

Estratégias Evolutivas EEs

Prof. Juan Moisés Mauricio Villanueva

jmauricio@cear.ufpb.br

www.cear.ufpb.br/juan

P₀

P₁

P_{final}

X₁
X₂
...
x₂₀

X₁
X₂
...
x₂₀

X₁
X₂
...
x₂₀

X₁
X₂
...
x₂₀

fitness

Estratégias Evolutivas

- Desenvolvidas por Rechenberg e Schwefel, e estendida por Herdy, Kursawe e outros autores, foram inicialmente propostas com o objeto de solucionar problemas de **otimização** de parâmetros, tanto discretos como contínuos.

Estrutura

Algoritmo EE's

Início

- $t \leftarrow 0$;
- Inicialize $P(t)$;
- Avalie $P(t)$;

Enquanto (não condição de parada) Faça

Início

- $t \leftarrow t + 1$;
- Selecione $P(t)$ a partir de $P(t-1)$;
- Cruzamento $P(t)$;
- Mutação $P(t)$;
- Avalie $P(t)$;

Fim

- Retorna a melhor solução;

Fim

Codificação da População

- Em contraste com a representação binária dos indivíduos em *algoritmos genéticos*, as **estratégias evolutivas** são diretamente baseadas em representações com vetores de **valores reais**.
- Os algoritmos evolutivos com codificação real são geralmente aplicados para resolver problemas de otimização:
 - multi-variável
 - não lineares
 - com restrições
 - com máximos ou mínimos locais e/ou globais.

Codificação da População

- Em problemas de otimização restrita, a codificação adotada pode fazer com que **indivíduos** modificados pelos operadores de cruzamento e mutação sejam **inválidos**.
- Nestes casos, cuidados especiais devem ser tomados na definição da codificação e na aplicação dos operadores, observando-se as **restrições** no domínio das variáveis.

$$x \in [x_{\min}, x_{\max}]$$

$$x_{\text{random}} = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}$$

rand é um número aleatório entre [0, 1]

$$x \in [x_{\min}, x_{\max}]$$

$$x_{\text{random}} = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}$$

rand é um número aleatório entre [0, 1]

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

```
xmin = 10
xmax = 20
xrand = xmin + (xmax-xmin)*np.random.rand(3,1)
xrand
```

```
array([[12.91908555],
       [15.39193245],
       [10.98211015]])
```

$$x \in [x_{\min}, x_{\max}]$$

$$x_{\text{random}} = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}$$

rand é um número aleatório entre [0, 1]

```
xmin = 10
xmax = 20
ymin = 5
ymax = 8
xrand = xmin + (xmax-xmin)*np.random.rand(5,1)
yrand = ymin + (ymax-ymin)*np.random.rand(5,1)
Po = np.zeros((5,2))
Po[0:5,0] = xrand[0:5,0]
Po[0:5,1] = yrand[0:5,0]
Po
```

```
array([[18.88289922,  7.23484432],
       [11.24538064,  6.89262497],
       [16.73466933,  5.55250999],
       [16.47313038,  6.58827317],
       [19.86690649,  6.50022079]])
```

X

Y

$$Z = f(x,y)$$

```
Po = [x11  y11
      x21  y21
      x31  y31
      x41  y41
      x51  y51]
```

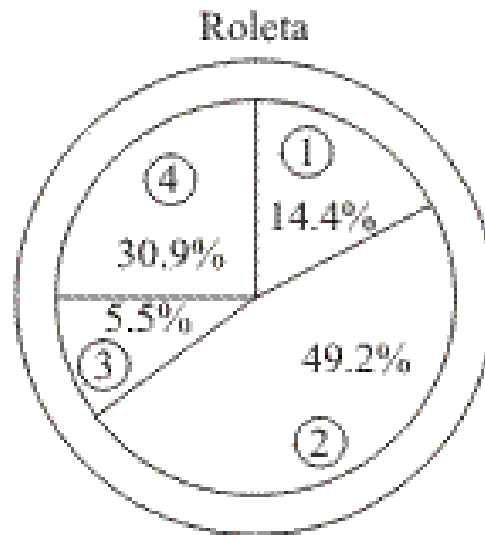

Seleção Proporcional ao Fitness

- Também conhecido como roleta ou *Roulette Wheel*, atribui-se a cada indivíduo de uma população uma probabilidade de passar à próxima geração proporcional à sua aptidão ou *fitness*.
- Indivíduos com maior valor do *fitness* terão maior probabilidade de passar à próxima geração.
- Este método pode fazer com que **indivíduos bem adaptados sejam perdidos**, ou seja, não passem para a próxima geração, isto devido a que este método é probabilístico.

Seleção Proporcional ao Fitness

- Para uma população de n indivíduos, na qual o i -ésimo indivíduo tem associado a ele uma medida de aptidão positiva e não nula, a probabilidade deste indivíduo ser selecionado é dada por:

$$Prob_{xi} = \frac{fitness(x_i)}{\sum_{j=1}^n fitness(x_j)}$$



Nº	String	<i>fitness</i>	%Total
$F(h_1')$	01101	169	14.4
$F(h_2')$	11000	576	49.2
$F(h_3')$	01000	64	5.5
$F(h_4')$	10011	361	30.9
Total		1100	100

Operador de Cruzamento

- É responsável pelo intercâmbio de informação genética entre os indivíduos de uma população, produzindo novas soluções candidatas ou potencias com algumas características já existentes dos pais.
- O processo de cruzamento depende da escolha dos pares (ou grupos) de indivíduos. A probabilidade de ocorrência de recombinação entre dois indivíduos de uma população é denominada *taxa de cruzamento* que varia entre 0,5 e 1,0.

Operador de Cruzamento

- No entanto, uma **alta probabilidade** de cruzamento faz com que indivíduos com uma maior aptidão, sejam eliminados antes que o processo de seleção possa produzir aperfeiçoamento.
- Por outro lado, uma **baixa probabilidade** de cruzamento pode convergir lentamente devido à baixa taxa de exploração das características genéticas.

Operador de Cruzamento Aritmético

- Operador de cruzamento desenvolvidos para uso com codificação em ponto flutuante.
- Este operador esta definido como uma **combinação linear** de dois vetores (cromossomos): sejam x_1 e x_2 dois indivíduos selecionados para cruzamento, então os dois filhos resultantes serão:

$$\begin{aligned}x'_1 &= a.x_1 + (1-a).x_2 \\x'_2 &= (1-a).x_1 + a.x_2\end{aligned}$$

sendo a uma variável aleatória definida no intervalo $a \in [0, 1]$.

Operador de Cruzamento Aritmético

- Este operador é particularmente apropriado para problemas de otimização numérica com restrições, onde a região factível é convexa.
- Isto porque, se x_1 e x_2 pertencem à região factível, combinações convexas de x_1 e x_2 serão também factíveis. Assim, garante-se que o operador de cruzamento não gera indivíduos inválidos.

$$F = f(x, y, z)$$

$$x \in [x_{\min}, x_{\max}]$$

$$x_{\text{random}} = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}$$

Indivíduos
Selecionados

x_{11}	y_{12}	z_{13}
x_{21}	y_{21}	z_{23}
x_{31}	y_{32}	z_{33}
x_{41}	y_{42}	z_{43}
x_{51}	y_{52}	z_{53}
x_{61}	y_{62}	z_{63}

 \Rightarrow

x_{f11}	y_{f12}	z_{f13}
x_{f21}	y_{f21}	z_{f23}
x_{f31}	y_{f32}	z_{f33}
x_{f41}	y_{f42}	z_{f43}
x_{f51}	y_{f52}	z_{f53}
x_{f61}	y_{f62}	z_{f63}

Cruzamento

$$x_{f11} = \alpha x_{11} + (1 - \alpha) x_{21}$$

$$x_{f21} = \alpha x_{21} + (1 - \alpha) x_{11}$$

$$y_{f11} = \alpha y_{11} + (1 - \alpha) y_{21}$$

$$y_{f21} = \alpha y_{21} + (1 - \alpha) y_{11}$$

$$z_{f11} = \alpha z_{11} + (1 - \alpha) z_{21}$$

$$z_{f21} = \alpha z_{21} + (1 - \alpha) z_{11}$$

Estrutura

Algoritmo EE's

Início

- $t \leftarrow 0$;
- Inicialize $P(t)$;
- Avalie $P(t)$;

Enquanto (não condição de parada) Faça

Início

- $t \leftarrow t + 1$;
- Selecione $P(t)$ a partir de $P(t-1)$;
- Cruzamento $P(t)$;
- Mutação $P(t)$;
- Avalie $P(t)$;

Fim

- Retorna a melhor solução;

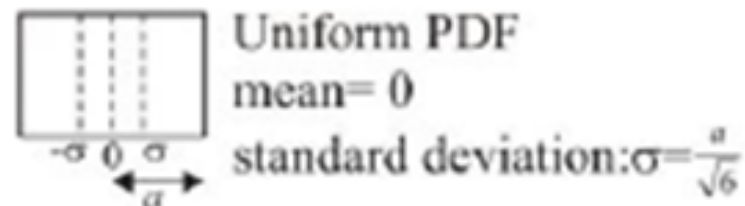
Fim

Operador de Mutação

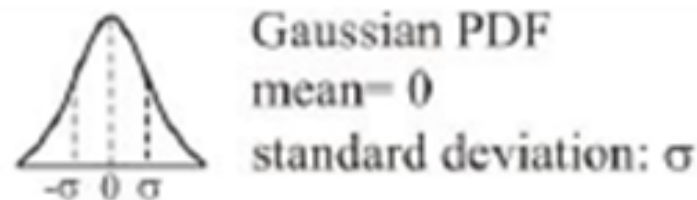
- O operador de mutação modifica aleatoriamente um ou mais genes de um cromossomo.
- A probabilidade de ocorrência de mutação em um gene é denominada *taxa de mutação*. Usualmente, são atribuídos valores pequenos para a taxa de mutação.
- A principal *contribuição* do operador de mutação é *criar uma variabilidade extra na população*, mas sem destruir o progresso já obtido com a busca.

Operador de Mutação

- No caso de problemas com codificação em ponto flutuante, os operadores de mutação mais populares são:
 - mutação uniforme (rand)



- mutação gaussiana (randn)

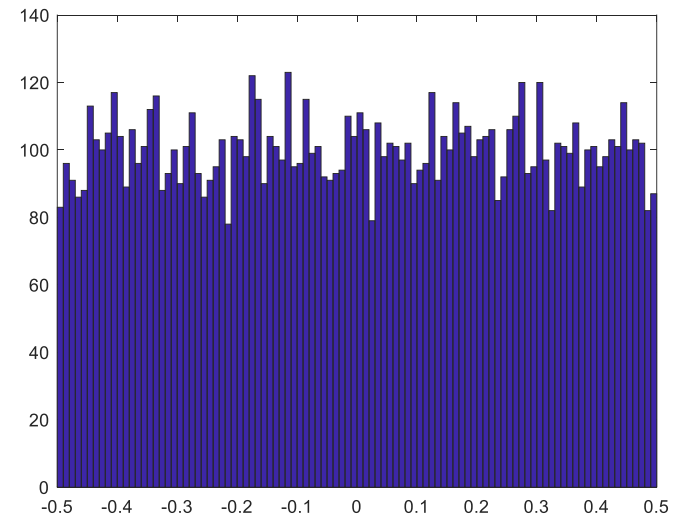


```

clc
close all
clear

%PDF Uniforme
minimo = -0.5;
maximo = 0.5;
desviopadrao = maximo/sqrt(6);
u = minimo + (maximo-
minimo)*rand(1,10000);
figure
hist(u,100)

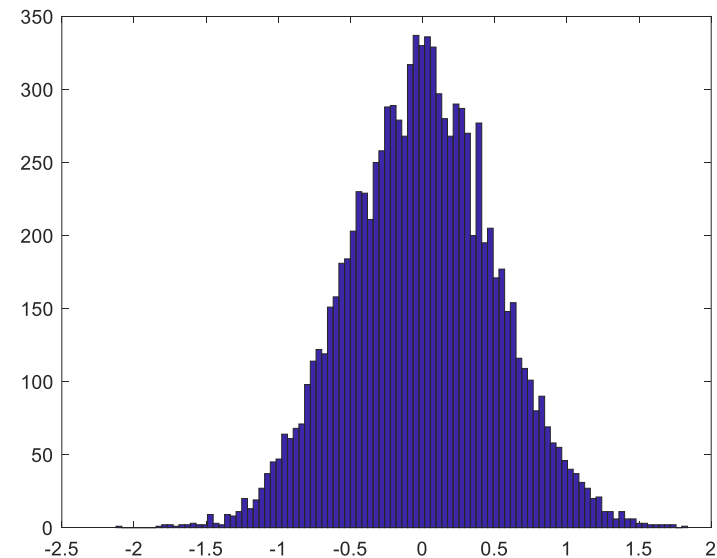
```



```

%PDF Gaussianna
desviopadrao = 0.5;
valormedio = 0;
g = valormedio +
desviopadrao*randn(1,10000);
figure
hist(g,100)

```



Operador de Mutação Uniforme

- O operador mutação uniforme seleciona aleatoriamente um componente $k \in \{1, 2, \dots, n\}$ do cromossomo

$$x = [x_1 \dots \mathbf{x}_k \dots x_n]$$

e gera um indivíduo :

$$x' = [x_1 \dots \mathbf{x}'_k \dots x_n],$$

sendo \mathbf{x}'_k é um número aleatório (com **distribuição de probabilidade uniforme**) amostrado no intervalo:

$$x_k \in [x_{kmin}, x_{kmax}]$$

Operador de Mutação Não Uniforme

- O operador mutação não uniforme foi especialmente desenvolvido para problemas de otimização com restrições e codificação em ponto flutuante, destinada a realizar a sintonia fina aos indivíduos da população.

$x = [x_1 \dots \mathbf{x}_k \dots x_n]$ e gera um indivíduo :
 $x' = [x_1 \dots \mathbf{x}'_k \dots x_n],$



Neste caso, x_k sofrerá uma **variação** para gerar um novo valor. Esta variação dependerá do problema.

Operador de Mutação Gaussiana

- No caso do operador de mutação gaussiana, todos os componentes de um cromossomo $\mathbf{x} = [x_1, \dots, x_n]$ são modificados na forma.

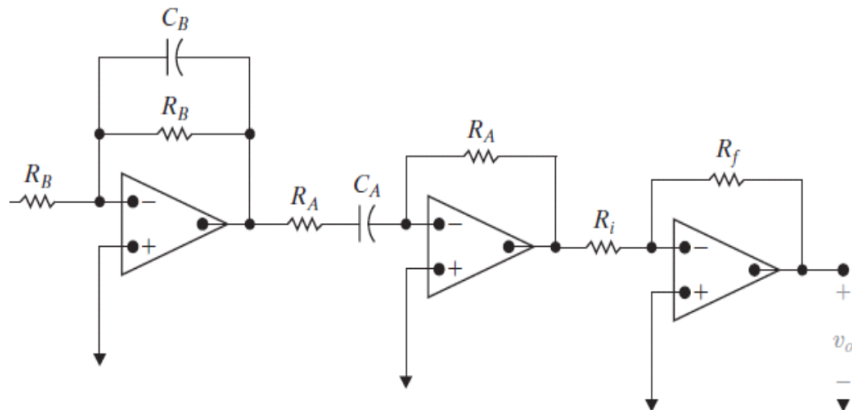
$$\mathbf{x}' = \mathbf{x} + N(0, \sigma)$$

sendo $N(0, \sigma)$ um vetor de **variáveis aleatórias gaussianas independentes**, com média zero e desvio padrão σ .

Exemplo

Exemplo: Projeto de um Filtro Passa-Faixa

- Filtro Passa-Faixa
- Largura de banda [100 Hz , 10 kHz]
- Ganho no estágio $k=2$
- Fator de Qualidade $Q = \frac{\sqrt{f_{corteLOW} \times f_{corteHIGH}}}{f_{corteHIGH} - f_{corteLOW}} = 0,101$



Estratégias Evolutivas



Solução= $[R_A \ R_B \ R_f \ R_i \ C_A \ C_B]$

Exemplo: Projeto de um Filtro Passa-Faixa

- Estrutura do Filtro Passa-Faixa com frequencia de corte em [100 Hz , 10 kHz].

Solução= $[R_A \ R_B \ R_f \ R_i \ C_A \ C_B]$

25

Restrições dos valores dos componentes eletrônicos:

$$R_{\min} = 10 \text{ ohm}$$

$$R_{\max} = 10\text{k ohm}$$

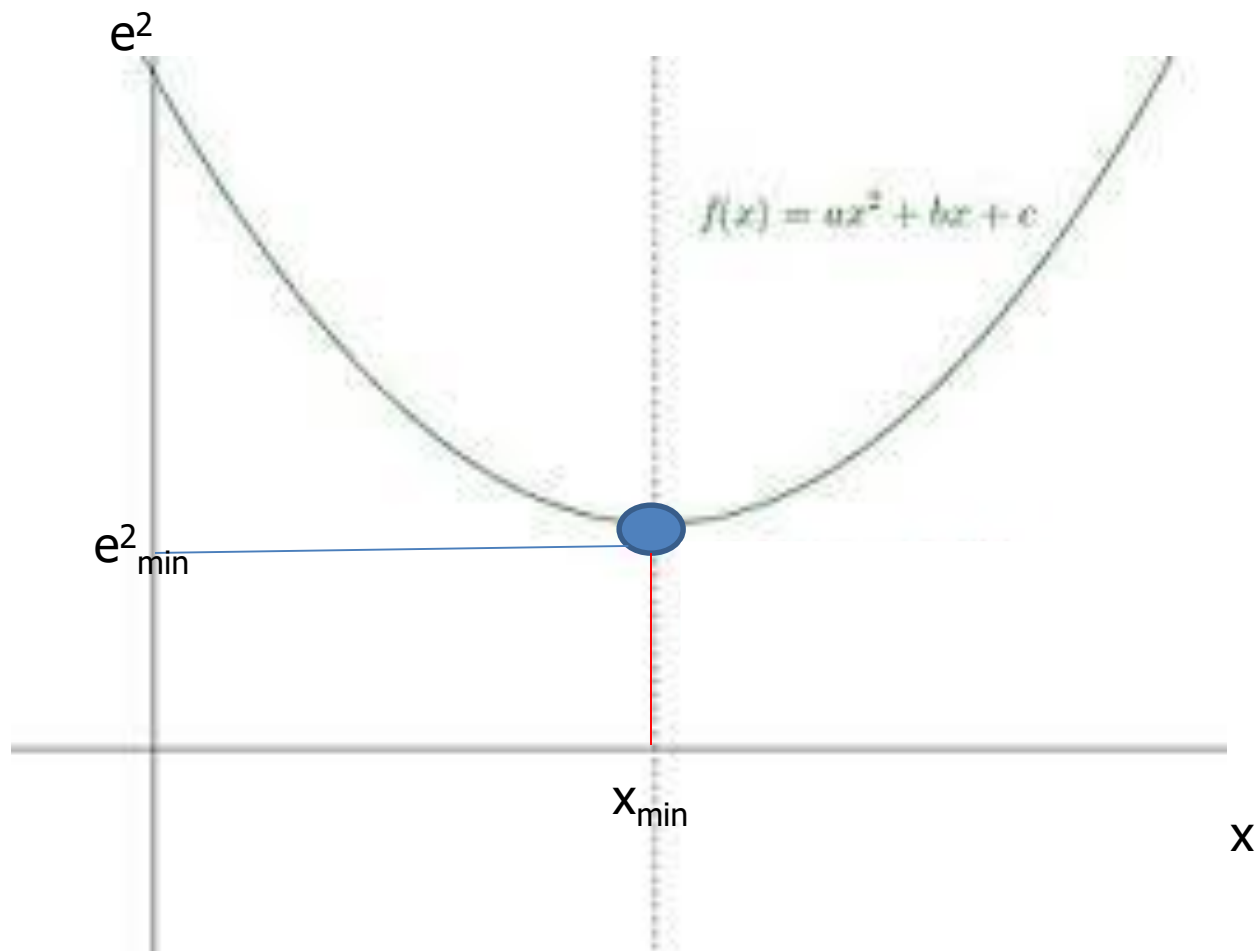
$$C_{\min} = 10 \text{ nF}$$

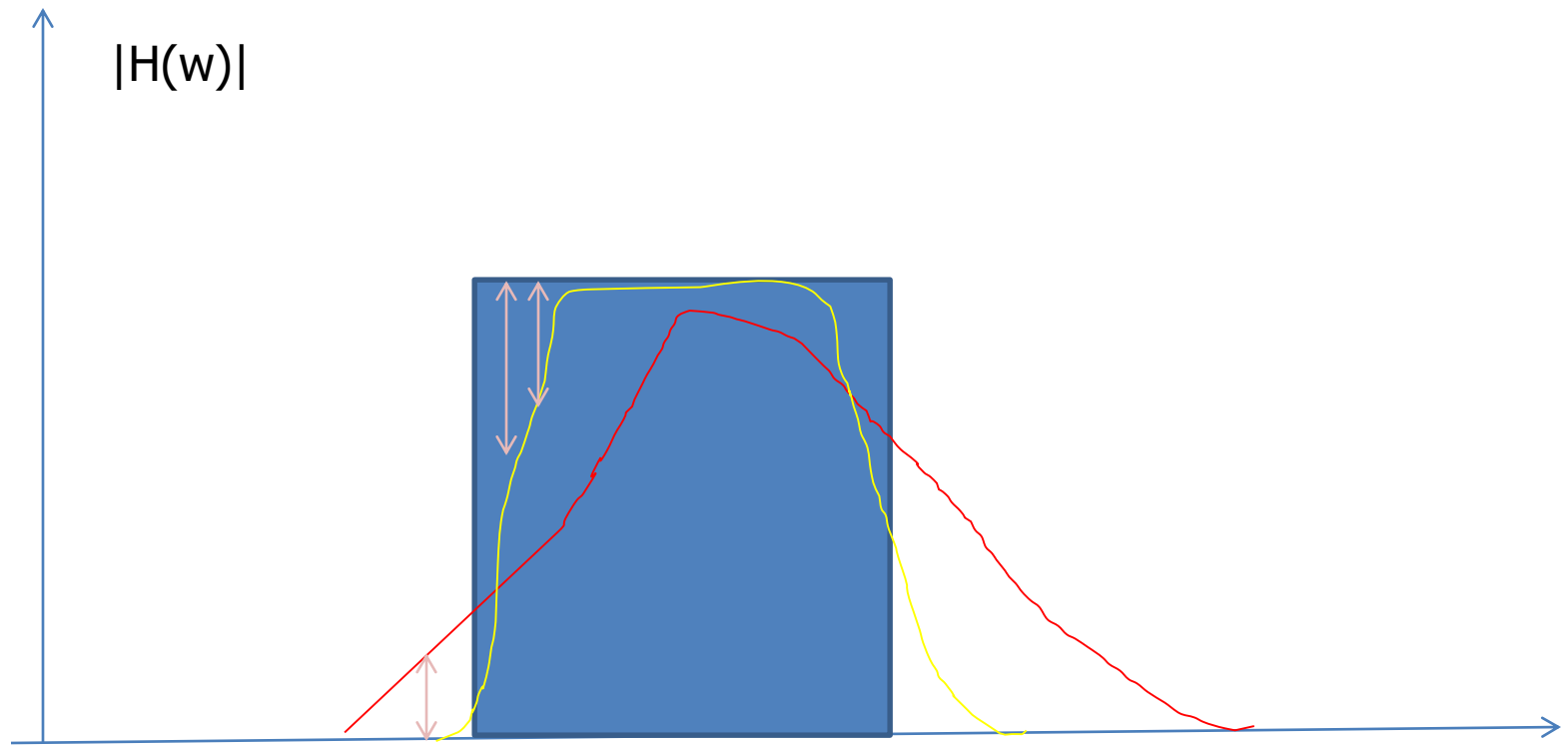
$$C_{\max} = 500 \text{ uF}$$

Função de Fitness

- Pode ser definida uma função de aptidão minimizando os erros quadráticos entre o valor especificado e o valor obtido, para cada conjunto de soluções:
 - e_{fL}^2 = Erro Quadrático da Frequência de corte inferior
 - e_{fH}^2 = Erro Quadrático da Frequência de corte superior
 - e_Q^2 = Erro Quadrático do Fator de qualidade
 - e_k^2 = Erro Quadrático do Ganho

$$fitness = \frac{1}{4} \left(e_{fL}^2 + e_{fH}^2 + e_Q^2 + e_k^2 \right)$$





$$fitness = \frac{1}{4} \left(e_{fL}^2 + e_{fH}^2 + e_Q^2 + e_k^2 \right)$$

$$fitness = \sum \left(e_H^2 \right)$$

Solução= $[R_A \ R_B \ R_f \ R_i \ C_A \ C_B]$

População Inicial

- Geração de números aleatórios dos componentes eletrônicos de acordo às restrições do problema.
- Determinar um tamanho da população, por exemplo 100

$$P_0 = \begin{bmatrix} R_{A1} & R_{B1} & R_{f1} & R_{i1} & C_{A1} & C_{B1} \\ R_{A2} & R_{B2} & R_{f2} & R_{i2} & C_{A2} & C_{B2} \\ \dots & & & & & \\ R_{A100} & R_{B100} & R_{f100} & R_{i100} & C_{A100} & C_{B100} \end{bmatrix}_{100 \times 6}$$

Seleção e Reprodução

- Seleção por Roleta
- Cruzamento Aritmético
- Mutação Gaussiana com valor médio zero e:
 - Desvio padrão para resistores 1k ohm
 - Desvio padrão para capacitores de 1 μ F
- Elitismo:
 - Os 5 melhores descendentes substituirão aos 5 piores pais
- Parâmetros:
 - $P_c = 0,85$ $P_m = 0,01$ #Gerações = 10000

Tópicos Avançados

Seleção

- Métodos de seleção com baixa pressão seletiva tendem a produzir progressos muito lentos no processo evolutivo.
- Métodos de Seleção:
 - *Seleção Proporcional à Função de Aptidão (Fitness)*
 - *Seleção Baseada na Classificação (Rank)*

Seleção Baseada na Classificação (Rank)

- A seleção proporcional considerando a aptidão dos indivíduos pode ser problemática se os indivíduos da população apresentarem desempenhos muito próximos entre si.
- Além disso, se o tamanho da população é pequeno, a perda de diversidade genética pode levar à convergência prematura, pois a busca fica reduzida a poucos pontos, causando uma diminuição no poder de exploração do algoritmo evolutivo.

Seleção Baseada na Classificação (Rank)

- Uma opção para se evitar o surgimento de super-indivíduos (relativos aos demais existentes na população atual) e a ocorrência de convergência prematura é reduzir as diferenças entre estes, a través de um mecanismo de seleção baseado em *rank*.
- Esta estratégia utiliza as **posições dos indivíduos** quando são ordenados de acordo com o *fitness* para determinar a **probabilidade de seleção**. Podem ser usados mapeamentos lineares ou não-lineares para determinar a probabilidade de seleção.

Seleção Baseada na Classificação (Rank)

- Para uma população de n indivíduos $P=\{x_1, x_2, \dots, x_n\}$, são arranjados em ordem crescente de aptidão ou fitness, tal que $fitness(x_i) < fitness(x_j)$, para todo $1 \leq i < j \leq n$.
- Define-se SP como a pressão de seleção do ambiente sobre os indivíduos. Neste caso, o valor da função de aptidão pode ser calculado por:
- Ranking Linear: O ranking linear permite valores de pressão de seleção SP entre [1 ; 2].

$$F(i) = 2 - SP + \frac{2 \cdot (SP - 1) \cdot (i - 1)}{n - 1}$$

Seleção Baseada na Classificação (Rank)

- **Ranking Não Linear:** O ranking não linear permite valores de pressão de seleção entre $SP \in [1 ; n-2]$.

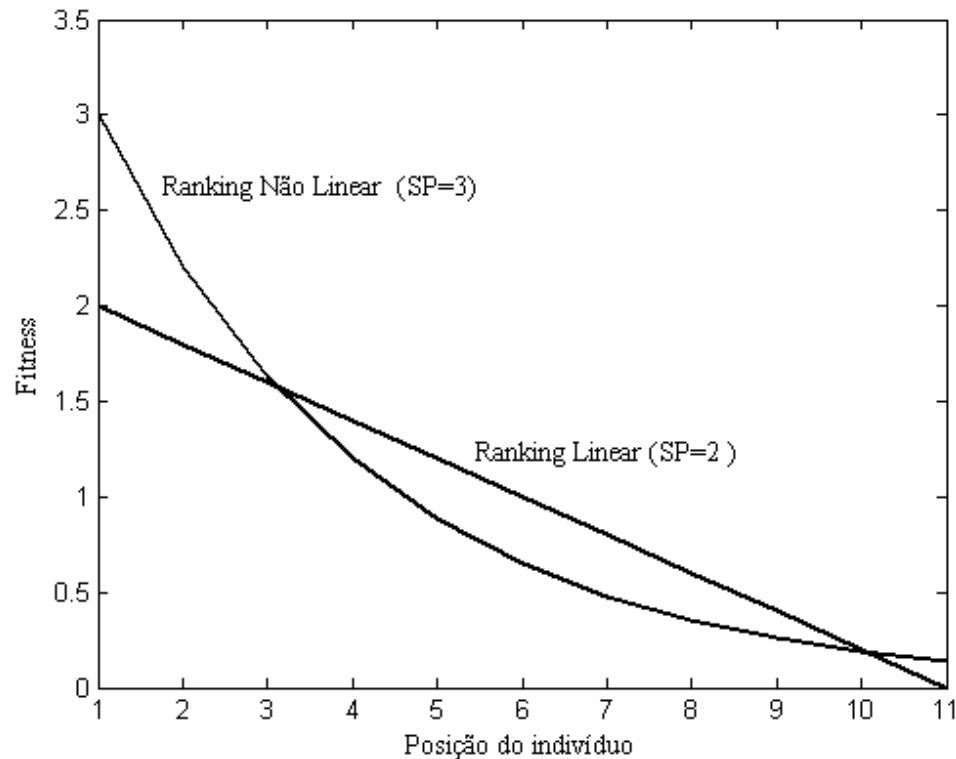
$$F(i) = \frac{n.v^{n-1}}{\sum_j^n v^{j-1}} \quad \text{Para: } i = 1, \dots, n$$

sendo que v é calculado como a raiz do polinômio:

$$(SP - n)v^{n-1} + SP.v^{n-2} + \dots + SP = 0$$

Seleção Baseada na Classificação (Rank)

- A seleção por **Rank** tem a **desvantagem** de exigir a ordenação de toda a população, o que pode representar um **custo computacional** em algumas aplicações.



Elitismo

- O termo elitismo está associado à adoção de uma operação adicional junto aos métodos de seleção, que **força o algoritmo evolutivo a reter o melhor indivíduo** ou um número de melhores indivíduos, a cada geração.
- Estes indivíduos **poderiam ser perdidos** se não fossem selecionados de forma determinística para compor a próxima geração, ou então fossem modificados por operadores de cruzamento ou mutação.
- Em grande parte dos casos, estratégias elitistas associadas aos métodos de seleção melhoram o desempenho do algoritmo evolutivo.