1. **Introdução:**

O algoritmo A\* é uma das maneiras mais conhecidas de busca pela melhor escolha. Cada novo nodo *n*  é avaliado de acordo com o custo para alcançar o mesmo a partir da raiz da árvore de busca (g(n)) e e uma heurística que estima o custo para ir de *n* até o nodo objetivo (h(n)). Ou seja, sendo f(n) a função que avalia n, procuramos o mínimo de:

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

Uma das propriedades que A\* possui é que ele garante encontrar sempre a melhor solução. Sendo assim este relatório trata da aplicação do algoritmo A\* juntamente com uma heurística, aplicado sobre o problema das 15 peças, ou 15 *puzzle.*

O problema é, dado um estado inicial na configuração de um tabuleiro com 15 peças, encontrar o menor número de movimentos que levam até o estado final, no nosso caso mostrado pela figura 1.

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

Figura 1: estado final do *15 puzzle*.

O hardware utilizado para fazer os testes foi um notebook com 6 GB de RAM, processador core i5-2430M 2.4 GHz e sistema operacional Windows 8.1, além disso a linguagem utilizada na implementação foi Java através da IDE Netbeans.

1. **Algoritmo A\***

O funcionamento do algoritmo aplicado ao problema das 15 peças é dado da seguinte forma, descobre novos nodos a serem explorados a cada nodo analisado. Por exemplo, ao expandir um nodo que representa uma configuração *c* não estado final das 15 peças, são criados de dois a quatro novos nodos, cada um representando um possível movimento a serem feitos para alterar *c*. Dessa forma o algoritmo define uma lista de nodos não explorados e outra com os já explorados.

1. **Heurísticas utilizadas:**

Foram implementados 5 heurísticas são elas *h’1*, *h’2*, *h’3*, *h’4*, *h’5* abaixo é descrito o funcionamento de cada uma:

*h’1(n)*: Essa heurística verifica o número de peças foras de seu lugar no tabuleiro para o estado final.

*h’2(n):* número de peças fora de ordem na sequência numérica das 15 peças, seguindo a ordem de posições do tabuleiro.

*h’3(n):* somar a quantidade de deslocamento para cada peça fora do lugar, para isso é considerado que o caminho esteja livre para que se possa realizar o menor número de movimentos.

*h’4(n):* é aplicado um peso pra cada heurística tal que a soma desses pesos *c1, c2* e *c3* seja igual a 1.

*h’5(n):* verifica quais das heurísticas anteriores *h’1, h’2, h’3* vai ser utilizado a partir da maximização.

1. **Comparação de desempenho:**

Os critérios que serão avaliados serão quatro:

1. Quantidade de árvores geradas pela heurística;
2. Quantidade de árvores processadas;
3. Quantidade de movimentos necessários para atingir o estado final;
4. Tempo gasto no processamento do caso.

Foram utilizados cinco casos como bases para verificar o desempenho. Cada caso é representado abaixo:

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

Figura 3: caso base 1 do *15puzzle.*

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 5 | 0 | 11 |
| 2 | **9** | **13** | **5** |
| 3 | **7** | **14** | **10** |
| 4 | **8** | **12** | **15** |

Figura 4: caso base 2 do *15 puzzle*.

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

Figura 5: caso base 3 do *15 puzzle*.

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

Figura 6: caso base 4 do *15 puzzle*.

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

Figura 7: caso base 5 do *15 puzzle*.

Primeiramente será feito uma comparação dos casos a partir de uma mesma heurística, em seguida a comparação será entre as heurísticas a partir dos critérios de avaliação já pré-estabelecidos anteriormente.

* 1. **Comparação dos casos na mesma heurística:**

Como já havia menciona nessas próximas seções iremos compara os dados capturados pelos casos bases somente de uma heurística de cada vez.

* 1. **Comparação dos casos base aplicando *h’1(n):***

Tabela 1: Árvores geradas em cada caso pela *h’1(n).*

Tabela 2: Árvores processadas em cada caso pela *h’1(n).*

Tabela 3: Quantidade de movimentos em cada caso pela *h’1(n).*

Tabela 4: Tempo de processamento em cada caso pela *h’1(n).*

Obs: O caso 3, caso 4, caso 5 não foi possível de serem gerados devido ao fato de consumirem muita memória.

* 1. **Comparação dos casos base aplicando *h’2(n):***

Tabela 5: Quantidade de árvores geradas em cada caso pela *h’2(n).*

Tabela 6: Quantidade de Árvores Processadas em cada caso pela *h’2(n).*

Tabela 7: Quantidade de Movimentos em cada caso pela *h’2(n).*

Tabela 8: Tempo de processamento em cada caso pela *h’2(n).*

* 1. **Comparação dos casos base aplicando *h’3(n):***

Tabela 9: Árvores geradas em cada caso pela *h’3(n).*

Tabela 10: Árvores Processadas em cada caso pela *h’3(n).*

Tabela 11: Quantidade de movimentos em cada caso pela *h’3(n).*

Tabela 12: Tempo de Processamento em *milissegundos* em cada caso pela *h’3(n).*

* 1. **Comparação dos casos bases aplicando a *h’4(n):***
     1. **Comparação com valores das constantes médio:**

Os gráficos a seguir apresentam uma comparação dos dados bases aplicando um valor médio para as constantes, portanto utilizamos os seguintes valores: C1 = 0,33; C2 = 0,33; C3 = 0,34.

Tabela 13: Quantidade de árvores geradas em cada caso com constantes iguais pela *h’4(n).*

Tabela 14: Quantidade de árvores geradas em cada caso com constantes iguais pela *h’4(n).*

Tabela 15: Quantidade de movimentos em cada caso com constantes iguais pela *h’4(n).*

Tabela 16: Tempo de processamento em cada caso com constantes iguais pela *h’4(n).*

* + 1. **Comparação com valor de constante C1 maior:**

Para os casos a seguir utilizamos os seguintes valores para as constantes: C1 = 0,6; C2 = 0,2; C3 = 0,2.

Tabela 17: Quantidade de árvores geradas em cada caso com constante C1 maior pela *h’4(n).*

Tabela 18: Quantidade de árvores processadas em cada caso com constante C1 maior pela *h’4(n).*

Tabela 19: Quantidade de movimentos em cada caso com constante

C1 maior pela *h’4(n).*

Tabela 19: Tempo de processamento em cada caso com constante

C1 maior pela *h’4(n).*

* + 1. **Comparação com valor de constante C2 maior:**

Para os casos a seguir utilizamos os seguintes valores para as constantes: C1 = 0,2; C2 = 0,6; C3 = 0,2.

Tabela 20: Quantidade de árvores geradas em cada caso com constante

C2 maior pela *h’4(n).*

Tabela 21: Quantidade de árvores processadas em cada caso com constante

C2 maior pela *h’4(n).*

Tabela 22: Quantidade de movimentos em cada caso com constante

C2 maior pela *h’4(n).*

Tabela 23: Tempo de processamento em cada caso com constante

C2 maior pela *h’4(n).*

* + 1. **Comparação com valor constante C3 maior:**

Para os casos a seguir utilizamos os seguintes valores para as constantes: C1 = 0,2; C2 = 0,6; C3 = 0,2.

Tabela 24: Tempo de processamento em cada caso com constante

C3 maior pela *h’4(n).*

Tabela 25: Tempo de processamento em cada caso com constante

C3 maior pela *h’4(n).*

Tabela 26: Quantidade de movimentos em cada caso com constante

C3 maior pela *h’4(n).*

Tabela 27: Tempo de Processamento em cada caso com constante

C3 maior pela *h’4(n).*

* 1. **Comparação dos casos bases aplicando a h’5(n):**

Tabela 27: Quantidade de árvores geradas em cada caso pela h’5(n).

Tabela 28: Quantidade de árvores processadas em cada caso pela h’5(n).

Tabela 28: Quantidade de movimentos em cada caso pela h’5(n).

Tabela 29: Tempo de processamento em cada caso pela h’5(n).

1. **Comparação dos casos bases entre as heurísticas existentes:**
   1. **Parâmetro de comparação árvores geradas:**

Os gráficos abaixo utilizam os dados para heurística 4 é com a constante C3 maior.

Tabela 28: Quantidade de Árvores Geradas por cada heurística.

* 1. **Parâmetro de comparação árvores processadas:**

Tabela 29: Comparação quantidade de árvores processadas por cada heurística.

* 1. **Parâmetro de comparação quantidade de movimentos:**

Este parâmetro percebemos através dos gráficos anteriores que ele se mantém independente da heurística que é aplicada, portanto não há necessidade de um novo gráfico para fazer a sua comparação.

* 1. **Parâmetro de comparação Tempo de processamento:**

Tabela 30: Comparação Tempo de processamento por cada heurística.

1. **Conclusão:**

As heurísticas diferenciam na quantidade de árvores geradas, árvores processadas e o tempo gasto no processamento já que a quantidade de movimentos é igual para todas até que consiga alcançar o estado final. Dessa forma podemos notar que a heurística h’3(n) se demostrou ser a melhor escolha dentre as cincos apresentados, pois, foi a que gerou menos árvores consequentemente teve menos árvores para serem processadas e menor tempo de processamento. A heurística h’5(n) também obteve bons resultados.

Já as heurísticas h’1(n), h’2(n) não foram capazes de chegarem ao resultado final para o caso 3, caso 4 e caso 5 utilizando os recursos de hardware disponível, isso demonstra que, essas duas heurísticas geram uma quantidade maior de árvores do que as demais.

1. **Referência bibliográfica:**

Araujo. R, *et al,* A\* aplicado ao problema das 15 peças, Belo Horizonte – MG.