

Algoritmos Genéticos

Revisão dos conceitos

Sumário

- Representação
- Operadores
- Seleção
- Algoritmos
- Atividade
- Aplicação I
- Aplicação II

Representação do Indivíduo

- Definir qual será a representação do indivíduo é uma das primeiras decisões que se deve tomar no desenvolvimento de um AG.
- Codificação x Decodificação

Representação Binária

0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

 Representação Binária (8bits)

0	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 Representação Binária (16 bits)

0	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Representação Binária (32 bits)

Representação Binária

cromossomo com 2 variáveis

1	0	0	1	1	0	0	1	X_1
0	0	1	1	1	0	1	0	X_2

cromossomo com 3 variáveis

1	0	0	1	1	0	0	1	X_1
0	0	1	1	1	0	1	0	X_2
1	1	0	0	0	1	0	1	X_3

Representação Binária

- Duas importantes questões na representação de números reais em binários são:
 - Intervalo de domínio de cada uma das variáveis
 - Precisão desejada

Exemplo - Decodificação

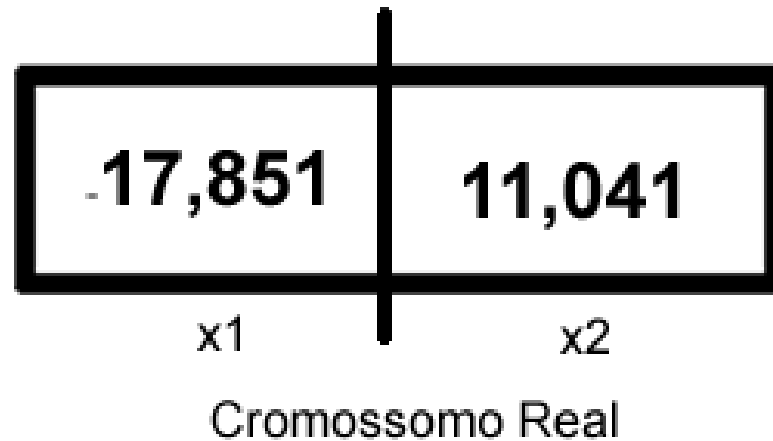
$$f(x) = x \sin(10\pi x) + 1 \quad -1 \leq x \leq 2.$$

$$s_1 = 1000101110110101000111$$

$$b_{10} = (1000101110110101000111)_2 = 2288967$$

$$x = \min + (\max - \min) \frac{b_{10}}{2^l - 1} \quad x_1 = -1 + (2 + 1) \frac{2.288.967}{(2^{22} - 1)} = 0,637197$$

Representação Real



Representação Binária x Real

- Binária

- ↑ Tradicional e fácil de utilizar
- ↓ Cromossomos longos para representar parâmetros contínuos com boa precisão.
- ↓ Longas cadeias podem levar a uma convergência lenta do método.
- ↓ Não uniformidade dos operadores. Por exemplo, mutação nos primeiros bits tem mais impacto que nos últimos.

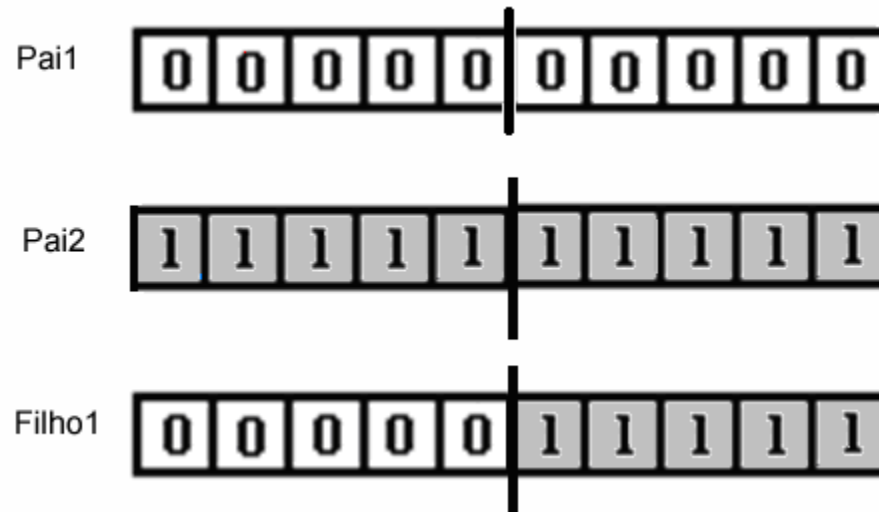
- Real

- ↑ Cadeias menores
- ↑ Compreendida mais naturalmente.
- ↑ Facilidade para criar novos operadores.

Operadores

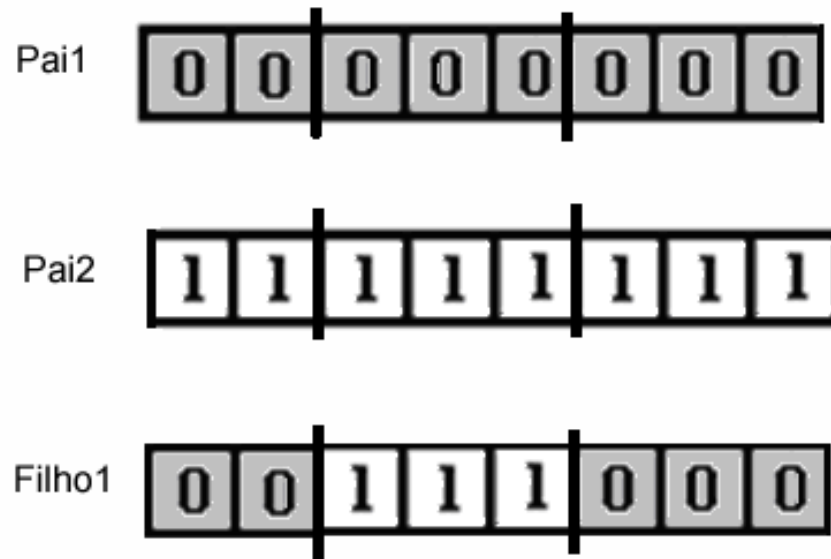
Crossover de um ponto

Representação Binária



Crossover de dois pontos

Representação Binária



Crossover uniforme

Representação Binária

Pai1	1	1	1	1	1	1	1
Pai2	0	0	0	0	0	0	0
Filho1	1	0	1	1	1	0	0

Números sorteados (0, 1, 0, 0, 0, 1, 1, 1)

Mutação

Representação Binária

Mutação

Antes:

0	0	1	0	0	0	1
---	---	---	---	---	---	---

Depois:

1	0	1	0	0	1	1
---	---	---	---	---	---	---

Crossovers

Representação Real

Pais

$$\mathbf{p}_1 = (p_{11}, p_{12}, \dots, p_{1l})$$

$$\mathbf{p}_2 = (p_{21}, p_{22}, \dots, p_{2l})$$



Filho

$$\mathbf{c} = (c_1, c_2, \dots, c_l).$$

Crossover média (Davis, 1991) $\mathbf{c} = (\mathbf{p}_1 + \mathbf{p}_2)/2$

Crossover média geométrica, $c_i = \sqrt{p_{1i} p_{2i}}$

Crossovers - Representação Real

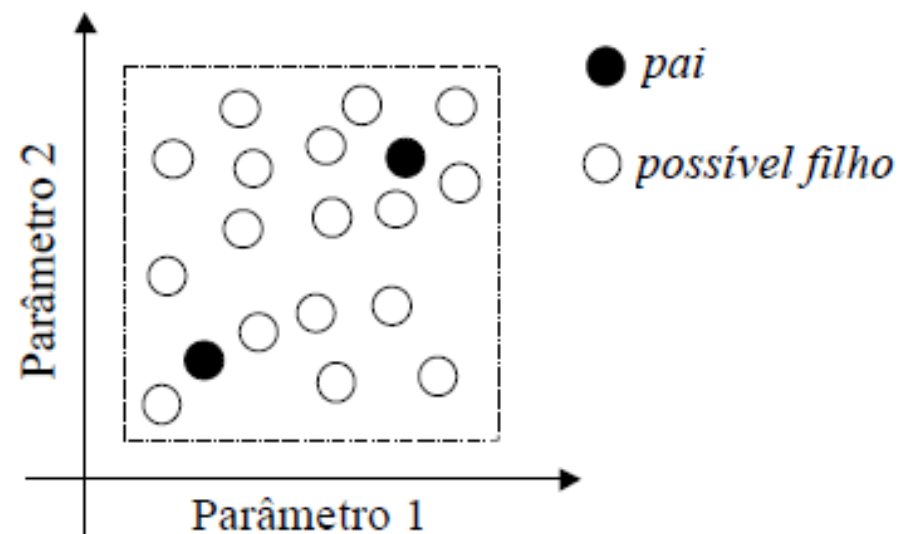
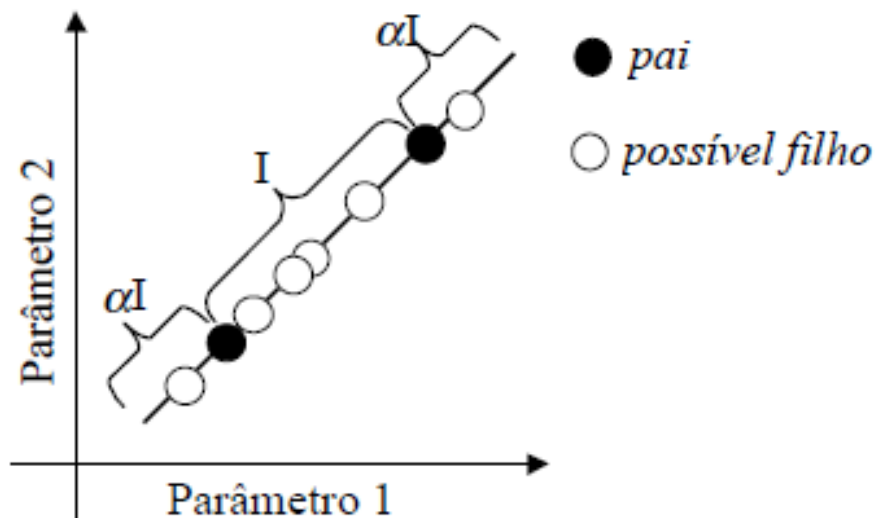
Crossover BLX- α

(Eshelman e Shaffer, 1993)

$$c = p_1 + \beta(p_2 - p_1) \quad \beta \in U(-\alpha, 1 + \alpha).$$

$$p_1 = (30,173; 85,342) \quad c_1 = 30,173 + 1,262(75,989 - 30,173) = 87,993$$

$$p_2 = (75,989; 10,162) \quad c_2 = 85,342 + 1,262(10,162 - 85,342) = -9,535$$



Crossovers - Representação Real

$$c_1 = 0,5p_1 + 0,5p_2$$

Crossover linear (Wright, 1991):

$$c_2 = 1,5p_1 - 0,5p_2$$

$$c_3 = -0,5p_1 + 1,5p_2$$

Crossover aritmético (Michalewicz, 1994)

$$c_1 = \beta p_1 + (1 - \beta)p_2$$

$$\beta \in U(0,1)$$

$$c_2 = (1 - \beta)p_1 + \beta p_2$$

Crossover heurístico (Michalewicz, 1994)

$$c = p_1 + r(p_1 - p_2), \text{ onde } f(p_1) > f(p_2)$$

$$r \in U(0,1)$$

Mutação

Representação Real

Mutação uniforme:
$$c_i = \begin{cases} U(a_i, b_i), & \text{se } i = j \\ p_i & \text{caso contrário} \end{cases}$$

Mutação gaussiana:
$$c_i = \begin{cases} N(p_i, \sigma), & \text{se } i = j \\ p_i & \text{caso contrário} \end{cases}$$

Mutação *creep*:

Mutação

Representação Real

Mutação limite (Michalewicz, 1994)
$$c_i = \begin{cases} a_i & \text{se } r < 0,5 \text{ e } i = j \\ b_i & \text{se } r \geq 0,5 \text{ e } i = j \\ p_i & \text{caso contrário} \end{cases} \quad r \in U(0,1)$$

Mutação não-uniforme (Michalewicz, 1994):

$$c_i = \begin{cases} p_i + (b_i - p_i)f(G) & \text{se } r_1 < 0,5 \text{ e } i = j \\ p_i - (p_i - a_i)f(G) & \text{se } r_1 \geq 0,5 \text{ e } i = j \\ p_i & \text{caso contrário} \end{cases}$$

$$f(G) = \left(r_2 \left(1 - \frac{G}{G_{\max}} \right) \right)^b$$

$$r_1 \text{ e } r_2 \in U(0,1)$$

$$b = 6$$

Seleção

- **Torneio**

- Proposto por Goldberg e Deb em Goldberg and Deb (1991).
- Seleciona aleatoriamente um número fixo de indivíduos Q e o melhor entre esses indivíduos é escolhido para cruzamento.

- **Truncamento**

- Truncamento foi proposto por Mühlenbein e Schlierkamp-Voose em Mühlenbein and Schlierkamp-Voosen (1993).
- Um número $M \geq N$ de indivíduos é gerado e os melhores N indivíduos são selecionados para formar a próxima população.

Seleção

disputa	indivíduo	cromossomo	<i>fitness</i>	vencedor
1	1	110111	5	1
	4	011001	3	
2	6	010100	2	8
	8	111011	5	
3	9	101101	4	9
	11	010001	2	
4	13	111000	3	13
	15	000001	1	

Seleção

- **Roleta**

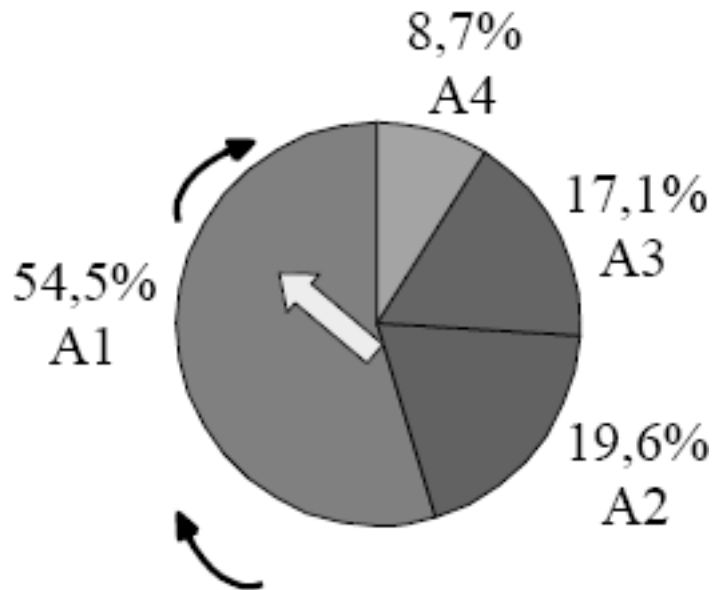
Algoritmo 3.2: Roda da Roleta

```
total  $\leftarrow \sum_{i=1}^N f_i$  /* a soma das aptidões de todos os cromossomos da
população */
rand  $\leftarrow$  randômico(0, total)
total parcial  $\leftarrow 0$ 
 $i \leftarrow 0$ 
repetir
     $i \leftarrow i + 1$ 
    total parcial  $\leftarrow$  total parcial +  $f_i$ 
até total parcial  $\geq$  rand
retornar o cromossomo  $s_i$ 
```

Seleção

- Mais chance para os mais aptos

- Problemas com valores negativos



$$f_i = 2(N - i)/(N - 1)$$

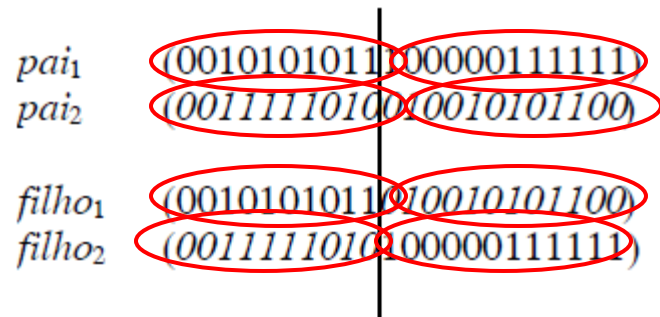


Figura 3.3 - Crossover

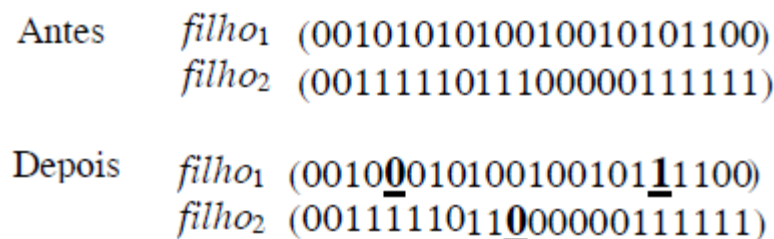


Figura 3.4 - Mutação

Índice Original	População Intermediária (<i>Mating pool</i>)	Crossover e Mutação	Primeira Geração (o sublinhado indica mutação)	Ponto de Corte
1	1101000000011110110111		1101000000011000011100	12
5	10011101101111000011100		1001110110111110110111	12
12	0001101001100010101111		000110100110 <u>0</u> 010010110	12
6	0000110011111010010110		00 <u>1</u> 01100111100 <u>0</u> 0101111	12
19	0010001101001100101100	•	0 <u>1</u> 10001101000100011101	9
17	0100011001000100011101		0100011001001100101100	9
15	1000100011110001000011		1000100011110001010010	17
7	0011000000100111010010		00 <u>0</u> 1000000100111000011	17

$$s_1 = 1000101110110101000111 \quad b_{10} = (1000101110110101000111)_2 = 2288967$$

$$x = \min + (\max - \min) \frac{b_{10}}{2^l - 1} \quad x_1 = -1 + (2 + 1) \frac{2.288.967}{(2^{22} - 1)} = 0,637197$$

E tem mais....

- Elitismo
- Reinicialização
- Migração

Algoritmo Genético - I

```
1: procedure Algoritmo Genético
2:    $t \leftarrow 0$ 
3:   Gerar População Inicial  $P(t)$ 
4:   Avaliar  $P(t)$ 
5:   while not criteriodeparada do
6:     for  $i \leftarrow 1$  to  $N/2$  do
7:       Selecionar dois indivíduos de  $P(t)$ .
8:       Aplicar crossover aos dois indivíduos com probabilidade  $p_c$ .
9:       Mutar os novos indivíduos gerados com probabilidade  $p_m$ .
10:      Inserir os novos indivíduos em  $P(t + 1)$ .
11:    end for
12:     $t \leftarrow t + 1$ 
13:  end while
14: end procedure
```

Figure: Pseudocódigo para AG.

Algoritmo Genético - II

Algoritmo 3.1: Algoritmo Genético típico

Seja $S(t)$ a população de cromossomos na geração t .

$t \leftarrow 0$

inicializar $S(t)$

avaliar $S(t)$

enquanto o critério de parada não for satisfeito **faça**

$t \leftarrow t + 1$

 selecionar $S(t)$ a partir de $S(t-1)$

 aplicar *crossover* sobre $S(t)$

 aplicar mutação sobre $S(t)$

 avaliar $S(t)$

fim enquanto

Algoritmo Genético - III

Método AG

Início

repita

 inicializaPopulações();

repita

para i = 1 **até** numeroDePopulações **faça**

 crossoverPop(i);

 mutacaoPop(i);

 estruturaPop(i);

até convergir;

 migração();

até critério de parada;

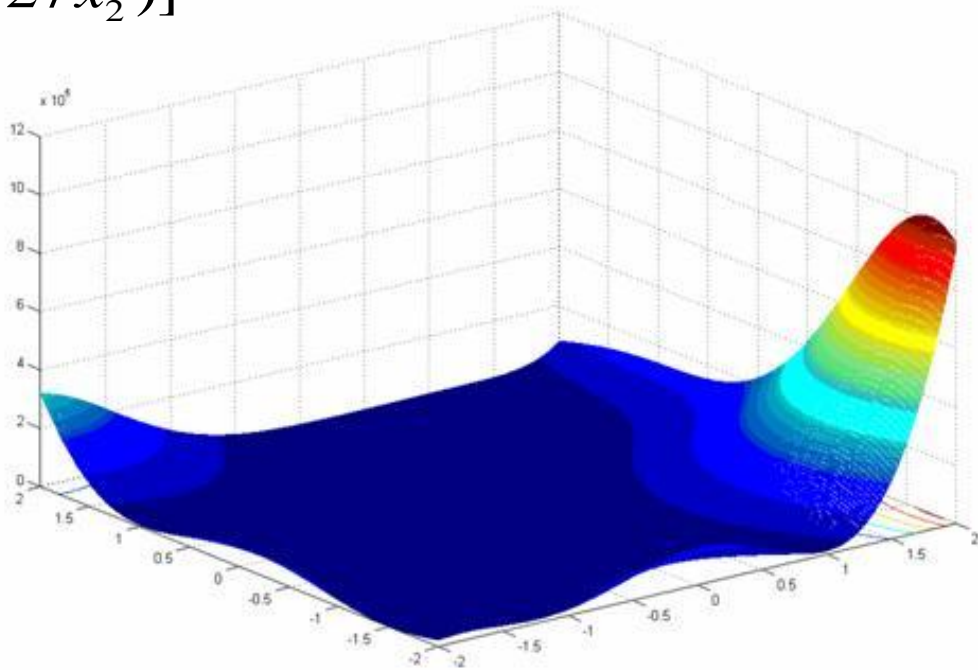
fim

Atividade

Funções

Goldstein & Price

$$GP(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$



domínio $[-2, 2]$

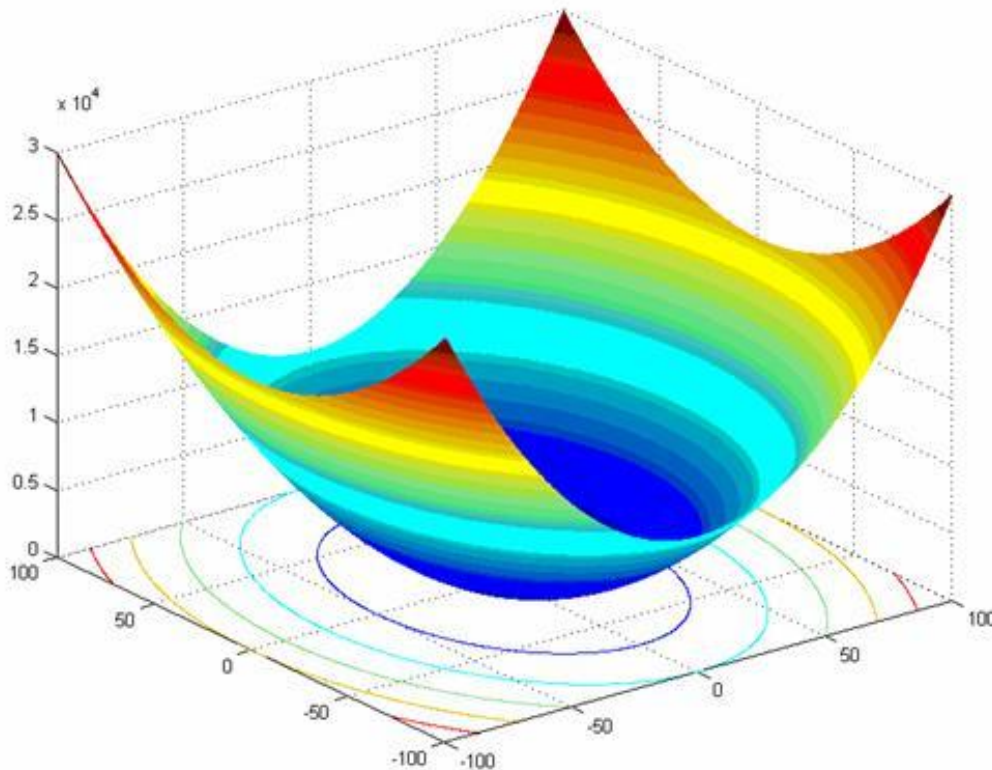
4 mínimos locais

1 mínimo Global

$$GP(0, -1) = 3.$$

Funções Bohachevsky

$$B2(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$



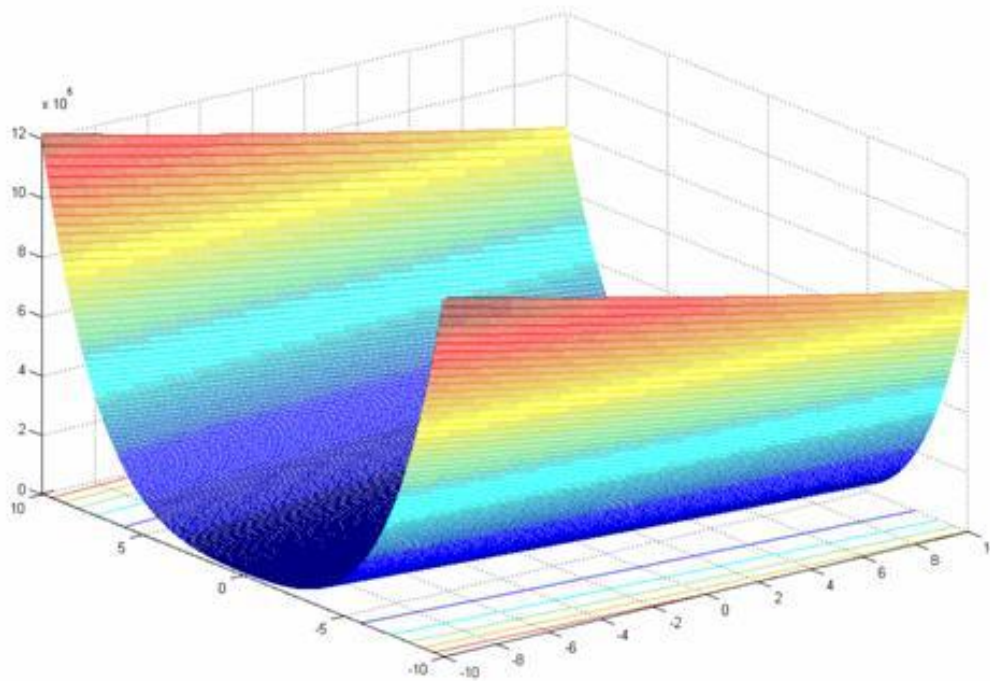
domínio $[-50, 50]$

1 ótimo global

$B2(0,0) = 0$.

Funções Rosenbrock

$$R_n(x) = \sum_{j=1}^{n-1} [(1 - x_j)^2 + 100(x_{j+1} - x_j^2)^2]$$



domínio $[-30, 30]$

$$R_2(1, 1) = 0$$

Resultados

- Métricas para comparação:
 - Número de funções avaliadas
 - Quantidade de fitness determinado até que o critério de parada seja atendido.
 - Critério de parada definido pelo método. **Qual devo utilizar???**
 - Taxa de sucesso após **10** (se possível 100) execuções.
 - Melhor indivíduo e ótimo devem satisfazer:
$$/f_{\text{Best}} - f_{\text{Ótimo}}/ < \mathbf{0.01}$$
- Definir pelo menos um tipo de AG.
 - Implementar com representação real, se possível, binária.
- Avaliar os AGs em todas as funções apresentadas.

Referências

- Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.
- Eshelman, L. J.; Shaffer, D. J. (1992). Real-coded genetic algorithms and interval-schemata. In: WHITLEY, D. L. (ed). *Foundations of Genetic Algorithms 3*. San Mateo, CA : Morgan Kaufman, p.187- 203.
- Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. 3.ed. Springer-Verlag .
- Goldberg, D. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic Algorithms. Morgan Kaufmann.