

---

# MÉTODOS DE BUSCA

---

INE5430 - INTELIGÊNCIA ARTIFICIAL

**Lukas Derner Grüdtner**

Departamento de Informática e Estatística - INE  
Universidade Federal de Santa Catarina - UFSC  
Florianópolis - Santa Catarina - Brasil  
lukas.grudtner@grad.ufsc.br

22 DE MAIO DE 2019

## 1 GUIA DE UTILIZAÇÃO

### 1.1 GERANDO MINAS

Para gerar minas de tamanho qualquer, basta digitar o seguinte comando:

```
$ make generate <nome do arquivo> <tamanho da mina>
```

Exemplo:

```
$ make generate test 20
```

Uma mina de tamanho 20 irá ser gerada e armazenada no diretório **input/generated**.

*Não existe a necessidade da pré-geração de minas.  
Qualquer mina devidamente estruturada deverá ser lida pelo programa.*

### 1.2 COMPILANDO

Para iniciar o processo de compilação, basta digitar o seguinte comando:

```
$ make
```

### 1.3 EXECUTANDO

Para executar o programa, basta digitar o seguinte comando:

```
$ make run
```

### 1.4 ARQUIVOS DE ENTRADA

Na inicialização do programa, será necessário informar o caminho do arquivo de entrada. A seguir temos um exemplo de entrada:

```
input/generated/test1.mine
```

**Observe que o caminho é relativo à raiz do projeto.**

Existem cinco arquivos contendo minas para teste no diretório **input/generated**, os quais foram utilizados nos experimentos e que serão analisados mais adiante.

Logo após, será necessário informar os parâmetros que serão utilizados nos métodos de busca, sendo eles:

Tabela 1: Parâmetros dos métodos de busca

BUSCA	PARÂMETRO
<i>DFS limitada</i>	nível máximo de profundidade
<i>DFS iterativa</i>	número máximo de iterações
<i>A* com heurística</i>	-
<i>DFS iterativa com heurística</i>	número máximo de iterações

## 2 RESULTADOS

A seguir mostramos os dados obtidos em alguns experimentos. Todos os quatro métodos de busca foram testados utilizando minas de tamanho igual a 4, 5, 6, 7 e 8.

Tabela 2: Mina de tamanho 4

	PONTUAÇÃO	TEMPO DE EXECUÇÃO	ESTADOS EXAMINADOS
<i>DFS limitada</i>	41	0.9 ms	21
<i>DFS iterativa</i>	43	1.4 ms	62
<i>A* com heurística</i>	45	3.5 ms	39
<i>DFS iterativa com heurística</i>	42	2.0 ms	71

Tabela 3: Mina de tamanho 5

	PONTUAÇÃO	TEMPO DE EXECUÇÃO	ESTADOS EXAMINADOS
<i>DFS limitada</i>	309	6.7 ms	135
<i>DFS iterativa</i>	744	0.5 ms	28
<i>A* com heurística</i>	841	0.8 ms	15
<i>DFS iterativa com heurística</i>	744	0.3 ms	28

Tabela 4: Mina de tamanho 6

	PONTUAÇÃO	TEMPO DE EXECUÇÃO	ESTADOS EXAMINADOS
<i>DFS limitada</i>	1568	8.21 ms	366
<i>DFS iterativa</i>	3268	13.6 ms	1372
<i>A* com heurística</i>	3594	03.55 s	87643
<i>DFS iterativa com heurística</i>	3605	58.4 ms	3929

Tabela 5: Mina de tamanho 7

	PONTUAÇÃO	TEMPO DE EXECUÇÃO	ESTADOS EXAMINADOS
<i>DFS limitada</i>	19208	11.23 ms	21
<i>DFS iterativa</i>	93196	5.265 ms	62
<i>A* com heurística</i>	91821	117.9 ms	39
<i>DFS iterativa com heurística</i>	92560	9.147 ms	71

Tabela 6: Mina de tamanho 8

	PONTUAÇÃO	TEMPO DE EXECUÇÃO	ESTADOS EXAMINADOS
<i>DFS limitada</i>	289562	22.24 ms	726
<i>DFS iterativa</i>	994476	26.76 ms	2217
<i>A* com heurística</i>	986924	54.384 s	1080294
<i>DFS iterativa com heurística</i>	988112	417.1 ms	13944

A função *score* foi definida da seguinte maneira:

$$score = gold * (battery + 1) + (size_{problem}^2 / n_{actions}) + (size_{problem}^2 / n_{exploredstates})$$

## 2.1 DISCUSSÃO

Como podemos observar pelos dados acima, o método de busca em profundidade limitada é sempre mais rápido e examina menos estados do que os outros métodos, porém obtém pontuação menor por pegar caminhos que, apesar de chegar ao destino, fazem um percurso maior do que o esperado.

O método de busca em profundidade iterativa é um pouco mais lento que o anterior, além de examinar mais estados, mas a pontuação obtida faz compensar seu maior esforço, apresentando caminhos menores até o objetivo.

O método  $A^*$  é conhecido por retornar uma solução ótima. Como ele examina muito mais estados que os outros métodos, ele tende a ter um desempenho pior. Porém, a solução ótima muitas vezes compensa.

Por fim, o método de busca em profundidade iterativa com heurística é, em geral, melhor do que a sua versão sem o uso de heurística, porém acaba se tornando mais lento e examinando mais estados. Isto depende muito da eficácia da heurística utilizada em relação ao problema.