

UMASOLUÇÃO DO PROBLEMA DAS N RAINHAS ATRAVÉS DE ALGORITMOS GENÉTICOS

A SOLUTION OF N-QUEENS PROBLEM BY GENETIC ALGORITHMS

Eliane Vendramini de Oliveira

Mestre em Engenharia Elétrica – UNESP/ Ilha Solteira

RESUMO

Este artigo analisa o problema das N-Rainhas a ser resolvido por um algoritmo evolutivo, o algoritmo genético. Este algoritmo resolve o problema através de uma função objetivo que calcula as colisões entre as rainhas no tabuleiro e melhora (redução do número de colisões entre rainhas) a configuração de modo que o número de colisões se reduza a zero. Passando por todas as etapas de um algoritmo genético clássico (seleção, recombinação, mutação) e com algumas modificações para melhor atender o problema, é obtido resultados de boa qualidade.

Palavras-chave: Algoritmo Genético. Problema N Rainhas. Metaheurística.

ABSTRACT

This paper analyzes the problem of the N-queens being solved by an evolutionary algorithm, the Genetic Algorithm. This algorithm solves the problem through a function objective that calculates the collisions among the queens in the board and improvement (it reduces the number of collisions among queens) the configuration so that the number that collisions reduce to zero.

Going by all of the stages of a classic genetic algorithm (selection, recombination, mutation) and with some modifications for best to assist the problem, it is obtained results of good quality.

Key-words: Genetic Algorithms. N-Queens Problem. Metaheuristics

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, segundo Laguna (1994).

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.

A representação deste problema caracteriza-se por ser de fácil codificação.

O Algoritmo Genético por sua vez é um algoritmo que imita o processo de seleção natural, por isto o nome Algoritmo Genético, e que para imitar este processo trabalha com algumas etapas pré-definidas: seleção, recombinação e mutação.

Este algoritmo por seu poder de combinação de resultados e otimização dos mesmos pode ser empregado na solução do problema das N Rainhas, mas sua formação tradicional pode ser melhorada se adequada ao problema aqui exposto.

Assim sendo, o estudo realizado tem como principal objetivo contribuir com uma solução para o problema das N Rainhas propondo o Algoritmo Genético modificado como o instrumento de solução e melhora de resultados, comparando com o Algoritmo Genético tradicional.

MATERIAIS E MÉTODOS

Existem várias codificações para representação do problema, utilizaremos a seguinte.

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 a codificação será a seguinte.

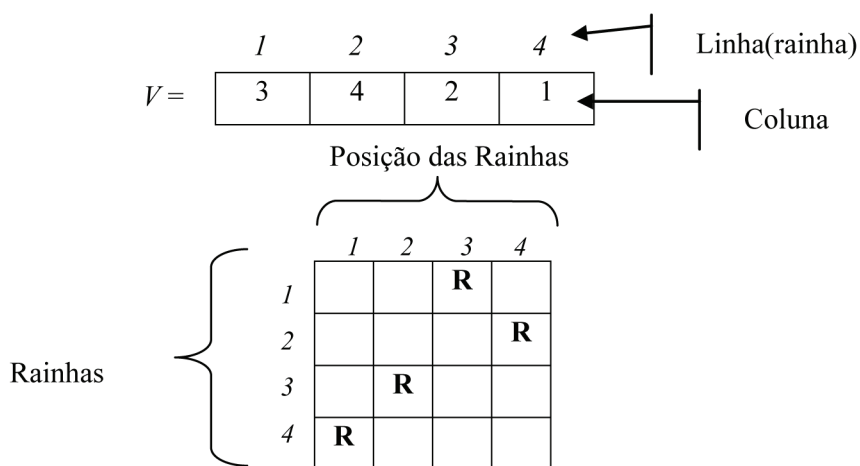


Fig. 1: Configuração inicial de 4 rainhas

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. Estas características são mostradas nas figuras 2 e 3.

1	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0

Fig. 2: Constante das Diagonais Positivas

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8

Fig. 3: Constante das Diagonais Negativas

Pode-se verificar facilmente que existem 2 colisões percorrendo as diagonais positivas e negativas. Na verdade existe uma colisão na diagonal positiva -2 e uma colisão na diagonal negativa 5.

Finalizando assim o processo de codificação e representação do problema das N-Rainhas.

Algoritmo Genético

Nesta seção verifica-se o funcionamento das principais etapas do algoritmo genético sendo aplicado ao problema das N-Rainhas, abordado neste estudo. Além das etapas tradicionais, serão relatados os tópicos relacionados com a melhoria dos descendentes e a forma como estes são aproveitados nas próximas gerações.

Tipo de Algoritmo Genético

O algoritmo genético aplicado neste trabalho é baseado no Algoritmo Genético de Chu-Beasley, onde o controle da diversidade é mais acentuado do que o tradicional, segundo Alencar (2004) e utilizando além das etapas tradicionais uma melhora local.

Codificação do Problema

A codificação foi feita segundo a representação vista anteriormente.

Por se tratar de um algoritmo genético, trabalha-se com vários indivíduos (configurações) 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.

A população escolhe aleatoriamente um valor para cada elemento de uma linha da matriz, de maneira que eles não se repitam.

Veja a ilustração da representação deste problema na figura 4.

		Número de Rainhas														
		1	2	3	4	5	6	7	8	9	10	11	...	n		
Número de Indivíduos	1	3	5	4	1	2	9	7	6	8	11	19	10
	2	20	11	15	3	4	2	12	5	7	13	18	6
	...	3	5	4	2	1	7	9	6	8	11	19	10
	m	20	1	16	3	4	2	7	6	8	11	5	14

Fig 4: Proposta de codificação do problema das N-Rainhas sendo resolvido por Algoritmo Genético

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

A soma do número de colisões existentes nas diagonais positiva e negativa geram o calculo da função objetivo.

Seleção

O método de seleção escolhido para ser aplicado no problema foi o método de seleção por torneio com $k=2$, ou seja, é escolhido para participar de cada jogo dois indivíduos (configurações) da população corrente que são analisados, o calculo da função objetivo é feito e o indivíduo que detém o menor número de colisões é escolhido.

Recombinação

Optou-se 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 artigo os dois indivíduos geram somente um descendente, onde é escolhido aleatoriamente em qual dos dois indivíduos a recombinação será executada para a geração de um descendente. Na figura 5 é exemplificado como a recombinação é realizada.

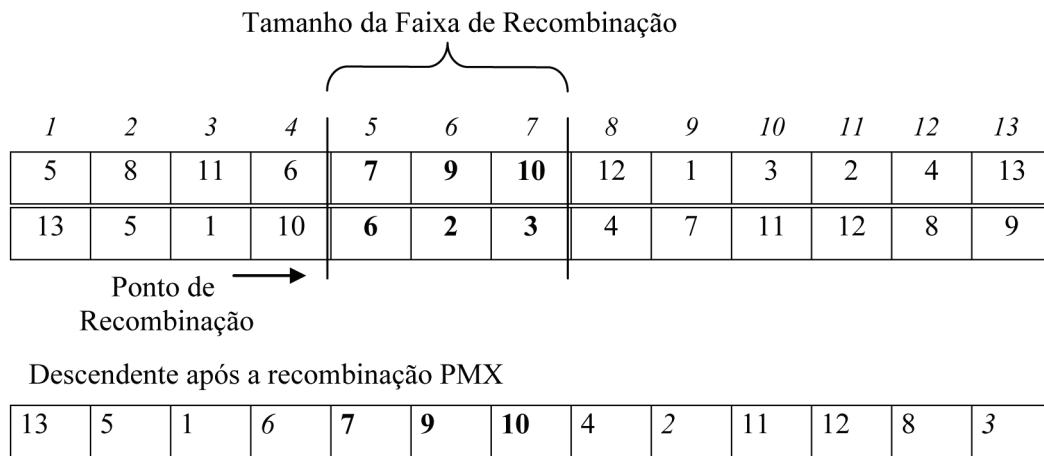


Fig. 5: Exemplo de Recombinação

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. As figuras 6 e 7 mostram como o processo de mutação ocorre.

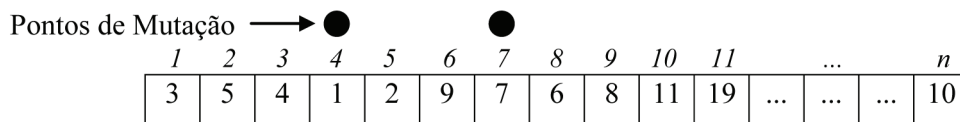


Fig 6: População antes da Mutação

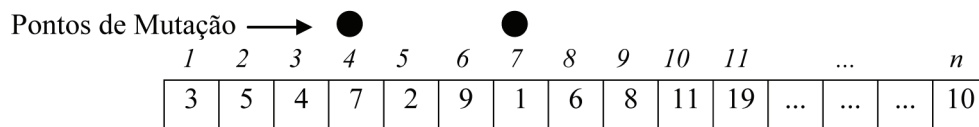


Fig 7: População depois da Mutação

Melhorando os descendentes

Procurando melhorar o descendente antes de verificar se ele será aproveitado na próxima geração, este descendente passa por uma etapa de melhora local.

Verifica-se quais das diagonais, positivas ou negativas, deste indivíduo (configuração) está mais carregada, ou seja, com mais colisões. Logo após é simulada cada troca de posição no tabuleiro das rainhas, e a suposta troca que gerar um número menor de colisões é efetivada mudando as posições das rainhas no tabuleiro.

Substituição da População

Somente será aproveitado o indivíduo gerado na etapa de seleção, mutação e melhora local se este detém um número

menor de colisões do que o indivíduo que detém o maior número de colisões da população corrente.

Além disso, é verificado se esta configuração já existe na população corrente, se existir o mesmo indivíduo (configuração) na população, o indivíduo descendente será descartado, caso contrário ele será aproveitado na população corrente e entrará no lugar do indivíduo que tinha o número maior de colisões.

Algoritmo

O algoritmo assume a seguinte forma:

1. Gera-se a população inicial aleatoriamente.
2. Transforma-se a população inicial na população corrente.
3. Calcula-se o valor da função objetivo.
4. Implementa-se o processo de seleção de dois indivíduos aleatoriamente;
5. Escolhe-se aleatoriamente o tamanho da faixa de recombinação e a posição de início.
6. Escolhe-se aleatoriamente qual indivíduo sofrerá a recombinação PMX.
7. Implementa-se o processo de recombinação PMX.
8. Implementa-se mutação escolhendo dois pontos de mutação aleatoriamente.
9. Verifica-se a possibilidade de melhorar o indivíduo gerado na etapa anterior. Neste caso o descendente passa por uma série de comparações.
 - Primeiro localiza-se a diagonal mais carregada (número maior de rainhas).
 - Troca-se de posição as rainhas dessa diagonal e verifica-se qual troca teve o menor número de colisões, a troca de resultado melhor é consumada.
9. Verifica-se o valor da função objetivo do indivíduo descendente, se este número for menor do que o maior da população corrente o indivíduo pode ser aproveitado caso contrário volta-se para o passo 3.
10. Se o indivíduo descendente é melhor do que o pior encontrado na população corrente, ele passa por uma nova comparação. Este novo indivíduo (configuração) deve ser diferente de todos os outros da população, se já existir na população corrente a mesma configuração, esta será descartada e volta-se para o passo 3, caso contrário ela será aproveitada, fazendo parte da nova geração.
11. Atualiza-se a população corrente gerando a próxima geração e volta ao passo 2.
12. O algoritmo para em um número pré-determinado de iterações.

RESULTADOS E DISCUSSÃO

O algoritmo genético pode ser melhorado no seguinte aspecto.

Propõe-se que ao invés de detectar qual a diagonal mais carregada na configuração descendente e melhorá-la, deve-se detectar a rainha desta configuração que sofre o maior número de ataques, e trocá-la de posição no tabuleiro com todas as rainhas que se confrontam com ela. A troca que resultar em um número menor de colisões é consumada.

Os resultados obtidos com a proposta de melhoria, são consideráveis.

O critério de parada teve duas opções a seguir, ou parar quando encontrar zero colisões na população corrente ou estipular um número de iterações para o algoritmo parar. Depois de muitos testes optou-se por pré determinar um número de iterações como critério de parada.

O tamanho da faixa de recombinação foi outro ponto importante para o sucesso do algoritmo. Estipulo-se que em uma

população de 1000 indivíduos com 1000 rainhas o tamanho da faixa de recombinação ficasse entre 30 e 80 rainhas.

Ao implementar o problema das N-Rainhas usando algoritmos genéticos os resultados foram os seguintes:

Com 1000 Rainhas e uma população de 1000 indivíduos

Sem Melhoria do descendente: 150.000 iterações para achar a solução ótima.

Com melhoria pela diagonal mais carregada: 150.000 iterações para achar a solução ótima, ou seja, não houve melhora.

Com melhoria pela rainha mais atacada: 1 iteração para achar a solução ótima.

O primeiro indivíduo a retornar zero colisões no teste com 1000 rainhas e 1000 indivíduos foi o de número 930.

CONCLUSÕES

O algoritmo melhorado neste artigo encontrou resultados de melhor qualidade que o algoritmo genético tradicional, principalmente quando a melhoria vem da troca da rainha com maior número de ataques.

REFERÊNCIAS BIBLIOGRÁFICAS

LAGUNA, M. A Guide to Implementing Tabu Search. **Investigacion Operativa**, 4:159-178, 1994.

ALENCAR, M. P. L. Análise Crítica do Algoritmo Genético de Chu-Beasley para o Problema Generalizado de Atribuição. **SBPO**, 36:1391-1399, 2004.

DIAZ, A.; GLOVER, F.; CHAZIRI, H. M.; LAGUNA, M.; MOSCATO, P.; TSENG, F. T. **Optimización Heurística y Redes Neuronales**. 1 ed. MADRID: Paraninfo, 1996.

SOSIE, R.; GU, J. Fast Search Algorithms for the N-Queens Problem. **IEEE Transactions on Systems, Man, and Cybernetics**, 21:1572-1576, 1991.

CHU, P.; BEASLEY, J. E. A Genetic Algorithm for the Generalized Assignment Problem, **Computers and Operation Research**, 24:17-23, 1997.