

Descrição do problema e possível solução

Para se encontrar uma solução para o problema das 8 Rainhas basta ir alternando a posição das rainhas, em suas respectivas linhas, até encontrar uma posição onde a rainha não entre em conflito com outra rainha.

Um exemplo:

```
tabuleiro = [5, 8, 6, 7, 1, 3, 4, 2]
```

O array tabuleiro na posição 1 (tabuleiro[0]) indica que a primeira rainha está disposta na primeira linha do tabuleiro, e na quinta coluna. Seguindo esta lógica, a representação gráfica do array tabuleiro seria a seguinte:

```
- - - - 0 - - -  
- - - - - - - 0  
- - - - - 0 - -  
- - - - - - 0 -  
0 - - - - - - -  
- - 0 - - - - -  
- - - 0 - - - -  
- 0 - - - - - -
```

Pode-se notar que algumas rainhas estão em conflito na diagonal, porém nunca as rainhas estarão dispostas na mesma linha ou na mesma coluna.

Uma das formas de resolver os conflitos nas diagonais das rainhas é mudá-las de coluna até se encontrar uma solução que não cause conflitos. É neste momento que entra o algoritmo de permutação.

Descrição da solução implementada

O R fornece, através do CRAN, o pacote GA para o uso de algoritmos genéticos. Assim, para resolver o problema das 8 rainhas, basta passar como parâmetro ao pacote GA que se quer a solução por permutação, os limites mínimo e máximo que os valores podem ter, na solução proposta serão entre 1 e o tamanho do tabuleiro, o número máximo de iterações, e a função fitness.

O algoritmo de permutação será responsável por reorganizar o array através do crossover, mutação ou elitismo. Já a função fitness será a responsável por validar se há conflitos nas diagonais das rainhas. Abaixo está a fórmula usada para validar se há conflito entre as diagonais de duas rainhas (fonte: <https://stackoverflow.com/a/3209201>):

```
temConflito = abs(linhaRainha1 - linhaRainha2) == abs(colunaRainha1 -  
colunaRainha2)
```

Desta forma, o algoritmo continua realizando as permutações até se encontrar uma solução sem conflitos, ou atingir o limite máximo de iterações.

Melhores soluções encontradas

Com o limite máximo de iterações definido em 20, foi rodado o algoritmo três vezes. Em todas as vezes o algoritmo encontrou soluções sem conflitos entre as rainhas:

Primeiro teste:

[5, 8, 4, 1, 3, 6, 2, 7]

[8, 4, 1, 3, 6, 2, 7, 5]

Segundo teste:

[6, 3, 1, 7, 5, 8, 2, 4]

[8, 2, 4, 1, 7, 5, 3, 6]

Terceiro teste:

[6, 4, 7, 1, 8, 2, 5, 3]

[5, 8, 4, 1, 3, 6, 2, 7]

Quando o número máximo de iterações é diminuído para 10, por exemplo, é possível notar que o algoritmo encontrará soluções sem conflito com menos frequência.

Bônus

Da forma que o algoritmo está estruturado hoje, é possível transformar o problema das 8 Rainhas em um problema das N Rainhas. Ao invés de se dispor rainhas sem conflitos em um tabuleiro 8x8, como o problema original se propõem, é possível encontrar uma solução que dispõem rainhas em um tabuleiro NxN sem conflitos.

Para isso basta alterar o tamanho do tabuleiro na linha 3 do algoritmo.