

Universidade Estadual de Maringá  
Departamento de Informática

## **Relatório: N-PUZZLE**

Aluna: Thais Aparecida Silva Camacho (93807)  
Professor: Dr. Wagner Igarashi

Dezembro  
2017

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Apresentação</b>	<b>1</b>
2.1	Algoritmo A* . . . . .	1
2.2	N-Puzzle . . . . .	1
<b>3</b>	<b>Análise dos Resultados</b>	<b>2</b>
3.1	8-puzzle . . . . .	2
3.2	15-puzzle . . . . .	3
3.3	24-puzzle . . . . .	4
3.4	35-puzzle . . . . .	4

# 1 Resumo

O presente relatório apresenta uma comparação entre dois algoritmos usados na resolução do n-puzzle. A comparação buscou encontrar os melhores resultados entre os algoritmos analisados. Para a análise um conjunto de casos de teste é executado.

## 2 Apresentação

O jogo escolhido para o trabalho da disciplina foi o n-puzzle. Implementou-se o  $A^*$  para a resolução do jogo junto com duas heurísticas  $h'1$  e  $h'2$ . Onde

1.  $h'1(n)$  = número de peças fora de seu lugar na configuração final.
2.  $h'2(n)$  = para cada peça fora de seu lugar somar a distância Manhattan (quantidade de deslocamentos) para colocar em seu devido lugar. Neste caso considera-se que o caminho esteja livre para fazer o menor número de movimentos.

O objetivo do presente relatório é analisar o desempenho das duas heurísticas na resolução do n-puzzle.

### 2.1 Algoritmo $A^*$

O algoritmo  $A^*$ , proposto por Peter Hart, Nils Nilsson e Bertram Raphael, é um algoritmo de busca em grafos que utiliza uma função  $f(n)$  para estimar o custo do caminho mais curto entre origem e destino que passa pelo nó  $n$ . A função é definida por

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

em que  $g(n)$  representa o custo exato do caminho entre o nó inicial e qualquer nó  $n$  e  $h(n)$  representa uma estimativa heurística. O algoritmo explora sempre o nó de menor  $f(n)$ . Se a heurística escolhida  $h(n)$  é admissível, o  $A^*$  retorna um caminho ideal.

### 2.2 N-Puzzle

O N-puzzle consiste de um tabuleiro com  $N + 1$  posições, onde uma posição é vazia e as outras  $N$  possuem uma peça numerada. O objetivo é movimentar as peças e chegar no tabuleiro final. O tabuleiro final do 8-puzzle, por exemplo, é dado por

1	2	3
4	5	6
7	8	0

### 3 Análise dos Resultados

Os critérios considerados na análise são: consumo de memória(quantidade de nós expandidos) e tempo de processamento(milisegundos). Realizou-se casos de teste para o 8-puzzle, 15-puzzle, 24-puzzle e 35-puzzle. Os resultados são expostos a seguir.

#### 3.1 8-puzzle

Os testes considerados para o 8-puzzle foram

```
(define 8-inst-1 (list 6 4 2 1 5 3 7 0 8))
(define 8-inst-2 (list 0 2 5 1 8 3 7 4 6))
(define 8-inst-3 (list 8 0 7 6 5 4 3 2 1))
(define 8-inst-4 (list 8 3 2 5 1 0 7 4 6))
(define 8-inst-5 (list 6 4 7 8 5 0 3 2 1))
```

Figura 1: Casos de Teste para o 8-puzzle

E os resultados obtidos foram os apresentados a seguir:

Caso	Tempo	
	$h'1$	$h'2$
1	8.17	7.82
2	110.88	20
3	3769.73	19.8
4	1578.57	263.84
5	11084.59	621.97

Caso	Memória	
	$h'1$	$h'2$
<b>1</b>	88	34
<b>2</b>	2702	751
<b>3</b>	98191	432
<b>4</b>	48665	5942
<b>5</b>	235150	12284

### 3.2 15-puzzle

Os testes considerados para o 15-puzzle foram

```
(define 15-inst-1 (list 1 0 3 4 11 2 13 14 10 12 15 5 8 9 7 6))
(define 15-inst-2 (list 2 13 3 14 10 1 12 4 5 11 15 8 0 9 7 6))
(define 15-inst-3 (list 1 11 5 13 0 10 7 3 14 6 4 2 9 12 8 15))
(define 15-inst-4 (list 12 1 2 3 11 13 14 4 10 15 6 5 0 9 8 7))
(define 15-inst-5 (list 1 2 3 4 11 12 13 14 10 9 15 5 0 8 7 6))
```

Figura 2: Casos de Teste para o 15-puzzle

E os resultados obtidos foram os apresentados a seguir:

Caso	Tempo	
	$h'1$	$h'2$
<b>1</b>	595.91	160.97
<b>2</b>	-	1076.33
<b>3</b>	-	8001.73
<b>4</b>	-	7866.44
<b>5</b>	-	38089.02

Caso	Memória	
	$h'1$	$h'2$
<b>1</b>	17971	4339
<b>2</b>	-	24831
<b>3</b>	-	178484
<b>4</b>	-	136372
<b>5</b>	-	726085

### 3.3 24-puzzle

Os testes considerados para o 24-puzzle foram

```
(define 24-inst-1 (list 2 3 0 4 5 1 7 8 9 10 6 11 12 13 15 16 17 18 14 20 21 22 23 19 24))
(define 24-inst-2 (list 1 2 3 5 10 6 7 8 4 0 11 12 13 9 15 16 17 18 14 20 21 22 23 19 24))
(define 24-inst-3 (list 2 3 5 4 10 12 1 6 7 15 13 8 9 0 20 11 17 18 14 24 16 21 22 23 19))
(define 24-inst-4 (list 11 6 1 2 3 21 7 8 4 5 17 16 12 10 15 0 13 14 18 9 22 23 19 20 24))
(define 24-inst-5 (list 2 3 4 5 10 6 1 7 8 9 16 11 12 13 15 17 18 14 0 20 21 22 23 19 24))
```

Figura 3: Casos de Teste para o 24-puzzle

E os resultados obtidos foram os apresentados a seguir:

Caso	Tempo	
	$h'1$	$h'2$
1	2.41	3.07
2	2.39	4.40
3	-	42.023
4	-	4097.83
5	-	12.05

Caso	Memória	
	$h'1$	$h'2$
1	24	24
2	17	17
3	-	657
4	-	66212
5	-	142

### 3.4 35-puzzle

Os testes considerados para o 35-puzzle foram

```
(define 35-inst-1 (list 1 2 4 5 6 12 7 8 3 17 10 18 14 9 16 28 11 24 13 19 20 22 29 23 25 26 0 21 30 15 31 32 27 33 34 35))
(define 35-inst-2 (list 1 2 4 5 6 12 7 0 3 17 10 18 14 8 16 28 11 24 13 9 20 22 29 23 25 19 26 21 30 15 31 32 27 33 34 35))
(define 35-inst-3 (list 7 1 4 5 6 12 0 2 3 17 10 18 14 8 16 28 11 24 13 9 20 22 29 23 25 19 26 21 30 15 31 32 27 33 34 35))
```

Figura 4: Casos de Teste para o 35-puzzle

E os resultados obtidos foram os apresentados a seguir:

Caso	Tempo	
	$h'1$	$h'2$
<b>1</b>	-	18918.41
<b>2</b>	-	40777.10
<b>3</b>	-	47531.65

Caso	Memória	
	$h'1$	$h'2$
<b>1</b>	-	191839
<b>2</b>	-	463531
<b>3</b>	-	444916

Os resultados com - obteve-se estouro de memória no *Dr.Racket*. Como ambas as heurísticas são admissíveis, todos os teste que não estouraram memória, obteram a quantidade mínima de movimentos, ou seja, todas deram a solução ótima. Contudo, tiveram uma diferença grande de desempenho.

A heurística com maior desempenho foi a  $h'2$  que resolveu grande parte dos casos de teste de forma rápida e com quantidade de memória satisfável. Diferente da  $h'1$  que, quando não obtém estouro de memória, resolve utilizando-se uma quantidade muito maior do que a  $h'2$ , como pode se observar melhor na figura 5.

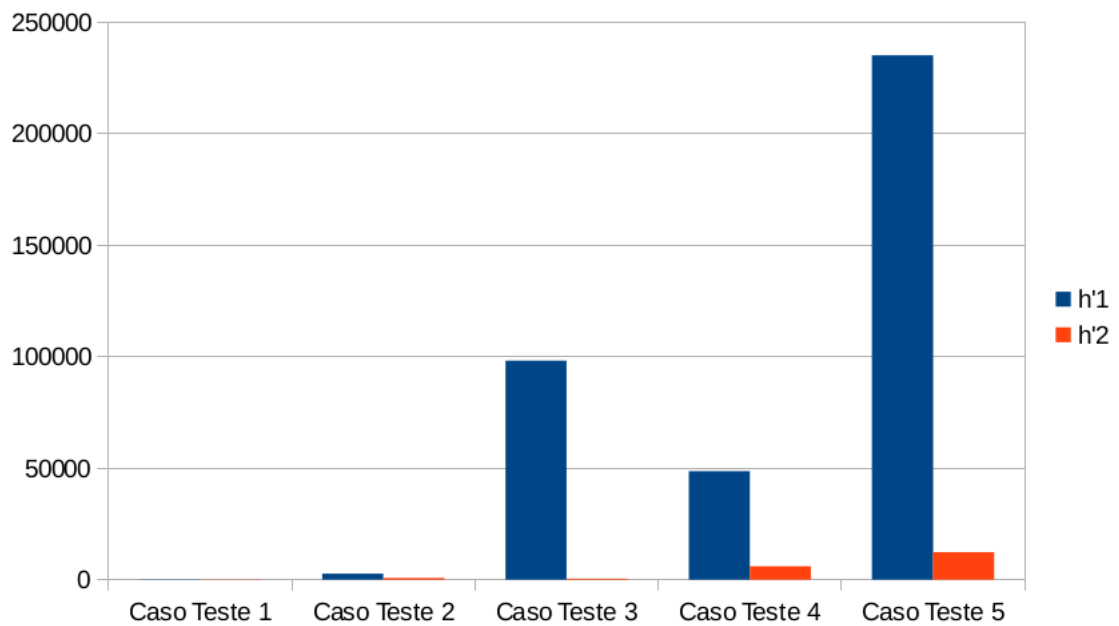


Figura 5: Comparação de consumo de memória entre  $h'1$  e  $h'2$  para os casos do 8-puzzle

Note ainda que, além de se ter uma diferença considerável no desempenho de acordo com a heurística utilizada, quanto maior o tamanho do problema, maior é o tempo de processamento e a memória utilizada, basta ver a grande diferença entre os resultados do 8-puzzle com o 35-puzzle.

Dessa forma, conclui-se que a  $h'2$  obteve um melhor desempenho.