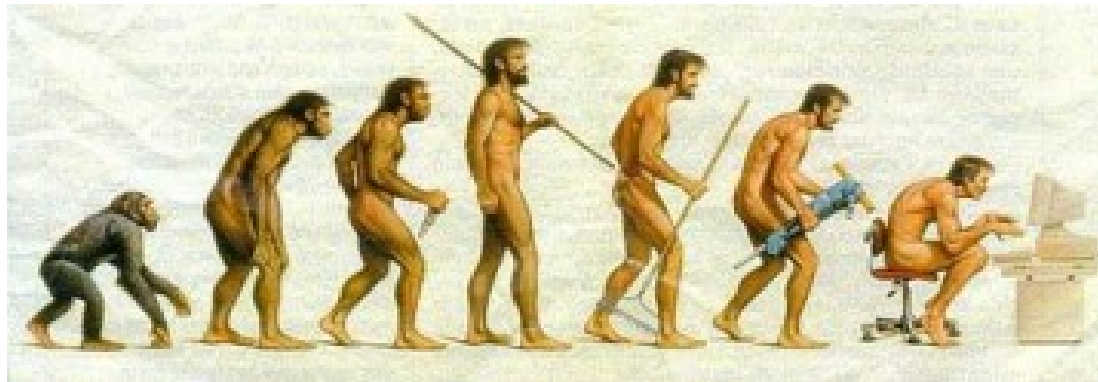


Exemplo de Aplicação de Algoritmos Genéticos



Prof. Juan Moisés Mauricio Villanueva

jmauricio@cear.ufpb.br

cear.ufpb.br/juan

Estrutura do Algoritmo Genético

Algoritmo genético

Início

$t = 0$

inicializar $P(t)$

avaliar $P(t)$

while (não cumpre a condição) **fazer**

$t = t + 1$

selecionar $P(t)$ de $P(t-1)$

reprodução $P(t)$

avaliar $P(t)$

fim

fim

Estrutura do Algoritmo Genético

- Componentes do AG:
 - Representação/Codificação dos indivíduos
 - População inicial
 - Função de avaliação (*fitness*)
 - Operador de Seleção
 - Reprodução- operadores genéticos
 - Cruzamento
 - Mutação

1. Representação de Indivíduos

- Cada indivíduo “*cromossoma*” representa um candidato potencial do problema
- Os indivíduos são codificados usando números **binários** de tamanho fixo.

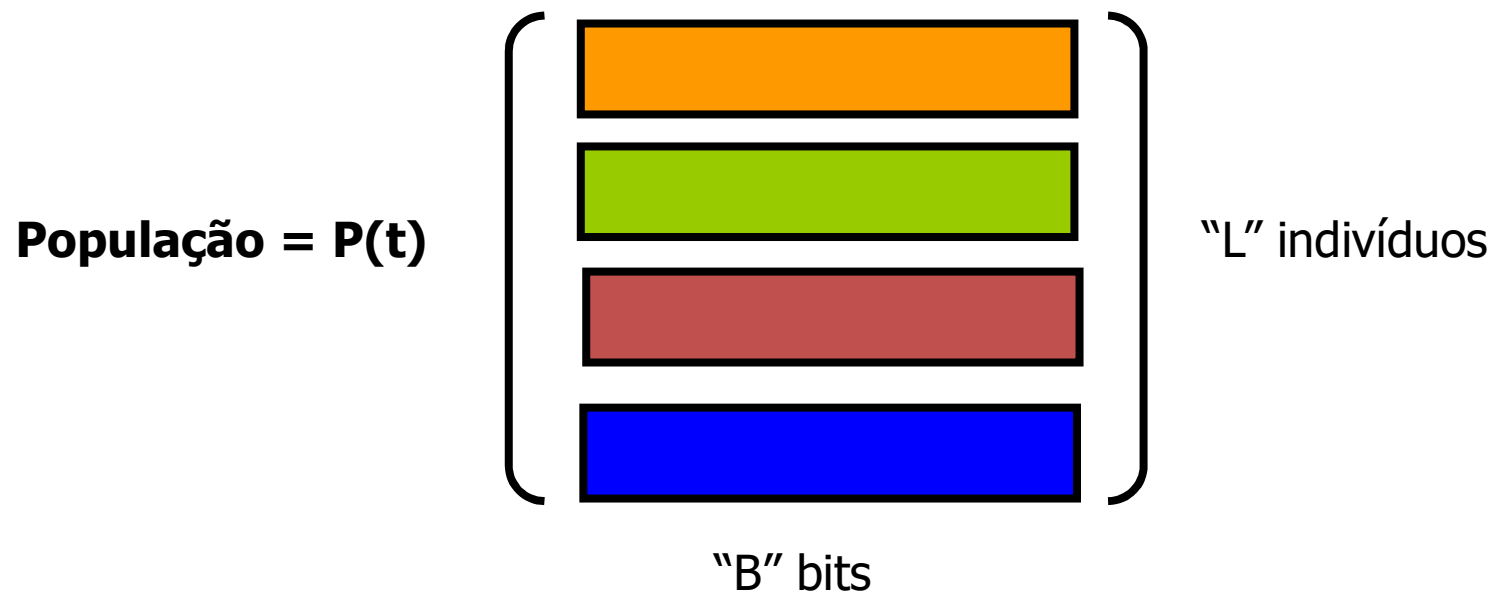
Indivíduo

1001010101001

“B” bits

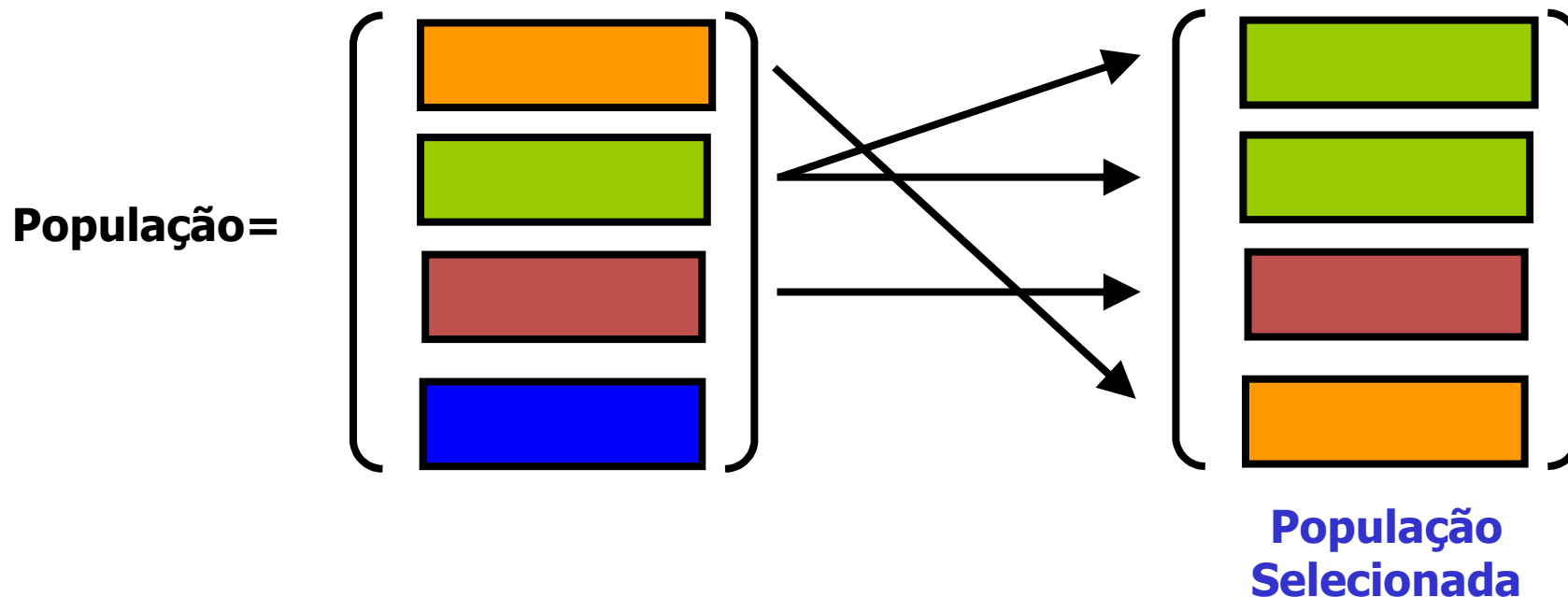
2. População Inicial

- A população inicial é um conjunto de indivíduos gerados aleatoriamente.



3. Seleção

- Consiste em **escolher** alguns indivíduos da população para criar descendentes.
- o objetivo da seleção consiste em **privilegiar** os indivíduos melhor adaptados, para criar descendentes.



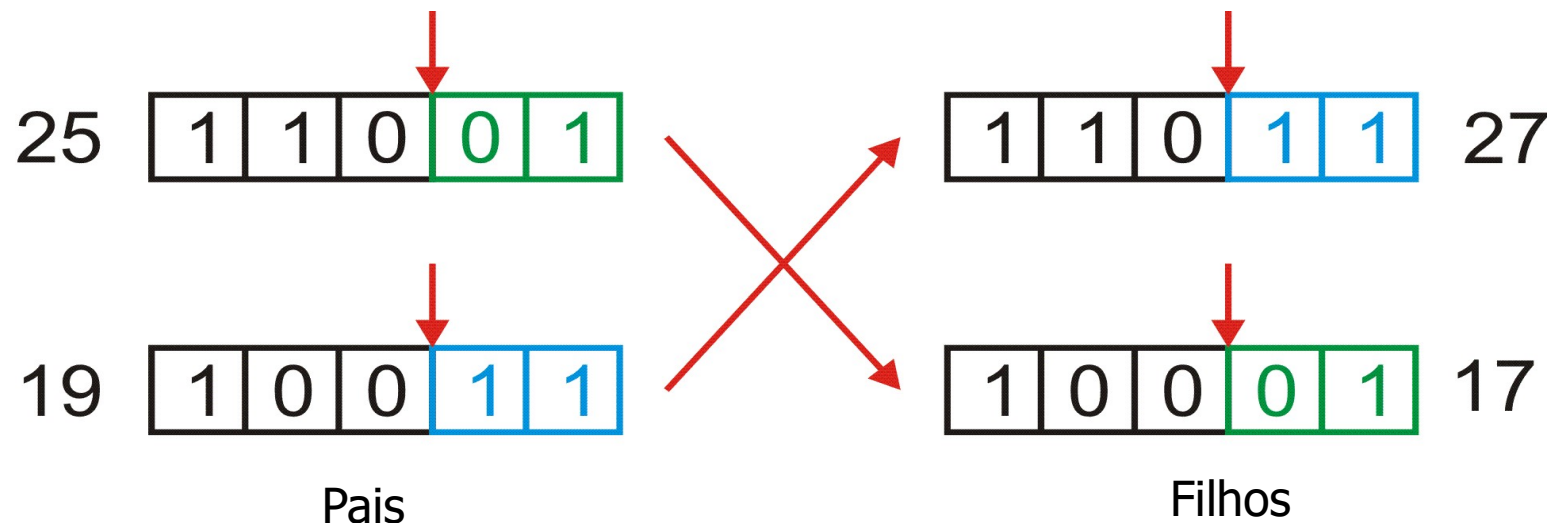
4. Operadores Genéticos

- O processo de reprodução gerará **novos indivíduos** da população selecionada.
- Os operadores genéticos representam uma fonte de **diversidade** e **variabilidade**
 - Cruzamento
 - Mutação

4. Operadores Genéticos

- **Cruzamento**

- Combina dois indivíduos intercambiando suas informações genéticas
- A probabilidade do cruzamento se denomina **taxa de cruzamento** que varia entre 0,5 a 1.



4. Operadores Genéticos

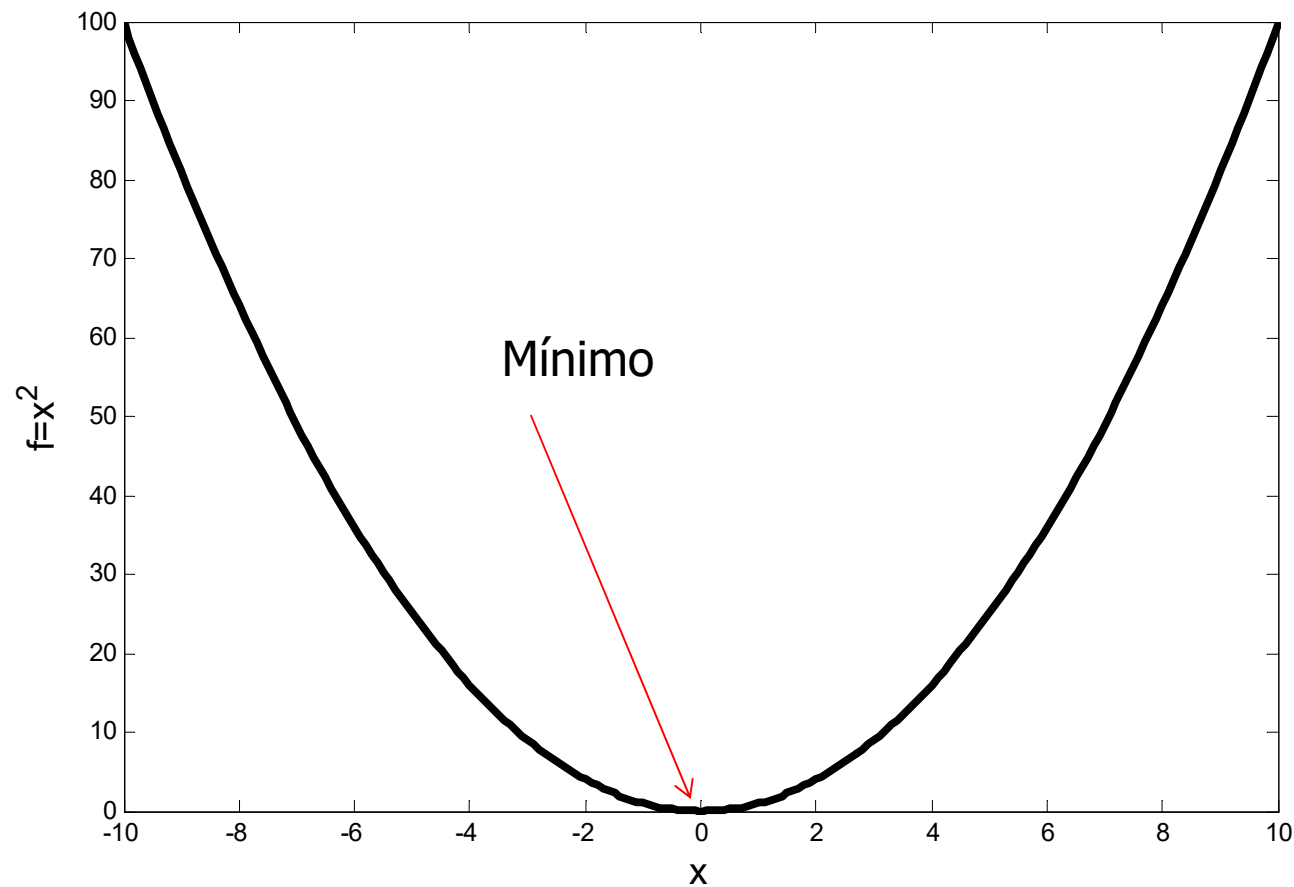
- **Mutação**

- Introduze novas combinações genéticas nos indivíduos.
- A probabilidade de mutação se denominada **taxa de mutação** que usualmente são valores pequenos 0.01, 0.02



Exemplo de Implementação do AG

- Minimizar a Função $f=x^2$ para $-10 \leq x \leq 10$



Exemplo de Implementação do AG

- Em problemas de **Optimização**, minimizar uma função f , é equivalente a maximizar uma função do tipo:

$$\min f(x) = \max \{C - f(x)\}$$

- *Em que: C é uma constante*

Usando AG para resolver o problema

Algoritmo genético

Início

$t = 0$

inicializar $P(t)$

avaliar $P(t)$

while (não cumpre a condição) **fazer**

$t = t + 1$

selecionar $P(t)$ de $P(t-1)$

reprodução $P(t)$

avaliar $P(t)$

fim

fim

AG para Problemas de Optimização

- Inicialmente se deve identificar:
 1. O intervalo de variação da variável x

$$a \leq x \leq b$$

2. Construir a função de fitness:

$$\max \{C - f\} = \max \{C - x^2\}$$

$$fitness = C - x^2$$

AG para Problemas de Optimiza  o

3. Se deve definir a precis  o com a qual ser   representada a vari  vel x:

$$\log_2 \left[(b - a) 10^N + 1 \right] \leq Bits$$

**N=Precis  o das
casas decimais**

N  mero de Bits

Indiv  duo x:

1001010101001

"B" bits

AG para Problemas de Optimização

4. O mapeamento da representação Binaria a Decimal é dada por:

$$x = a + decimal(10101...001) \cdot \frac{b - a}{2^{Bits} - 1}$$



**Representação
Decimal**



**Representação
Binaria**

Representação dos Indivíduos

- Para uma precisão de 3 casas decimais $N = 3$,

$$14.28 \leq Bits$$
$$B = 15$$

$$Indivíduo = [101010011100101]$$

"B=15" bits

População Inicial

- Conjunto de Indivíduos criados Aleatoriamente

$$P = \begin{bmatrix} \textit{Individuo 1} \\ \textit{Individuo 2} \\ \dots \\ \textit{Individuo L} \end{bmatrix} = \begin{bmatrix} 101010011100101 \\ 001010101010000 \\ \dots \\ 010101000111101 \end{bmatrix} \quad \begin{array}{l} \text{"L" indivíduos} \\ \\ \\ \text{"B=15" bits} \end{array}$$

População Inicial

- Para uma população inicial de tamanho L=10:

	3.9854	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1
	5.3661	1	1	0	0	0	1	0	0	1	0	1	0	1	1	1
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
x =	7.7960	1	1	1	0	0	0	1	1	1	1	0	0	1	0	0
	-6.0765	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0
	-5.7164	0	0	1	1	0	1	1	0	1	1	0	1	0	1	0
	5.6804	1	1	0	0	1	0	0	0	1	0	1	1	0	1	0
	9.7772	1	1	1	1	1	1	0	1	0	0	1	0	0	1	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0



**Indivíduos
Representação
Decimal**



**População
Representação
Binaria**

Avaliação (*fitness*)

- Cada indivíduo terá associada sua função de fitness:

$$P = \begin{bmatrix} \textit{Individuo 1} \\ \textit{Individuo 2} \\ \dots \\ \textit{Individuo L} \end{bmatrix} \qquad \textit{fitness} = \begin{bmatrix} \textit{fitness 1} \\ \textit{fitness 2} \\ \dots \\ \textit{fitness L} \end{bmatrix}$$

Função de
Fitness

$$\textit{fitness}(i) = C - x_i^2$$

Avaliação (*fitness*)

	3.9854		84.1165
	5.3661		71.2053
	-1.2693		98.3890
	3.3580		88.7242
	7.7960		39.2231
<code>x =</code>	-6.0765	<code>fitness =</code>	63.0757
	-5.7164		67.3225
	5.6804		67.7329
	9.7772		4.4061
	5.6767		67.7745



Indivíduo: x



$$fitness(i) = 100 - x_i^2$$

Seleção

Probabilidade de
Seleção

$$\text{Prob}(i) = \frac{\textit{fitness}(i)}{F}$$

$$F = \sum_{i=1}^L \textit{fitness}(i)$$

Seleção

$x =$	3.9854	84.1165	0.1290
	5.3661	71.2053	0.1092
	-1.2693	98.3890	0.1509
	3.3580	88.7242	0.1361
	7.7960	39.2231	0.0602
	-6.0765	63.0757	0.0967
	-5.7164	67.3225	0.1033
	5.6804	67.7329	0.1039
	9.7772	4.4061	0.0068
	5.6767	67.7745	0.1040

$$F = 651,9697$$

$$\text{Prob}(i) = \frac{\text{fitness}(i)}{F}$$

Seleção

	0.1290		0.1290
	0.1092		0.2382
	0.1509		0.3891
	0.1361		0.5252
ProbSelecao =	0.0602	q =	0.5854
	0.0967		0.6821
	0.1033		0.7854
	0.1039		0.8893
	0.0068		0.8960
	0.1040		1.0000

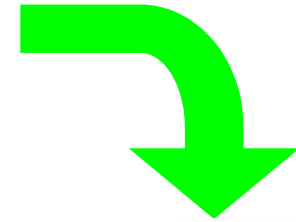
**Probabilidade
Seleção**

**Probabilidade
Acumulada**

Seleção

- Gera-se “L” números aleatórios

ProbSelecao =	0.1290	q =	0.1290	r =	0.4173
	0.1092		0.2382		0.0497
	0.1509		0.3891		0.9027
	0.1361		0.5252		0.9448
	0.0602		0.5854		0.4909
	0.0967		0.6821		0.4893
	0.1033		0.7854		0.3377
	0.1039		0.8893		0.9001
	0.0068		0.8960		0.3692
	0.1040		1.0000		0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =

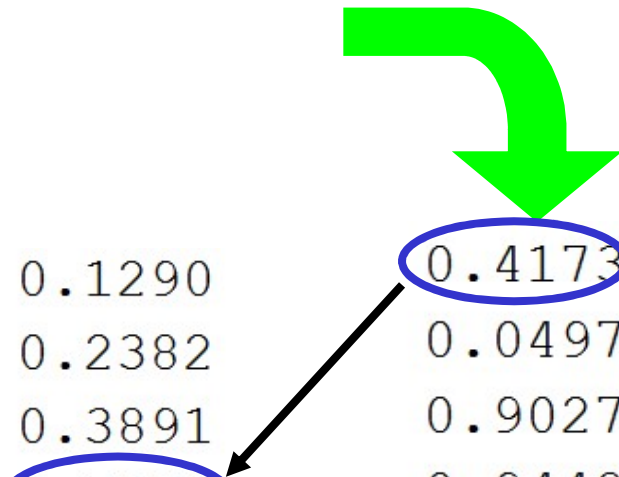
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

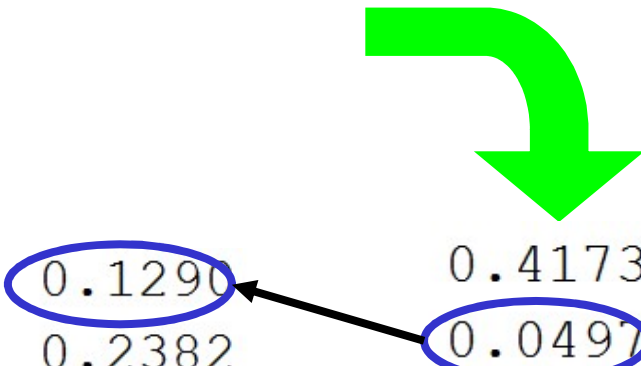
0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecção =	0.1290	q =	0.1290	r =	0.4173
	0.1092		0.2382		0.0497
	0.1509		0.3891		0.9027
	0.1361		0.5252		0.9448
	0.0602		0.5854		0.4909
	0.0967		0.6821		0.4893
	0.1033		0.7854		0.3377
	0.1039		0.8893		0.9001
	0.0068		0.8960		0.3692
	0.1040		1.0000		0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecção =

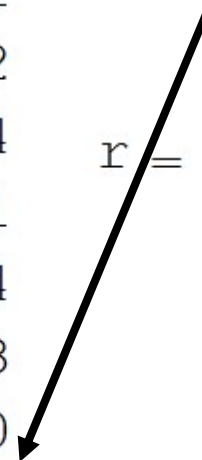
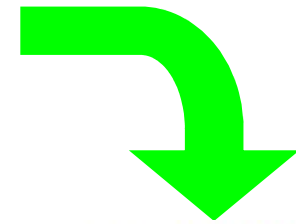
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =

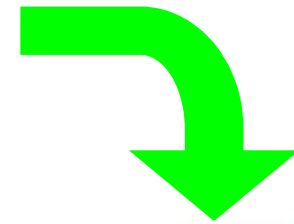
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =

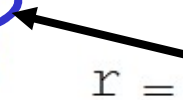
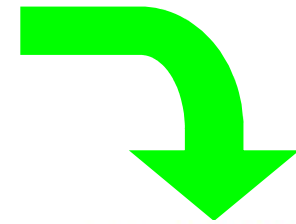
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =

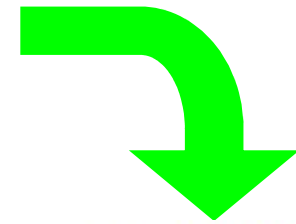
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

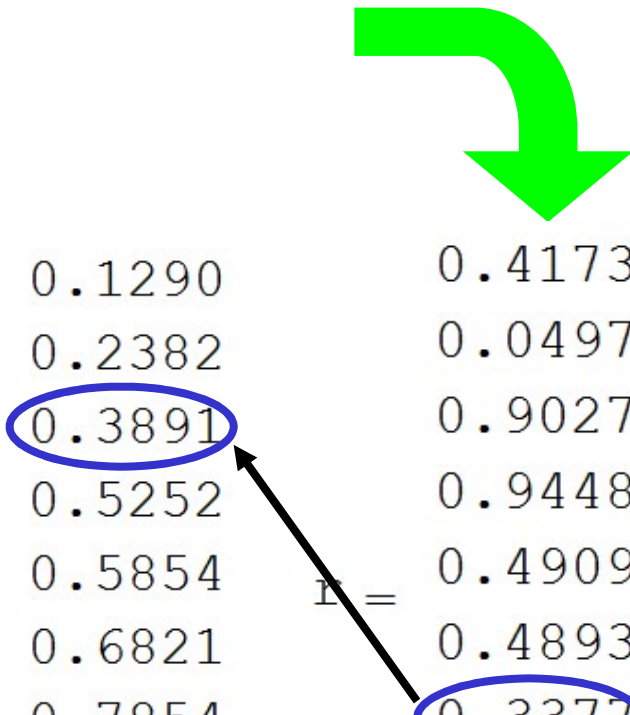
0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios


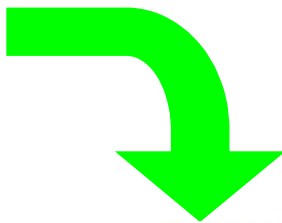
ProbSelecao =	0.1290	q =	0.1290	r =	0.4173
	0.1092		0.2382		0.0497
	0.1509		0.3891		0.9027
	0.1361		0.5252		0.9448
	0.0602		0.5854		0.4909
	0.0967		0.6821		0.4893
	0.1033		0.7854		0.3377
	0.1039		0.8893		0.9001
	0.0068		0.8960		0.3692
	0.1040		1.0000		0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =	0.1290	q =	0.1290	r =	0.4173
	0.1092		0.2382		0.0497
	0.1509		0.3891		0.9027
	0.1361		0.5252		0.9448
	0.0602		0.5854		0.4909
	0.0967		0.6821		0.4893
	0.1033		0.7854		0.3377
	0.1039		0.8893		0.9001
	0.0068		0.8960		0.3692
	0.1040		1.0000		0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =

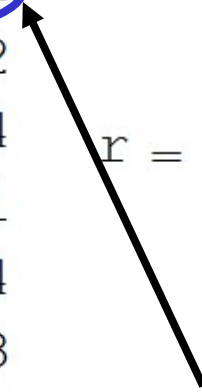
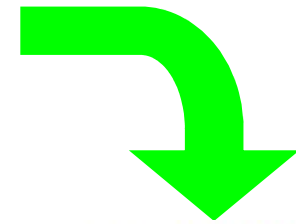
0.1290
0.1092
0.1509
0.1361
0.0602
0.0967
0.1033
0.1039
0.0068
0.1040

q =

0.1290
0.2382
0.3891
0.5252
0.5854
0.6821
0.7854
0.8893
0.8960
1.0000

r =

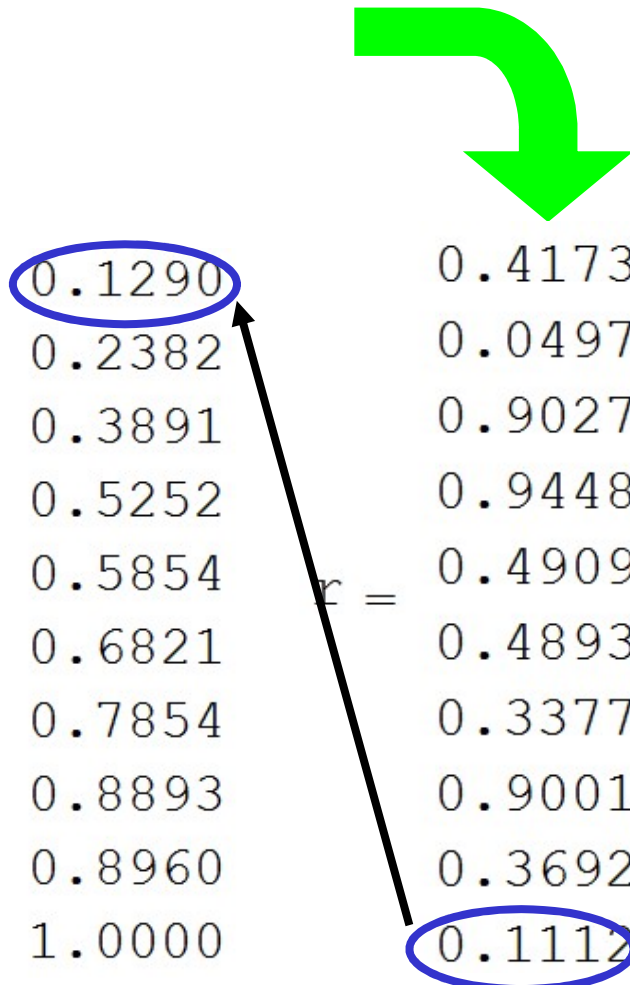
0.4173
0.0497
0.9027
0.9448
0.4909
0.4893
0.3377
0.9001
0.3692
0.1112



Seleção

- Gera-se “L” números aleatórios

ProbSelecao =	0.1290	q =	0.1290	r =	0.4173
	0.1092		0.2382		0.0497
	0.1509		0.3891		0.9027
	0.1361		0.5252		0.9448
	0.0602		0.5854		0.4909
	0.0967		0.6821		0.4893
	0.1033		0.7854		0.3377
	0.1039		0.8893		0.9001
	0.0068		0.8960		0.3692
	0.1040		1.0000		0.1112



Seleção

Índice dos indivíduos selecionados



4	3.3580	88.7242
1	3.9854	84.1165
10	5.6767	67.7745
10	5.6767	67.7745
4	3.3580	88.7242
4	3.3580	88.7242
3	-1.2693	98.3890
10	5.6767	67.7745
3	-1.2693	98.3890
1	3.9854	84.1165

x =

fitness =

**Indivíduos
Selecionados**

Fitness



Seleção

	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
	3.9854	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
x =	3.3580	P = 1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
	3.9854	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1



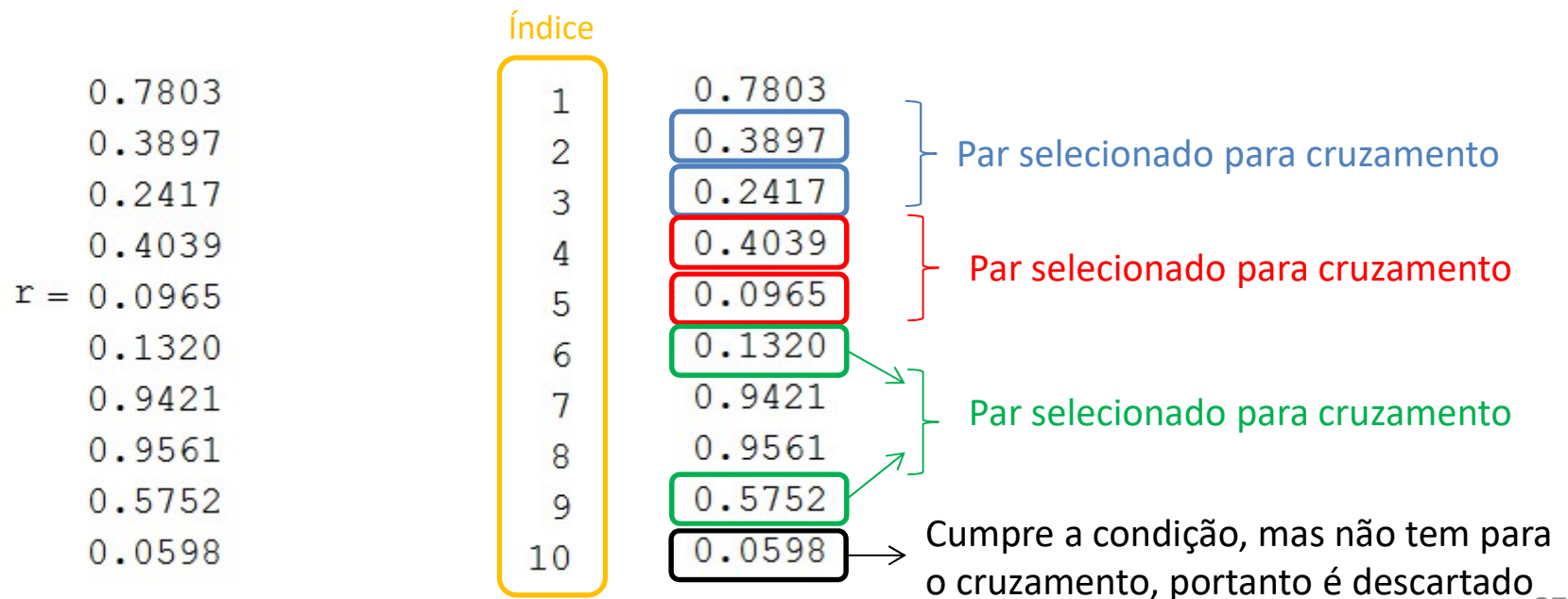
**Indivíduos
Selecionados**



**Codificação dos
Indivíduos Selecionados**

Operador de Cruzamento

- Se define uma probabilidade de cruzamento $P_c=0.7$
- Em seguida, se realiza a geração de 10 números aleatórios (r)
- Finalmente, se verifica que posições (índices) atendem a condição $r \leq P_c$, caso satisfeita a condição o individuo poderá ser usado na operação de cruzamento (neste caso os índices 2, 3, 4, 5, 6, 9, 10).



Operador de Cruzamento

- Selecionando o primeiro par para cruzamento, com índices das linhas 2 e 3

0.7803		1	0	1	0	1	0	1	1	1	1	1	1	0	1
0.3897		1	0	1	1	0	0	1	1	0	0	0	0	0	1
0.2417		1	1	0	0	1	0	0	0	1	0	1	0	1	0
0.4039		1	1	0	0	1	0	0	0	1	0	1	0	1	0
r = 0.0965	P = 1	0	1	0	1	0	1	0	1	1	1	1	1	1	0
0.1320		1	0	1	0	1	0	1	0	1	1	1	1	1	0
0.9421		0	1	1	0	1	1	1	1	1	0	0	0	0	0
0.9561		1	1	0	0	1	0	0	0	1	0	1	0	1	0
0.5752		0	1	1	0	1	1	1	1	1	0	0	0	0	0
0.0598		1	0	1	1	0	0	1	1	0	0	0	0	0	1

Operador de Cruzamento

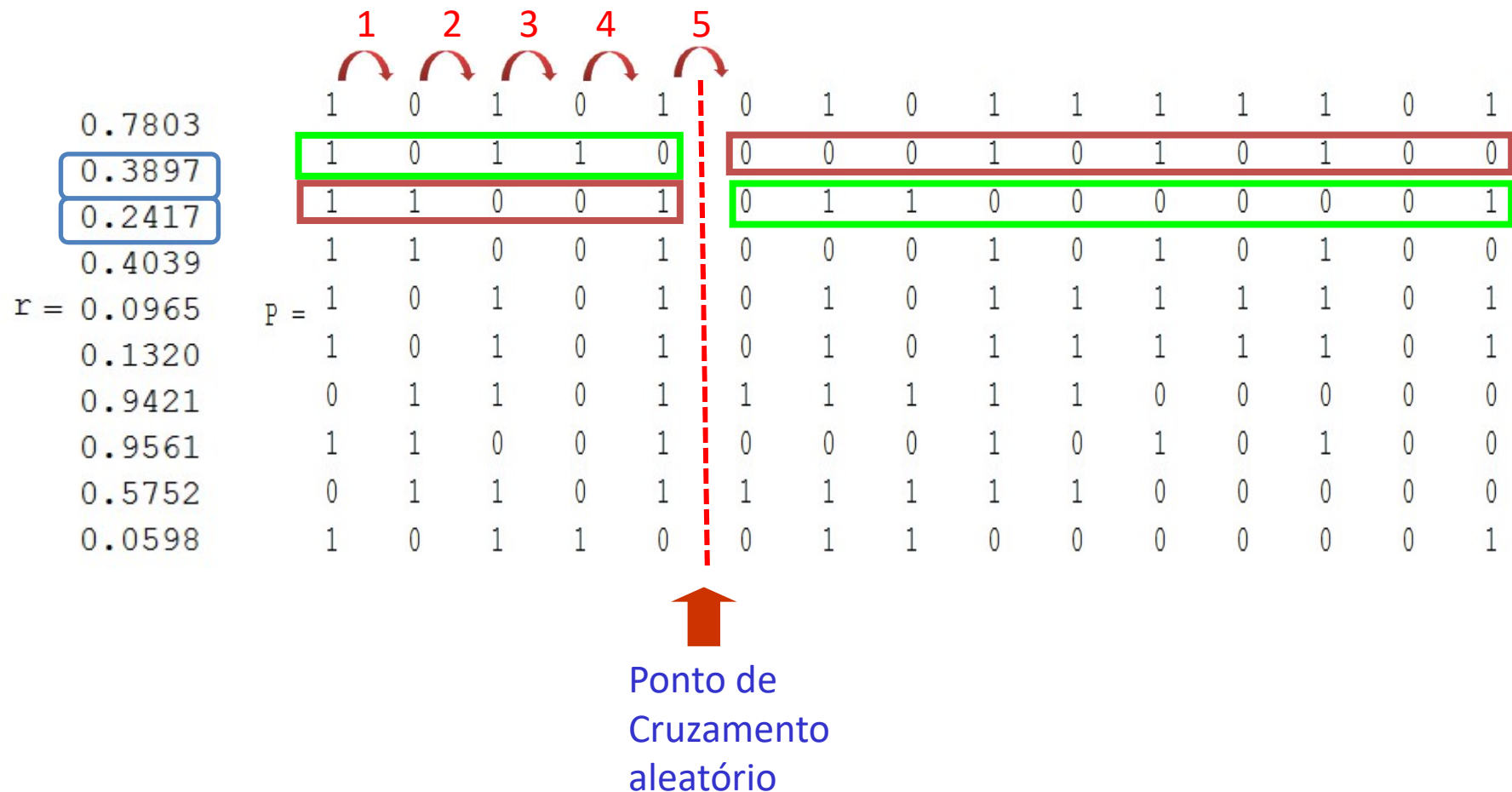
- Determinar aleatoriamente o ponto de cruzamento. Neste caso deverá ser gerado um número aleatório interior no intervalo de 1 a 14.
- Para o ponto de cruzamento igual a 5

0.7803	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	
0.3897	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1	
0.2417	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0	
0.4039	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0	
r = 0.0965	P = 1	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1
0.1320	1	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1
0.9421	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0
0.9561	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0
0.5752	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0
0.0598	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1

<

Operador de Cruzamento

- Após o cruzamento as linhas com índices 2 e 3 intercambiaram informações por meio da troca de bits



Operador de Cruzamento

- Este procedimento será repetido para os pares com linhas (4, 5) e (6, 9)
- Finalmente tem-se uma nova população modificada com o operador de cruzamento.

		1	0	1	0	1	0	1	0	1	1	1	1	0	1
	3.3580	1	0	1	1	0	0	1	1	0	0	0	0	0	1
	3.9854	1	1	0	0	1	0	0	0	1	0	1	0	1	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0
x =	3.3580	P = 1	0	1	0	1	0	1	0	1	1	1	1	1	0
	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0
	3.9854	1	0	1	1	0	0	1	1	0	0	0	0	0	1



	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0
	3.8017	1	0	1	1	0	0	0	0	1	0	1	0	1	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0
	5.8629	1	1	0	0	1	0	1	1	0	0	0	0	1	0
x =	3.3555	P = 1	0	1	0	1	0	1	0	1	1	1	1	0	0
	3.3580	1	0	1	0	1	0	1	0	1	1	1	1	1	0
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0
	5.6767	1	1	0	0	1	0	0	0	1	0	1	0	1	0
	-1.2693	0	1	1	0	1	1	1	1	1	1	0	0	0	0
	3.9854	1	0	1	1	0	0	1	1	0	0	0	0	0	1

Operador de Mutação

- Se define uma probabilidade de mutação $P_m=0.01$
- Em seguida, se realiza a geração uma matriz de 10x15 contendo números aleatórios (r). Este tamanho corresponde ao tamanho da matriz binária da população P .
- Em seguida, para cada número aleatório se verifica se cumpre a condição $r \leq P_m$, caso satisfeita a condição, o elemento da matriz P é permutado de 0 para 1 ou vice-versa.

	0.8034	0.9841	0.7379	0.5391	0.6692	0.4283	0.2653	0.2607	0.4709	0.8200	0.2665	0.9577	0.8444	0.4243	0.1527
	0.0605	0.1672	0.2691	0.6981	0.1904	0.4820	0.8244	0.5944	0.6959	0.7184	0.1537	0.2407	0.3445	0.4609	0.3411
	0.3993	0.1062	0.4228	0.6665	0.3689	0.1206	0.9827	0.0225	0.6999	0.9686	0.2810	0.6761	0.7805	0.7702	0.6074
	0.5269	0.3724	0.5479	0.1781	0.4607	0.5895	0.7302	0.4253	0.6385	0.5313	0.4401	0.2891	0.6753	0.3225	0.1917
$r =$	0.4168	0.1981	0.9427	0.1280	0.9816	0.2262	0.3439	0.3127	0.0336	0.3251	0.5271	0.6718	0.0067	0.7847	0.7384
	0.6569	0.4897	0.4177	0.9991	0.1564	0.3846	0.5841	0.1615	0.0688	0.1056	0.4574	0.6951	0.6022	0.4714	0.2428
	0.6280	0.3395	0.9831	0.1711	0.8555	0.5830	0.1078	0.1788	0.3196	0.6110	0.8754	0.0680	0.3868	0.0358	0.9174
	0.2920	0.9516	0.3015	0.0326	0.6448	0.2518	0.9063	0.4229	0.5309	0.7788	0.5181	0.2548	0.9160	0.1759	0.2691
	0.4317	0.9203	0.7011	0.5612	0.3763	0.2904	0.8797	0.0942	0.6544	0.4235	0.9436	0.2240	0.0012	0.7218	0.7655
	0.0155	0.0527	0.6663	0.8819	0.1909	0.6171	0.8178	0.5985	0.4076	0.0908	0.6377	0.6678	0.4624	0.4735	0.1887



Os elementos da matriz r , com índices (5,13) e (9,13) sofrerão mutação

Operador de Mutação

P =

1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
1	0	1	1	0	0	0	0	1	0	1	0	1	0	0
1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
1	1	0	0	1	0	1	1	0	0	0	0	1	0	1
1	0	1	0	1	0	1	0	1	1	1	1	0	0	1
1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
1	0	1	1	0	0	1	1	0	0	0	0	0	0	1

Mutação

P =

1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
1	0	1	1	0	0	0	0	1	0	1	0	1	0	0
1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
1	1	0	0	1	0	1	1	0	0	0	0	1	0	1
1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
1	0	1	0	1	0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	1	0	1	0	1	0	0
0	1	1	0	1	1	1	1	1	1	0	0	1	0	0
1	0	1	1	0	0	1	1	0	0	0	0	0	0	1

Processo de Evolução para 1 geração

	População Inicial	Seleção	Cruzamento	Mutação
Indivíduos	3.9854	3.3580	3.3580	3.3580
	5.3661	3.9854	3.8017	3.8017
	-1.2693	5.6767	5.6767	5.6767
	3.3580	5.6767	5.8629	5.8629
	7.7960	3.3580	3.3555	3.3580
	-6.0765	3.3580	3.3580	3.3580
	-5.7164	-1.2693	-1.2693	-1.2693
	5.6804	5.6767	5.6767	5.6767
	9.7772	-1.2693	-1.2693	-1.2668
	5.6767	3.9854	3.9854	3.9854
fitness	84.1165	88.7242		88.7242
	71.2053	84.1165		85.5471
	98.3890	67.7745		67.7745
	88.7242	67.7745		65.6263
	39.2231	88.7242		88.7242
	63.0757	88.7242		88.7242
	67.3225	98.3890		98.3890
	67.7329	67.7745		67.7745
	4.4061	98.3890		98.3952
	67.7745	84.1165		84.1165

Comparação do melhor indivíduo na população inicial com o melhor indivíduo após a mutação

	População Inicial	Seleção	Cruzamento	Mutação
Indivíduos	3.9854	3.3580	3.3580	3.3580
	5.3661	3.9854	3.8017	3.8017
	-1.2693	5.6767	5.6767	5.6767
	3.3580	5.6767	5.8629	5.8629
	7.7960	3.3580	3.3555	3.3580
	-6.0765	3.3580	3.3580	3.3580
	-5.7164	-1.2693	-1.2693	-1.2693
	5.6804	5.6767	5.6767	5.6767
	9.7772	-1.2693	-1.2693	-1.2668
	5.6767	3.9854	3.9854	3.9854
fitness	84.1165	88.7242		88.7242
	71.2053	84.1165		85.5471
	98.3890	67.7745		67.7745
	88.7242	67.7745		65.6263
	39.2231	88.7242		88.7242
	63.0757	88.7242		88.7242
	67.3225	98.3890		98.3890
	67.7329	67.7745		67.7745
	4.4061	98.3890		98.3952
	67.7745	84.1165		84.1165

Melhor
solução

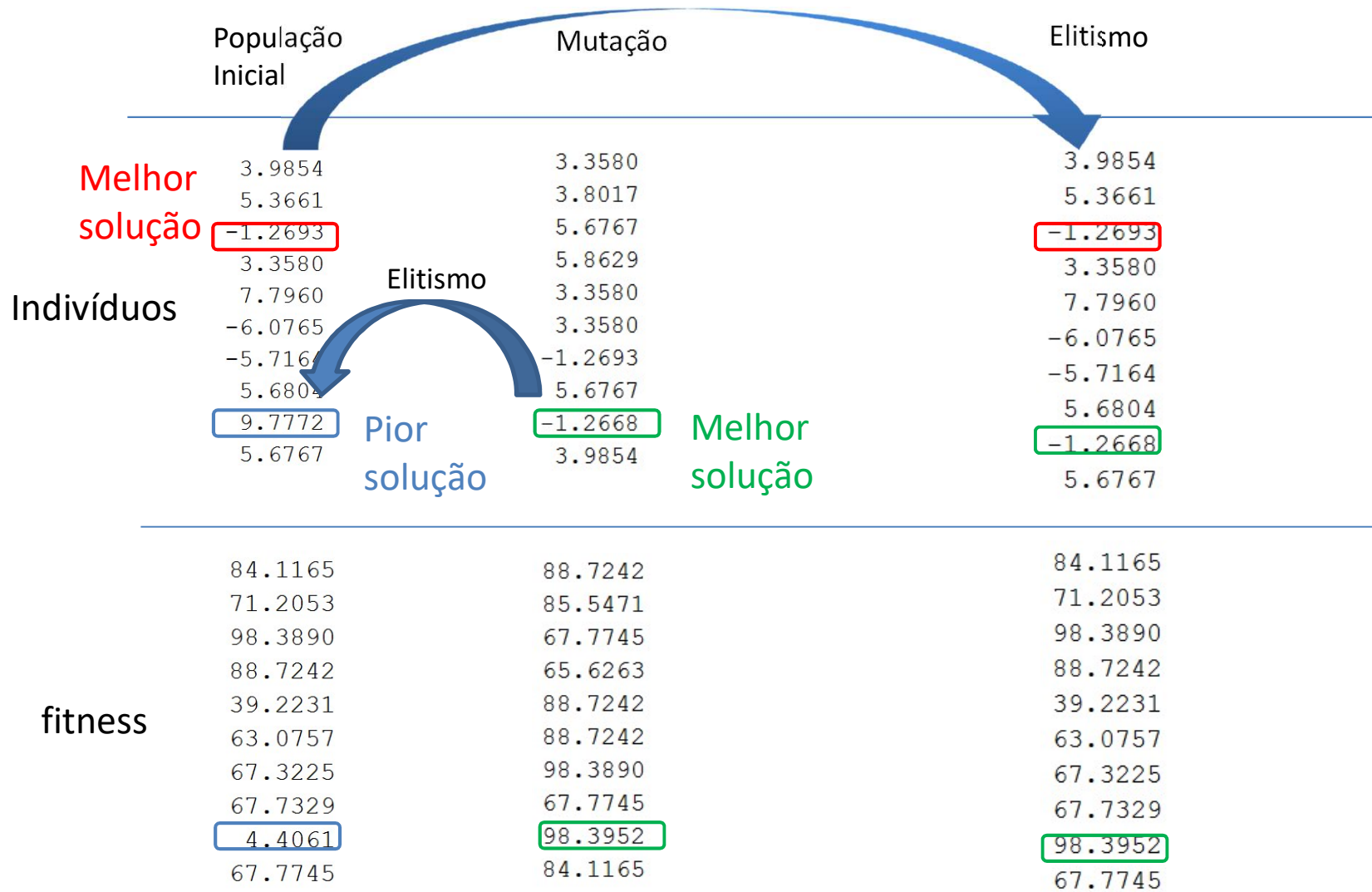
Melhor
solução

Neste passo
não é
necessário
avaliar o
fitness

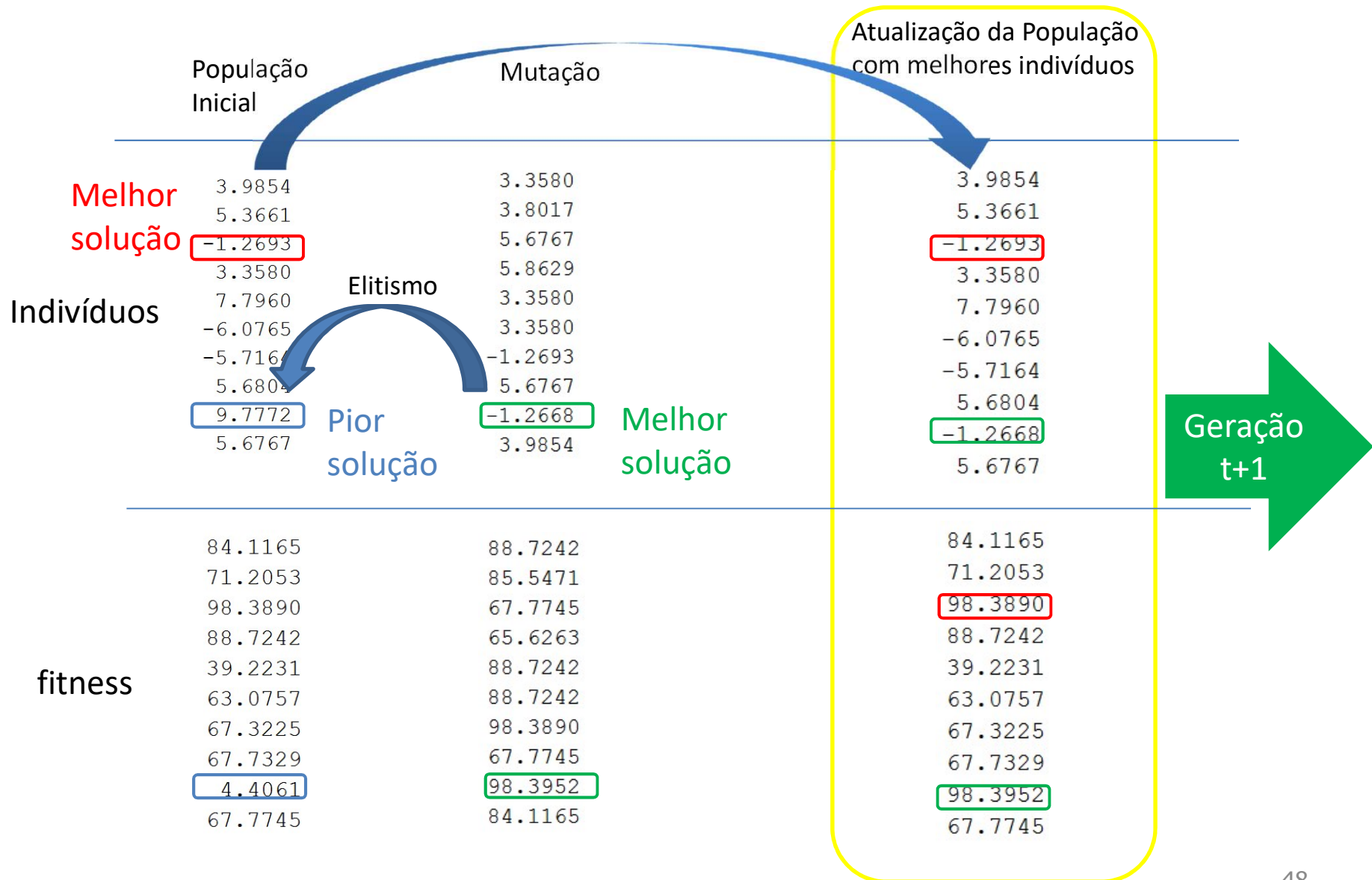
Elitismo: Substituição do melhor pelo pior

	População Inicial	Mutação
Indivíduos	3.9854	3.3580
	5.3661	3.8017
	Melhor solução -1.2693	5.6767
	3.3580	5.8629
	7.7960	3.3580
	-6.0765	3.3580
	-5.7164	-1.2693
	5.6804	5.6767
	9.7772	-1.2668
	5.6767	3.9854
	Pior solução	Melhor solução
fitness	84.1165	88.7242
	71.2053	85.5471
	98.3890	67.7745
	88.7242	65.6263
	39.2231	88.7242
	63.0757	88.7242
	67.3225	98.3890
	67.7329	67.7745
	4.4061	98.3952
	67.7745	84.1165

Elitismo: Substituição do melhor pelo pior

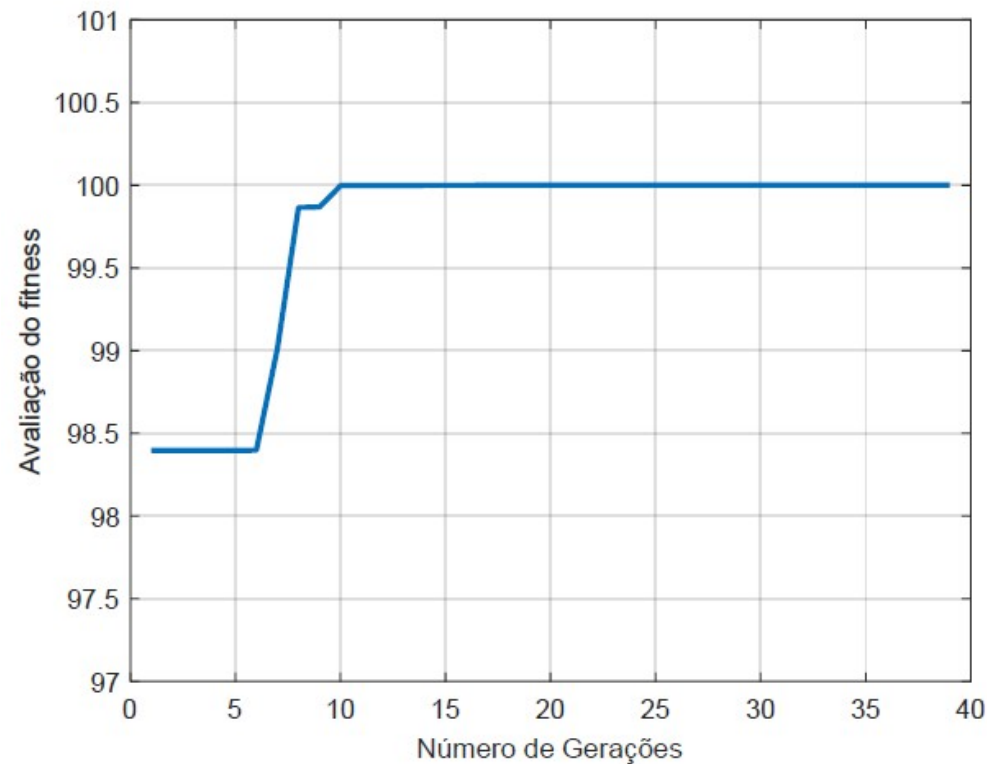


Elitismo: Substituição do melhor pelo pior



Processo de Evolução vs #Gerações

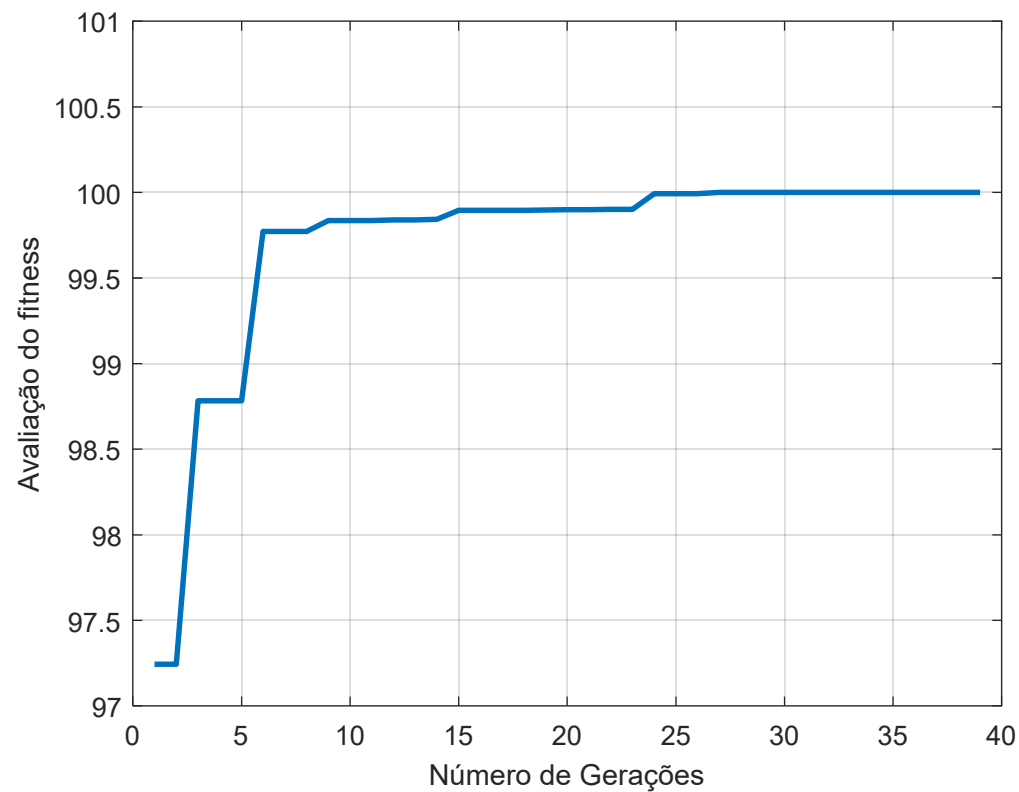
- Para 40 Gerações:



Solução: $x = -9.1556e-04$ $y = f(x) = 8.3824e-07$ **fitness=100**

Processo de Evolução vs #Gerações

- Para 40 Gerações:



Solução: $x = 3.0519e-04$ $y=f(x) = 9.3138-08$ **fitness=100**

Conclusões e Observações

- O processo de evolução de um AG realiza a busca em um espaço de soluções potenciais para o problema (População).
- Os AGs são mais robustos, devido a que existe uma direção ótima no processo de busca da solução.
- Os AGs mantem populações de soluções potenciais intercambiando informações.
- Os AGs não garantem uma solução GLOBAL, entretanto a solução obtida pode se encontrar próxima da global (solução local).

Conclusões e Observações

- A velocidade de convergência do AGs dependerá dos parâmetros de configuração, tais como: Tamanho da população, probabilidade de cruzamento P_c , probabilidade de mutação P_m , e operação de elitismo.
- Para sair de um estágio de estagnação poderia se implementar um operador de mutação com probabilidade P_m aumentando em função ao número de gerações.
- Observa-se que para cada nova execução do AGs a solução obtida tem variações assim como sua curva de convergência.