

METAHEURISTICA EFICIENTE USADA PARA RESOLVER O PROBLEMA DAS N RAINHAS.

Samuel Custódio de Oliveira, Rubén Romero.
Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia de Ilha Solteira – Campus de Ilha Solteira.

1. Introdução

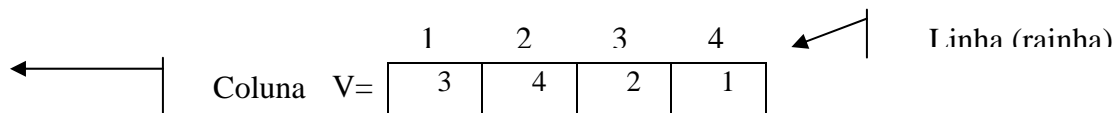
O problema das N-Rainhas é muito conhecido na literatura de pesquisa operacional. O problema consiste em colocar n rainhas em um tabuleiro de xadrez de dimensão $n \times n$ de maneira que não haja colisões entre as rainhas ou que minimize o número de colisões.

A colisão entre rainhas ocorre quando duas ou mais rainhas se encontram na mesma linha, coluna, ou diagonal do tabuleiro. A solução ótima para este problema obviamente será o número de zero colisões. Portanto a solução ótima do problema se encontra em dispor n rainhas em um tabuleiro $n \times n$ onde não exista mais de uma rainha na mesma linha, coluna ou diagonal do tabuleiro. Uma vez sabido que a solução ótima, ou seja, zero colisões, não existe para uma determinada configuração, não há como otimizar o problema.

Alternativamente o problema pode ser visto não mais para minimizar colisões, mas sim para maximizar o número de rainhas em um tabuleiro sujeito a um número de colisões, onde não deve existir mais de uma rainha na mesma linha, coluna ou diagonal do tabuleiro. Neste caso a solução ótima será o número de rainhas n .

No problema das N-Rainhas a codificação de uma configuração utiliza um vetor V de tamanho n , onde um elemento desse vetor, chamado $V(i)$ representa a coluna do tabuleiro que a rainha i está ocupando. A rainha i sempre ocupará a linha i .

Para representar a configuração mostrada na figura 1.0 a codificação será a seguinte.



A codificação apresentada é interessante por garantir a factibilidade da configuração inicial e por assegurar que não haverá colisões entre rainhas através de linhas e colunas do tabuleiro. Isto pode ser facilmente verificado, pois os elementos do vetor V são diferentes o que significa que sempre uma coluna é ocupada por uma única rainha, também não existe colisão através das linhas por que as rainhas sempre permanecem numa mesma linha do tabuleiro.

Sabendo que não haverá colisões entre rainhas através das linhas e colunas do tabuleiro, resta agora calcular o número de colisões através da diagonal. Uma maneira também muito

simples de codificar o número de colisões entre as rainhas através da diagonal é encarar o tabuleiro como uma matriz $n \times n$. É estabelecido o conceito de diagonal positiva e negativa no arranjo matricial.

Para encontrar o número de colisões na diagonal de uma configuração é suficiente percorrer as diagonais positivas e negativas de uma matriz e fazer a contagem do número de colisões.

Associa-se a cada diagonal um número que será uma constante para cada diagonal, número este que será calculado diferentemente para diagonais positivas e negativas.

Para a diagonal negativa soma-se o índice da linha com o índice da coluna de cada elemento da matriz e obtém-se a constante desta diagonal, já para a diagonal positiva subtrai-se o índice da linha com o índice da coluna de cada elemento da matriz e obtém-se a constante desta outra diagonal, percorrendo as diagonais, se mais de uma rainha estiver com o mesmo número associado à célula onde ela se encontra na matriz existe colisão do contrário não.

2. Algoritmo Genético de Chu-Beasley

Neste item analisaremos os principais componentes do algoritmo genético desenvolvido por Chu-Beasley para solucionar o problema das N-Rainhas.

2.1 Codificação do problema

A codificação foi feita segundo a representação vista anteriormente. Por se tratar de um algoritmo genético trabalha-se com vários indivíduos formando uma população. Esta população de tamanho m será representada por uma matriz onde cada linha da matriz representa uma configuração de n rainhas em um tabuleiro. O tabuleiro de xadrez será representado por uma linha da matriz com dimensões n , onde cada elemento da matriz representa a coluna onde a rainha se encontra no tabuleiro e cada índice de coluna da matriz representa a linha onde a rainha está localizada no tabuleiro ou propriamente à rainha.

A vantagem de trabalhar com esta proposta de codificação facilita no cálculo do número de colisões entre as rainhas, pois garante que não haverá colisões entre rainhas na mesma linha e coluna, restando somente o cálculo de colisões entre rainhas nas diagonais. Outra vantagem da codificação proposta é garantir sempre a factibilidade da população inicial.

2.2 Função Objetivo

A função objetivo é calculada segundo a proposta vista anteriormente de verificar o número de colisões entre rainhas percorrendo as diagonais positiva e negativa do tabuleiro.

É agregada a cada diagonal uma constante que a identifica, se existir mais de uma rainha em uma diagonal é contabilizado mais uma colisão.

A constante da diagonal positiva é calculada através da diferença entre o índice de coluna da matriz de cada célula e o elemento que ocupa cada célula.

Já a constante da diagonal negativa é calculada através da soma entre o índice de coluna da matriz de cada célula e o elemento que ocupa cada célula.

As somas do número de colisões existentes nas diagonais positivas e negativas geram o cálculo da função objetivo.

2.3 Substituição da população

A proposta a ser apresentada consiste em substituir, em cada passo, apenas um elemento da população. O descendente gerado entra na população se for diferente dos elementos da população e se for de melhor qualidade que o elemento de pior qualidade da população.

2.4 Seleção

A seleção utilizada é baseada em torneio com $k = 2$, onde k é o número de topologias da população que vão participar de cada jogo. A seleção por torneio é uma das formas mais eficientes de realizar a seleção desde que o valor de k seja adequadamente escolhido. Os valores geralmente utilizados de k variam entre 2 e 3, no nosso caso 2.

2.5 Recombinação

A recombinação para este tipo de problema não poder ser a recombinação tradicional de um algoritmo genético, onde são escolhidos aleatoriamente dois indivíduos da população corrente, estes indivíduos são recombinados a partir de um ponto também aleatoriamente escolhido e gerariam dois descendentes.

Se aplicássemos este tipo de recombinação correríamos o risco de perder a factibilidade do problema, ou seja, ter colisões entre rainhas na mesma coluna, isto é, ter elementos com valores repetidos dentro de uma linha da matriz que representa a população corrente. Isto não seria bom, pois perderia o sentido de trabalhar com a proposta de codificação vista anteriormente sendo ela tão vantajosa. Optou-se, portanto, em trabalhar com a recombinação PMX que recombina os dois indivíduos sem perder o que já havíamos conquistado (configuração sem colisões entre rainhas na mesma linha e coluna).

Este tipo de recombinação trabalha com dois indivíduos escolhidos aleatoriamente. É escolhido também aleatoriamente o tamanho da faixa que será recombinada e onde a recombinação irá começar. Neste estudo os dois indivíduos geram somente um descendente.

2.6 Mutação

A mutação se dá de uma maneira muito simples. Serão escolhidas aleatoriamente duas rainhas em cada indivíduo da população corrente para se fazer a troca de posição destas rainhas no tabuleiro.

2.7 Proposta de Melhoria

Propõe-se que ao invés de detectar qual a diagonal mais carregada na configuração inicial e melhorá-la, detectar a rainha desta configuração que sofre o maior número de ataques, e conseqüentemente trocá-la de posição no tabuleiro com outra rainha que sofre o menor número de ataques possível.

Verificou-se também que as configurações ótimas dos problemas apresentavam certa variabilidade das rainhas no tabuleiro, e com isto, geraram-se as configurações iniciais com certa variabilidade, de modo que cada rainha que era gerada não colidisse com nenhuma rainha já disposta no tabuleiro, ou melhor, que gerasse o menor número de colisões possível.

3. Testes e Resultados

Ao implementar o problema das N-Rainhas usando o algoritmo genético de Chu-Beasley, o resultado utilizando uma população de 40 indivíduos foi considerado satisfatório.

Gerando uma população aleatoriamente, verificou-se que a média de colisões para cada configuração era de aproximadamente 66,5% do número de rainhas. Com a proposta de melhoria, verificou-se um resultado satisfatório visto que a média de colisões para cada configuração é de aproximadamente 9,0% do número de rainhas. Com isto, o número de iterações reduziu consideravelmente.

3. Conclusão

O algoritmo encontrou resultados de melhor qualidade que o algoritmo genético tradicional. Com uma população relativamente pequena e com um grande número de rainhas envolvidas no problema, o programa teve um ótimo desempenho chegando ao resultado ótimo em aproximadamente 1.0 min para 400 rainhas e um pouco mais de 16.0 min para 1000 rainhas.

Bolsa: CNPq