

Inteligência Artificial

Nome: Vinícius Morais Silva

Matricula: 201205007

Diferenças dos Algoritmos

A* com heurísticas:

Peças fora do lugar: Nessa heurística o custo de cada nó expandido será o custo de profundidade do nó pai somado ao novo custo +1, esse novo custo será calculado de acordo com o número de peças fora do lugar, para cada peça fora do lugar o custo será incrementado em 1.

Distância de Manhattan: Aqui o custo de cada nó expandido será calculado a partir da soma do custo de profundidade do nó pai ao custo calculado de acordo com a distância de Manhattan +1. Esse cálculo será feito somando quantos movimentos cada peça terá que fazer para chegar ao seu respectivo lugar.

OBS: A quantidade de movimentos na Distância de Manhattan é calculada da seguinte forma:

$$\sum |X_i - Z_i| + |Y_i - W_i|$$

Onde i varia de 1 a 9.

Y é a coluna atual de onde a peça está e X é a linha onde a peça está.

W é a coluna onde a peça deveria estar e Z é a linha onde a peça deveria estar.

Busca em largura:

O algoritmo de busca em largura irá expandir toda a árvore até encontrar o estado procurado. Nesse algoritmo é implementado uma fila, ou seja, todos os nós que são gerados pela expansão irão ser retirados da fila e analisados em ordem de chegada. Isso garante que todos os nós de custo 1 serão visto antes que qualquer nó de custo 2 seja visto, depois todos os nós de custo 2 serão analisados antes que qualquer um de custo 3 seja analisado, e assim por diante.

Esse algoritmo é ótimo para esse problema, pois quando achar o estado da resposta irá garantir que não há outro estado resposta de menor custo.

Busca em profundidade:

Esse algoritmo irá expandir a árvore usando uma pilha, ou seja, o algoritmo irá empilhar todos os nós encontrados, e irá retirar o nós da fronteira do topo da pilha. Apesar do algoritmo buscar a resposta, ele não irá garantir que ela seja ótimo, pois esse algoritmo pode encontrar uma nodo de resposta que seja o de pior custo, lembrando que há mais de uma solução ou nenhuma para cada caso.

Testes

Soluções para o estado: "123456_78"

Gráfico com o número de estados expandidos.

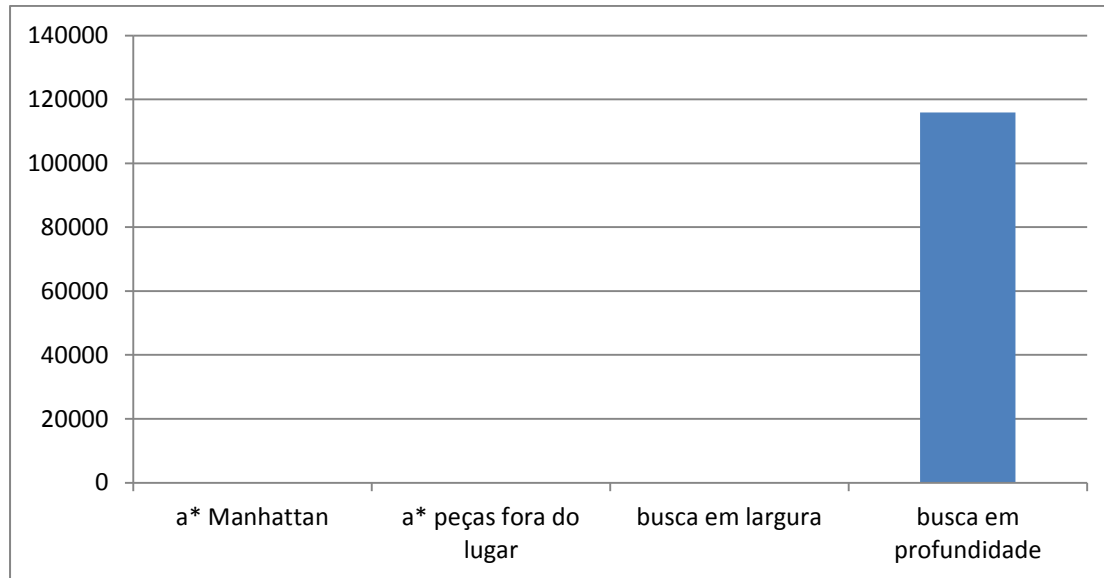


Gráfico com o tempo em segundos de cada algoritmo.

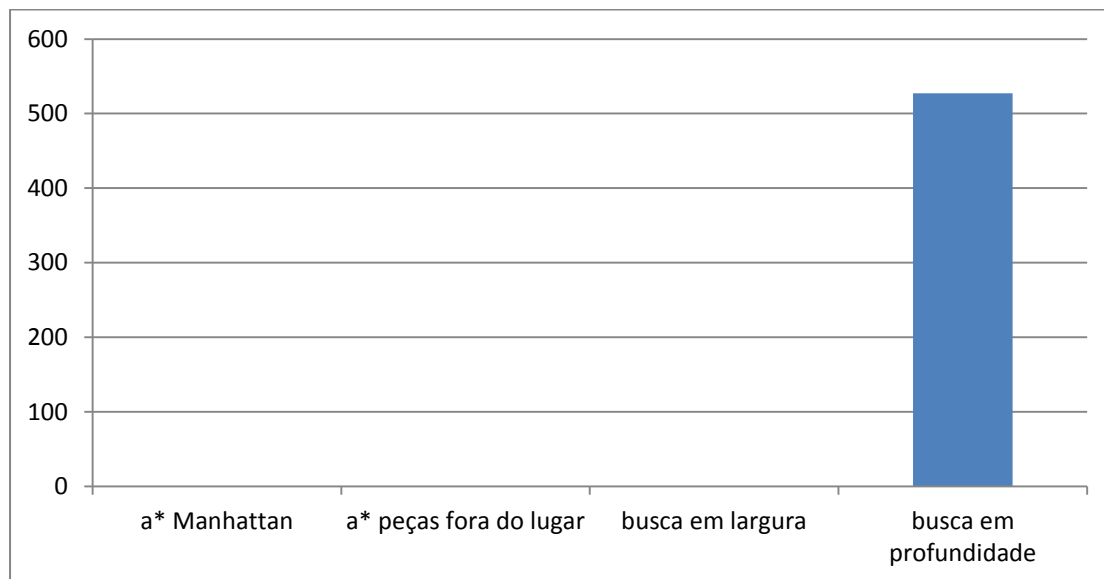


Tabela:

Estado: 123456_78	Número de Ações	Numero de estados expandidos	Tempo em segundos
A* Manhattan	2	2	0,3
A* peças fora do lugar	2	2	0,3
Busca em largura	2	3	0,37
Busca em profundidade	106330	115907	527

Soluções para o estado: "87654321_"

Gráfico com o número de estados expandidos.

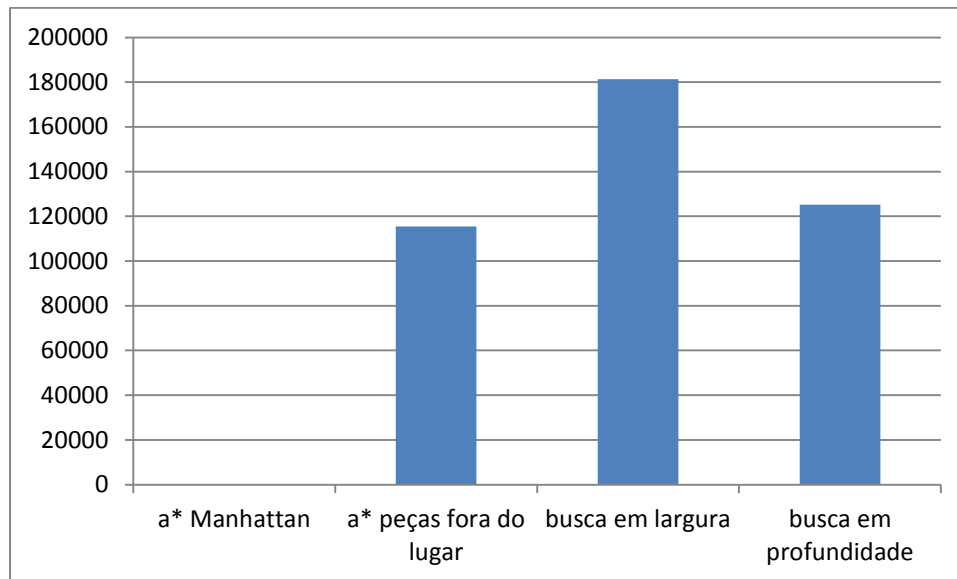


Gráfico com o tempo em segundos de cada algoritmo.

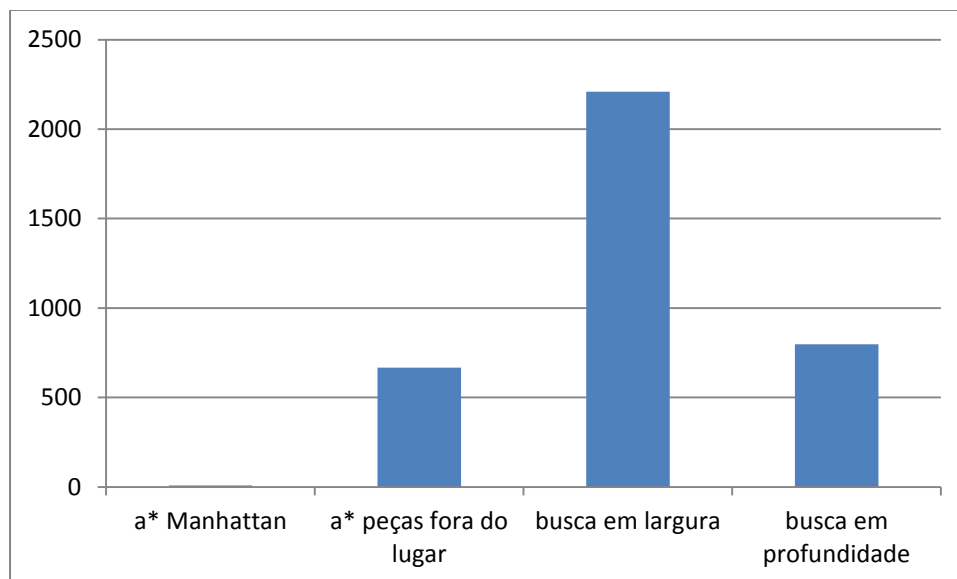


Tabela:

Estado: 87654321_	Número de ações	Numero de estados expandidos	Tempo em segundos
A* Manhattan	30	2	8,374
A* peças fora do lugar	30	115447	666,162
Busca em largura	30	181384	2209,553
Busca em profundidade	111890	125188	796,854

Soluções para o estado: “_12345678”

Gráfico com o número de estados expandidos.

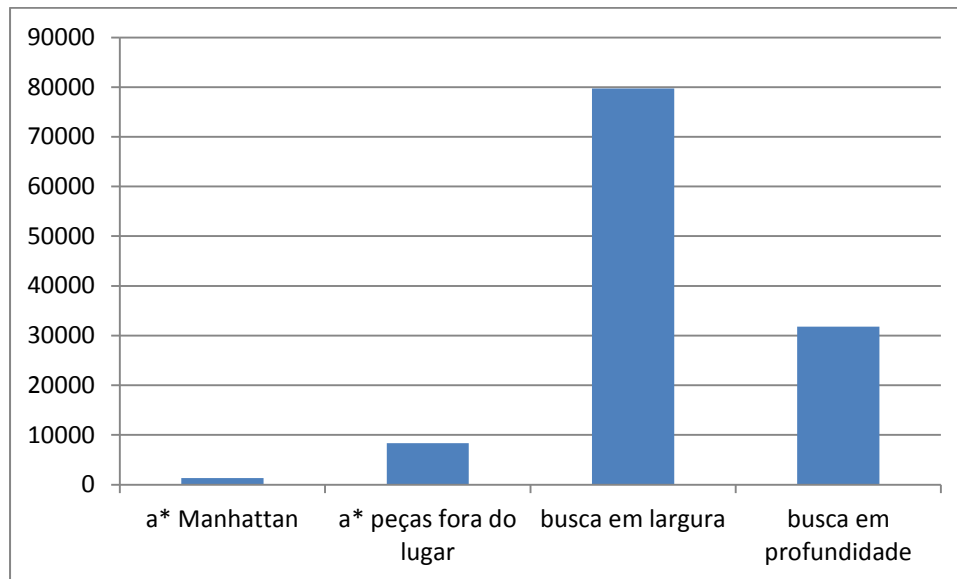


Gráfico com o tempo em segundos de cada algoritmo.

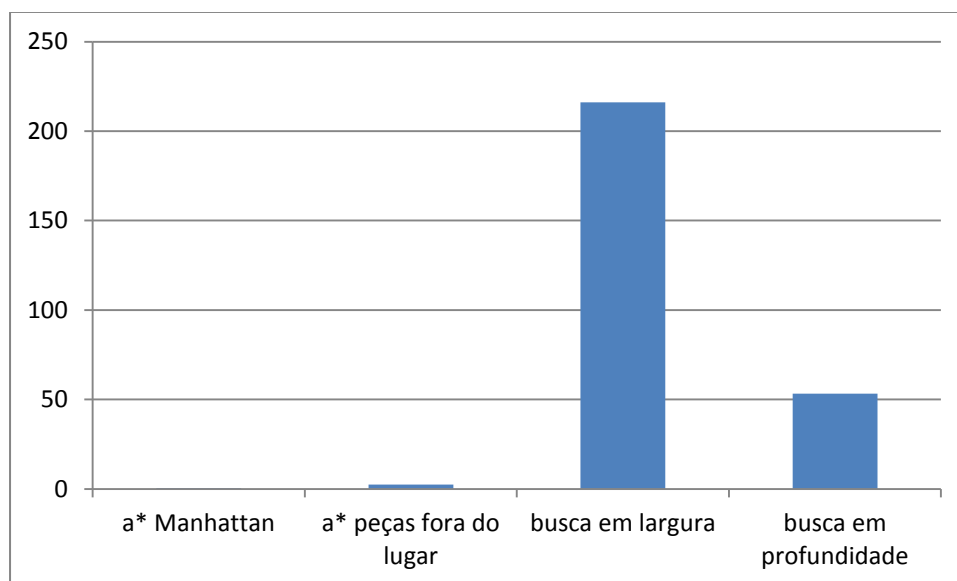


Tabela:

Estado: _12345678	Número de ações	Numero de estados expandidos	Tempo em segundos
A* Manhattan	22	1347	0,477
A* peças fora do lugar	22	8328	2,5
Busca em largura	22	79711	216,12
Busca em profundidade	30988	31789	53,17

Soluções para o estado: "2_3541687"

Gráfico com o número de estados expandidos.

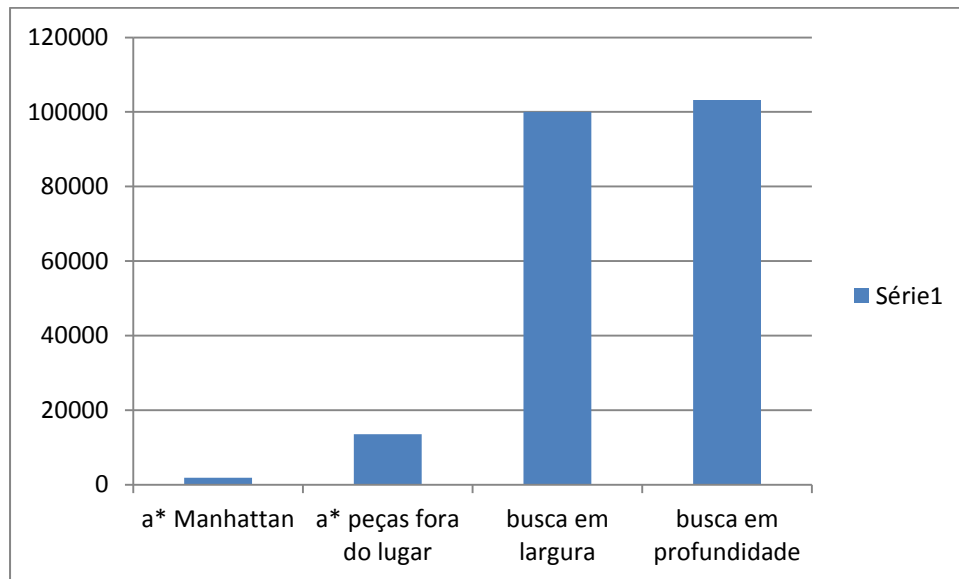


Gráfico com o tempo em segundos de cada algoritmo.

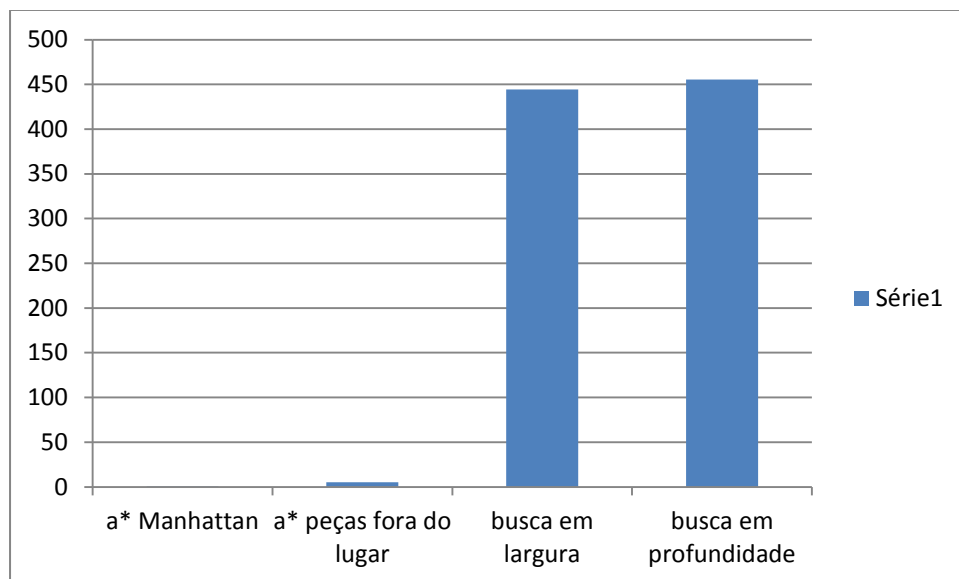


Tabela:

Estado: 87654321_	Número de ações	Numero de estados expandidos	Tempo em segundos
A* Manhattan	23	1830	0,521
A* peças fora do lugar	23	13506	5,249
Busca em largura	23	100002	444,331
Busca em profundidade	96453	103195	455

Custo de solução:

A*:

O custo de solução para o A* é mais satisfatório que os algoritmos de busca em profundidade e busca em largura. O custo continua ser $O(N!)$, pois se caso não houver resposta a fronteira irá visitar todos nós alcançáveis. Apesar disso o tempo de execução para o A* é muito menor que os algoritmos de busca em largura e o de busca em profundidade.

A grande vantagem do A* é que a heurística implementada fará que o algoritmo sempre caminhe para a resposta certa, eliminando vários estados que possuem menos possibilidade de levar à resposta.

O número de nós expandidos no A*, tanto usando a heurística Distância de Manhattan como de Peças fora do lugar, é pequeno comparado aos outros algoritmos.

A diferença entre as heurísticas foram maiores quando usado em um caso mais difícil: "87654321_". Onde a diferença entre o tempo de execução chegou a ser mais do que minutos, isso inclui também para o números de nós expandidos. A distancia de Manhattan é mais eficiente para estados mais custosos, mas para casos mais simples a diferença entre as heurísticas é pequena.

Busca em largura:

O número de nós expandidos é sempre alto para estados que requer um número de movimentos alto, pois o algoritmo irá visitar todos os nós de custos menores. Para estados que necessitam de pouquíssimas jogadas o algoritmo é rápido.

Em geral o tempo de execução é alto.

Busca em profundidade:

É o algoritmo que precisa de mais tempo, mesmo quando o resultado precisa de apenas dois movimentos. A complexidade é $O(N!)$.

Em todos os testes o número de estados expandidos é superiormente maior quando comparado aos outros algoritmos. Além disso a resposta não foi ótima para nenhum caso testado.

Detalhes de implementação

Todos os programas foram implementados em C++. Para a pilha foi usada a biblioteca Stack, e para o heap foi usado a biblioteca Set. No caso da fila foi usado o heap, como o custo de cada aresta é um para o caso da busca em largura então o heap serviu como uma fila, pois todos os nós de custos menores serão visitados antes dos custos maiores.

Para a comparação dos valores no heap foi usado sobrecarga de operador assim o struct nodo será comprado pelo custo.

Os testes foram realizados em uma maquina com as seguintes características.

- CPU intel i3 3.40 GHz
- 8 GB RAM
- Sistema operacional Windows 7 64bits

Conclusão

O algoritmo A* em ambas as heurísticas são admissíveis.