

Algoritmos para otimização do problemas das N-Rainhas e da função Rastrigin

Marcio Roberto Andrade Souza Filho
Univesidade Federal de Minas Gerais - UFMG
Belo Horizonte - Minas Gerais, Brasil
marciorasf@gmail.com

Resumo—O presente trabalho propõe dois algoritmos genéticos, GA-Rainhas e GA-Rastrigin, o primeiro para encontrar uma solução para o problema das N-rainhas e o segundo para a minimização da função Rastrigin. O GA-Rainhas é um algoritmo de otimização de problemas de permutação inteira, devido a modelagem do problema das N-rainhas. O foco principal desse trabalho para o GA-Rainhas foi a otimização do problema das N-rainhas com $N = 8$. O GA-Rastrigin tem suas bases no Simple Genetic Algorithm (SGA) [1], com modificações feitas para melhor atender ao seu propósito multimodal, tendo como foco a minimização da função Rastrigin. Neste trabalho é apresentada a estrutura geral algoritmos, os resultados e análise crítica dos testes realizados.

Palavras-chave—Algoritmo genético, N-rainhas, Multimodal, Rastrigin.

I. INTRODUÇÃO

A. Problema das N-rainhas

O problema das n-rainhas consiste em, dado um tabuleiro de xadrez $N \times N$, posicionar N rainhas de forma que nenhuma delas seja capaz de atacar outra em somente uma jogada. A Figura 1 mostra o alcance de captura de uma rainha e ilustra uma solução válida para o problema em um tabuleiro 8×8 . Para um tabuleiro 8×8 , há ${}^{64}C_8 = 4426165368$ maneiras distintas de posicionar as 8 rainhas, no entanto se considerarmos que as rainhas ocupam linhas e colunas distintas, o número de possibilidades diminuiria a $8! = 40320$. Existem 92 [2] soluções válidas, sendo assim a probabilidade de alocar as oito rainhas aleatoriamente seria $\frac{92}{40320} \approx 0.00228 \equiv 0.228\%$.

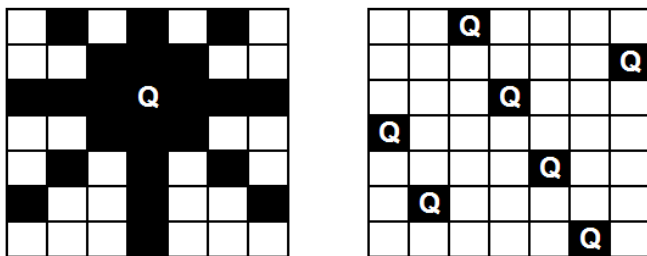


Figura 1. Alcance de captura de uma rainha e uma solução válida para um tabuleiro 8×8 , respectivamente da esquerda para direita.

Fonte: <https://www.dcc.fc.up.pt/~ricroc/aulas/0405/soII/prat/exerc2.html>.

B. Função Rastrigin

A função Rastrigin é uma função não convexa multimodal utilizada na avaliação de algoritmos de otimização. Inicialmente

foi proposta por Rastrigin [3] como uma função bidimensional e posteriormente generalizada por Mühlhennbein et al [4]. É definida para n dimensões por:

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

com $x_i \in [-5.12, 5.12]$, mínimo local em $x^* = 0$ e $f(x^*) = 0$. A Figura 2 ilustra as curvas de nível da função Rastrigin, para o caso bidimensional. Nela é possível observar os vários mínimos locais presentes na superfície. Essa característica é a principal dificuldade encontrada pelos algoritmos de otimização para encontrar seu mínimo global.

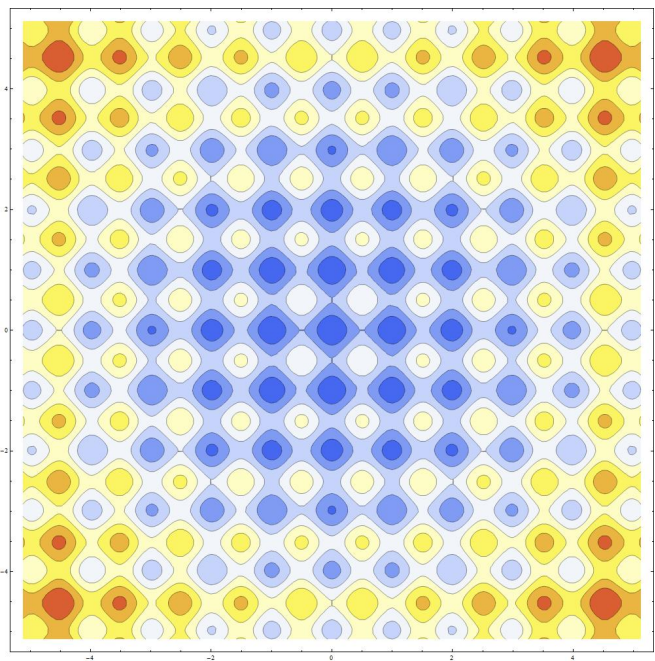


Figura 2. Curvas de nível da função Rastrigin para o caso bidimensional. Fonte: By Tos - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=16082548>.

II. GA-RAINHAS

A. Estrutura Geral

O algoritmo GA-Rainhas possui a mesma estrutura geral dos algoritmos genéticos. Essa estrutura está ilustrada na Figura 3.

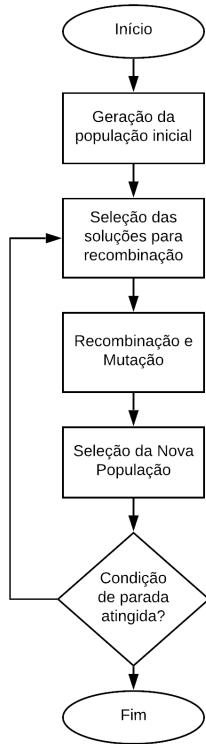


Figura 3. Estrutura geral de algoritmos genéticos

B. Representação

Como apresentado na seção I, permitir que uma rainha só possa ser posicionada de forma que não haja outra na mesma linha ou coluna reduz significativamente o espaço de busca do problema. Essa restrição pode ser alcançada representando as soluções candidatas por vetores de permutação de dimensão N , onde o índice da coluna representa a coluna e o valor atribuído ao índice representa a linha da posição da rainha. O exemplo a seguir ilustra uma solução candidata, considerando o problema com $N = 4$.

$$[0, 3, 1, 2]$$

O vetor de permutação representa a seguinte configuração: rainhas nas posições $[0, 0]$, $[3, 1]$, $[1, 2]$, $[3, 2]$.

C. Função de Aptidão - Fitness

A função de aptidão mais direta é utilizar a quantidade de conflitos entre as rainhas da solução avaliada. Devido a maneira que são representadas as soluções, os conflitos acontecem apenas em diagonais. Para contabilizá-los, para cada rainha é verificado quantas outras estão na mesma diagonal, i.e., a distância entre as linhas é igual a distância entre as colunas. Para que o objetivo do algoritmo fosse maximizar a função de aptidão, ela foi definida como:

$$fitness = \frac{1}{QuantidadeConflitos + 1}$$

D. Operadores de Variação

1) *Order Crossover - OX*: O operador OX gera duas novas soluções candidatas, denominadas filhos, a partir de duas outras soluções, denominadas pais, mantendo a ordem em que os valores aparecem. Para gerar uma solução, o operador sorteia um inteiro, pos_corte no intervalo $[0, N - 1]$, esse número representa a posição de corte. O filho1 receberá o segmento $[0, pos_corte]$ do pai1, e o restante do vetor será preenchido com os valores que faltam obedecendo a ordem em que aparecem no pai2. Para a geração do filho2 é realizado o mesmo procedimento porém trocando a ordem que os pais são utilizados.

2) *SWAP*: Para uma dada solução candidata o operador de mutação SWAP sorteia duas posições do vetor de permutação, os valores correspondentes a essas posições são trocados.

E. Modelo de população

O modelo de população adotado foi o Steady-State, a população é formada por 50 soluções candidatas. A cada geração as duas piores soluções são substituídas por novas soluções.

F. Mecanismos de Seleção

1) *Seleção das Soluções para Recombinação*: A seleção das soluções utilizadas na recombinação é feita a partir de um torneio. São sorteados cinco soluções distintas da população, aquelas duas com melhores aptidões são as escolhidas.

2) *Seleção da Nova População*: Após a geração de duas novas soluções, estas são inseridas na população substituindo as duas piores soluções.

G. Condição de Parada

A condição de parada define quando o algoritmo deve parar sua execução. A condição de parada trivial é a existência de uma solução válida, i.e., uma solução com função de aptidão igual a um, que representa uma solução com zero conflitos entre as rainhas. Além da condição anterior foi definido um número de gerações máximas para impedir algoritmo executar por tempo indeterminado. Esse limite é 4975 gerações, que resulta em um orçamento máximo de 10000 avaliações da função de aptidão.

III. GA-RASTRIGIN

A. Estrutura Geral

Assim como o GA-Rainhas, a estrutura geral do GA-Rastrigin segue o esquema ilustrado na Figura 3.

B. Representação

As soluções são representadas utilizando codificação binária utilizando 16 bits por variável. Por conta de facilidades na implementação, ao invés de uma única *String* binária para representar todas as variáveis, a solução foi representada como uma lista de *Strings*, na qual cada *String* representa uma variável. Para diminuir o problema de Hamming Cliff a codificação binária foi implementada seguindo o código de Gray.

C. Função de Aptidão - Fitness

A função de aptidão utilizada é a própria função Rastrigin. Dessa forma, a avaliação da função de aptidão para uma solução é tão melhor quanto menor seu valor.

D. Operadores de Variação

1) Recombinação com 1 Ponto de Corte por Variável:

Dado duas soluções candidatas (pais), para cada variável é sorteado um ponto de corte (*pos_corte*), o filho1 é composto pelos primeiros *pos_corte* bits do pai1, os outros bits são completados pelos bits do pai2. O filho2 é formado alterando a ordem em que os pais são utilizados em relação ao filho1.

2) *Mutação Bit-Flip*: A mutação Bit-Flip possui duas formas conhecidas de implementação. Na primeira segundo uma probabilidade p_m uma variável terá um bit trocado, esse bit é selecionado aleatoriamente. Na segunda, a mutação é aplicada para cada bit do indivíduo segundo uma probabilidade p_m . Por usar dezesseis bits para cada variável na representação, a segunda implementação é mais vantajosa para possibilitar maior exploração do espaço de busca.

E. Modelo de população

O modelo de população adotado foi geracional, a população é formada por 100 soluções candidatas. A cada geração essas 100 soluções são substituídas pelas soluções escolhidas pelos mecanismo de seleção da nova população.

F. Mecanismos de Seleção

1) *Operador baseado em Ranking*: Para implementar o método da roleta, é necessário calcular as probabilidades de seleção de cada indivíduo e posteriormente mapear essas probabilidades em um segmento de reta $[0, 1]$. Por utilizar a própria função Rastrigin como função de aptidão, o cálculo das probabilidades de seleção foi feito utilizando o ranqueamento da população. Para uma população ordenada pela função rastrigin de forma decrescente, as probabilidades de seleção são calculadas por:

$$p_i = \frac{2 - s}{\mu} + \frac{2i(s - 1)}{\mu(\mu - 1)}$$

sendo p_i a probabilidade de seleção da solução i , s é um parâmetro que determina a pressão de seleção imposta e μ o tamanho da população.

2) *Seleção das Soluções para Recombinação*: A seleção das soluções para recombinação tem 50% de probabilidade de ser feita via torneio e o restante via roleta. No torneio, dois pares de soluções são escolhidas aleatoriamente da população e as melhores entre os pares são escolhidas. No método da roleta, é utilizado o segmento de reta gerado pelo operador descrito na subseção anterior. Dois número no intervalo $[0, 1]$ são gerados, os pais escolhidos correspondem aos pais cujas posições na reta contém os valores gerados.

3) *Seleção da Nova População*: Considerando o foco do GA-Rastrigin, em otimizar a função Rastrigin com um orçamento de 10000 avaliações da função Rastrigin, foi utilizada uma estratégia elitista para acelerar sua convergência. A população nova a cada geração é formada por 65% das melhores soluções das populações originais da geração (pais) e da gerada (filhos), os 35% restante da população é completada escolhendo aleatoriamente, soluções ainda não escolhidas dentre a população gerada (filhos). Essa proporção foi definida a partir de testes.

G. Condição de Parada

A condição de parada utilizada é o algoritmo alcançar o número de avaliações máximo ou encontrar uma solução cujo valor da função Rastrigin é menor que 0.0001, esse valor é um valor aproximado do mínimo valor que a função consegue alcançar utilizando 16 bits para a codificação.

H. Hiper-Parâmetros

A probabilidade de cruzamento é 1.0 durante toda a execução do algoritmo. A probabilidade de mutação é definida no começo do algoritmo é definida por $\frac{8}{N_{bits}}$, a cada 33% do número total de gerações essa probabilidade é multiplicada por 0.25, essa estratégia se mostrou útil ao aumentar a exploração do espaço de busca em uma primeira fase do algoritmo e ao decair aumentar o refinamento das soluções encontradas.

IV. RESULTADOS

Nesta seção estão apresentados os resultados obtidos pelos algoritmos GA-Rainhas e GA-Rastrigin. Como o problema das n-rainhas é relativamente mais simples que o problema da função Rastrigin, foram feitos menos testes para o algoritmo GA-rainhas.

A. GA-Rainhas

A Figura 4 e a Figura 5, ilustram evoluções típicas das populações do algoritmo GA-Rainhas. A Figura 4 mostra a média dos conflitos entre rainhas por geração enquanto a Figura 5 mostra a quantidade de conflitos para a melhor solução por geração.

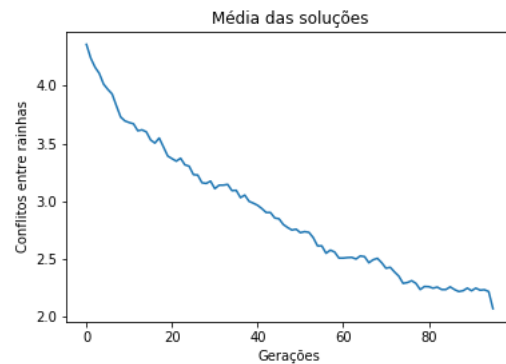


Figura 4. Evolução da população em relação ao número médio de conflitos entre rainhas.

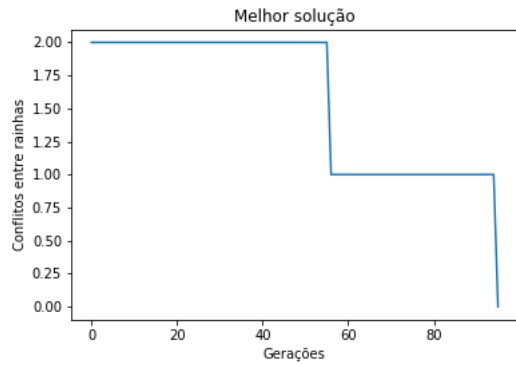


Figura 5. Evolução da população em relação ao número de conflitos entre rainhas da melhor solução.

Para determinar em quantas gerações em média o GA-Rainhas encontra uma solução válida, foram realizadas 10000 execuções do algoritmo, os resultados obtidos estão sumarizados na Tabela I e no histograma da Figura 6.

Tabela I
RESULTADOS OBTIDOS A PARTIR DE 10000 EXECUÇÕES DO GA-RAINHAS..

Nº de Execuções	Média de gerações	Desvio padrão
10000	109.79	118.98

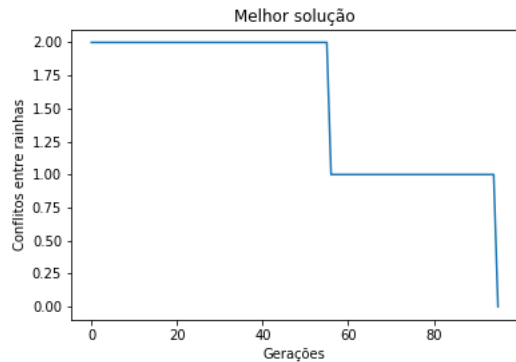


Figura 6. Histograma relativo a 10000 execuções do GA-Rainhas.

B. GA-Rastrigin

Para avaliação do GA-Rastrigin foram definidos duas configurações básicas de teste: considerando o orçamento máximo de avaliações da função Rastrigin igual a $10000 \cdot n$ número de variáveis ou considerando o orçamento máximo igual a 10000 avaliações. O primeiro teste do algoritmo visou avaliar sua qualidade para o padrão comum de benchmarks que define o orçamento máximo de avaliações da função rastrigin em $10000 \cdot n$ avaliações, sendo n o número de variáveis. O número de variáveis foi variado de 1 a 10, para cada valor foram feitos 20 execuções do algoritmo. Os resultados estão sumarizados na Tabela II.

Tabela II
RESULTADOS DO GA-RASTRIGIN PARA ORÇAMENTO DE $10000 \cdot n$ AVALIAÇÕES.

Nº de variáveis	Média da função Rastrigin	Nº de convergências
1	4.48e-5	20
2	4.17e-5	20
3	6.51e-5	20
4	6.83e-5	20
5	7.34e-5	20
6	7.75e-5	20
7	8.94e-5	20
8	8.82e-5	20
9	9.03e-5	20
10	9.96e-2	18

O segundo teste, foi realizado da mesma forma que o primeiro porém limitando o orçamento máximo em 10000 avaliações. Os resultados obtidos estão na tabela III.

Tabela III
RESULTADOS DO GA-RASTRIGIN PARA ORÇAMENTO DE 10000 AVALIAÇÕES.

Nº de variáveis	Média da função Rastrigin	Nº de convergências
1	2.06e-5	20
2	4.60e-5	20
3	5.84e-5	20
4	4.98e-2	19
5	5.97e-1	8
6	6.97e-1	9
7	1.25e+0	0
8	2.54e+0	0
9	4.10e+0	0
10	4.40e+0	0

O terceiro e quarto teste buscou comparar a implementação utilizando codificação binária simples e a implementação utilizando código de Gray. Para isso foram realizados os dois testes descritos acima, porém para o algoritmo utilizando código de Gray. Os resultados estão na tabela IV e V.

Tabela IV
RESULTADOS DO GA-RASTRIGIN UTILIZANDO CÓDIGO DE GRAY PARA ORÇAMENTO DE $10000 \cdot n$ AVALIAÇÕES.

Nº de variáveis	Média da função Rastrigin	Nº de convergências
1	3.85e-5	20
2	4.99e-5	20
3	5.21e-5	20
4	6.49e-5	20
5	6.76e-5	20
6	8.43e-5	20
7	8.26e-5	20
8	8.43e-5	20
9	4.98e-2	19
10	7.46e-5	20

V. ANÁLISE DE RESULTADOS

Nesta seção será feita uma análise crítica dos resultados apresentados na seção anterior.

A. GA-Rainhas

Os resultados referentes ao algoritmo GA-Rainhas ilustram que o algoritmo é capaz de resolver o problema para o caso

Tabela V
RESULTADOS DO GA-RASTRIGIN UTILIZANDO CÓDIGO DE GRAY PARA
ORÇAMENTO DE 10000 AVALIAÇÕES.

Nº de variáveis	Média da função Rastrigin	Nº de convergências
1	2.08e-5	20
2	5.18e-5	20
3	6.18e-5	20
4	5.72e-5	20
5	1.99e-1	16
6	7.59e-1	5
7	1.99e+0	0
8	2.86e+0	0
9	3.60e+0	0
10	5.41e+0	0

de oito rainhas e possivelmente para mais rainhas também. Porém pelos gráficos é possível observar que o algoritmo fica estagnado em uma melhor solução por várias gerações, isso pode representar um problema para dimensões maiores do problema, pois existiriam vários patamares onde o algoritmo estagnaria. O estudo de outros operadores de variação seria fundamental para melhorar a convergência do algoritmo, ou até mesmo adotar ao invés de um algoritmo genético, uma heurística para tentar encontrar soluções válidas.

B. GA-Rastrigin

Os resultados referentes ao algoritmo GA-Rastrigin mostram que ele é capaz de minimizar a função Rastrigin na maioria de suas execuções para até dez variáveis, contudo que seja fornecido um orçamento de 10000*n avaliações da função Rastrigin. Para o objetivo de minimização com orçamento de 10000 avaliações, o algoritmo implementado se mostrou adequado para até quatro variáveis, apesar de ter convergido algumas vezes para cinco e seis variáveis, essa limitação ocorre devido ao tradeoff entre rápida convergência e exploração do espaço de busca, pois para atingir resultados melhores para 10000 avaliações é necessário acelerar o processo de convergência do algoritmo, o que acaba levando-o para mínimos locais.

Em relação à utilização do código de Gray, analisando os resultados obtidos, não é possível perceber melhoras significativas no comportamento do algoritmo que justifiquem o aumento de seu esforço computacional para utilizar o código de Gray.

VI. CONCLUSÃO

O seguinte trabalho teve como objetivo demonstrar a estrutura e o desempenho dos algoritmos genéticos GA-Rainhas e GA-Rastrigin na resolução dos problemas das N-rainhas e minimização da função Rastrigin respectivamente.

Em relação ao desempenho dos algoritmos, eles são capazes de atender aos seus propósitos dentro de orçamentos computacionais razoáveis.

Em trabalhos futuros, para o GA-Rainhas, poderiam ser testados outros operadores de mutação de cruzamento e mutação a fim de melhorá-lo. Sobre o GA-Rastrigin, poderia ser feita uma abordagem utilizando nichos para maximizar a convergência do algoritmo para o máximo global.

REFERÊNCIAS

- [1] J. H. A. in Natural and M. . s. e. . T. U. o. M. P. A. A. Artificial Systems. MIT Press, Cambridge.
- [2] B. . A. s. o. k. r. Bell, Jordan; Stevens and research areas for n-queens". Discrete Mathematics. 309 (1): 1–31. doi:10.1016/j.disc.2007.12.043.
- [3] M. . Rastrigin, L. A. "Systems of extremal control." Mir.
- [4] D. S. H. Mühlenbein and p. . J. Born. "The Parallel Genetic Algorithm as Function Optimizer ". Parallel Computing, 17.