

# **Relatório de Modelagem e Otimização de Algoritmos**

## **A\* aplicado no jogo 15-Puzzle**

**Fernando E. A. de Carvalho, Vinícius P. de Camargo**

<sup>1</sup>Departamento de Informática – Universidade Estadual de Maringá (UEM)  
Av. Colombo, 5790 - UEM - Bloco C56 – Maringá – PR – Brazil

{ra88408, ra88063}@uem.br

### **1. Introdução e resumo do problema**

Criado nos Estados Unidos da América no final do século XIX e conhecido no Brasil desde 1995, o Jogo dos 15 ou 15-Puzzle consiste de realizar sequências de movimentos que levem a uma determinada sequência final, como demonstrado pela Figura 1.



**Figura 1. Tabuleiro em estado final para o objetivo deste presente trabalho.**

Para a resolução do problema, o algoritmo A\* foi indicado; sendo o mesmo inicialmente proposto por Peter Hart, Nils Nilsson, e Bertram Raphael em 1968 e chamado apenas de A, recebendo posteriormente o nome de A\* quando este possuía uma heurística admissível, ou seja, levava ao resultado ótimo.

### **2. Objetivos**

Segundo o objetivo do trabalho proposto para disciplina de Modelagem e Otimização Algorítmica do curso de Bacharelado em Informática, o algoritmo A\*, que pode ser desenvolvido em qualquer linguagem a qual a plataforma run.codes suporte, deve ser executado e os resultados presentes na Seção 3 devem levar em consideração a resposta obtida, a memória consumida na execução do algoritmo A\* e também seu tempo de execução.

### 3. Resultados

A presente seção apresenta os resultados dos testes, sendo estes conduzidos em dois computadores distintos segundo a necessidade de memória de cada instância do problema, sendo os respectivos hardwares identificados a seguir:

- Computador Pessoal (PC)<sup>1</sup>
  - Processador: Intel i5 3ª geração CPU @ 2.50Ghz;
  - Memória RAM: 6Gb (4Gb alocados para os testes);
  - Sistema Operacional: Windows 10 Home Edition;
  - Versão JVM: 1.8.
- Servidor do Departamento de Informática - UEM (Pitomba)
  - Processador: Intel Xeon®5000 CPU @ 2.30Gzh;
  - Memória RAM: 128Gb (45Gb alocados para os testes);
  - Sistema Operacional: Debian GNU/Linux 8 (jessie);
  - Versão JVM: 1.9;

As subseções apresentam as instâncias do problema, uma breve explicação de cada heurística e os resultados de cada caso de execução<sup>2</sup>.

#### 3.1. Instâncias do problema

1. 5 13 6 10 1 7 2 9 4 3 15 14 8 0 11 12
2. 2 10 11 9 3 1 0 13 4 6 7 14 5 8 12 15
3. 5 9 13 10 2 6 14 15 1 4 7 12 0 3 11 8
4. 7 11 4 5 0 6 15 8 14 1 3 13 9 12 10 2
5. 5 10 9 14 7 3 13 6 1 15 0 12 8 2 4 11
6. 0 9 3 7 1 14 6 4 2 11 12 15 13 8 10 5
7. 3 9 0 7 2 1 6 5 11 13 4 12 8 14 15 10
8. 9 6 7 4 2 1 5 12 8 3 11 0 14 15 10 13
9. 2 9 4 5 0 7 11 12 14 6 3 13 1 8 15 10
10. 7 11 5 12 9 8 6 13 2 3 4 10 14 1 15 0

#### 3.2. Heurística 1

Esta heurística baseia-se na quantidade de peças que estão fora do lugar em comparação com a solução final, já que cada peça errada necessitará de no mínimo um movimento para que se chegue na conclusão, assim esta heurística torna-se uma heurística aceitável.

---

<sup>1</sup>Este hardware apresenta não só economia de tempo devido à frequência do processador, mas economia de memória para uma mesma instância, devido possivelmente a alguma otimização feita nas instruções do processador.

<sup>2</sup>Testes que ocuparam mais de 45 Gigabytes de memória foram interrompidos.

<b>Instância</b>	<b>Passos</b>	<b>Tempo (milisegundos)</b>	<b>Memória (mb)</b>	<b>Hardware utilizado</b>
1	20	31	3,36	PC
2	27	733	123,92	PC
3	27	391	81,22	PC
4	—	—	—	Pitomba
5	34	57.800	4.914,34	Pitomba
6	—	—	—	Pitomba
7	—	—	—	Pitomba
8	—	—	—	Pitomba
9	—	—	—	Pitomba
10	—	—	—	Pitomba

**Tabela 1. Aplicação da heurística 1 nas instâncias do problema.**

### 3.3. Heurística 2

Na segunda heurística é contado a quantidade de peças que estão fora de ordem crescente das quinze peças. Esta heurística segue o princípio da primeira em que no mínimo um movimento será necessário para posicionar uma peça errada ao seu devido lugar.

<b>Instância</b>	<b>Passos</b>	<b>Tempo (milisegundos)</b>	<b>Memória (mb)</b>	<b>Hardware utilizado</b>
1	20	31	4,80	PC
2	27	6.048	426,01	PC
3	27	1.671	438,52	PC
4	—	—	—	Pitomba
5	34	148.016	12.222,72	Pitomba
6	—	—	—	Pitomba
7	—	—	—	Pitomba
8	—	—	—	Pitomba
9	—	—	—	Pitomba
10	—	—	—	Pitomba

**Tabela 2. Aplicação da heurística 2 nas instâncias do problema.**

### 3.4. Heurística 3

A terceira heurística calcula a distância manhattan, que é o cálculo da distância em que a peça está, para o seu devido lugar na posição de solução supondo um caminho livre. Como será necessário pelo menos a quantidade retornada pelo cálculo da distância manhattan para posicionar a peça em sua posição de solução, tal heurística se mostra válida e apresenta o melhor desempenho entre todas as heurísticas testadas.

Instância	Passos	Tempo (milisegundos)	Memória (mb)	Hardware utilizado
1	20	15	1,92	PC
2	27	46	8,64	PC
3	27	62	20,64	PC
4	57	58.175	2.795,67	PC
5	34	250	15,48	PC
6	56	96.368	9.225,79	Pitomba
7	44	859	111,98	PC
8	51	24.407	1.482,84	PC
9	49	39.531	2.047,94	PC
10	50	8.703	518,72	PC

**Tabela 3. Aplicação da heurística 3 nas instâncias do problema.**

### 3.5. Heurística 4

Na heurística quatro é utilizada a soma das 3 primeiras heurísticas, cada uma com um peso respectivo, a soma dos pesos deve ser igual a um, e os mesmos foram selecionados de forma empírica.

Instância	Passos	Tempo (milisegundos)	Memória (mb)	Hardware utilizado
1	20	15	1,92	PC
2	27	94	19,68	PC
3	27	78	21,60	PC
4	—	—	—	Pitomba
5	34	297	43,17	PC
6	—	—	—	Pitomba
7	44	10.094	824,55	PC
8	51	245.277	17.856,90	Pitomba
9	49	202.342	15.801,71	Pitomba
10	50	102.268	8.300,00	Pitomba

**Tabela 4. Aplicação da heurística 4 nas instâncias do problema.**

### 3.6. Heurística 5

A quinta e última heurística implementada usa o maior valor das três primeiras heurísticas, como todas as três são heurísticas aceitáveis, esta por sua vez também se torna. Um problema encontrada no cálculo das heurísticas quatro e cinco, é que para realizar o mesmo é necessário o cálculo das heurísticas um, dois, e três, desta forma tendo um custo computacional mais alto, no entanto, escolhas entre as melhores opções das heurísticas um, dois e três otimizam o processo, de modo que as duas últimas heurísticas apresentam desempenho muito melhor que as heurísticas um e dois.

Instância	Passos	Tempo (milisegundos)	Memória (mb)	Hardware utilizado
1	20	16	1,92	PC
2	27	63	19,68	PC
3	27	78	20,16	PC
4	57	129.826	12.213,87	Pitomba
5	34	281	19,37	PC
6	56	378.224	28.772,61	Pitomba
7	44	3.094	218,44	PC
8	51	44.690	2.204,32	PC
9	49	83.199	7.053,88	Pitomba
10	50	24.486	1.352,284	PC

**Tabela 5. Aplicação da heurística 5 nas instâncias do problema.**

#### **4. Instruções de execução**

As instruções de execução assumem que o leitor tenha um compilador de java instalado em seu computador.

1. Abra a pasta que contém o arquivo Main.java
2. Abra o terminal
3. Digite `javac Main.java`
4. Digite `java -Xmx4G Main < arquivoDesejado.in` //Neste exemplo a memória máxima alocada para a Heap da JVM (*Java Virtual Machine*) é 4G, mas esta quantidade não resolve todas as instâncias e o `arquivoDesejado.in` deve ser substituído pelo arquivo que deseja-se testar.