

Relatório de Busca Heurística no Problema do 8-Puzzle

Pedro Magno, Breno Moreira, Lucas Lopes, Guilherme Meyer e Pedro Hosken

07/12/2025

1 Introdução

O objetivo deste trabalho é comparar diferentes algoritmos de busca aplicados ao problema clássico do **8-Puzzle**. O problema consiste em um tabuleiro 3x3 contendo oito peças numeradas e um espaço vazio. O estado final desejado é:

1	2	3
4	5	6
7	8	0

A tarefa do programa é encontrar a sequência mínima de movimentos que leva do estado inicial fornecido até o estado objetivo. Foram implementados e avaliados três algoritmos de busca:

- Busca em Largura (BFS)
- Busca Gulosa
- A* (com heurística Manhattan)

Todos os algoritmos foram analisados com respeito ao tempo de execução, número de nós visitados (expandidos), número de nós gerados, profundidade da solução e caminho obtido.

2 Descrição dos Algoritmos

2.1 Busca em Largura (BFS)

A BFS expande primeiro os nós mais superficiais da árvore de busca. Ela garante a **solução ótima**, pois encontra sempre o menor número de movimentos até o objetivo.

Vantagens:

- Sempre encontra a menor solução.
- Fácil de implementar.

Desvantagens:

- Consome muito espaço em memória.
- Se o estado estiver muito distante da solução, pode se tornar inviável.

2.2 Busca Gulosa

A busca gulosa utiliza uma heurística para estimar quão perto está do objetivo. Em cada passo, seleciona o estado que **parece estar mais próximo da solução**. A avaliação é feita somente pelo valor de $h(n)$.

Vantagens:

- Rápida em muitos casos.
- Consome menos memória que BFS.

Desvantagens:

- Não garante solução ótima.
- Pode entrar em ciclos ou caminhos ruins.

2.3 A*

O algoritmo A* utiliza a função:

$$f(n) = g(n) + h(n)$$

onde:

- $g(n)$ = custo do caminho percorrido até o nó
- $h(n)$ = estimativa da distância até o objetivo

Quando a heurística é admissível (nunca superestima), o A* é garantidamente **ótimo e completo**.

3 Heurísticas Escolhidas

Foram utilizadas duas heurísticas:

3.1 Heurística das Peças Fora do Lugar (Misplaced Tiles)

Conta quantas peças não estão na posição correta.

$$h(n) = \#\{i \mid estado[i] \neq objetivo[i]\}$$

É simples, rápida e admissível.

3.2 Distância de Manhattan

Soma a distância horizontal e vertical de cada peça até sua posição correta:

$$h(n) = \sum_i |x_i - x_i^*| + |y_i - y_i^*|$$

É mais precisa que a heurística anterior e também é admissível.

4 Resultados Experimentais

Os testes foram realizados utilizando o estado inicial:

714083265

Os dados coletados seguem abaixo.

Algoritmo	Tempo (s)	Nós Expandidos	Nós Gerados	Profundidade
BFS	0.08177	35155	49791	19
Gulosa	0.00299	391	661	31
A* (Manhattan)	0.00293	241	394	19

Table 1: Resultados experimentais dos três algoritmos.

5 Análise dos Resultados

5.1 Tempo de Execução

O algoritmo A* apresentou o melhor desempenho geral, pois consegue equilibrar custo real e custo estimado para guiar a busca. A busca gulosa foi rápida, porém encontrou caminhos não ótimos. A BFS, por sua vez, teve o pior tempo devido ao crescimento exponencial da árvore.

5.2 Nós Visitados

A BFS visitou muito mais nós por não utilizar heurística. A busca gulosa visitou poucos nós, mas pode tomar decisões ruins. O A* visitou significativamente menos nós que a BFS enquanto manteve a qualidade da solução.

5.3 Profundidade da Solução

O A* encontrou a solução ótima, assim como a BFS. A busca gulosa encontrou uma solução mais profunda (não ótima).

6 Caminho da Solução

Abaixo está o caminho retornado pelo algoritmo A* e BFS:

714
083
265

714
803
265

704
813
265

740
813
265

743
810
265

743
815
260

743
815
206

743
815
026

743
015
826

043
715
826

403
715
826

413
705
826

413
725
806

413
725
086

413
025
786

013
425
786

103
425
786

123
405
786

123
450
786

123
456
780

Abaixo está o caminho retornado pelo algoritmo Gulosa:

714
083
265

014
783
265

104
783
265

140
783
265

143
780
265

143
785
260

143
785
206

143
705
286

143
750
286

143
756
280

143
756
208

143
756
028

143
056
728

143
506
728

143
526
708

143
526
780

143
520
786

140
523
786

104
523
786

124
503
786

124
053
786

024
153
786

204
153
786

240
153
786

243
150
786

243
105
786

203
145
786

023
145
786

123
045
786

123
405
786

123
450
786

123
456
780

7 Conclusão

Com base no experimento:

- O A^* foi o algoritmo mais eficiente e escalável.
- A heurística Manhattan forneceu excelente desempenho.
- A BFS serve como referência para solução ótima, mas é impraticável em estados muito difíceis.
- A busca gulosa pode ser rápida, porém não é confiável para obter caminhos ótimos.

O A^* demonstrou ser a melhor abordagem para o problema do 8-Puzzle.