

Minimização da função Sphere De Jong's

William Sdayle Marins Silva

Abril 2019

1 Introdução

Para resolução da atividade número 2 da matéria de Tópicos em Inteligência Computacional, foram utilizadas a linguagem python 3 para desenvolvimento dos algoritmos necessários e a IDE pycharm. Visando otimizar a função (1), mais conhecida como função esfera, a mais simples das funções de De Jong, é contínua, convexa, unimodal, lisa, simétrica e possui o mínimo $f(0) = 0$, tomando como princípio que computação evolutiva se interessa em otimização há muitos anos. [1]

$$\sum_{x=1}^N x^2 \quad (1)$$

[2] classifica a função (1) como classe (a), unimodal, convexa e multidimensional, afirmando que são "enganosas", causando lentidão ou lentidão para convergencia global.

2 Desenvolvimento

Foram definidos valores dos intervalos de cada gene como $[-5.12, 5.12]$. Passos para criação do algoritmo

1. Codificação dos cromossomos
2. Método de Crossover
3. Método de Mutação
4. Cálculo do fitness
5. Nova Geração

2.1 Codificação dos cromossomos

A maneira de codificação escolhido para o trabalho foi por números reais, tratando-se da otimização para o mínimo global de uma função somatória, foi preferido utilizar valores reais, até para ser mais pratico verificar o quão boa está sendo o algoritmo para a otimização da função, sem ser necessário a transformação de valores, como seria preciso caso fosse uma codificação com números binários.

2.2 Método de Crossover

O método de crossover escolhido foi aritmético, visando uma maior diversificação dos valores obtidos. Ao iniciar a geração, todos os genes são preenchidos com valores aleatórios trazidos de uma função específica de uma biblioteca python, para ser possível uma melhor diversidade entre os indivíduos foi utilizado o algoritmo de crossover aritmético.

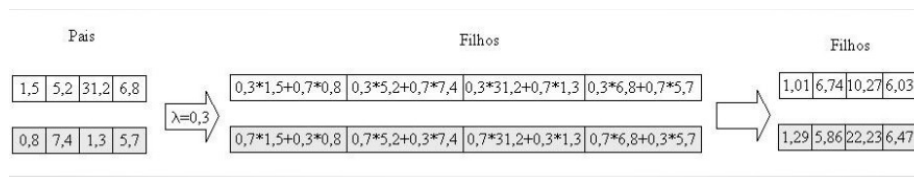
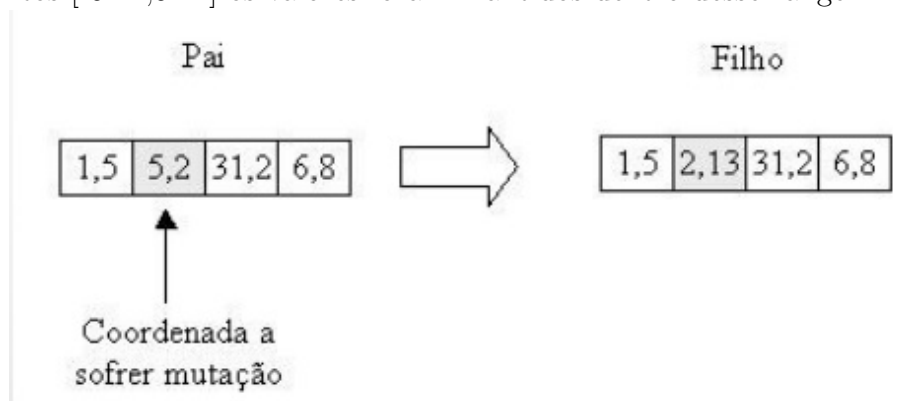


Figura 1: Processo de crossover aritmético.

2.3 Método de Mutação

A mutação utilizada na solução do algoritmo foi a mutação real aleatória, o processo dela consiste em escolher um valor aleatório em uma posição aleatória do cromossomo e troca-lo por um outro valor aleatório, como no caso da sphere o valor deveria ser entre os pontos $[-5.12, 5.12]$ os valores foram mantidos dentro desse range.

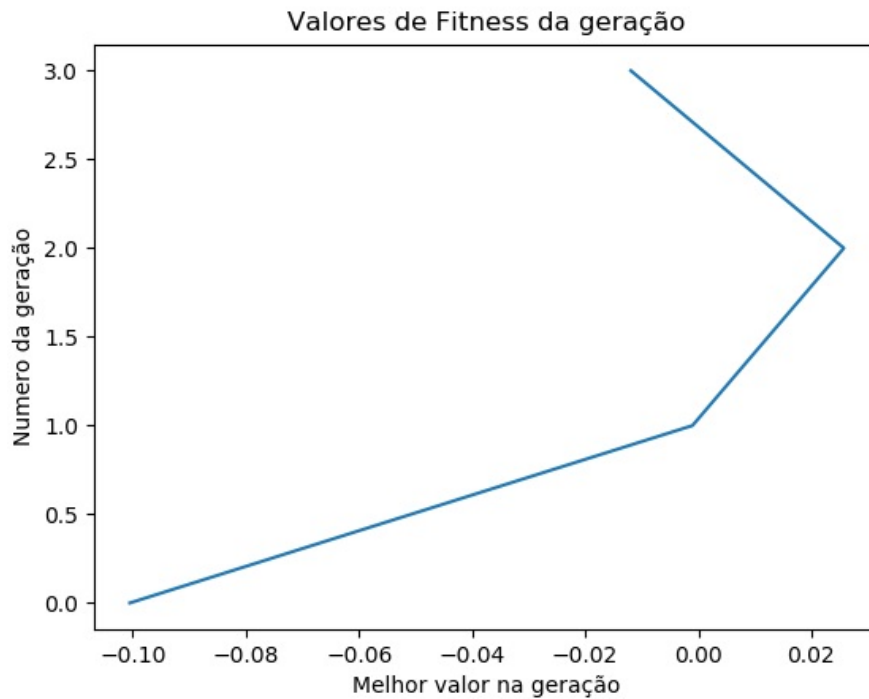


2.4 Cálculo do fitness

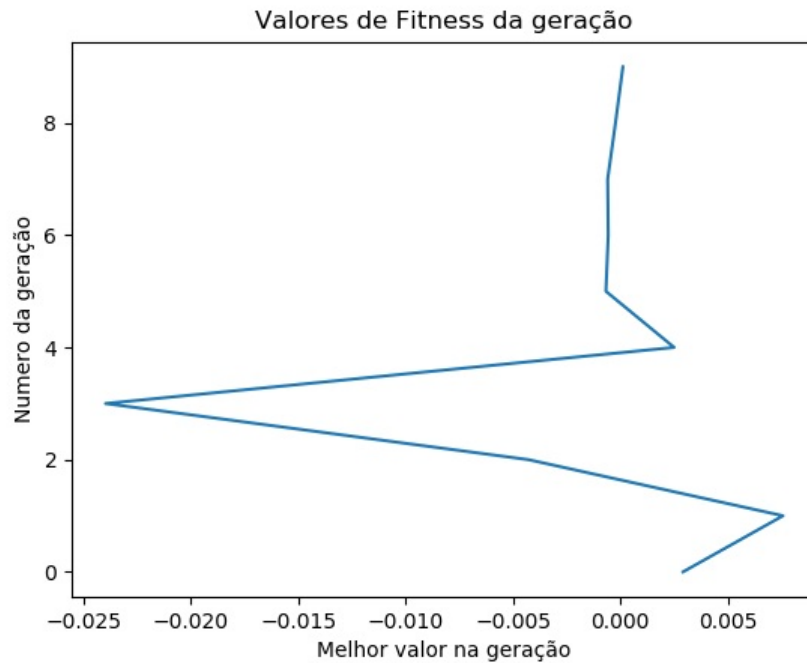
Cálculo do fitness foi feito baseado na função De Jong's mostrado na função (1), visando aumentar o valor da função, a medida que o valor da função vai aumentando, mais perto de zero o resultado chega.

$$1 / \sum_{x=1}^N x^2 + 1 \quad (2)$$

3 Valores obtidos



Esse gráfico indica os valores obtidos de acordo com as gerações, correspondente a uma geração de 12 indivíduos, realizado o algoritmo em 4 iterações.



Esse gráfico indica os valores obtidos de acordo com as gerações, correspondente a uma geração de 100 indivíduos, realizado o algoritmo em 10 iterações. A cada iteração é formada uma nova geração, guardando o melhor cromossomo para apresentação em um gráfico.

Referências

- [1] V. Jakob, T. Rene, *A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems*
- [2] M. Marcin, S. Czesław, *Test functions for optimization needs*