

# Trabalho de Inteligência Computacional

Diego de Oliveira  
Escola de Artes, Ciências e Humanidades (EACH)  
Universidade de São Paulo (USP)  
Email: diego@diegooliveira.com

**Resumo**—Iremos demonstrar o uso de algoritmos genéticos na resolução de uma série de funções matemáticas, apresentando os resultados comparativos de várias parametrizações.

## I. INTRODUÇÃO

Esse trabalho tem como objetivo apresentar o uso de algoritmos genéticos (AG) na resolução de uma série de funções matemáticas, exercitando o conteúdo apresentado em sala, complementando quando necessário. Será usado a função de Rastrigin como problema de maximização enquanto o de minimização será feita com funções de alta dimensionalidade - unimodais e multimodais.

No AG serão utilizados os operadores de mutação, seleção e cruzamento, aplicando valores diferentes para as chances de utilização de cada um deles. Iremos avaliar o uso de torneio e roleta para o processo de cruzamento além do uso de elitismo na evolução das gerações dos indivíduos.

A forma escolhida para representação do cromossomo é um fator importante na implementação dos algoritmos genéticos tanto no quesito de performance, temporal e espacial, quanto na capacidade de encontrar o ótimo global. Nas sessões seguintes serão discutidos formas de representação do cromossomo para os problemas tratados, explicando vantagens e desvantagens das possíveis abordagens.

Para elaboração do trabalho foram usados uma série de ferramentas. O código do AG foi desenvolvido em Rust, os gráficos foram gerados com *scripts* R e foram usados alguns *scripts* bash pra executar os testes. Rust é uma linguagem de programação elaborada pela Mozilla Foundation e foi escolhida devido ao conhecido desafio de performance dos AGs.

## II. OPERADORES GENÉTICOS

Existe uma grande quantidade de operadores que podem ser aplicados no AG tanto para seleção, cruzamento e mutação. Foram codificados para o trabalho os seguintes operadores:

### A. Seleção

Os operadores de seleção são usados na etapa escolha do par de indivíduos que irão participar da etapa de cruzamento.

- **Torneio** - Simula o processo natural de torneio. Deve ser definido um parâmetro  $n$  de amostras que serão escolhidas aleatoriamente para duelarem entre si, os dois mais aptos são selecionados pra cruzamento, os demais são reestabelecidos a população.
- **Roleta** - Processo de seleção que simula um a roleta da sorte onde os indivíduos com maior aptidão possuem maior chance de serem sorteados.

### B. Cruzamento

O operador de cruzamento é responsável por mesclar dois indivíduos. Foram implementados os seguintes operadores:

- **Cruzamento de um ponto** - Esse operador de cruzamento cria dos filhos, um com parte dos cromossomos do primeiro pai até um ponto  $x$  e o resto com cromossomos do segundo pai, e outro com o no sentido inverso.
- **Cruzamento de dois pontos** - Parecido com o primeiro cruzamento porém são selecionados dois pontos de cruzamento.

### C. Mutação

O operador de mutação é responsável por adicionar variabilidade na população de indivíduos. Foram implementados os seguintes operadores:

- **Mutação simples** - Escolhe aleatoriamente um ponto do cromossomo e troca por um valor aleatório.

## III. MODELO DE AVALIAÇÃO

Foram feitas várias execuções para avaliar a parametrização e os operadores aplicados aos problemas abordados nesse trabalho. Inicialmente os valores de mutação e cruzamento foram escolhidos empiricamente, em seguida foram feitos testes para validar sua estabilidade, ajustando quando necessário.

Durante a avaliação dos parâmetros executamos testes com o valor de mutação, variando de 0 a 100, incrementando 5 por vez. Para cada valor de mutação rodamos o AG cinco vezes para avaliar a estabilidade do valor encontrado. Nas funções de maximização iremos tomar o menor valor da série de cinco, nas de minimização o maior, como o as funções são conhecidas será possível a estabilidade.

Para todos os problemas foi usado como critério de parada o número máximo de iterações. Os valores foram ajustados de acordo com os parâmetros usados no AG, formato do cromossomo e sua estabilidade.

## IV. A FUNÇÃO RASTRIGIN

A função de Rastrigin 1 foi um problema elaborada especificamente para teste de performance em algoritmos de maximização, seu principal desafio se da por conta do amplo espaço de busca e pela existência de vários mínimos locais como pode ser observado na figura 1.

$$f(\mathbf{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)] \quad (1)$$

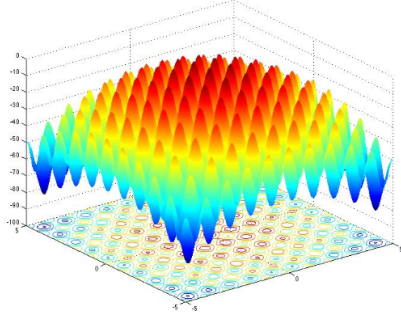


Figura 1. Função Rastrigin para duas variáveis

Para o trabalho foi proposto a inversa dessa função, o que irá transformar em um problema de maximização, com parâmetros  $n = 2$ ,  $A = 10$  e  $x_i \in [-5, 5]$ . Para esse conjunto de parâmetros o valor máximo já é conhecido, vale 0.

#### A. Modelagem da solução

Para esse problema foi comparado a performance e precisão da solução ao representar o cromossomo como uma arranjo de pontos flutuantes ou um conjunto de bits.

Ao representar o cromossomo como um arranjo de pontos flutuantes de duas posições, onde a posição zero guarda o valor de  $x_1$  e o índice um  $x_2$ , temos um modelo mais direta e natural ao problema.

Na representação como um conjunto de bits será necessário uma codificação que permita representar valores de ponto flutuante. Nos testes foram usados trinta e dois bits para representação de  $x$ , dezesseis bits para cada  $x_i$ . A função de transformação para o espaço de busca será feito conforme a equação 2, dessa forma o erro será de 0,000015259.

$$g(x) = -5 + \frac{x}{2^{16}} * 10 \quad (2)$$

Para seleção do melhor formado do cromossomo e operador de seleção foi feito uma rodada com parâmetros escolhidos empiricamente. A análise dos resultados mostrou que o formato binário tem melhor taxa de convergência para o ponto correto de maximização. Quanto aos algoritmos de seleção, roleta ou torneio, é possível observar que o seletor de torneio se sai melhor. Os melhores resultados estão apresentados na figura 2, a figura 5 apresenta os resultados para outras combinações de formatos e operadores de seleção.

#### B. Estabilidade dos parâmetros

As figuras 3 e 4 mostram o resultado da série de cinco execuções nas variações do valor de mutação. Como pode ser observado, sem elitismo o valor máximo é encontrado sempre que o percentual de mutação é maior que 30, com elitismo em algumas execuções ele fica preso em um máximo local.

#### C. Resultados

A tabela I apresenta os melhores parâmetros encontrados para a resolução desse problema.

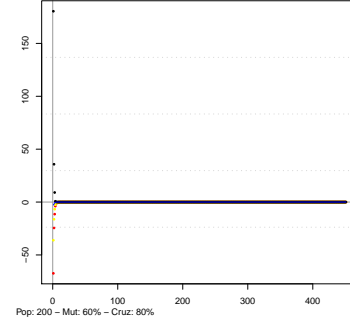


Figura 2. Distribuição do *fitness* nas gerações. A linha azul mostra o maior *fitness*, o vermelho o menor, a amarela o médio e o preto o desvio padrão

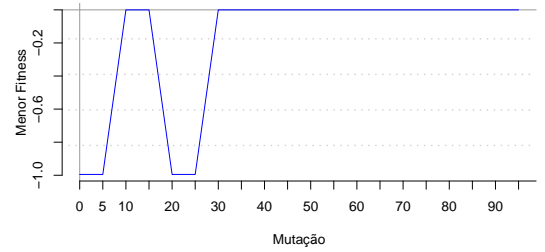


Figura 3. Fitness máximo para variação do percentual de mutação e cruzamento sem elitismo

## V. FUNÇÃO UNIMODAL

Uma função é denominada unimodal quando existe um valor  $m$  tal que  $f(x)$  cresce quando  $x \leq m$  e diminui quando  $x \geq m$ . Essa propriedade faz com que o ponto de máxima da função  $f(x)$  seja  $f(m)$ . Para o estudo foram usados as funções 3(FU1) e 4(FU2) com  $D = 30$  e espaço de busca  $[-100, 100]$ .

$$f(x) = \sum_{i=1}^D x_i^2 \quad (3)$$

$$f(x) = \sum_{i=1}^D (|x_i + 0.5|)^2 \quad (4)$$

Observando as funções é possível dizer empiricamente seus pontos de mínimos. Para a função FU1 será 0 e para FU2 será 7.5.

#### A. Modelagem da solução

Foram feitos testes representando o cromossomo como um arranjo de inteiros ou vetor binário.

A representação em arranjo utilizou-se de 30 posições na memória, uma para cada valor.

A codificação do cromossomo no formato binário é mais fácil uma vez que não há a necessidade de armazenar valores de ponto flutuante. É possível representar cada valor do espaço

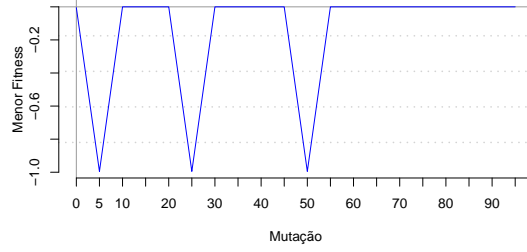


Figura 4. Fitness máximo para variação do percentual de mutação e cruzamento com elitismo

Tabela I  
MELHORES PARÂMETROS PARA A FUNÇÃO RASTRIGIN

Parâmetro	Valor
população	200
gerações	450
elitismo	Não
cruzamento	80%
mutação	60%
seleção	Torneio
formato do cromossomo	Binário

de busca com 10 bits, como  $D = 30$  serão necessários 300 bits, porém não há um tipo primitivo desse tamanho em Rust, dessa forma escolhemos por implementá-la com uso de uma arranjo de *boolean*(formato binário). Foi necessário a implementação de uma função de conversão do formato binário para o decimal, mas dessa vez sem perda de precisão.

Nos testes executados o formato de cromossomo que se comportou melhor foi o arranjo de 30 posições para as duas funções.

A FU1 encontrou o ponto mínimo mais rápido usando torneio com uso de elitismo como demonstrado na figura 6 e 7. O uso de elitismo não é decisivo para localização do mínimo global, porém trabalha para a manutenção dele quando encontrado.

#### B. Estabilidade dos parâmetros

Para avaliar a estabilidade na busca da solução foi executado o teste em uma série de rodadas como descrito anteriormente. O resultado é apresentado na figura 8 e 9.

A mutação contribui para a solução do problema em qualquer ponto a partir de 40% para qualquer uma das duas funções e ela é estável em encontrar a resposta em uma série de rodadas.

#### C. Resultados Unimodal

A tabela II apresenta os melhores parâmetros encontrados para a resolução das funções FU1 e FU2.

Tabela II  
MELHORES PARÂMETROS PARA AS FUNÇÕES UNIMODAIS

Parâmetro	Valor
população	250
gerações	600
elitismo	Sim
cruzamento	80%
mutação	55%
seleção	Torneio
formato do cromossomo	Arranjo

## VI. FUNÇÃO MULTIMODAL

Uma função multimodal possui vários máximos locais, como por exemplo a função Rastrigin. Nessa sessão iremos abordar a função 5(FM1).

$$f(x) = \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|}) \quad (5)$$

A grande diferença em relação a função de Rastrigin se da pela quantidade de universos e o espaço de busca. Para esse problema teremos  $D = 30$  com espaço de busca  $[-500, 500]$

#### A. Modelagem da solução

Novamente para esse problema foi feito a comparação entre a representação do cromossomo como um arranjo de 30 posições ou um binário 30 conjunto de 10 bits.

A representação como arranjo se mostrou melhor em todos os testes. A figura 10 apresenta os melhores resultados encontrados.

#### B. Estabilidade dos parâmetros

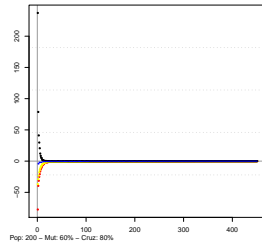
Observando a figura 11 é possível notar que os parâmetros selecionados se apresentaram estáveis em uma série de execuções, além do que a mutação não se mostrou diferencial na busca da solução.

#### C. Resultado Multimodal

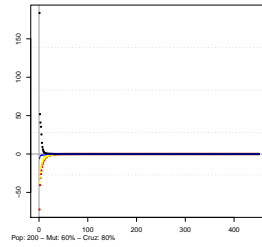
O uso de elitismo ajuda na busca pela solução mas não é um diferencial.

Tabela III  
MELHORES PARÂMETROS PARA A FU1

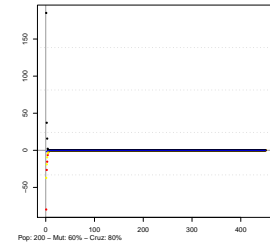
Parâmetro	Valor
população	250
gerações	500
elitismo	Sim
cruzamento	90%
mutação	30%
seleção	Torneio
formato do cromossomo	Arranjo



(a) Seleção por roleta, cromossomo arranjo



(b) Seleção por roleta, cromossomo binário



(c) Seleção por torneio, cromossomo arranjo

Figura 5. Testes com formato do cromossomo e funções de seleção para a função Rastrigin

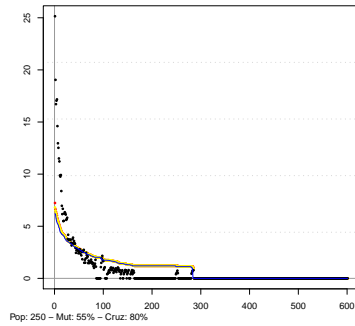


Figura 6. Função fitness para a FU1 com uso de elitismo e arranjo como cromossomo

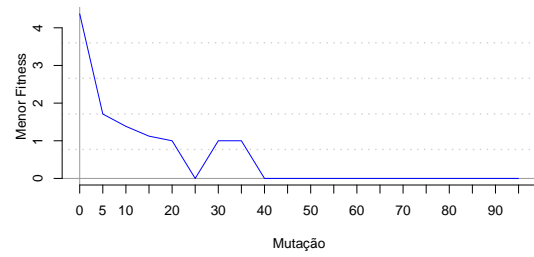


Figura 8. FU1

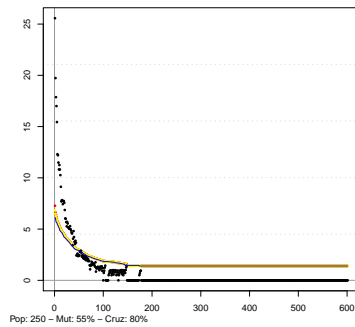


Figura 7. Função fitness para a FU2 com uso de elitismo e arranjo como cromossomo

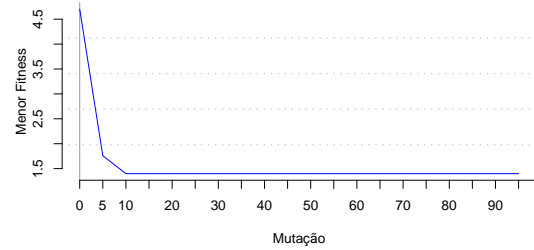


Figura 9. FU2

diferenças significativa na busca da solução.

Seleção por torneio se mostrou melhor para todos os problemas propostos.

## VII. CONCLUSION

A avaliação do uso do Algoritmo Genético nos problemas propostos mostra sua força em problemas de maximização e minimização de funções de forma heurística. É possível perceber que a utilização de formates de cromossomos diferentes podem facilitar na busca da solução e que a combinação dos percentuais de cruzamento e mutação devem ser ajustados de forma fina e iterativa.

A utilização de cruzamento de um ou dois pontos não gera

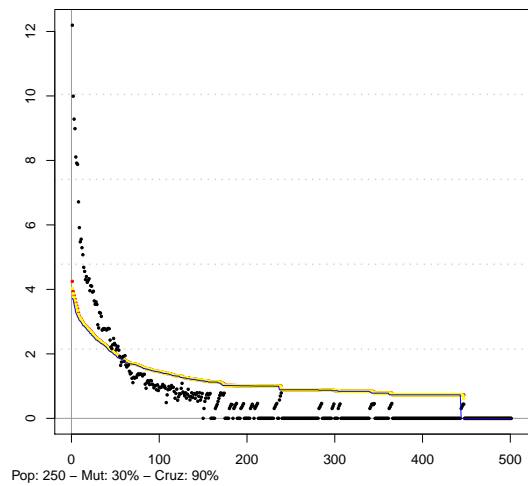


Figura 10. Distribuição do *fitness* entre as gerações para a função multimodal FM1

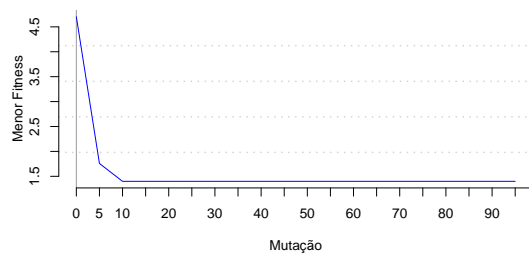


Figura 11. Distribuição do *fitness* entre as gerações para a função FM1