



UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

ENGENHARIA ELÉTRICA

COMPUTAÇÃO EVOLUCIONÁRIA

Relatório 3 – Evolução Diferencial

Alunos:

Gabriel Saraiva Espeschit - 2015065541

04 de novembro de 2019

1. Introdução

Algoritmos de evolução diferencial são métodos de otimização que utilizam de representações e operações vetoriais para encontrar mínimos e máximos das funções objetivos. Esse tipo de algoritmo segue o pseudocódigo a seguir:

$t \leftarrow 1$

Inicializar população $X_t = \{x_{t,i}; i = 1, \dots, N\}$

Enquanto algum critério de parada não for satisfeito, **faça**:

Para $i = 1$ até N , **faça**:

 Selecione aleatoriamente $r1, r2, r3 \in \{1, \dots, N\}$

 Selecione aleatoriamente $\delta_i \in \{1, \dots, N\}$

Para $j = 1$ até n , **faça**:

Se $\mu_{[0,1]} \leq C$ **E** $j == \delta_i$, **então**:

$u_{t,i,j} = x_{t,r1,j} + F * (x_{t,r2,j} - x_{t,r3,j})$

Senão:

$u_{t,i,j} = x_{t,r1,j}$

Fim se

Fim para

Se $f(u_{t,i}) \leq f(x_{t,i})$, **então**:

$x_{t+1,i} \leftarrow u_{t,i}$

Senão:

$x_{t+1,i} \leftarrow x_{t,i}$

Fim se

Fim para

Fim enquanto

Sendo assim, foram selecionadas duas funções, *rastrigin* e *peaks*, para serem submetidas a um algoritmo de evolução diferencial.

2. Metodologia

A metodologia de implementação consistiu na tradução das funções repassadas pelo professor para linguagem Python, e da implementação, em Python, do pseudocódigo acima apresentado.

3. Resultados

Para as cada uma das funções testadas, o resultado ótimo esperado era:

- *Peaks*: mínimo global em $x^* = [0.228, -1.625]$ com $f(x^*) = -6.5511$;
- *Rastrigin*: mínimo global em $x^* = [0, 0]$ com $f(x^*) = -20$.

Abaixo algumas imagens que mostram as gerações 0, 50 e 100 do algoritmo convergindo na função *peaks*:

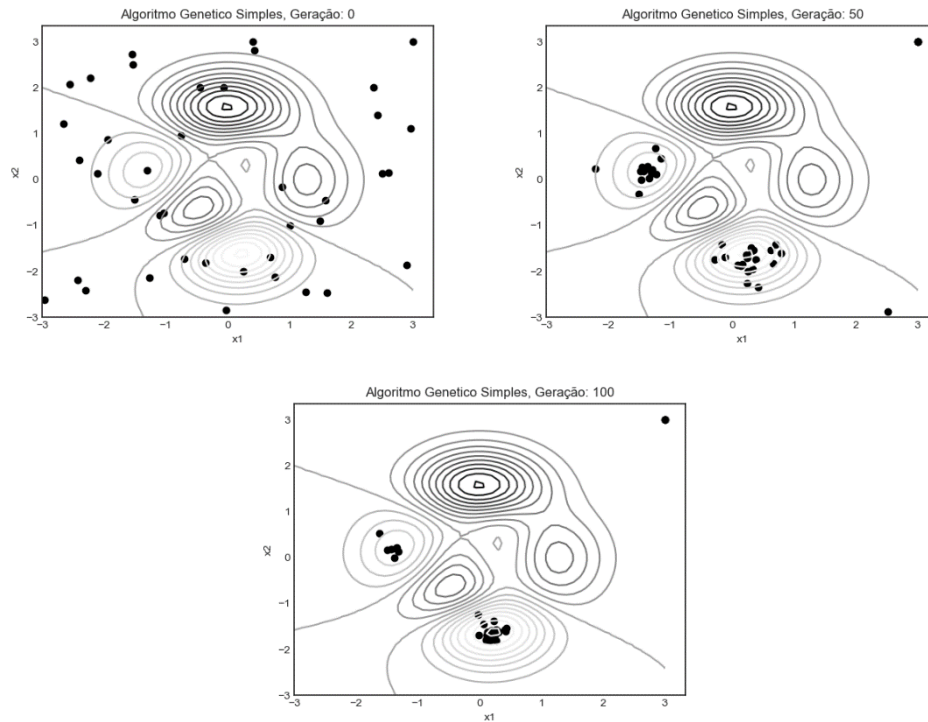


Figura 1 - Gráfico mostrando a convergência do algoritmo na função peaks.

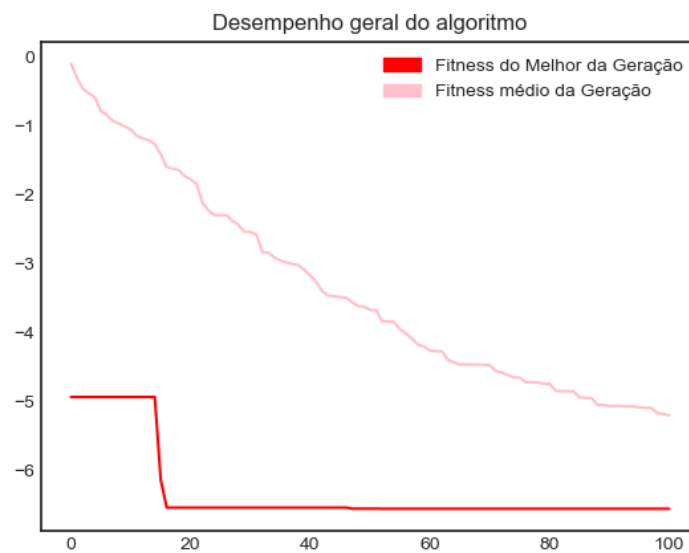


Figura 2 - Desempenho do algoritmo. A convergência no caso média mostra que o comportamento do algoritmo foi conforme esperado.

Abaixo algumas imagens que mostram as gerações 0, 50 e 100 do algoritmo convergindo na função *rastrigin*:

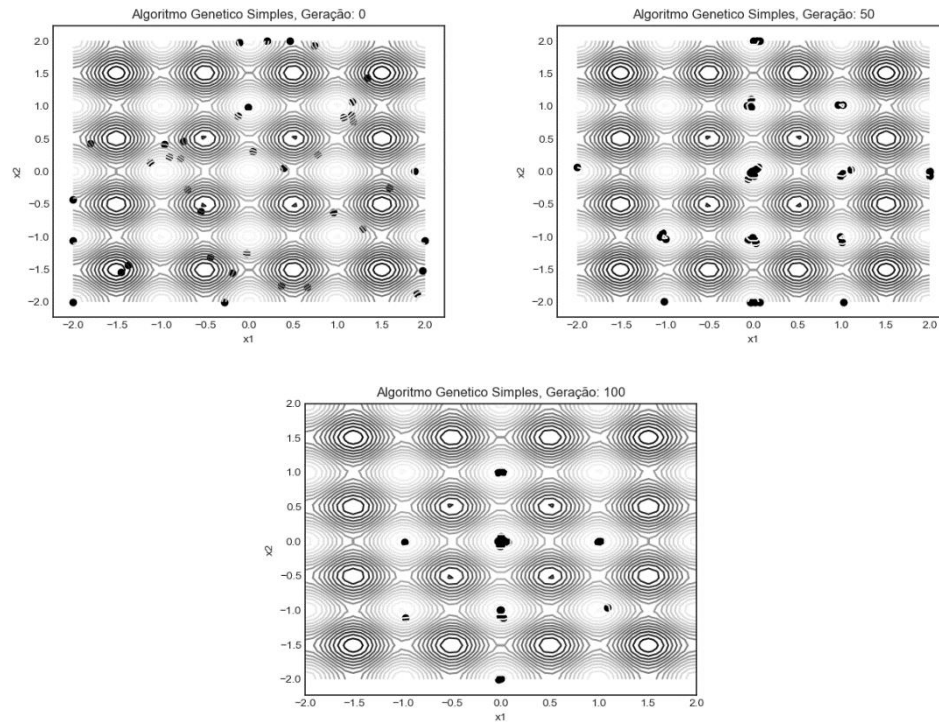


Figura 3 - Gráfico mostrando a convergência do algoritmo na função rastrigin.

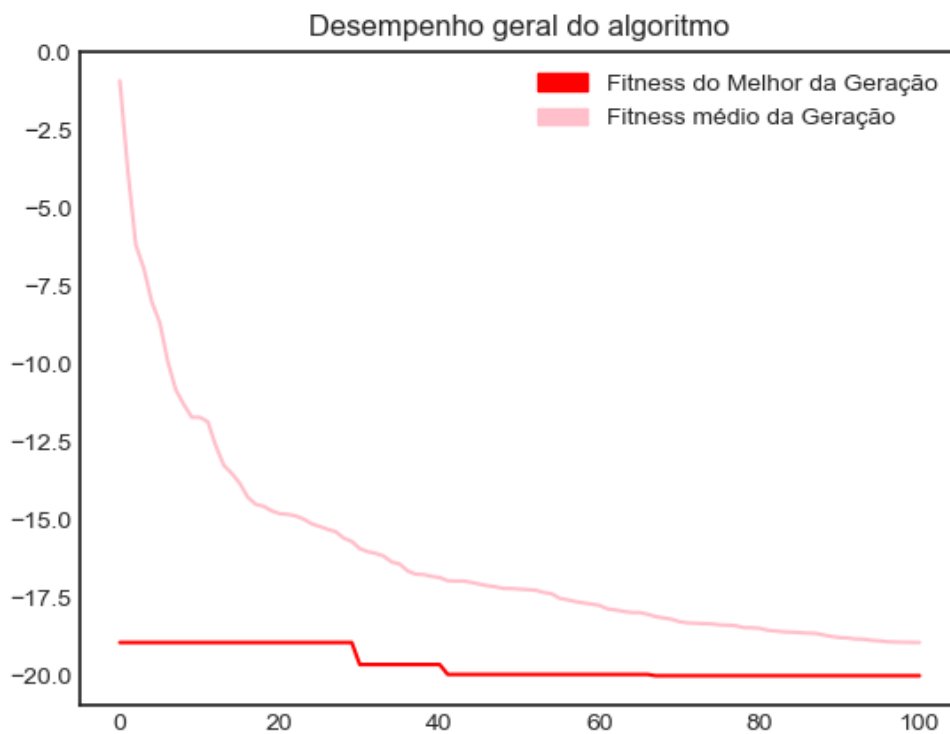


Figura 4 - Desempenho do algoritmo. A convergência no caso média mostra que o comportamento do algoritmo foi conforme esperado.

4. Código

O código desenvolvido em Python foi anexado na pasta zip enviada.