



## EEE882 – Computação Evolucionária

Prof. Frederico Gadelha Guimarães

### 1 O problema das N-Rainhas

Dado um tabuleiro de xadrez regular ( $N \times N$ ) e  $N$  rainhas, o problema das N-rainhas consiste em posicioná-las no tabuleiro de forma que nenhuma rainha seja capaz de capturar outra rainha. A Figura 1 ilustra a área de captura de uma rainha posicionada no tabuleiro.

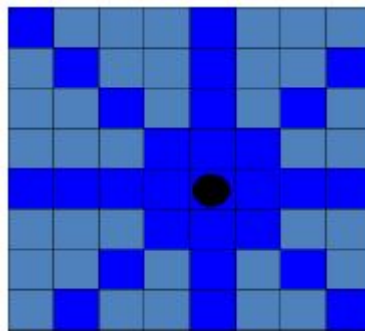


Figura 1: Área de captura de uma rainha considerando um tabuleiro de dimensões  $8 \times 8$ .

O problema pode ser formulado como um problema de Satisfação de Restrições, no qual o objetivo é encontrar uma configuração das rainhas sobre o tabuleiro que não viole nenhuma restrição.

Para resolver o problema das N-rainhas utilizando Algoritmos Genéticos (AG) deve-se, primeiramente, definir:

- uma representação (codificação) para os indivíduos (i.e. solução candidata);
- uma função de aptidão (objetivo) para avaliar a qualidade de uma solução candidata;
- um ou mais operadores de variação;
- um ou mais mecanismos de seleção;
- uma condição de término, por exemplo, número de gerações ou um valor de *fitness*.

Algumas sugestões a serem utilizadas na implementação são:

- **Representação:**
  - Fenótipo ( $f$ ): uma dada configuração do tabuleiro com  $N$  rainhas;
  - Genótipo ( $g$ ) ou cromossomo: permutação de inteiros  $1, 2, 3, \dots, N$ . Isto é,  $g = \{i_1, i_2, \dots, i_N\}$  denotando uma configuração onde a  $k$ -ésima coluna (posição do vetor) contém uma rainha posicionada na  $i_k$ -ésima linha, conforme ilustrado na Figura 2 e 3;

Para o tipo de representação adotado (permutação de  $N$  inteiros), a quantidade máxima de xeques que podem ocorrer é  $q(f) = N(N - 1)/2$  e a quantidade mínima é  $q(f) = 0$  - ver arquivo `fitness_nq.m`.

- **Operadores:**

- Cruzamento: cria duas novas soluções candidatas (filhos) a partir da combinação de dois cromossomos pais. Os filhos produzidos devem ser permutações válidas de  $N$  inteiros - ver arquivo exemplo `CutandCrossfill_CrossOver.m`.
- Mutação: seleciona de forma aleatória duas posições em um cromossomo (permutação) e troca os valores destas posições.

- **Seleção:**

- Seleção dos pais: escolha aleatoriamente dentre os indivíduos da população e os dois melhores são sugeridos para cruzamento.
- Seleção dos sobreviventes: ordene todos os indivíduos (população original mais 2 descendentes) e elimine os dois piores a cada geração.

- **Condição de parada:** finalize quando a solução ótima for encontrada, i.e.  $q(f) = 0$ , **ou** quando um número máximo de gerações tiver sido alcançado.

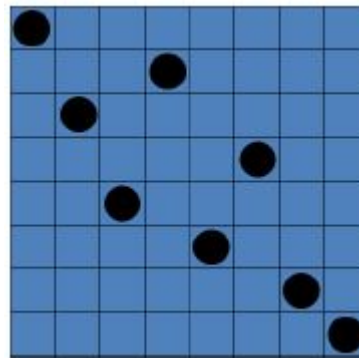


Figura 2: Fenótipo de uma solução candidata

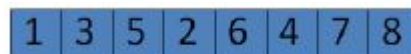


Figura 3: Genótipo correspondente ao fenótipo da Figura 2

Com relação a esse problema, deve-se implementar as tarefas a seguir.

### Tarefa 1

Desenvolva um AG para o problema das N-rainhas, considerando um tabuleiro de dimensão fixa de 8x8. Verifique com quantas gerações em média o seu algoritmo está alcançando a melhor solução. Ilustre com gráficos e/ou tabelas. Em sua implementação, você é livre para modificar ou acrescentar outras formas de seleção, cruzamento e mutação.

## 2 Função Rastrigin

A função Rastrigin é uma função não convexa utilizada para avaliar o desempenho de algoritmos de otimização. Ela foi proposta inicialmente por Rastrigin [1] como uma função bidimensional e posteriormente generalizada por Muhlenbein et al. [2]. O objetivo é desenvolver um AG e aplicá-lo na minimização da **função Rastrigin** para  $n=10$  variáveis.

$$\min_{\mathbf{x}} f(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (1)$$

com  $x_i \in [-5.12, 5.12]$ . O mínimo global ocorre em  $\mathbf{x}^* = 0$ , onde  $f(\mathbf{x}^*) = 0$ .

### Tarefa 1

Desenvolva um AG com as seguintes características:

- i. **Representação** das variáveis por **codificação binária**;
- ii. **Operador de seleção** dos indivíduos por meio de:
  - Método da **roleta** com 50% de probabilidade;
  - Método do **torneio** com 50% de probabilidade.
- iii. Operador de **cruzamento**: **1 ponto de corte por variável**;
- iv. Operador de **mutação**: **bit flip**.
- v. **Outros operadores** de livre escolha do aluno.

Defina os valores dos hiper-parâmetros  $p_c$  e  $p_m$  ou faça um ajuste dinâmico desses parâmetros. O orçamento computacional é de 10,000 avaliações de função.

### Observações importantes:

- Os parâmetros de entrada e saída do programa devem ser conforme modelo abaixo:  
 $[\mathbf{x}^*, f^*] = \text{nome\_sobrenome}[nvar, ncal]$ , sendo:  
 $\mathbf{x}^*$ : vetor das variáveis de decisão do melhor indivíduo;  
 $f^*$ : valor da função objetivo avaliada em  $\mathbf{x}^*$   
 $ncal$ : número total de chamadas da função de cálculo da *fitness* do problema;  
 $nvar$ : número de variáveis de decisão.
- O programa não deve imprimir nenhum gráfico ou resultado no prompt de comando durante o processo iterativo. O aluno que não respeitar esta premissa será penalizado em sua avaliação.

## 3 Entrega

**Entrega:** Enviar para o e-mail marcosalves@ufmg.br até **25/04/2019 as 23h59**.

Conteúdo do e-mail:

- Um arquivo do tipo **.m** ou **.py** com todo o código do programa em formato de função para o problema das N-rainhas e um arquivo do tipo **.m** ou **.py** com todo o código do programa em formato de função para o problema da função Rastrigin. O nome dos arquivos deve ser, necessariamente: nome\_sobrenome\_rainhas.m ou nome\_sobrenome\_rastrigin.m (por exemplo, frederico\_guimaraes\_rainhas.m e frederico\_guimaraes\_rastrigin.m).
- Um relatório em PDF no formato da IEEE com análise crítica dos resultados (máximo 5 páginas). Foque no que foi implementado e quais estratégias utilizou para obter os resultados.

O programa deve ser implementado ou em Matlab, em versão compatível com a R2015a, ou em Python, em versão compatível com a IDE PyCharm 2018.2.1.

## Referências

- [1] L. A. Rastrigin. Systems of extremal control. Nauka, 1974.
- [2] H. Mühlenbein, D. Schomisch and J. Born. "The Parallel Genetic Algorithm as Function Optimizer ". Parallel Computing, 17, 1991.
- [3] Agoston E. Eiben, and James E. Smith. Introduction to evolutionary computing. Vol. 53. Berlin: springer, 2003.