

Trabalho II

Algoritmo Genético aplicado para N-Rainhas

Alisson Jaques
Engenharia da Computação
Universidade do Estado de
Minas Gerais-UEMG
Divinópolis, MG
alissonjaquesrmq@gmail.com

Guilherme Gomes Noronha
Engenharia da Computação
Universidade do Estado de
Minas Gerais-UEMG
Divinópolis, MG
ggn.noronha@gmail.com

Abstract—Application of Genetic Algorithm to solve the n-queens problem.

Palavras-chaves—algoritmo-genetico, n-rainhas, N-Queens, mutação, crossover

1. INTRODUÇÃO

Os algoritmos genéticos utilizam técnicas baseadas nos princípios da seleção natural e reprodução genética de Charles Darwin, sendo normalmente empregados em problemas complexos de otimização. Neste trabalho abordamos a utilização de algoritmos na resolução do problema das N-Rainhas.

O problema consiste em encontrar uma configuração para posicionamento das N rainhas em um tabuleiro N por N, de forma que nenhuma rainha possa atacar a outra. Tendo em vista que no xadrez, a rainha é a peça mais poderosa, movendo-se em todas direções, incluindo diagonais e com qualquer número de casas, a resolução torna-se não trivial assim que levamos nosso N a uma ordem de grandeza maior. A Figura 1 mostra um exemplo de uma possível configuração em um tabuleiro 8x8.

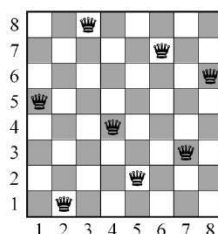


Figura 1: Uma resolução para o problema

Conforme descrito em [1], otimização é a busca da melhor solução para um dado problema. Consiste em tentar várias soluções e utilizar a informação obtida neste processo de forma a encontrar soluções cada vez melhores.

Não existe uma forma de prever o exato número de soluções para determinado N assim, aplicar um algoritmo se torna crucial, dado a complexidade do problema. A Figura 2 demonstra alguns números de soluções dado N.

n	Size of solution space (n!)	Number of solutions
1	1	1
2	2	0
3	6	0
4	24	2
5	120	10
6	720	4
7	5040	40
8	40320	92
9	362880	352
10	3628800	724
11	39916800	2680
12	479001600	14200
13	6227020800	73712
14	87178291200	365596
15	1307674368000	2279184
16	20922789888000	14772512
17	355687428096000	95815104
18	6402373705728000	666090624
19	121645100408832000	4968057848
20	2432902008176640000	39029188884
21	51090942171709440000	314666222712
22	112400072777607680000	2691008701644
23	25852016738884976640000	24233937684440
24	620448401733239439360000	227514171973736
25	1551121004330985984000000	2207893435808352
26	403291461126605635584000000	22317699616364044

Figura 2: Número de soluções dado N

Tendo como base a situação problema acima, serão feitas comparações entre os métodos de busca cega e busca heurística.

II. ALGORITMO GENÉTICO

Assim como apresentado em [3]

“Os AGs se baseiam principalmente na evolução das espécies como forma de alcançar a solução para um problema. A abordagem evolutiva é utilizada para compreender como a natureza competitiva em que se encontram ajuda a buscar a ser "o mais forte", que no caso da heurística não significa apenas força, mas sim um conjunto de informações que, quando processadas no meio em que se encontram, trazem melhores resultados que outras combinações. O processo de evolução é aleatório porém, guiado por um mecanismo de seleção baseado na adaptação de estruturas individuais.”

De forma introdutória, podemos descrever a formulação de um algoritmo genético como apresentado em [1]:

“O primeiro passo de um Algoritmo Genético típico é a geração de uma **população inicial de cromossomos**, que é formada por um conjunto aleatório de cromossomos que representam possíveis soluções do problema a ser resolvido. Durante o **processo evolutivo**, esta população é avaliada e cada cromossomo recebe uma nota (denominada de **aptidão** no jargão da literatura de AGs), refletindo a qualidade da solução que ele representa. Em geral, os cromossomos mais aptos são selecionados e os menos aptos são descartados (Darwinismo). Os **membros** selecionados podem sofrer modificações em suas características fundamentais através dos operadores de **crossover** e **mutação**, gerando descendentes para a próxima geração. Este processo é repetido até que uma solução satisfatória seja encontrada.”

Um pseudocódigo [1] é apresentado a seguir:

```
Seja  $S(t)$  a população de cromossomos na geração  $t$ .  
  
 $t \leftarrow 0$   
inicializar  $S(t)$   
avaliar  $S(t)$   
enquanto o critério de parada não for satisfeito faça  
     $t \leftarrow t+1$   
    selecionar  $S(t)$  a partir de  $S(t-1)$   
    aplicar crossover sobre  $S(t)$   
    aplicar mutação sobre  $S(t)$   
    avaliar  $S(t)$   
fim enquanto
```

Figura 3: pseudocódigo [1]

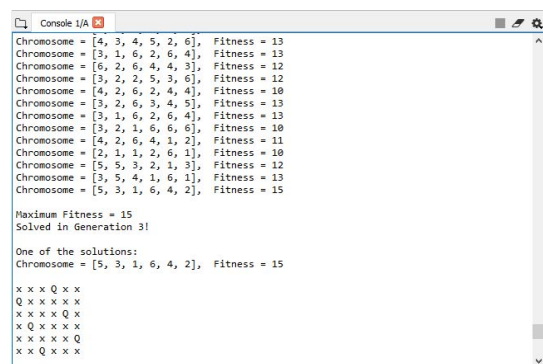
III. APLICANDO O ALGORITMO PARA SOLUCIONAR O PROBLEMA DAS N-RAINHAS

Foi feito um *fork* no repositório do GitHub intitulado “waqqasiq/n-queen-problem-using-genetic-algorithm”, tendo servido como base para a resolução geral do problema. O repositório pode ser conferido em:

<https://github.com/engenharia-da-computacao/n-queen-problem-using-genetic-algorithm>

O algoritmo foi desenvolvido em Python e segue basicamente os seguintes passos:

- Etapa 1: um cromossomo aleatório é gerado
- Etapa 2: O valor de adequação do cromossomo é calculado
- Etapa 3: Se o fitness não for igual a Fmax
- Passo 4: Reproduzir um novo cromossomo (cruzado) de dois melhores cromossomos selecionados aleatoriamente
- Passo 5: Mutação pode ocorrer
- Passo 6: Novo cromossomo adicionado à população
- Repita os passos 2 a 6 até que um cromossomo (solução) com valor de Fitness = Fmax seja encontrado



```
Console 1/A  
Chromosome = [4, 3, 4, 5, 2, 6], Fitness = 13  
Chromosome = [3, 1, 6, 2, 6, 4], Fitness = 13  
Chromosome = [6, 2, 6, 4, 4, 3], Fitness = 12  
Chromosome = [3, 2, 2, 5, 3, 6], Fitness = 12  
Chromosome = [4, 2, 6, 2, 4, 4], Fitness = 10  
Chromosome = [3, 2, 6, 3, 4, 5], Fitness = 13  
Chromosome = [3, 1, 6, 2, 6, 4], Fitness = 13  
Chromosome = [3, 2, 1, 6, 6, 6], Fitness = 10  
Chromosome = [4, 2, 6, 4, 1, 2], Fitness = 11  
Chromosome = [2, 1, 1, 2, 6, 1], Fitness = 10  
Chromosome = [5, 5, 3, 2, 1, 3], Fitness = 12  
Chromosome = [3, 5, 4, 1, 6, 1], Fitness = 13  
Chromosome = [5, 3, 1, 6, 4, 2], Fitness = 15  
  
Maximum Fitness = 15  
Solved in Generation 3!  
  
One of the solutions:  
Chromosome = [5, 3, 1, 6, 4, 2], Fitness = 15  
  
x x x Q x x  
Q x x x x  
x x x Q x  
x Q x x x  
x x x x Q  
x x Q x x
```

Figura 4: Solução do algoritmo para N = 6

REFERÊNCIAS

- [1] DE LACERDA, Estéfane GM; DE CARVALHO, ACPLF. Introdução aos algoritmos genéticos. Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais, v. 1, p. 99-148, 1999.
- [2] Genetic Algorithms: Solving the N-Queens problem Disponível em: <https://www.kaggle.com/mrknoott/genetic-algorithms-solving-the-n-queens-problem/data>. [Acesso: 15 Jun- 2019].