Laboratório I - Problema das N-Rainhas

Moises Mendes de Assis

Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica
ELE083 Computação Evolucionária
Engenharia de Sistemas

17 de abril de 2017

1 Introdução

Para este laboratório, propõe-se resolver o problema da N-rainhas, descrito da seguinte forma: "Dado um tabuleiro de xadrez regular (*NxN*) e N rainhas, posicione as N-Rainhas no tabuleiro de forma que elas não se coloquem em xeque".

A representação definida para realizar isso foi um vetor de inteiros *V* de tamanho N, em que cada posição *k* do vetor *V* representa a coluna *k* do tabuleiro. Isso é válido, pois duas rainhas não podem estar na mesma coluna, nem na mesma linha. As soluções geradas ao longo do processo são avaliadas de acordo com o número de xeques entre pares de rainhas, que é a função de aptidão. O objetivo é que o número de xeques seja zero. Caso seja encontrada uma solução que zera a função de aptidão, o algoritmo termina e retorna essa configuração, que é solução do problema.

2 Parte experimental

A primeira configuração de teste é:

- N = 8
- População = 30
- Número máximo de gerações = 100

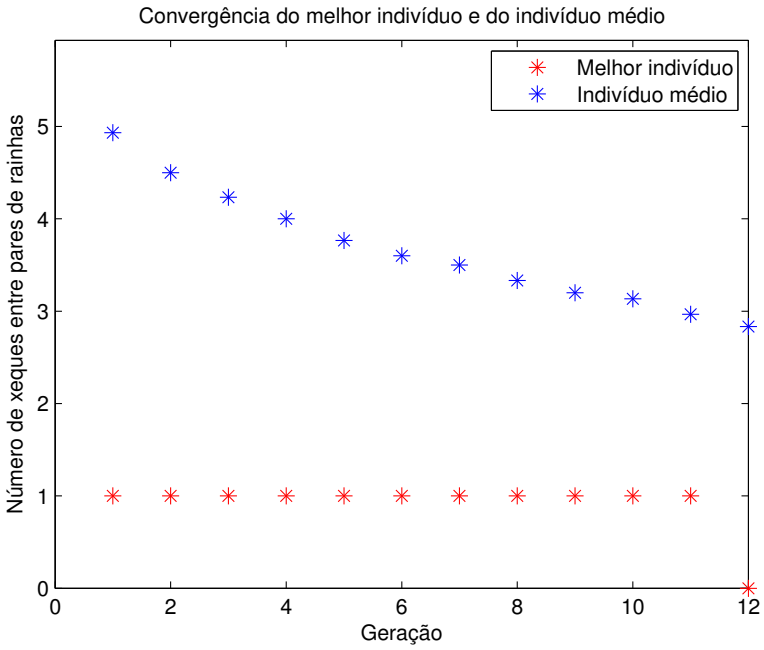

Figura 1: Resultado para a primeira configuração

A segunda configuração de teste é:

- N = 10
- População = 30
- Número máximo de gerações = 100

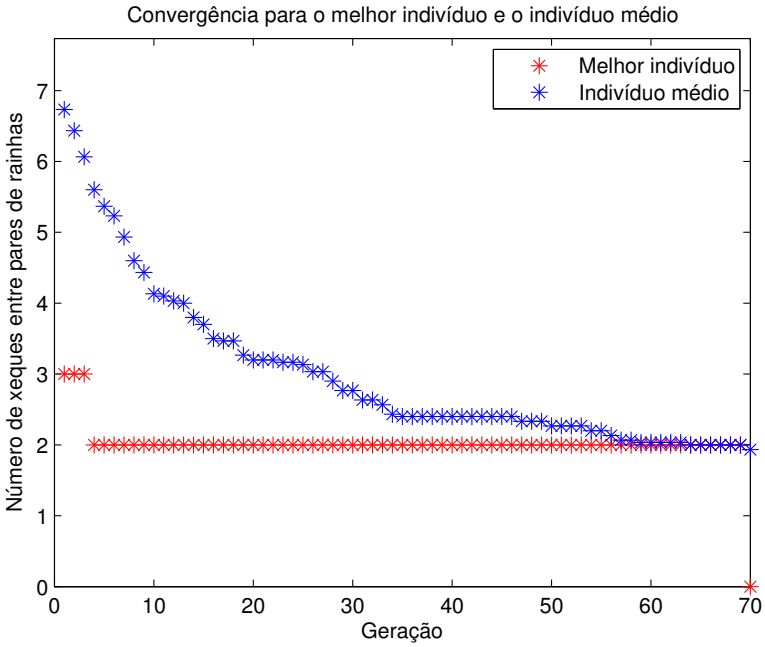

Figura 2: Resultado para a segunda configuração

A terceira configuração de teste é:

- N = 15
- População = 50
- Número máximo de gerações = 10000

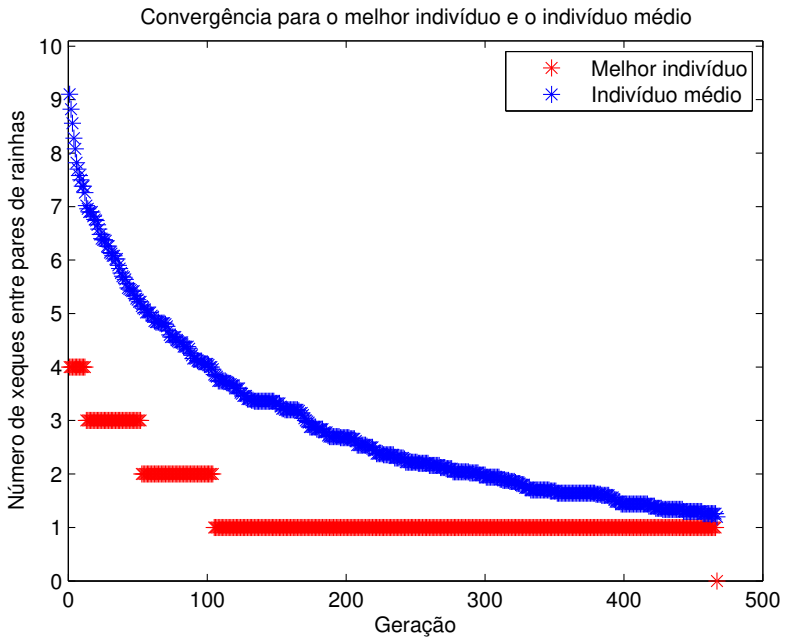

Figura 3: Resultado para a terceira configuração

Para o valor de N = 40, foram feitas dois testes, para mostrar a influência do aumento do tamanho do tabuleiro na complexidade:

Configuração A:

- N = 40
- População = 30
- Número máximo de gerações = 300

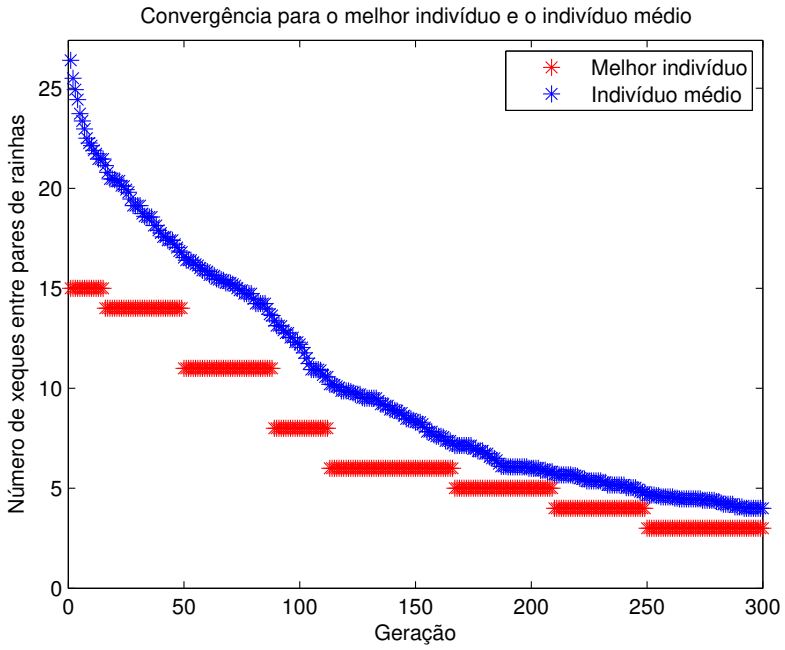

Figura 4: Resultado para o tamanho N = 40, primeira configuração

Configuração B:

- N = 40
- População = 100
- Número máximo de gerações = 10000

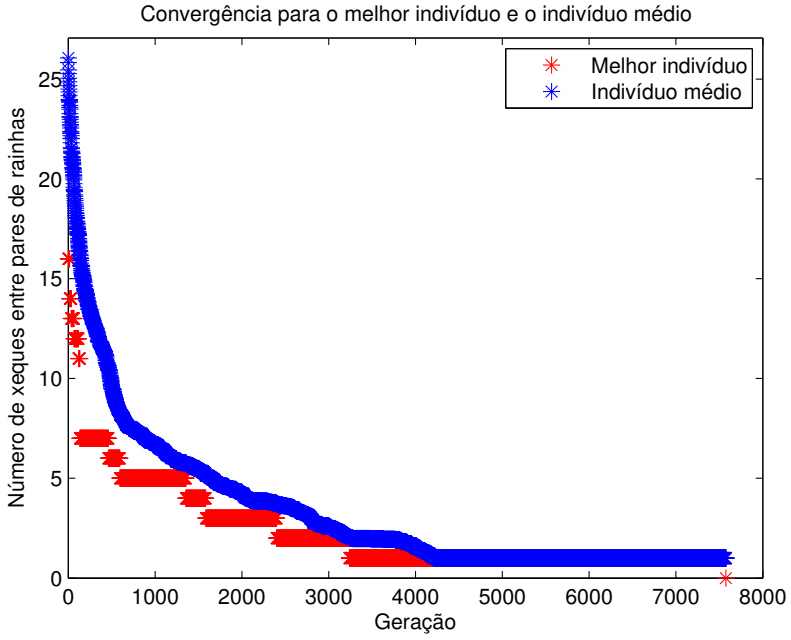

Figura 5: Resultado para o tamanho N = 40, segunda configuração

3 Análise dos resultados

Para os casos em que N é relativamente pequeno (como os casos 8x8 e 10x10), o algoritmo converge com um pequeno número de gerações e com uma população também relativamente pequena. Para o caso 15x15, o algoritmo já se torna mais lento. Foi preciso aumentar o número máximo de gerações, pois na grande maioria casos ele não convergia antes de 100 gerações.

Já para o caso extremo 40x40, o aumento no tamanho do tabuleiro foi considerável e não foi suficiente aumentar apenas o número máximo de gerações. A Figura 4 mostra que mesmo para um máximo de 300 gerações, o algoritmo não encontra solução para o problema. É necessário então aumentar tanto a população, quanto o número máximo de gerações, em relação aos casos mais simples. Percebe-se na Figura 5, que tem máximo de 10000 gerações e população igual a 100 indivíduos, são necessárias quase 8000 gerações para o algoritmo convergir.

Isso já era o resultado esperado. É interessante ver que ambos os parâmetros (população e número de gerações) influenciam diretamente na convergência do algoritmo, pois a ideia de que uma população maior pode convergir para uma solução usando o mesmo número de gerações parece lógica, porém com alguns testes é possível ver que isso não ocorre tanto. Muitas vezes será necessário um número maior de gerações para que o algoritmo encontre uma solução para o problema.

4 Código em MATLAB

A implementação do algoritmo foi feita por meio uma função do MATLAB chamada *nQueensProblem*. Ela recebe como argumentos de entrada *N*, *pop* e *ngenmax*, que são respectivamente, o tamanho do tabuleiro, o tamanho da população e o número máximo de gerações. Ela retorna a configuração de saída que zera a função de aptidão ou um vetor vazio caso não encontre solução.

Foram realizadas mudanças simples na função *CutAndCrosfill_Crossover.m* disponibilizada. Na linha 8 dessa função, mudou-se a posição inicial do corte para 2, para evitar que sejam gerados filhos idênticos aos pais. Por consequência, o valor calculado na linha seguinte pode ser maior que N, o que não é válido. Caso isso aconteça, é atribuído à variável o valor de N.