

Problema 8-Puzzle: Análise da solução usando Backtracking e Algoritmos Genéticos

Nelson Florêncio Junior

Orientador: Frederico Gadelha Guimarães

Departamento de Computação
UFOP

18 de agosto de 2011



Universidade Federal
de Ouro Preto

- 1 Introdução
 - 8-Puzzle
 - Busca de Espaço de Estados
 - Backtracking
 - Algoritmos Genéticos
 - Algoritmos Genéticos
 - Objetivos
- 2 Trabalhos Relacionados
- 3 Metodologia
 - Método
 - Backtracking
 - Pseudo-código
 - Representação do Problema
 - Avaliação e Seleção
 - Operadores Genéticos
 - Análise de Complexidade
- 4 Testes
- 5 Conclusões



8-Puzzle

1	2	3
8	0	4
7	6	5

Estado Inicial

0	1	2
3	4	5
6	7	8

Estado Final



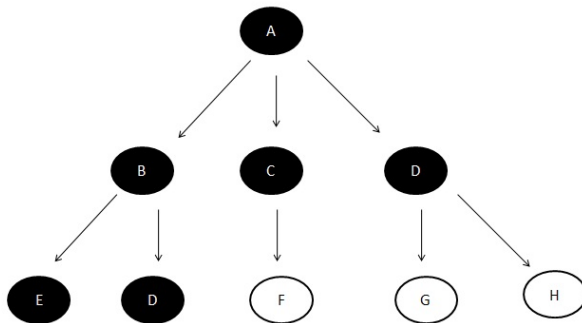
Busca de Espaço de Estados

- Estado Inicial
- Estado Final ou Objetivo
- Operadores
- Custo do caminho



Backtracking

- Tentativa e erro: decompor o processo em um número finito de sub-tarefas parciais que devem ser exploradas exaustivamente.
- É uma solução clássica para problemas de busca.



Algoritmos Genéticos

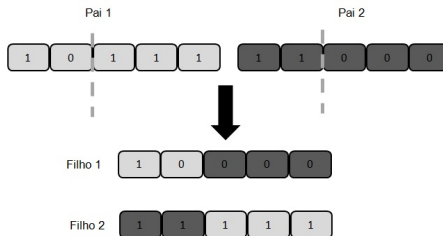
Para caracterizar os Algoritmos Genéticos deve-se levar em consideração:

- Representação das soluções do problema.
- Inicialização da População.
- Avaliação.
- Seleção.
- Operadores Genéticos.

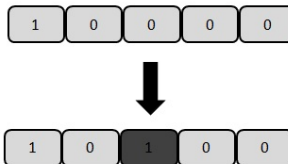


Operadores Genéticos

- *Crossover.*



- *Mutação.*



Objetivos

- Desenvolver dois algoritmos que resolvam o problema do 8-Puzzle, um baseado no paradigma de Backtracking e outro em Computação Evolutiva, especificamente Algoritmos Genéticos.
- Análise dos algoritmos propostos.



- 1 Introdução
 - 8-Puzzle
 - Busca de Espaço de Estados
 - Backtracking
 - Algoritmos Genéticos
 - Algoritmos Genéticos
 - Objetivos

- 2 **Trabalhos Relacionados**

- 3 Metodologia
 - Método
 - Backtracking
 - Pseudo-código
 - Representação do Problema
 - Avaliação e Seleção
 - Operadores Genéticos
 - Análise de Complexidade

- 4 Testes

- 5 Conclusões



Alex Fernandes da Veiga Machado et al.

TUTORIAL SBGAMES 2009: Criação de um Jogo de Quebra-Cabeças Usando Algoritmo Genético como Motor de Resolução.



Position 1	Position 2	Code	Position 1	Position 2	Code
1	2	1	1	4	10
2	3	2	4	7	11
3	1	3	7	1	12
4	5	4	2	5	13
5	6	5	5	8	14
6	4	6	8	2	15
7	8	7	3	6	16
8	9	8	6	9	17
9	7	9	9	3	18
0 means no move					

Ting Qian: State University of New York at Oswego

Using Genetic Algorithm to Solve Sliding Tile Puzzles

- Problemas na representação da população.
- Obteve resultados interessantes.



- 1 Introdução
 - 8-Puzzle
 - Busca de Espaço de Estados
 - Backtracking
 - Algoritmos Genéticos
 - Algoritmos Genéticos
 - Objetivos
- 2 Trabalhos Relacionados
- 3 Metodologia
 - Método
 - Backtracking
 - Pseudo-código
 - Representação do Problema
 - Avaliação e Seleção
 - Operadores Genéticos
 - Análise de Complexidade
- 4 Testes
- 5 Conclusões



Backtracking

- Foi implementado uma busca em profundidade recursiva.
- Baseada nos conceitos de busca espaço de estados.

BUSCAEMPROFUNDIDADE(*EstadoAtual*)

```
1  if EstadoAtual = Objetivo
2      return SUCESSO
3  Fechado[] := EstadoAtual
4  while Filhos[EstadoAtual] <> 0
5      Filho := ProximoFilho
6      if Filho <> Fechado[]
7          BuscaEmProfundidade(Filho)
8      return SUCESSO
9  return FALHA
```



Pseudo-código

ALGORITMOGENETICO()

```
1   $t = 0$ 
2  criar população  $P(t)$ ;
3  for cada indivíduo  $i$  de  $P(t)$ 
4      avaliar aptidão  $individuo(i)$ 
5  Fim-para
6  while condição de parada não satisfeita
7       $t := t + 1$ ;
8      Selecionar população  $P(t)$  de  $P(t - 1)$ ;
9      Aplicar operadores de cruzamento sobre  $P(t)$ ;
10     Aplicar operadores de mutação sobre  $P(t)$ ;
11     Avaliar  $P(t)$ ;
12 Fim-enquanto
```



Representação do Problema

- Cromossomos foram representados por uma cadeia de genes com valores aleatórios de (0 – 3).
- 0 - Cima, 1 - Baixo, 2 - Esquerda, 3 - Direita.

```

Pressione qualquer tecla para continuar. . .

----- algoritmos geneticos -----
  0      1      2      3      4      5      6      7      8
0      3      2      3      2      2      1      0      3
1      1      1      0      1      0      3      2      0
2      2      3      3      1      2      2      3      2
3      3      2      1      3      0      2      1      0
4      2      3      1      0      1      0      3      1
5      1      3      1      2      2      3      2      0
6      2      3      3      2      1      2      0      1
7      3      2      1      1      3      2      0      1
8      3      1      0      1      2      2      0      0
9      3      2      1      1      3      3      0      0
10     3      1      1      0      1      0      1      1
11     3      1      2      0      2      3      1      0
12     3      2      1      1      2      0      0      1
13     2      1      1      3      0      2      3      3
14     1      2      0      3      1      0      2      1
15     3      1      0      2      3      2      3      3
16     3      2      3      1      1      0      2      0
17     3      2      1      1      0      0      3      3
18     2      3      2      1      0      3      1      2
19     2      1      0      3      3      2      1      0
  
```



Avaliação e Seleção

- Função de avaliação:

$$f(n) = 36 * NumPecasTrocadas + 18 * DistanciaManhattan + 2 * NumInversoes$$

- Seleção: Método de Elitismo



Operadores Genéticos

- *Crossover*: Taxa de 30%
- *Mutação*: Taxa de 60%



Análise de Complexidade: Backtracking

- Espaço: $O(B * m)$
- Tempo: $O(B^m)$

Onde B é a profundidade máxima e m o número médio de filhos de cada nó.



Análise de Complexidade: Algoritmos Genéticos

Numa forma abstrata, definida por (Aguiar, 1998), podemos considerar uma função geral da complexidade como:

$$f(n) = O(Comp_{Inic} + [num_{Ger} * (Comp_{Reprod} + Comp_{Term})] + Comp_{Recupera})$$

Onde:

- num_{Ger} : Número de Gerações.
- $Comp_{Inic}$: Complexidade criar população inicial.
- $Comp_{Reprod}$: Complexidade dos Operadores e Método Seleção.
- $Comp_{Term}$: Complexidade de verificação se o objetivo foi alcançado e/ou condição de parada.
- $Comp_{Recupera}$: Complexidade de retornar os dados referentes a resolução ou não do problema.



Análise de Complexidade: Complexidade dos Operadores e Método Seleção

Complexidade dos Operadores e Método Seleção, merece um destaque especial, pois influencia todo o algoritmo dominando as outras complexidades, ela é expressa da seguinte forma:

$$Comp_{Reprod} = L * \sum (1/k * (Comp_{Escolher_{Individuo}} + Comp_{mod}))$$

Onde:

- L : número de vezes que a geração é executada.
- k : número de operadores genéticos existentes e/ou utilizados no problema.
- $Comp_{Escolher_{Individuo}}$: complexidade de escolher um indivíduo.
- $Comp_{mod}$: complexidade dos procedimentos de mutação e/ou crossover.



Complexidade do Algoritmo

- Complexidade do Algoritmo desenvolvido:

$$f = num_{ger} * O(n^2)$$

- num_{Ger} : Número de Gerações.
- n : número de indivíduos.



- 1 Introdução
 - 8-Puzzle
 - Busca de Espaço de Estados
 - Backtracking
 - Algoritmos Genéticos
 - Algoritmos Genéticos
 - Objetivos
- 2 Trabalhos Relacionados
- 3 Metodologia
 - Método
 - Backtracking
 - Pseudo-código
 - Representação do Problema
 - Avaliação e Seleção
 - Operadores Genéticos
 - Análise de Complexidade
- 4 Testes
- 5 Conclusões



Resultados

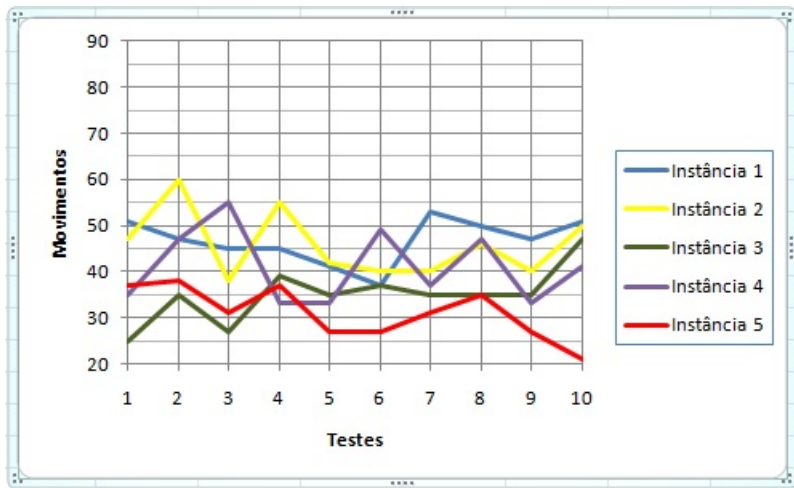
Foram realizados testes com 5 instâncias, devido a característica estocástica dos Algoritmos Genéticos cada instância foi testada 10 vezes, gerando uma média. Considere:

- Indivíduos: 1000
- Gerações: 4000
- Genes (tamanho dos indivíduos): 90

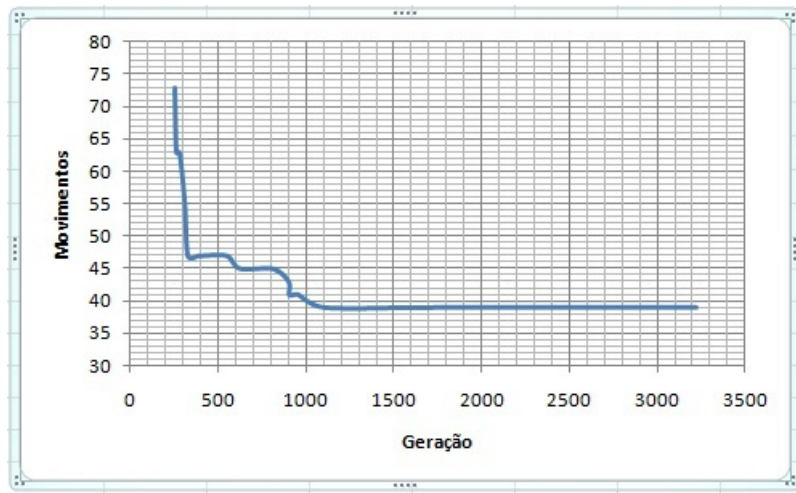
	Número de Movimentos		
Instância	<i>Backtracking</i>	Média(AG)	Melhor Resultado(AG)
1	94	46,7	37
2	137	45,8	38
3	84	35	25
4	300	41	33
5	316	31,1	21



Variação dos Resultados com Algoritmos Genéticos



Comportamento Evolutivo



- 1 Introdução
 - 8-Puzzle
 - Busca de Espaço de Estados
 - Backtracking
 - Algoritmos Genéticos
 - Algoritmos Genéticos
 - Objetivos
- 2 Trabalhos Relacionados
- 3 Metodologia
 - Método
 - Backtracking
 - Pseudo-código
 - Representação do Problema
 - Avaliação e Seleção
 - Operadores Genéticos
 - Análise de Complexidade
- 4 Testes
- 5 **Conclusões**



Conclusão

- Os Algoritmos Genéticos demonstraram um ótimo comportamento evolutivo.
- Os testes demonstraram como os Algoritmos Genéticos propiciam melhores resultados.
- Quanto maior o espaço de busca, menor a capacidade do *Backtracking* encontrar uma boa resposta ou mesmo encontrar uma resposta. Daí a necessidade de se conhecer outros métodos.
- A estrutura do algoritmo pode ser alterada para se obter resultados ainda melhores.

