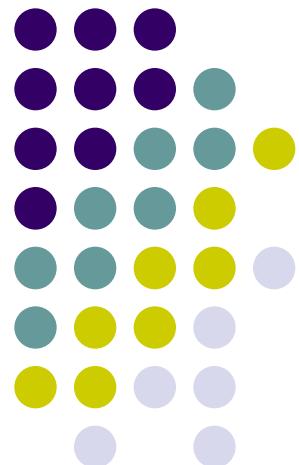


# **Inteligência Artificial**

---

**Prof. Celso Camilo**

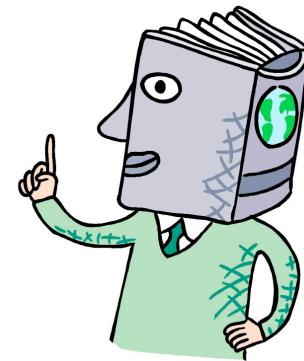


# Introdução



O que é Inteligência ?

# O que você considera inteligente ?



# Introdução

O que é Inteligência ?

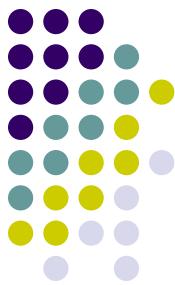


“capacidade de aprender, apreender e  
interpretar ”

Dicionário Aurélio

# Introdução

## O que é Inteligência ?



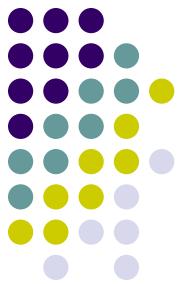
“A habilidade de aprender ou entender ou tratar com situações novas.”

“A habilidade de aplicar o conhecimento para manipular um ambiente ou pensar abstratamente como avaliado por critérios objetivos (como testes).”

Webster’s New Collegiate Dictionary, 1975

# Introdução

## O que é Inteligência ?



“Inteligência é a capacidade do sistema adaptar seu comportamento para encontrar seus objetivos em uma faixa de ambientes.”

David Fogel (1995)

# Introdução



## O que é Inteligência ?

- No livro do Andreas ele relata outras fontes que incluem aspectos como criatividade, habilidade, emoção e intuição na definição de inteligência

Andreas Book (CI – An Introduction)



# Intelligence Is...

- Capacity to learn from experience
- Ability to adapt to different contexts
- The use of metacognition to enhance learning



# Emotional Intelligence

- Mayer & Salovey (1997)

“The capacity to reason about emotions, and of emotions to enhance thinking. It includes the abilities to accurately perceive emotions, to access and generate emotions so as to assist thought, to understand emotions and emotional knowledge, and to reflectively regulate emotions so as to promote emotional and intellectual growth”



# Social Intelligence

- Ability to get along with others
- Knowledge of social matters
- Insight into moods or underlying personality traits of others

# Introdução



Os computadores podem ser inteligentes ?

# Introdução



## Os computadores podem ser inteligentes ?

- Em meados de 1900, Alan Turing acreditava que não há nada que a mente faz que um computador bem projetado não possa fazer.
- Depois de 20 anos, o seu pensamento ainda é considerado visionário.
- Em 1950 Turing publicou o seu teste para medir a inteligência do computador – Teste de Turing.

# Introdução



## Teste de Turing

- O teste consiste em uma pessoa fazer perguntas pelo teclado para uma outra pessoa e um computador.
- Se o interrogador não conseguir distinguir a máquina da pessoa, então o computador é inteligente.

# Introdução

- Inspiração na Natureza



*Lógica Fuzzy* - *processamento linguístico*

*Redes Neurais* - *neurônios biológicos*

*Algoritmos Genéticos* – *evolução biológica*

*Sistemas Híbridos* – *aspectos combinados*

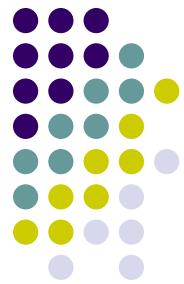
# Suporte a Decisão



- *Lógica Fuzzy*
- *Redes Neurais*
- *Algoritmos Genéticos*
- *Sistemas Híbridos*

- **Aquisição de Conhecimento**
- **Previsão**
- **Otimização**
- **Controle**
- **Planejamento**
- **Data Mining**
- **Análise de Risco**
- **Detecção de Fraude**

# Introdução – Áreas de Aplicação



Energia



Finanças



Telecomunicações



Medicina



Meio-Ambiente



Comércio



Indústria



# Algoritmos Genéticos



# Conceitos Básicos

→ Algoritmo de *busca/otimização* inspirado na *seleção natural* e *reprodução genética*.



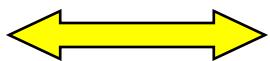
# Conceitos Básicos

- Algoritmo de *busca/otimização* inspirado na seleção *natural* e reprodução *genética*.
- Combina *sobrevivência do mais apto* e *cruzamento aleatório* de informação



# Analogia com a Natureza

Evolução Natural



Alg. Genéticos

- Indivíduo
- Cromossoma
- Reprodução Sexual
- Mutação
- População
- Gerações
- Meio Ambiente

- Solução
- Representação
- Operador Cruzamento
- Operador Mutação
- Conjunto de Soluções
- Ciclos
- Problema

# Qual a finalidade de Algoritmos Genéticos?

**Algoritmos Genéticos empregam um processo *adaptativo* e *paralelo* de busca de soluções em *problemas complexos*.**

# Qual a finalidade de Algoritmos Genéticos?



- **Adaptativo**

- informação corrente influencia a busca futura

- **Paralelo**

- várias soluções consideradas a cada momento

- **Problema Complexo**

- de difícil formulação matemática ou com grande espaço de busca (grande número de soluções)

# Problema Complexo

## Exemplo:

Maximizar  $f(x) = x^2$  : encontrar  $x \in (0 \dots 2^L - 1)$  para  $f(x) = \text{máx}$

$2^L$	<b>Número de Pontos no Espaço</b>	<b>Tempo de Busca</b> $10^9$ inst/seg
-------	---------------------------------------	--

# Problema Complexo

## Exemplo:

Maximizar  $f(x) = x^2$  : encontrar  $x \in (0 \dots 2^L - 1)$  para  $f(x) = \text{máx}$

<b><math>2^L</math></b>	<b>Número de Pontos no Espaço</b>	<b>Tempo de Busca</b> $10^9$ inst/seg
L=3	8	< 1 seg

# Problema Complexo

## Exemplo:

Maximizar  $f(x) = x^2$  : encontrar  $x \in (0 \dots 2^L - 1)$  para  $f(x) = \text{máx}$

<b><math>2^L</math></b>	<b>Número de Pontos no Espaço</b>	<b>Tempo de Busca</b> $10^9$ inst/seg
L=3	8	< 1 seg
L=10	1024	< 1 seg

# Problema Complexo

## Exemplo:

Maximizar  $f(x) = x^2$  : encontrar  $x \in (0 \dots 2^L - 1)$  para  $f(x) = \text{máx}$

<b><math>2^L</math></b>	<b>Número de Pontos no Espaço</b>	<b>Tempo de Busca</b> $10^9$ inst/seg
L=3	8	< 1 seg
L=10	1024	< 1 seg
L=30	1 bilhão	1 seg

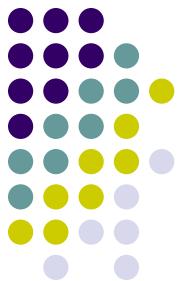
# Problema Complexo

## Exemplo:

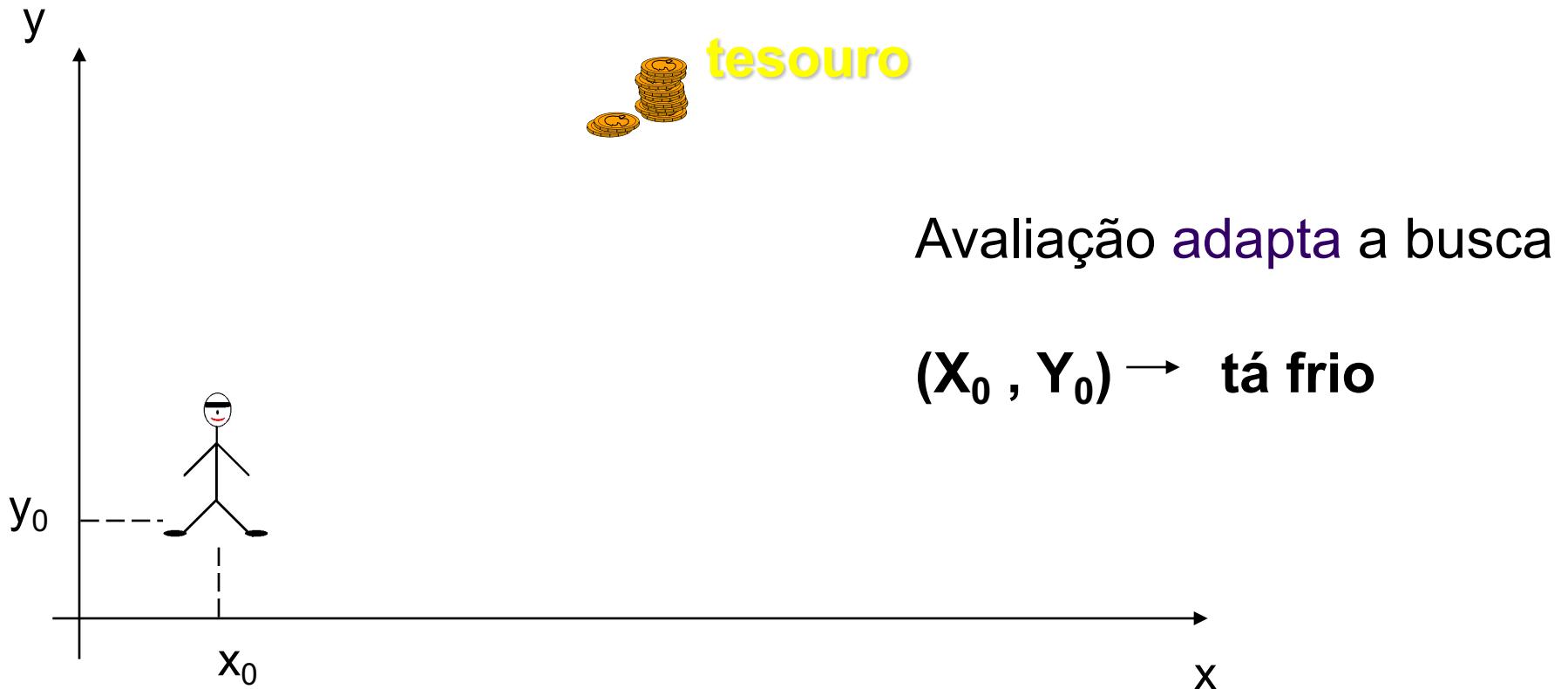
Maximizar  $f(x) = x^2$  : encontrar  $x \in (0 \dots 2^L - 1)$  para  $f(x) = \text{máx}$

<b><math>2^L</math></b>	<b>Número de Pontos no Espaço</b>	<b>Tempo de Busca</b> $10^9$ inst/seg
L=3	8	< 1 seg
L=10	1024	< 1 seg
L=30	1 bilhão	1 seg
L=90	$10^{27}$	15 bilhões de anos

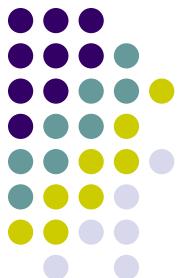
# Problema da Cabra Cega



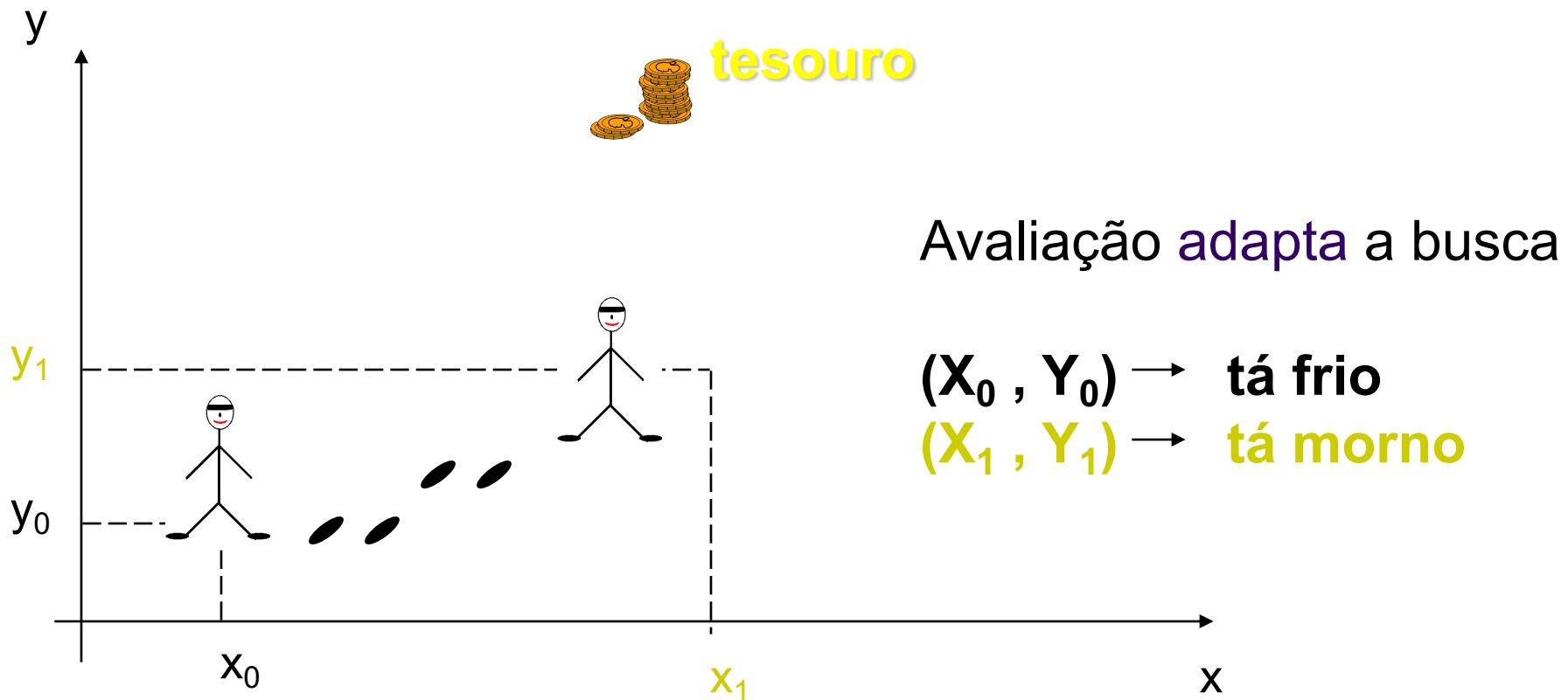
**Busca de objetivo escondido em uma área**



# Problema da Cabra Cega



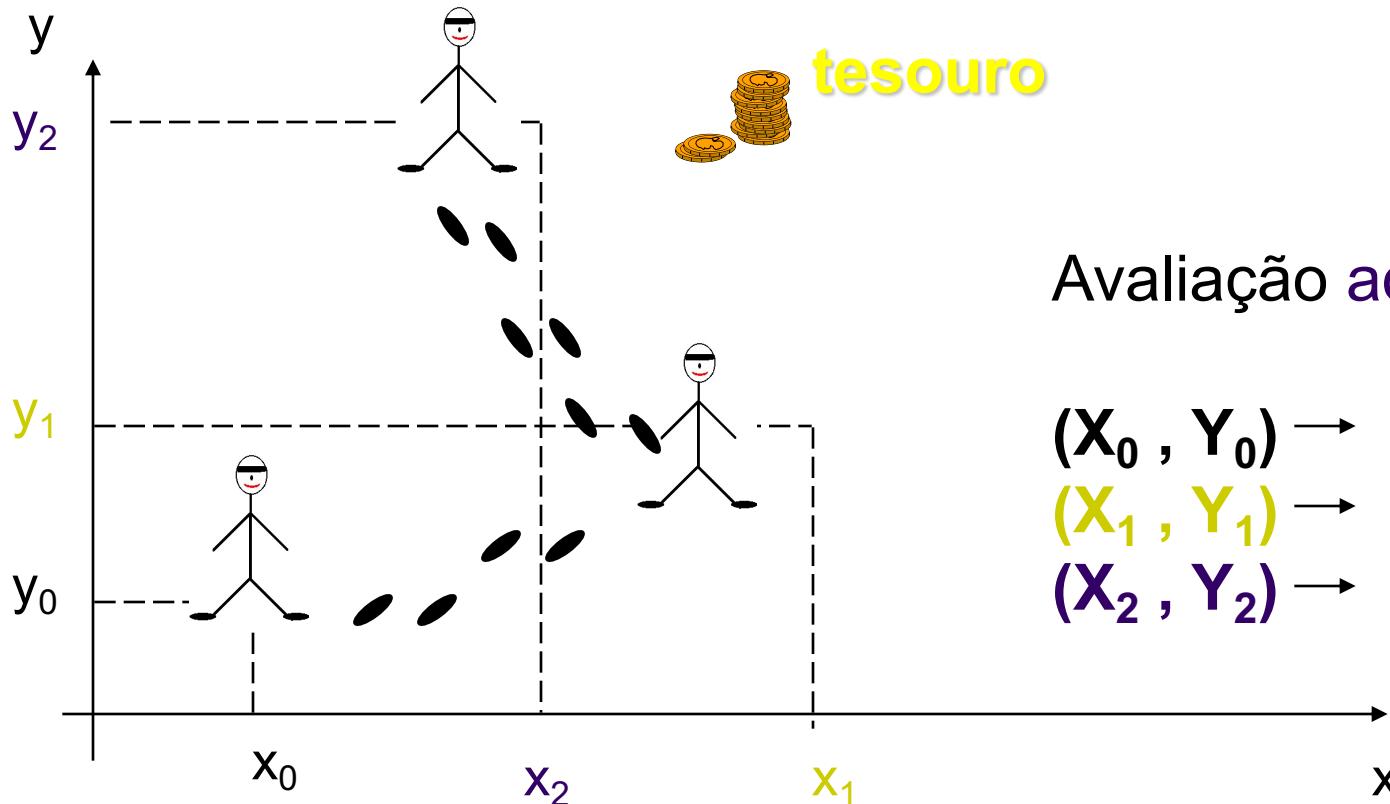
Busca de objetivo escondido em uma área





# Problema da Cabra Cega

Busca de objetivo escondido em uma área

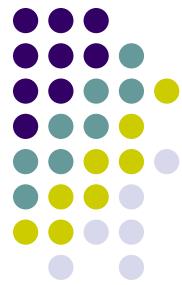


$(X_0 , Y_0) \rightarrow$  tá frio

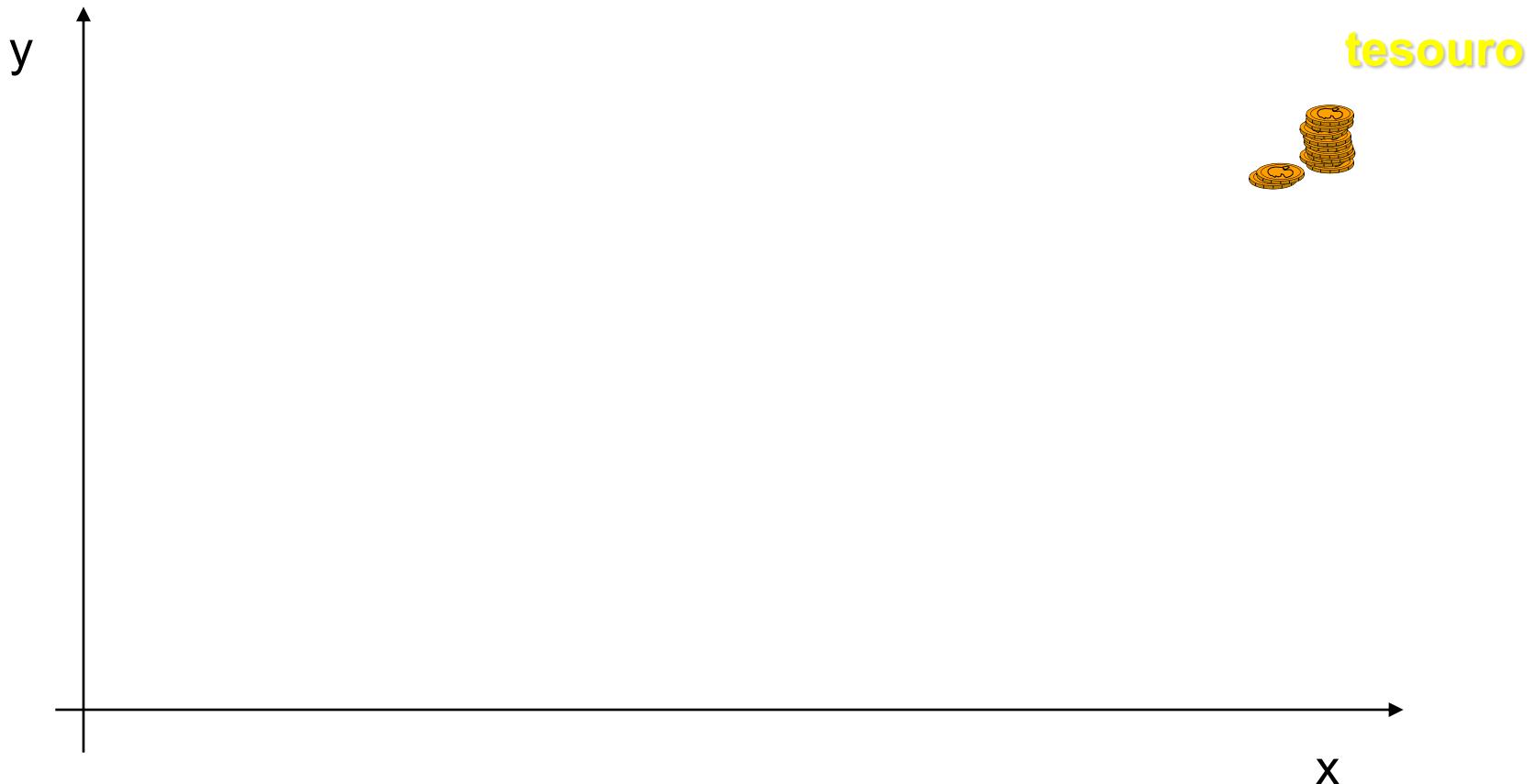
$(X_1 , Y_1) \rightarrow$  tá morno

$(X_2 , Y_2) \rightarrow$  tá quente!

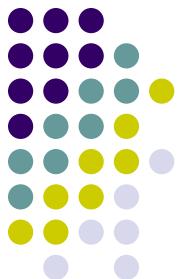
# Problema da Cabra Cega



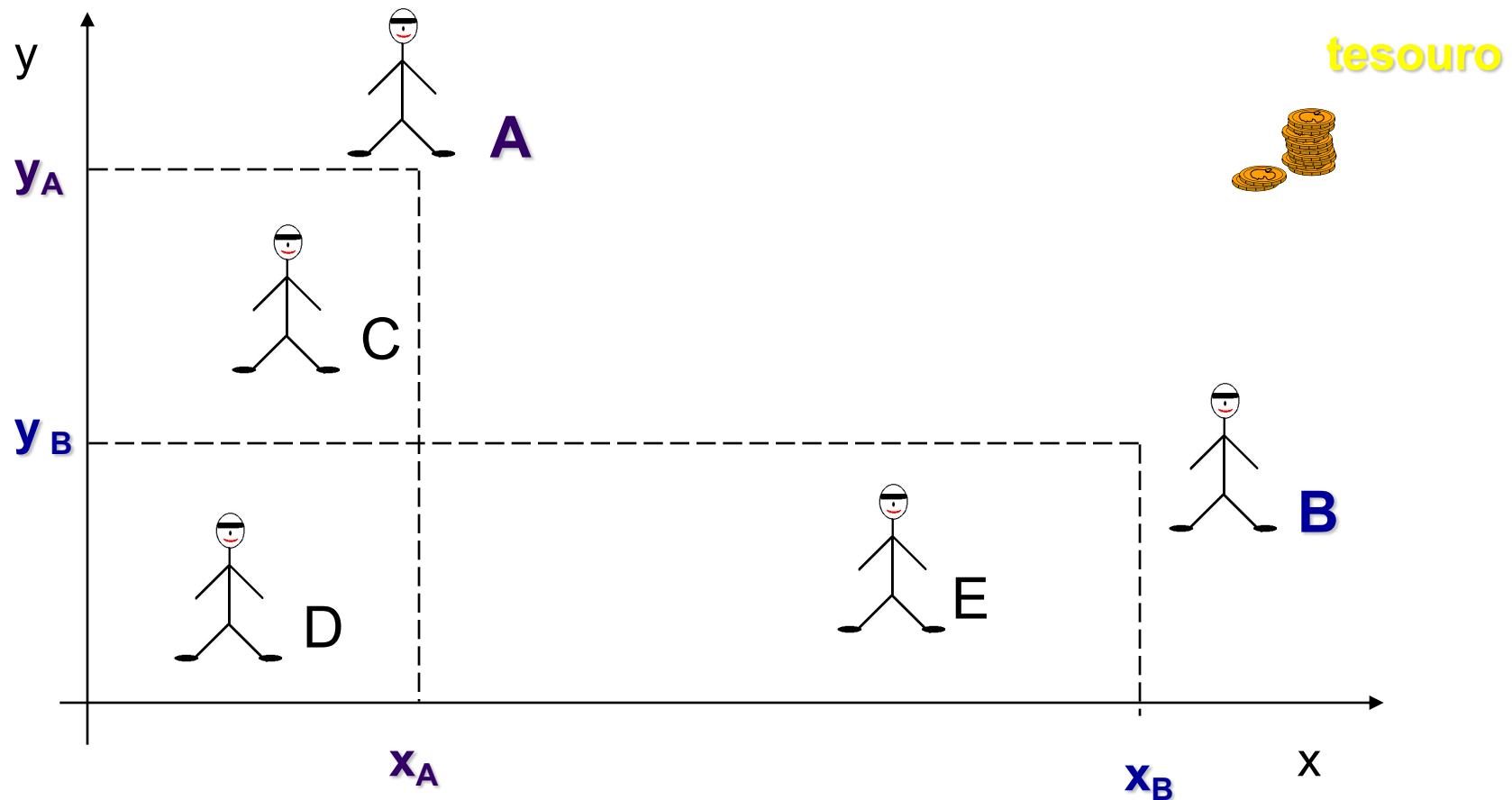
Área Muito Grande → Busca Paralela



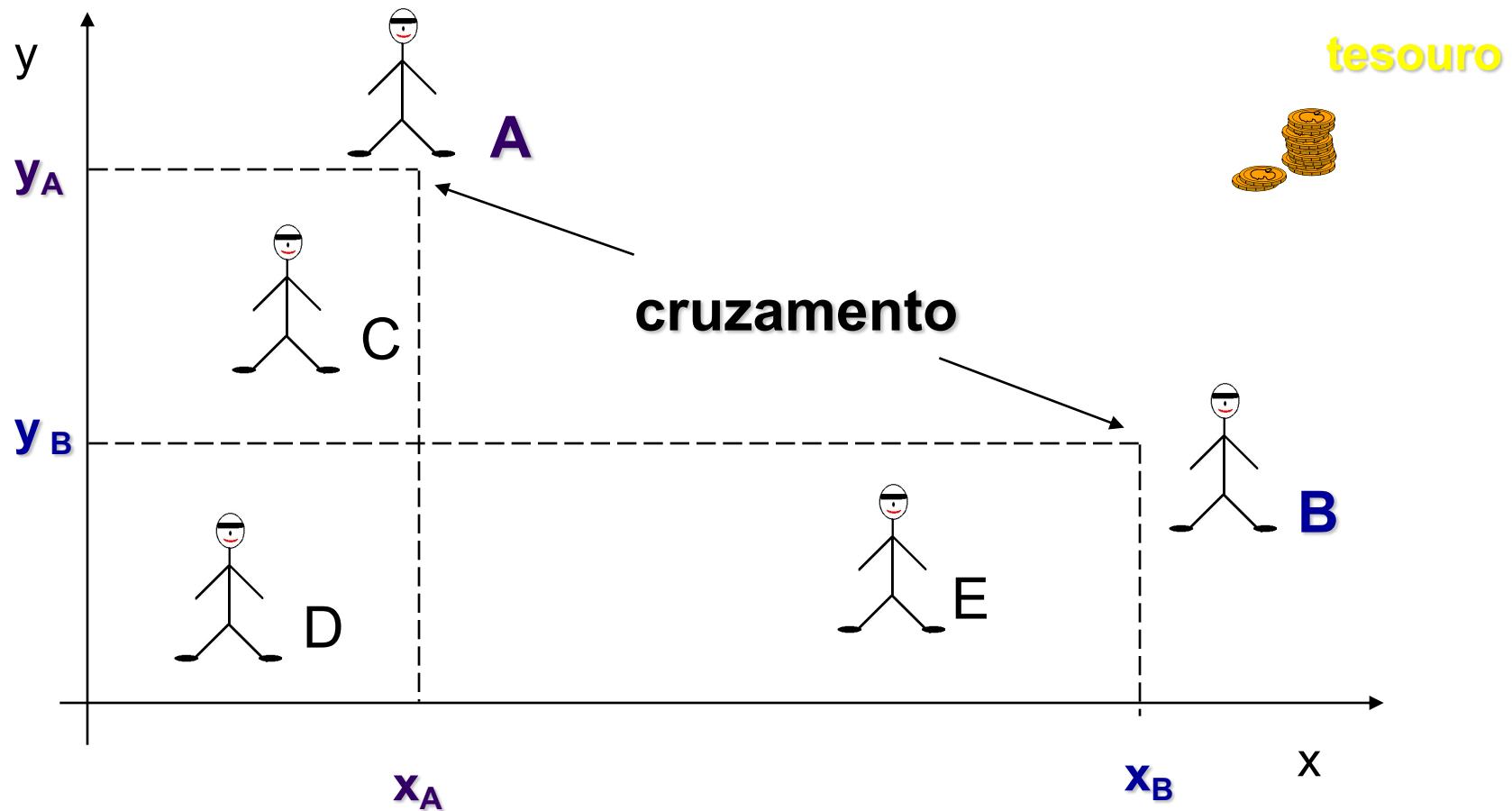
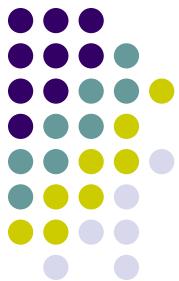
# Problema da Cabra Cega



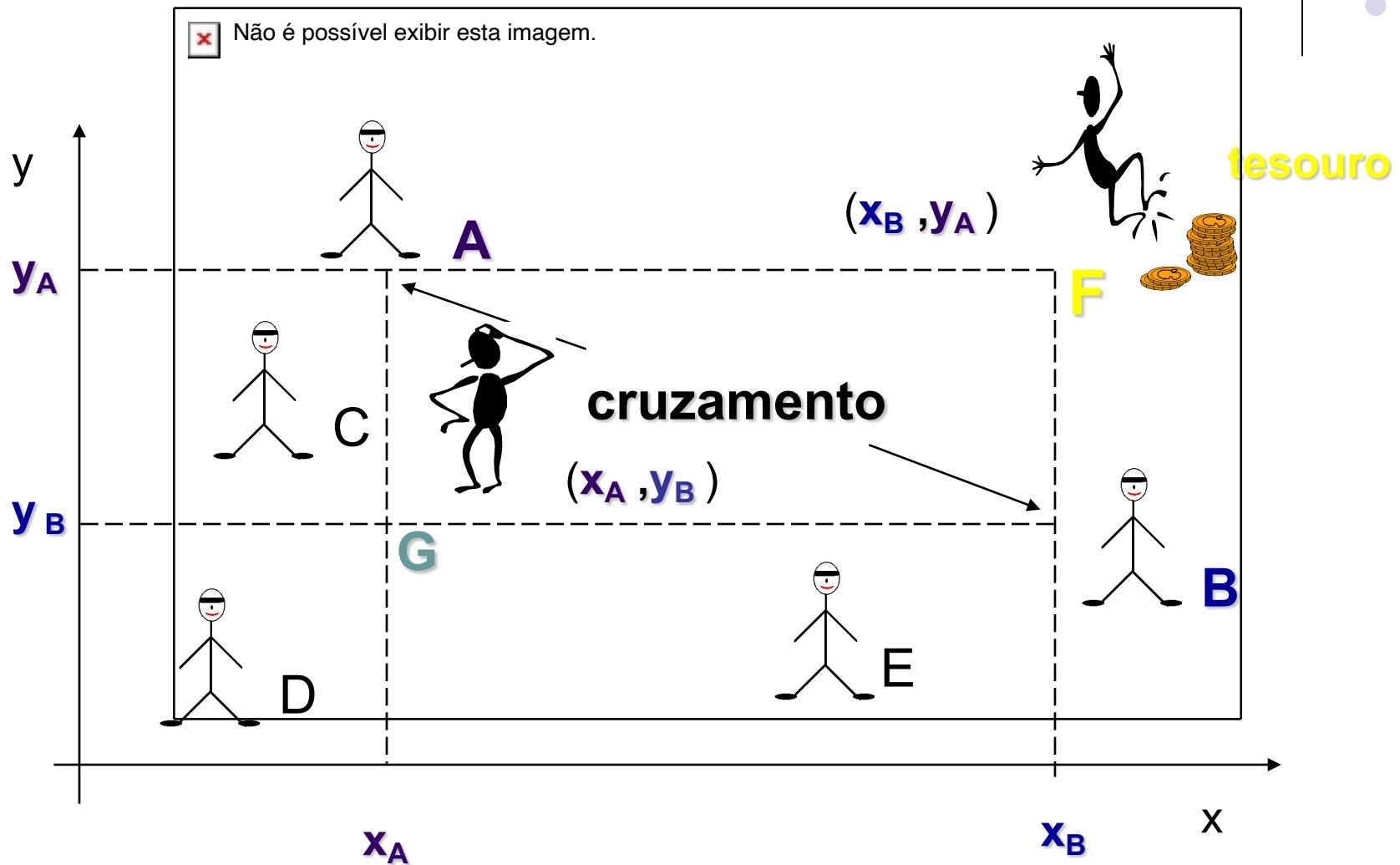
Área Muito Grande  $\rightarrow$  Busca Paralela

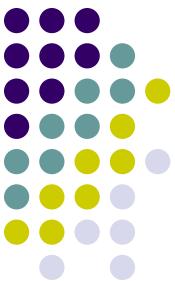


# Problema da Cabra Cega



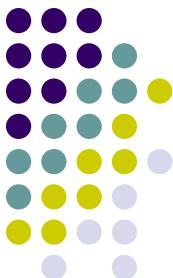
# Problema da Cabra Cega





# Operações Básicas

- **Seleção:** privilegia os indivíduos mais aptos



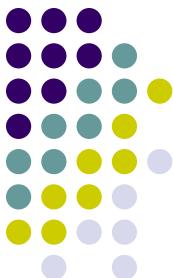
# Operações Básicas

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos (ex: palavras binárias) são reproduzidos com base na aptidão



# Operações Básicas

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos (ex: palavras binárias) são reproduzidos com base na aptidão
- **Crossover:** troca de genes (pedaços de palavras)



# Operações Básicas

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos (ex: palavras binárias) são reproduzidos com base na aptidão
- **Crossover:** troca de genes (pedaços de palavras)
- **Mutação:** troca aleatória de um gene (bit da palavra)



# Exemplo

## Problema:

→ Achar o valor máximo para  $f(x) = x^2$ ,  $x$  no limite de **0 a 63**.



# Exemplo

## Problema:

→ Achar o valor máximo para  $f(x) = x^2$ ,  $x$  no limite de 0 a 63.

## Representação da Solução:

→ Palavras binárias representando sucessivas potências de 2.

011100 => Representa 28

110101 => Representa 53 (uma solução mais apta)



# Seleção em Algoritmos Genéticos

## População

Cromossoma	Palavra	X	Aptidão ( $x^2$ )
A	100100	36	1296
B	010010	18	324
C	010110	22	484
D	000001	1	1

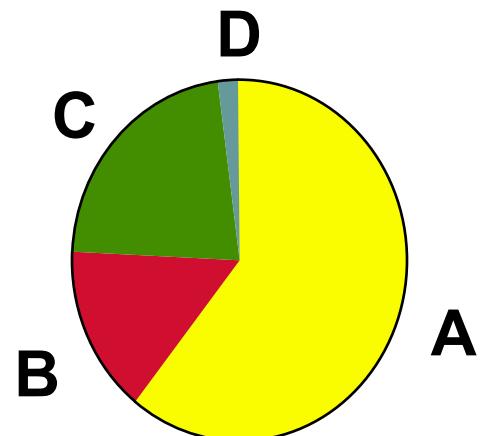


# Seleção em Algoritmos Genéticos

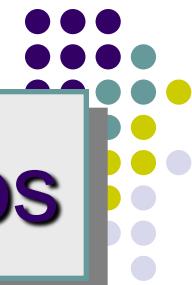
População

Cromossoma	Palavra	X	Aptidão ( $x^2$ )
A	100100	36	1296
B	010010	18	324
C	010110	22	484
D	000001	1	1

Seleção

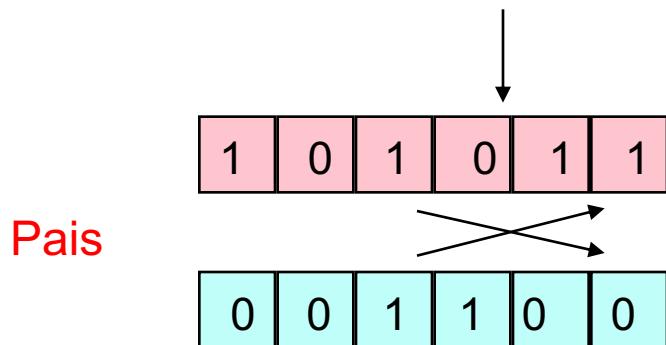


Probabilidade de Seleção  $\approx$  Aptidão do Cromossoma



# Operadores de Algoritmos Genéticos

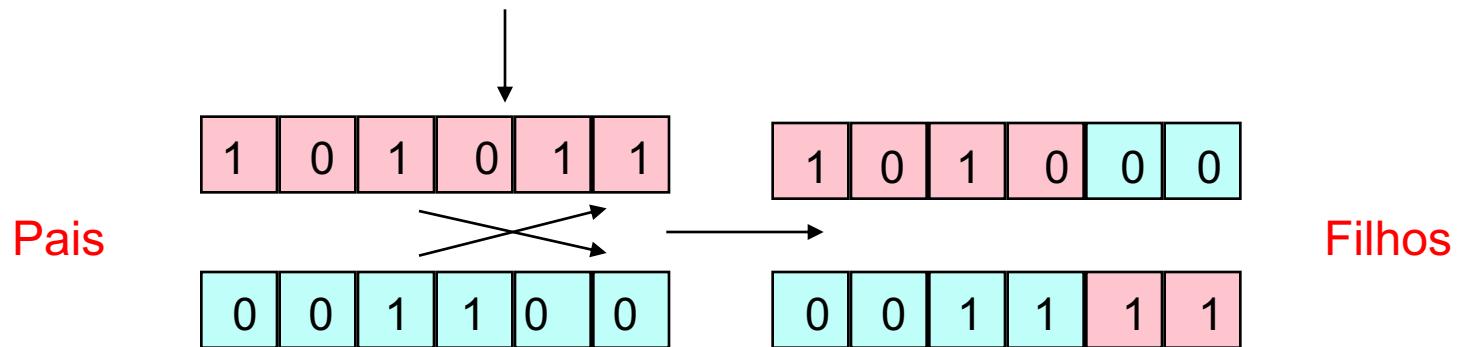
## Crossover





# Operadores de Algoritmos Genéticos

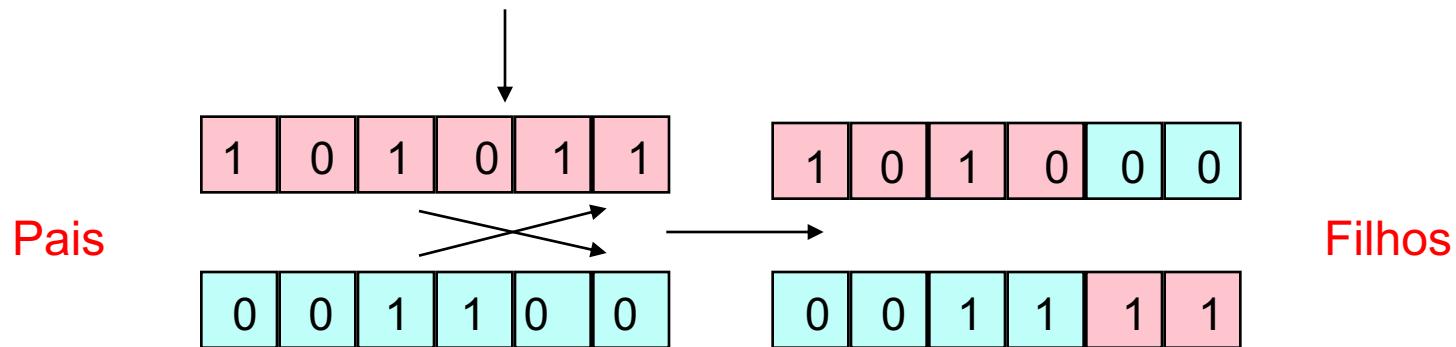
## Crossover



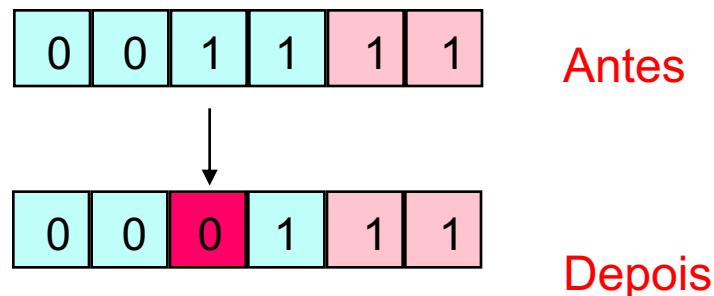


# Operadores de Algoritmos Genéticos

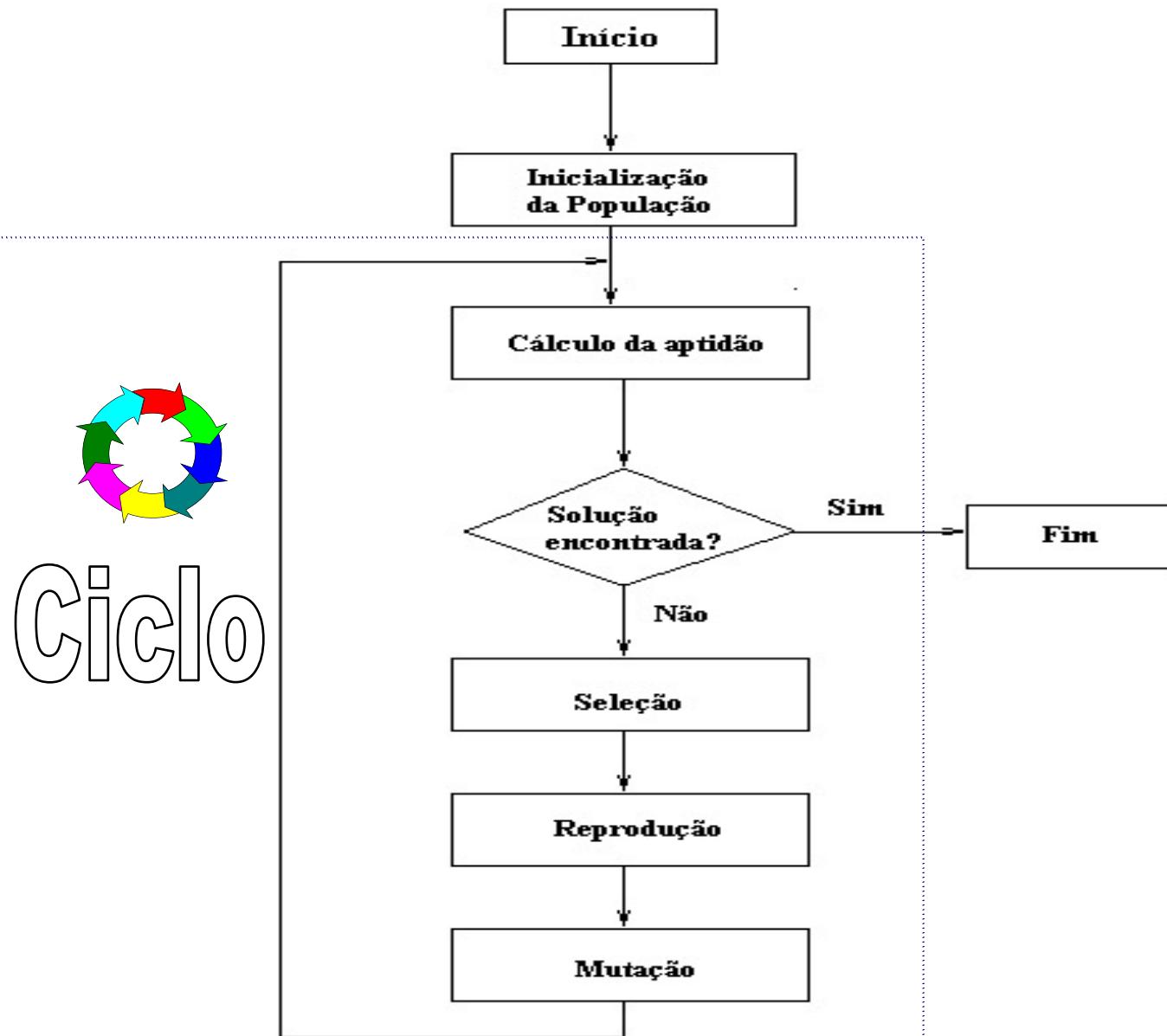
## Crossover

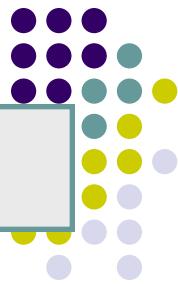


## Mutação



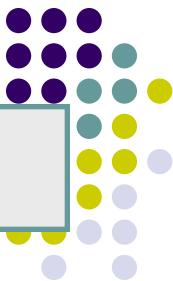
# Fluxo do Algoritmo Genético





# Ciclo do Algoritmo Genético

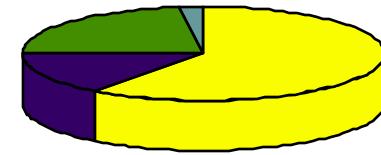
Cromossoma	Palavra	Aptidão
A	100100	1296
B	010010	324
C	010110	484
D	000001	1

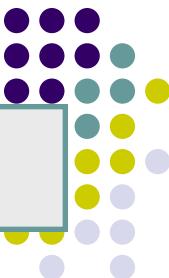


# Ciclo do Algoritmo Genético

Cromossoma	Palavra	Aptidão
A	100100	1296
B	010010	324
C	010110	484
D	000001	1

Pais

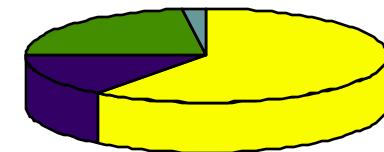




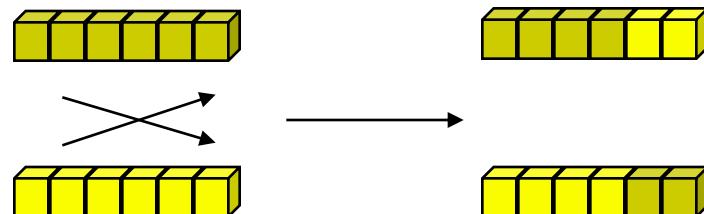
# Ciclo do Algoritmo Genético

Cromossoma	Palavra	Aptidão
A	100100	1296
B	010010	324
C	010110	484
D	000001	1

Pais



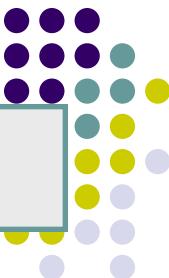
Crossover



Reprodução



Mutação



# Ciclo do Algoritmo Genético

Avaliação  
dos Filhos

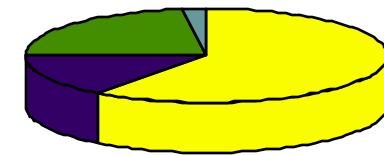
Cromossoma	Palavra	Aptidão
A	100100	1296
B	010010	324
C	010110	484
D	000001	1

Pais

$f( )$

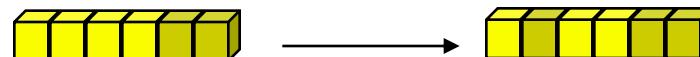
Crossover

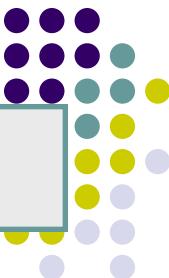
Filhos



Reprodução

Mutação





# Ciclo do Algoritmo Genético

Avaliação  
dos Filhos

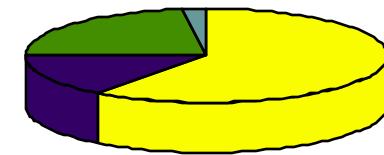
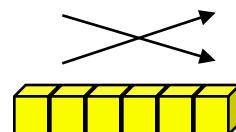
Cromossoma	Palavra	Aptidão
A	100100	1296
B	010010	324
C	010110	484
D	000001	1

Pais

$f( )$

Evolução

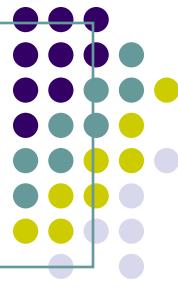
Filhos



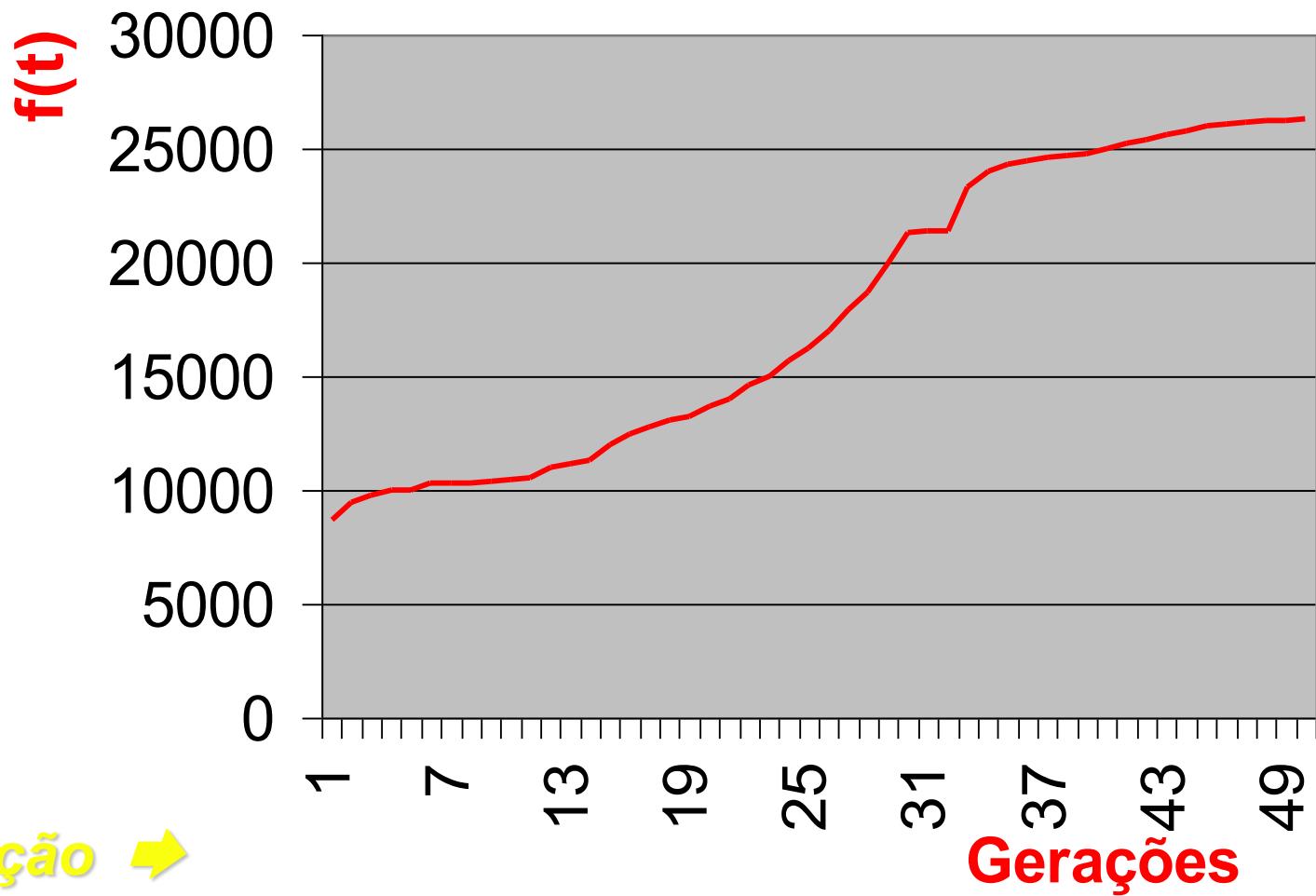
Reprodução



# Evolução dos Indivíduos



Melhor Indivíduo



Evolução ➔



# Avaliação

*Aplicado em problemas complexos de otimização – de difícil modelagem matemática, com variedade de regras e condições, ou com grande número de soluções a considerar.*



# Avaliação

## Vantagens

- Técnica de busca global (evita mínimos locais)
- Otimização de problemas complexos e mal estruturados
- Dispensa formulação matemática precisa do problema

## Desvantagens

- Precisão na representação do cromossoma
- Evolução demorada em alguns problemas
- Modelagem depende do habilidade do especialista em GA



# Aplicações Industriais

- ◆ **GENERAL ELECTRIC** - Otimização de Projeto de Motores DC
- ◆ **BRITISH GAS** - Otimização da Distribuição de Gás
- ◆ **BBN** - Roteamento de Telecomunicações
- ◆ **ATTAR** - Planejamento da Programação de TV

# Aplicações Comerciais



- ◆ **CAP VOLMAC** - Avaliação de Crédito e Análise de Risco
- ◆ **SEARCHSPACE** - Detecção de Fraude na Bolsa de Londres
- ◆ **IOC** - Planejamento dos Jogos Olímpicos
- ◆ **CAP Gemini** - Avaliação de Empréstimos e Financiamentos
- ◆ **GWI** - Modelagem Econômica
- ◆ **World Bank** - Geração de Regras de Negociação na Bolsa



# Desenvolvimento do AG



# Componentes de um Algoritmo Genético

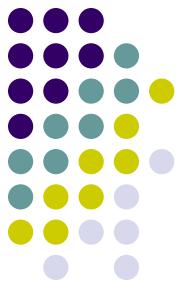
- 1. Problema**
- 2. Representação**
- 3. Decodificação**
- 4. Avaliação**
- 5. Operadores**
- 6. Parâmetros**



## 1. PROBLEMA

- **Estudo de Contexto do Problema:** Conhecer regras, restrições, objetivos, procedimentos em uso, etc.
- **GAs são indicados em problemas difíceis de otimização:**
  - ✿ muitos parâmetros e variáveis;
  - ✿ mal estruturados: com condições e restrições, difíceis de serem modeladas matematicamente;
  - ✿ grandes espaços de busca onde não é possível a busca exaustiva.

## 2. REPRESENTAÇÃO



Representação é fundamental na modelagem de um GA e deve:

- ✿ descrever o espaço de busca relevante ao problema;
- ✿ codificar geneticamente a “essência” do problema:  
evolução do “código” → evolução da solução
- ✿ ser compatível com os operadores (crossover e mutação)  
representação adequada → evolução, otimização



## 2. REPRESENTAÇÃO

Tipo de Problema     $\leftrightarrow$     Representação

- **Numérico**
- **Ordem**
- **Grupo**
- **Inteiro**
- **Misto**
- **Binário, Real**
- **Lista**
- **Vetor**
- **Inteiro**
- **Mista**



# Real representado por Binário

- Primeiro tipo de representação em Algoritmos Genéticos
- Número real é codificado através de um número binário de K bits
- Representação binária descreve um real em detalhes (genes):

$$13 \text{ em binário} = 1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 =$$

$$\begin{array}{r} \swarrow \quad \searrow \\ 8 + 4 + 1 \end{array}$$

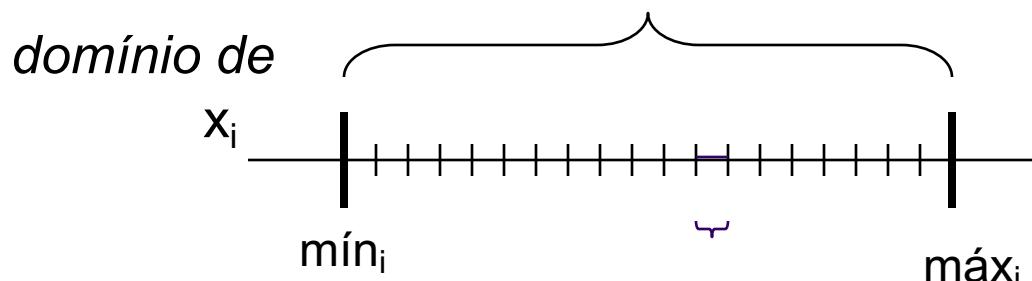


## BINÁRIO CODIFICANDO REAL

### Aspectos importantes:

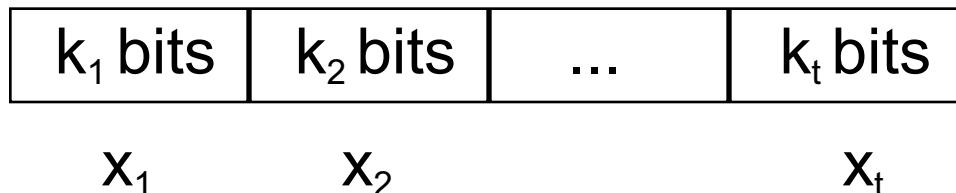
- ★ **variáveis** do problema ( $x_1, x_2, \dots, x_t$ )
- ★ **domínio** de valores:  $x_i \in (\min_i, \max_i)$  em  $\mathbb{R}$
- ★ **precisão**:  $p$  casas decimais

$(\max_i - \min_i) \times 10^p + 1$  diferentes soluções





## Representação:



onde,

$$2^{k_i} \geq (\text{máx}_i - \text{mín}_i) \times 10^p \rightarrow \text{Precisão} = \frac{(\text{máx}_i - \text{mín}_i)}{2^{k_i} - 1}$$

## Decodificação para Real:

$$x_{i \text{ real}} = x_{i \text{ bin}} \cdot \frac{(\text{máx}_i - \text{mín}_i)}{2^{k_i} - 1} + \text{mín}_i$$

Binário na base 10

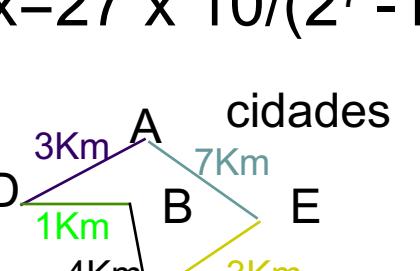
$$\begin{aligned} \text{se } x_{i \text{ bin}} = (0 \ 0 \dots 0) &\rightarrow x_{i \text{ real}} = \text{mín}_i \\ \text{se } x_{i \text{ bin}} = (1 \ 1 \dots 1) &\rightarrow x_{i \text{ real}} = \text{máx}_i \end{aligned}$$

### 3. DECODIFICAÇÃO



Construir a solução para o problema a partir de um cromossoma:

**Cromossomas “representam” soluções.**

Cromossoma	DECODIFICAÇÃO	Solução
0011011	bin → inteiro	x=27
0011011	$x=27 \times 10/(2^7 - 1)$	$x=2,1 \quad x \in [0,10]$ 1 casa decimal
ADBCE	 <p>A diagram showing five cities labeled A, B, C, D, and E arranged in a pentagonal shape. The distances between the cities are indicated by colored lines: A to D is purple (3Km), A to B is light blue (7Km), D to B is green (1Km), D to C is red (4Km), B to C is yellow (3Km), B to E is cyan, and E to C is orange. The word "cidades" is written above the diagram.</p>	<p>rota  <math>A \rightarrow D \rightarrow B \rightarrow C \rightarrow E</math></p>

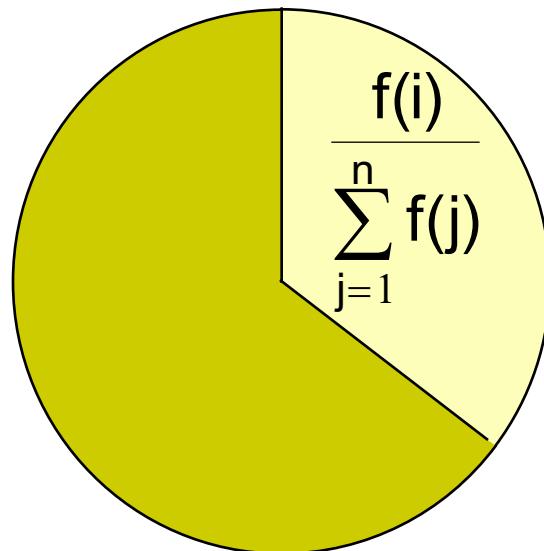


## 4. AVALIAÇÃO

Elo entre o algoritmo genético e o problema.

$f(\text{cromossoma})$  = medida numérica de aptidão

Chances de seleção são proporcionais à aptidão.





# Seleção pela Roleta

**Objetivo:** *Selecionar indivíduos aleatoriamente, proporcionando maiores chances de reprodução aos mais aptos.*

## Método por Computador

- Encontre a soma da aptidão de todos os membros da população  
 $A_T = \sum A_i \quad (0 \leq i \leq \text{pop\_size}-1)$
- Gere um número aleatório  $0 \leq \text{rand} \leq A_T$
- Pegue o primeiro membro da população  $I_k$  cuja aptidão somada às aptidões dos membros precedentes é maior ou igual a rand.

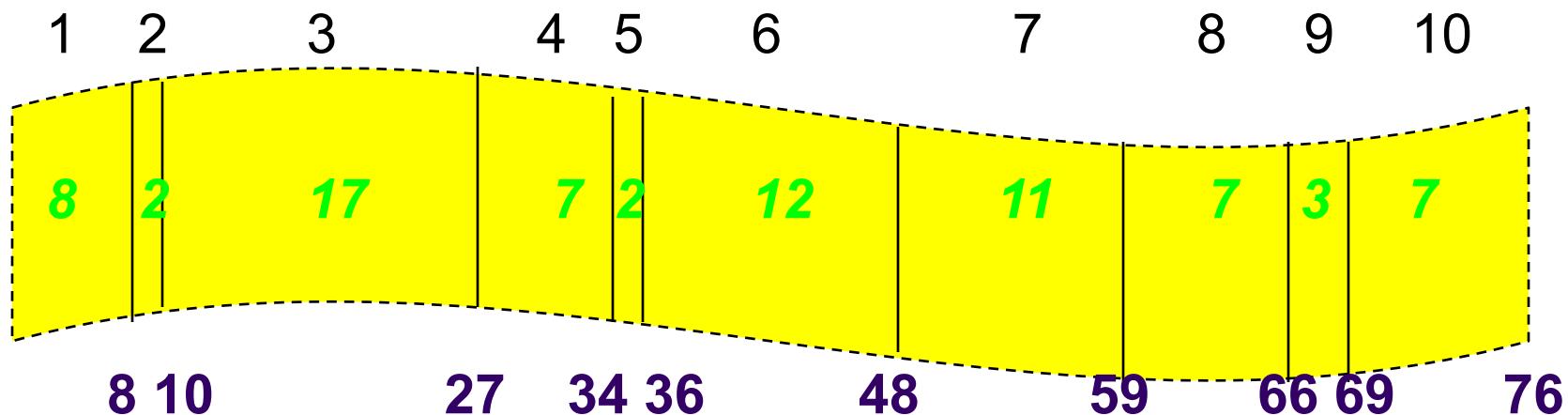
$$\sum A_i \geq \text{rand} \quad (i \leq k)$$

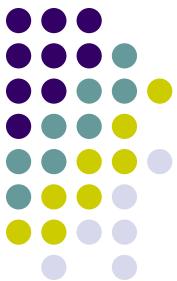


# Exemplo da Roleta

Cromossoma	1	2	3	4	5	6	7	8	9	10
Aptidão	8	2	17	7	2	12	11	7	3	7
$\Sigma A_i$	8	10	27	34	36	48	59	66	69	76

Número Aleatório	23	49	76	13	1	27	57
Selecionado	3	7	10	3	1	3	7





## 5. OPERADORES

Atuam no processo de criação de novos indivíduos (descendentes):

**1. Crossover**

**2. Mutação**

**3. Operadores específicos ao problema  
(heurísticos)**



# Crossover de 1 ponto de corte

- A cauda de dois cromossomas são trocadas a partir de posição escolhida aleatoriamente
- Taxa de Crossover : entre **60% e 90%**
  - Teste Verdadeiro → Efetua Cruzamento
  - Teste Falso → Copia os Genitores

	Cabeça			Cauda		
$P_1$	1	0	1	1	0	1
$P_2$	0	0	1	1	0	0
$F_1$	1	0	1	1	0	0
$F_2$	0	0	1	1	0	1

ponto de corte aleatório ↗



# Mutação por inversão de bit

- Inverte cada gene de um cromossoma se o teste de probabilidade for verdadeiro
- Taxa Mutação: entre **0,1% e 5%**
  - Teste Verdadeiro → troca bit
  - Teste Falso → mantém bit
  - **0,8% (0,008)**

Cromossoma

Número Aleatório

Novo Cromossoma

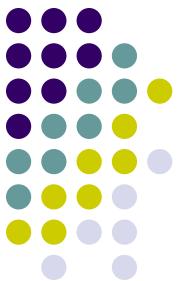
1	0	1	0	0,801	0,102	0,266	0,373	1	0	1	0
1	1	0	0	0,128	0,96	<b>0,005</b>	0,84	1	1	<b>1</b>	0
0	0	1	0	0,768	0,473	0,894	<b>0,001</b>	0	0	1	1



## 6. PARÂMETROS

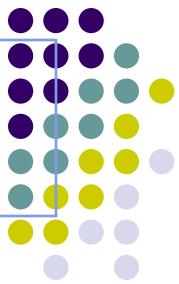
- TAMANHO\_POPULAÇÃO
- NÚMERO\_GERAÇÕES
- TAXA\_CROSSOVER
- TAXA\_MUTAÇÃO
- ...

# Desenvolvimento de um Algoritmo Genético



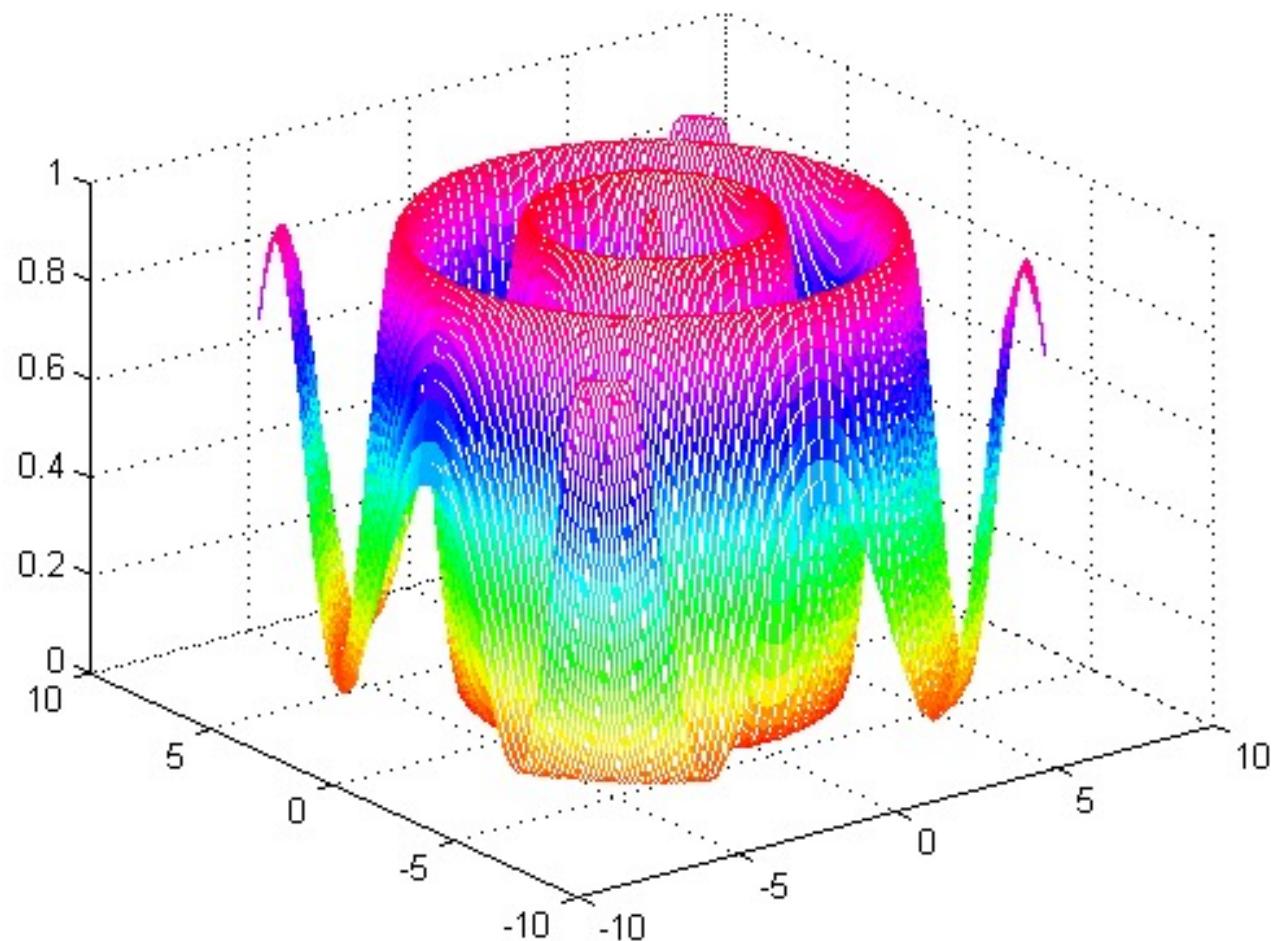
```
procedure algoritmo_genético
begin
    t = 0                                ; primeira geração
    inicializa P(t)                      ; população inicial aleatória
    avalia P(t)                          ; calcula f(i) p/ cada indivíduo
    while (not condição_parada) do
        begin
            t = t + 1                      ; próxima geração
            seleciona P(t) de P(t-1)
            altera P(t)                  ; crossover e mutação
            avalia P(t)                  ; calcula f(i) p/ cada indivíduo
        end
    end
```

# Sistemas de Desenvolvimento



-  • **ICADEMO**
  -  • **WinGenesis**
  - **GENOCOP**
  - **GeneHunter**
  - **Evolver 4.0**
  - **Open Beagle**
  - **Sugal**
  - **GACOM: Biblioteca de Componentes (C#)**
- 

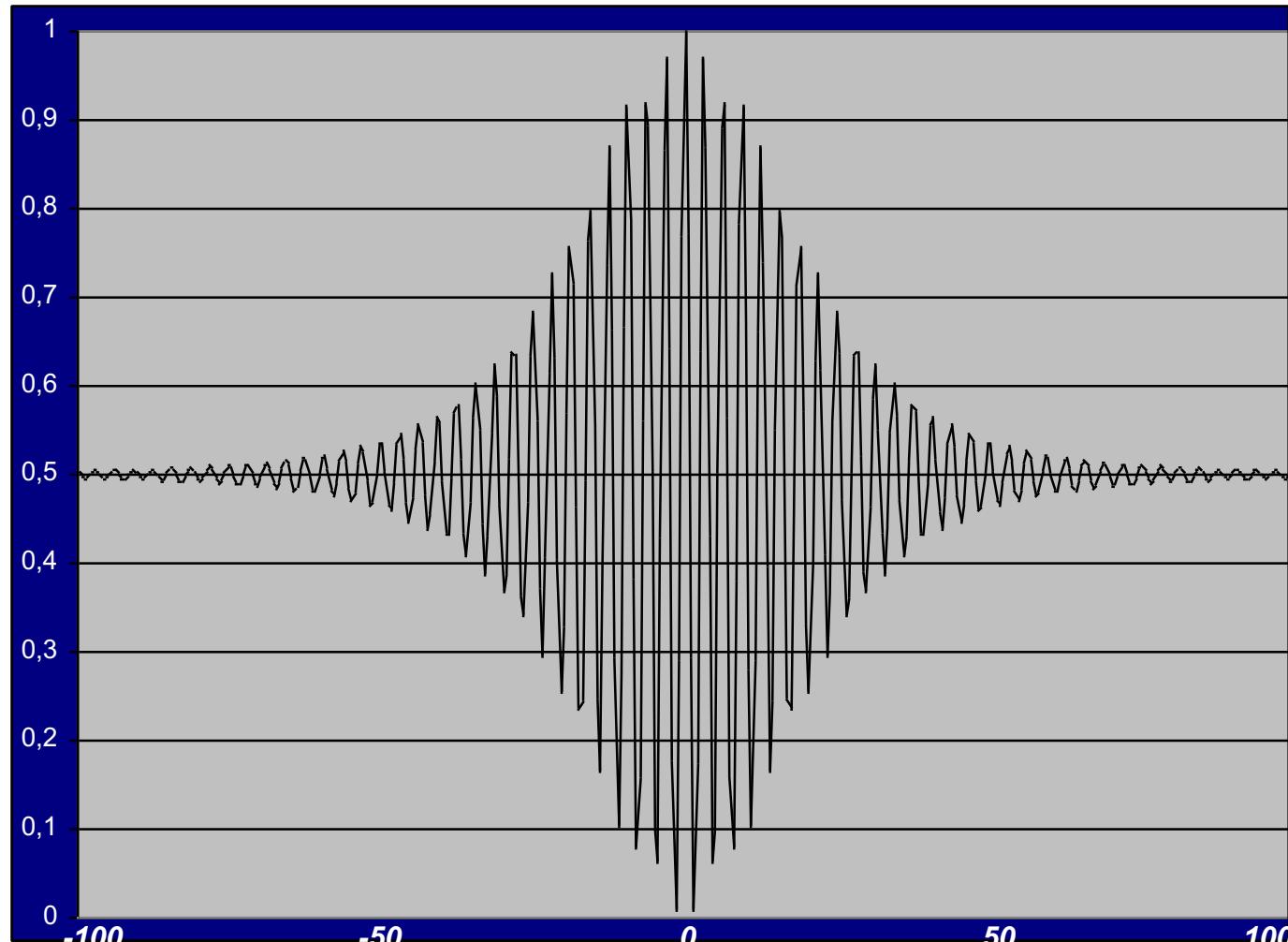
# Desenvolver um AG para Maximizar a função F6:





# Função $F6(x,y)$

$F6(x,0)$



$X$



- **Módulo de Avaliação**

**Função de Avaliação:**

**Função binária  $F_6$**

- **Módulo de População**

**Técnica de Representação:**

**Binária 44 bits**

**Técnica Inicialização da População:**

**Aleatória**

**Técnica Eliminação da População:**

**Elimina todos – o mais apto (Elitismo)**

**Técnica de Seleção de Genitores:**

**Roleta**

**Técnica de Aptidão:**

**Aptidão é a avaliação**

**Population Size:**

**100**

- **Módulo de Reprodução**

**Operadores:**

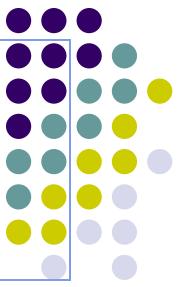
**Crossover 1 ponto & Mutação**

**Taxa Mutação:**

**0,008**

**Taxa Crossover:**

**0,65**



# Características da F6

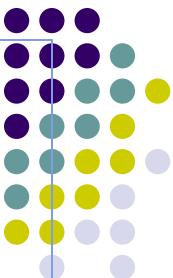
$$F6(x,y) = 0,5 - \frac{(\operatorname{sen} \sqrt{x^2 + y^2})^2 - 0,5}{(1,0 + 0,001(x^2 + y^2))^2}$$

- Objetivo: Maximizar F6
- Uma única solução ótima:  $F6(0,0)=1$
- Difícil de otimizar: vários mínimos locais



# Representação

- Binária codificando real
- 2 Variáveis: x, y
- Domínio:  $x, y \in [-100, +100]$
- Precisão: 4 a 5 casas decimais
- $K_i=22 \rightarrow$  total de 44 bits



# Exemplo

- **Cromossoma:**

00001010000110000000011000101010001110111011

- **Dividido em x e y:**

0000101000011000000001    1000101010001110111011

- **Convertidos para base 10:**

165377 e 2270139

- **Multiplicados por:  $200/2^{22}-1$**

7,885791751335085 e 108,24868875710696

- **Somados a mín:**

x=-92,11420824866492 e y=8,248688757106959

- **Aplicados a F6(x,y):**

F6(x,y)=0,5050708



# Módulo de População

- Técnica Inicialização da População: *Aleatoriedade*  
→ *Geração aleatória de palavras de 44 bits*
- Técnica de Aptidão: *Aptidão é a avaliação*  
→ *Aptidão é numericamente igual à avaliação*
- Técnica de Seleção de Genitores: *Roleta*



# Parâmetros

- Tamanho da População:  
**pop\_size** Exemplo  
100
- Número de Gerações:  
**num\_ger** 40
- Total de Indivíduos:  
**total\_ind = pop\_size x num\_ger** 4000



# Parâmetros

- Tamanho da População:  
**pop\_size** Exemplo  
10
- Número de Gerações:  
**num\_ger** 400
- Total de Indivíduos:  
**total\_ind = pop\_size x num\_ger** 4000