respectivas acabam por ter resultados que são muitos positivos por um lado (o tempo e espaço do greedy), porém não garantem a otimidade nem a completude do problema. Mesmo que as diferentes estratégias e métodos continuam por estar relacionadas entre si, elas são muito importantes para entender que várias tentativas e erros são necessários para a conquista de resultados cada vez melhores e um passo mais perto de uma solução ideal.

Sobre o N-Puzzle

O sistema foi planejado para suportar solução para o n-puzzle, porém, não fui capaz de aplicar alguma otimização para que seja possível executar em meu computador, a complexidade de espaço não foi superada.

Obs: dado o fato da minha inesperienza no conteúdo de algoritmos (fiz só até PDS2), a dedicação na frente do livro do Cormen para realizar essa atividade me encheu de satisfação após os resultados e a capacidade de realizar este TP com a máxima qualidade possível.