



Evolução Diferencial

Trabalho 08

Computação Evolutiva
at Universidade Federal de Uberlândia

Antonio Fernandes Valadares

15 de fevereiro de 2022

Número de matrícula:
Professor:

11711ECP015
Keiji Yamanaka

Objetivo

O objetivo desse trabalho é minimizar a seguinte função:

$$f(x) = \sum_{i=1}^{D-1} (1 - x_i)^2 + 100(x_i + 1 - x_i^2)^2$$

Com: $x_i \in [-1, 2]$.

Para minimizar essa função vamos utilizar uma técnica conhecida como evolução diferencial, esse é um algoritmo guloso que se baseia nos mecanismos de seleção natural e na genética de populações. Ele otimiza um problema iterando por gerações e melhorando a aptidão da sua população. Utiliza operadores de cruzamento, seleção e mutação.

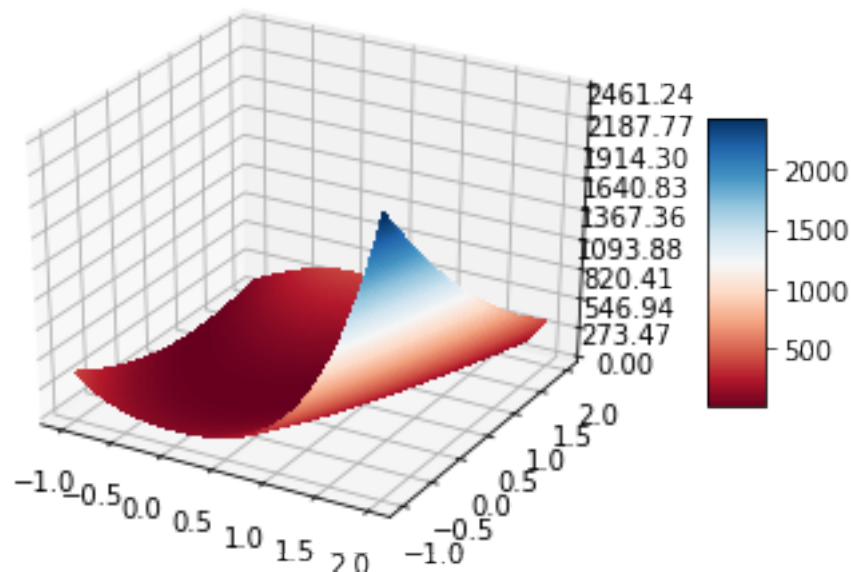


Figura 1: Função a ser minimizada

A função tem seu ponto mínimo no ponto $x_i = 0$ para todo i .

Construção do algoritmo

O algoritmo foi construído utilizando um notebook python.

Geração da população inicial

Para que nosso algoritmo funcione, primeiro é necessário gerar uma população inicial, essa população será o ponto inicial do algoritmo. A função responsável gera uma população aleatória inicial de um determinado tamanho configurado pelo usuário e com o tamanho especificado de x , ou seja, quantos elementos o array x possuirá.

Aptidão da população

Para o cálculo da aptidão dos indivíduos da população foi utilizado a própria função a ser minimizada, dessa forma cada indivíduo é aplicado a função e quanto menor seu valor, mais apto o indivíduo é. Logo podemos definir nossa função custo ou função objetivo como a própria função.

Crossover, Mutação e Seleção.

A nova população é gerada com base na teoria da evolução diferencial, para cada indivíduo(x) da população é selecionado outros 3 indivíduos aleatórios (x_{r1}, x_{r2}, x_{r3}). A partir disso um novo indivíduo é gerado a partir da seguinte função:

$$v = x_{r1} + F(x_{r2} - x_{r3}).$$

Onde F é um parâmetro configurável pelo usuário. Esse processo representa a mutação na evolução diferencial. Após isso é necessário realizar o cruzamento do indivíduo mutado com o indivíduo em questão x . O processo ocorre da seguinte forma:

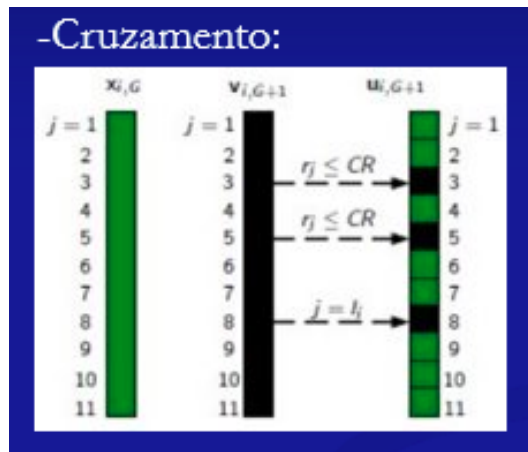


Figura 2: Crossover na evolução diferencial.

Um novo indivíduo u é gerado, onde um de seus valores é sempre de v e os outros valores é aplicados uma taxa de CR para valores de x e $1 - CR$ para v . CR também um parâmetro configurável do algoritmo.

A seleção é feita escolhendo o indivíduo mais apto, entre x e v , dessa forma o indivíduo que tiver um valor mais baixo na função objetivo é passado para a próxima geração.

Resultados

Foram realizados vários testes, e ps resultados foram bons para os parâmetros sugeridos pelo trabalho de um tamanho da população(NP)=100, CR=0.9 e F=0.5.

Para um vetor x com apenas dois valores e rodando o algoritmo por 1000 gerações tivemos os seguintes resultados:

	x0	x1	fitness
0	1.009193	1.016389	0.000517
1	0.953101	0.908525	0.002201
2	1.049695	1.102604	0.002525
3	1.052737	1.110783	0.003420
4	0.985832	0.979093	0.005425

Figura 3: Melhores 5 indivíduos gerados pela execução do algoritmo.

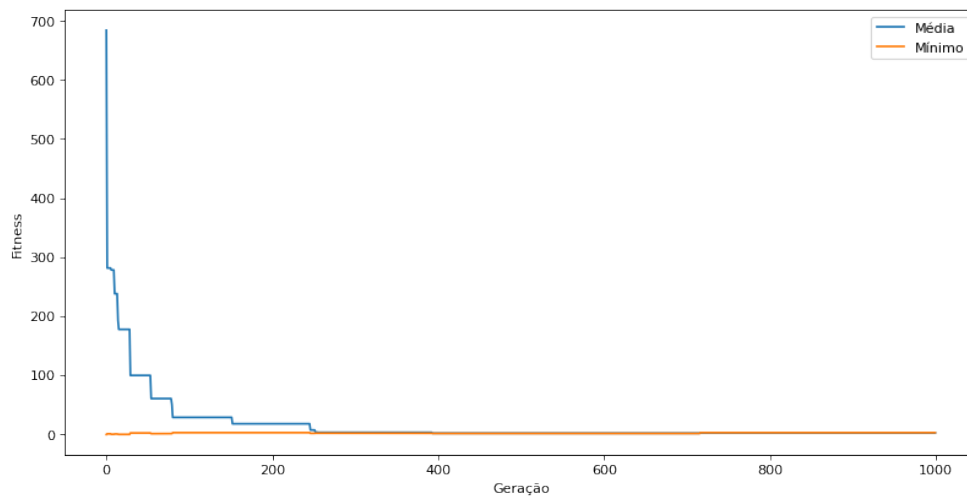


Figura 4: Gráfico de desempenho do algoritmo.

Para mais valores no vetor x tivemos resultados bons, mas logicamente piores.

	x0	x1	x2	x3	x4	fitness
0	0.997723	0.995884	0.991551	0.985007	0.970836	0.000715
1	1.002225	1.005118	1.010494	1.020359	1.042674	0.000897
2	1.002551	1.005953	1.010580	1.019613	1.038129	0.001290
3	0.995435	0.989953	0.982267	0.966930	0.936144	0.002704
4	0.998269	0.998898	0.997498	0.990061	0.979851	0.003128

Figura 5: Melhores 5 indivíduos gerados por 10000 gerações