

# COMPUTAÇÃO EVOLUCIONÁRIA

MANOELA KOHLER

[manoela@ele.puc-rio.br](mailto:manoela@ele.puc-rio.br)

# APRESENTAÇÃO

Engenheira de Computação – PUC-Rio

Mestre em Engenharia Elétrica (Métodos de Apoio à Decisão) – PUC-Rio

Doutora em Engenharia Elétrica (Métodos de Apoio à Decisão) – PUC-Rio

Professora do CCE – PUC-Rio – Pós-graduação Lato Sensu:

- R e Python
- Inteligência Artificial
- Data Mining
- Machine Learning
- Redes Neurais
- Deep Learning

Pós-graduação Stricto Sensu

- Computação Evolucionária

Coordenadora, pesquisadora e desenvolvedora no Laboratório de Inteligência Computacional Aplicada



**RECAPITULANDO**

# OTIMIZAÇÃO

O que é?

Objetivo da otimização: **Buscar** a solução ótima ou **melhorar** a solução que se possui.

# OTIMIZAÇÃO

## Para que serve?

Otimizar é essencial para:

1. aumentar o **desempenho e a eficácia** de qualquer processo em qualquer negócio;
2. Aumentar a competitividade nos negócios, implicando na **melhoria** de determinados processos em relação à concorrência;

A otimização deve ser **prática e flexível** de modo a ser empregada em qualquer processo, a qualquer tempo (dinâmica).

# QUAIS AS ENTIDADES DA OTIMIZAÇÃO?

1. O **problema**: suas características, restrições e variáveis.
2. As **variáveis do problema** cujos valores afetam a qualidade da solução.
3. A **função objetivo** que mede (calcula) quão boa é uma solução.
4. Um **método, algoritmo** ou **heurística** para buscar soluções.
5. O **espaço de busca**: o número total de soluções de um problema, que é determinado pelo número de variáveis e seus domínios.
6. **Recursos computacionais** para processamento do método, avaliação das soluções, tratamento das características do problema e escolha da melhor solução.

# OTIMIZAÇÃO

**Otimizar** é essencialmente melhorar a solução de problemas (em **negócios**, na **indústria** e em **processos operacionais**), o que pode trazer:

- **eficiência e rentabilidade**
- **redução de despesas, gastos e perdas**
- **aumento dos lucros.**



# OTIMIZAÇÃO

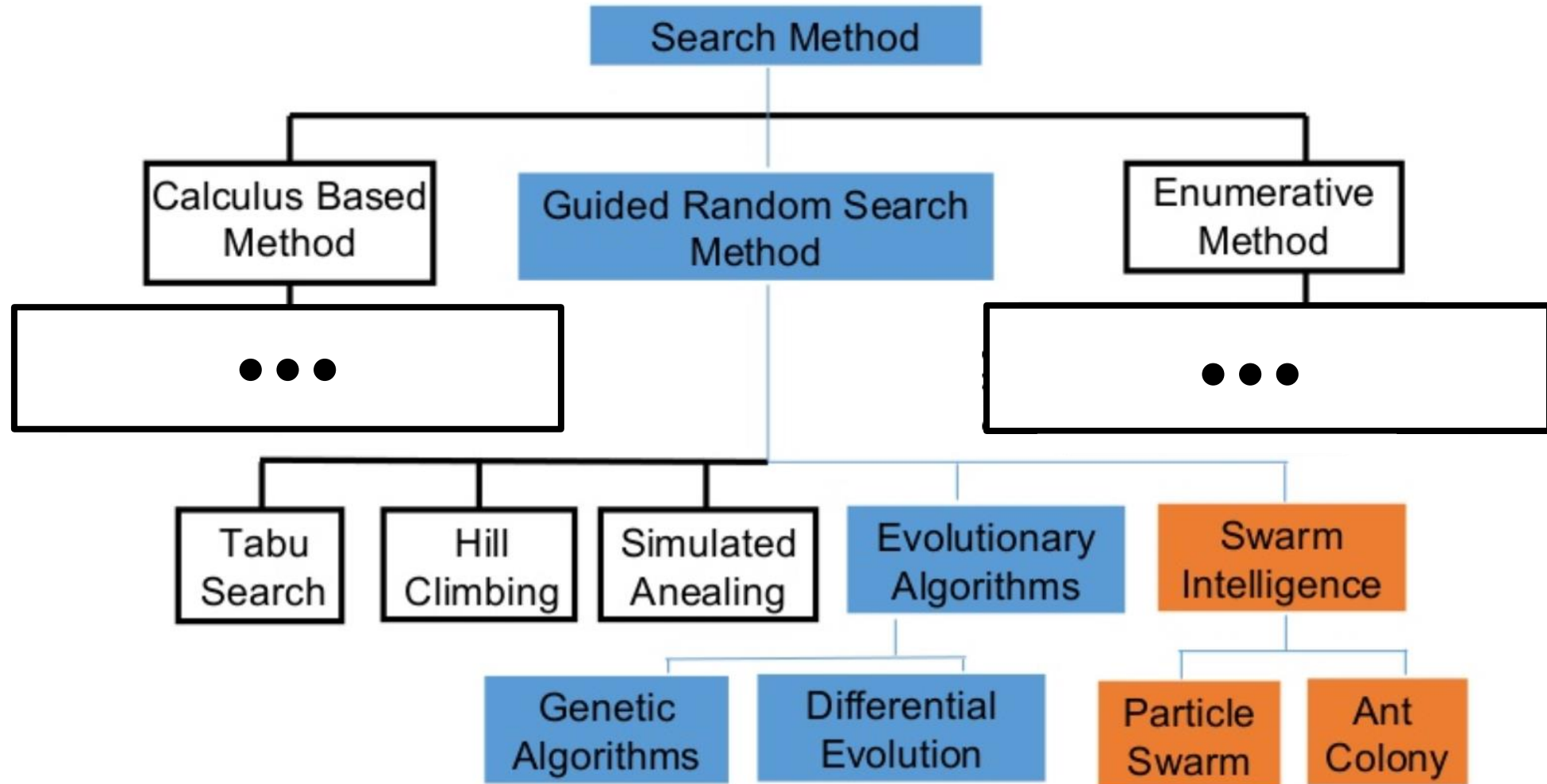
Muitos desses **problemas de otimização** são:

- **complexos**
- **com grandes espaços de busca**
- **de difícil modelagem**

Para tal, este curso visa, além de **introduzir os conceitos** fundamentais de **otimização**, apresentar **teoria e prática** de técnicas de otimização evolucionária e de inteligência de enxames.



# Classes of Search Methods



# ANALOGIA DA COMPUTAÇÃO EVOLUCIONÁRIA

## EVOLUÇÃO

Ambiente



Indivíduo



Aptidão



## SOLUÇÃO DE UM PROBLEMA

Problema

Solução candidata

Qualidade

Aptidão → chances de sobrevivência e reprodução

Qualidade → chances de gerar novas soluções

# ALGORITMOS EVOLUCIONÁRIOS

- ✓ Emular processos evolucionários;
- ✓ Operar em populações de indivíduos;

## **PARTES IMPORTANTES DE UMA OTIMIZAÇÃO EVOLUCIONÁRIA:**

- ✓ Uma função a ser otimizada;
- ✓ O domínio de operação;
- ✓ Um algoritmo evolucionário.



# BENCHMARKS CONHECIDOS

# *FUNÇÃO ESFERA*

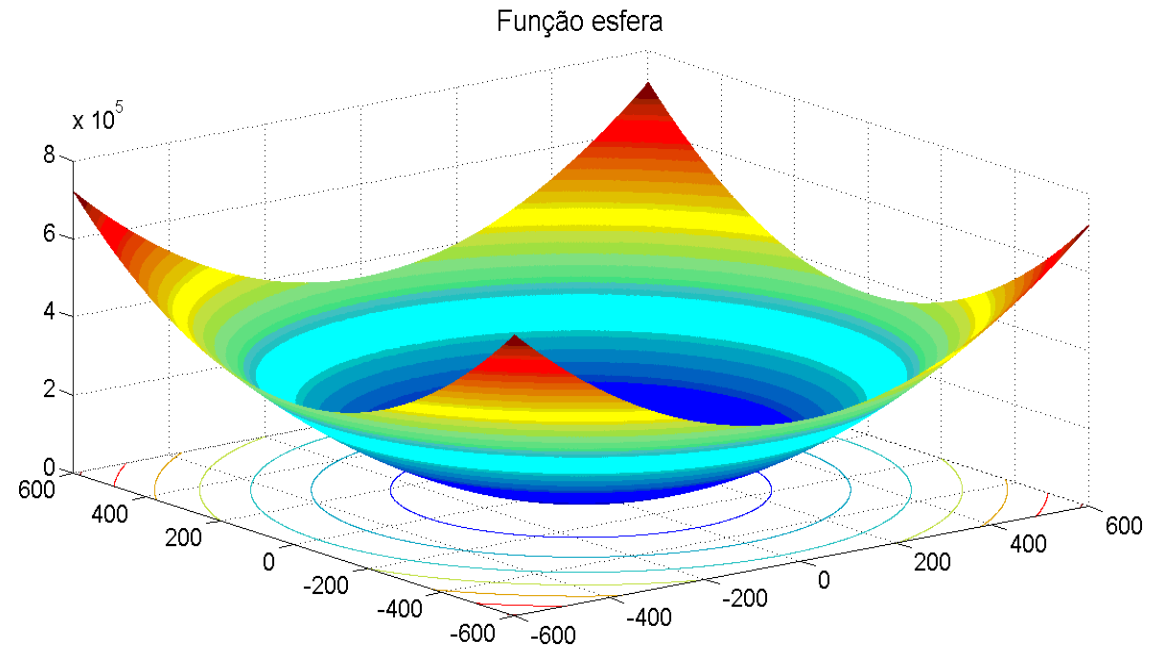
- Uma das funções mais simples utilizadas em testes de algoritmos evolucionários.

$$f_{sphere} = \sum_{i=1}^n x_i^2$$

$$f(x, y) = x^2 + y^2$$

$$[-600, 600]$$

$$o = (0, \dots, 0) = 0$$



Mapa 3D da função esfera bidimensional.

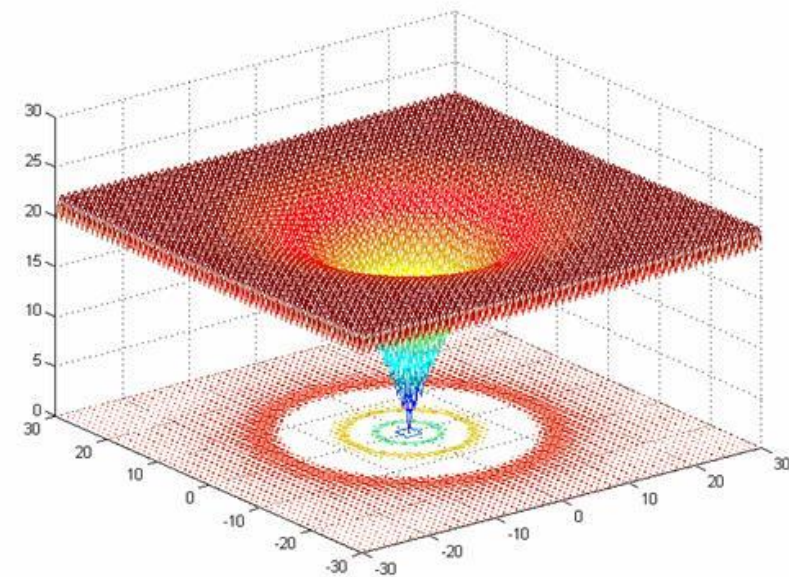
# *FUNÇÃO DE ACKLEY*

- Possui vários ótimos locais.

$$\begin{aligned} f_{Ackley} &= -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) \\ &\quad - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1) \end{aligned}$$

$[-30, 30]$

$\mathbf{o} = (0, \dots, 0) = 0$



Mapa 3D da função de Ackley bidimensional

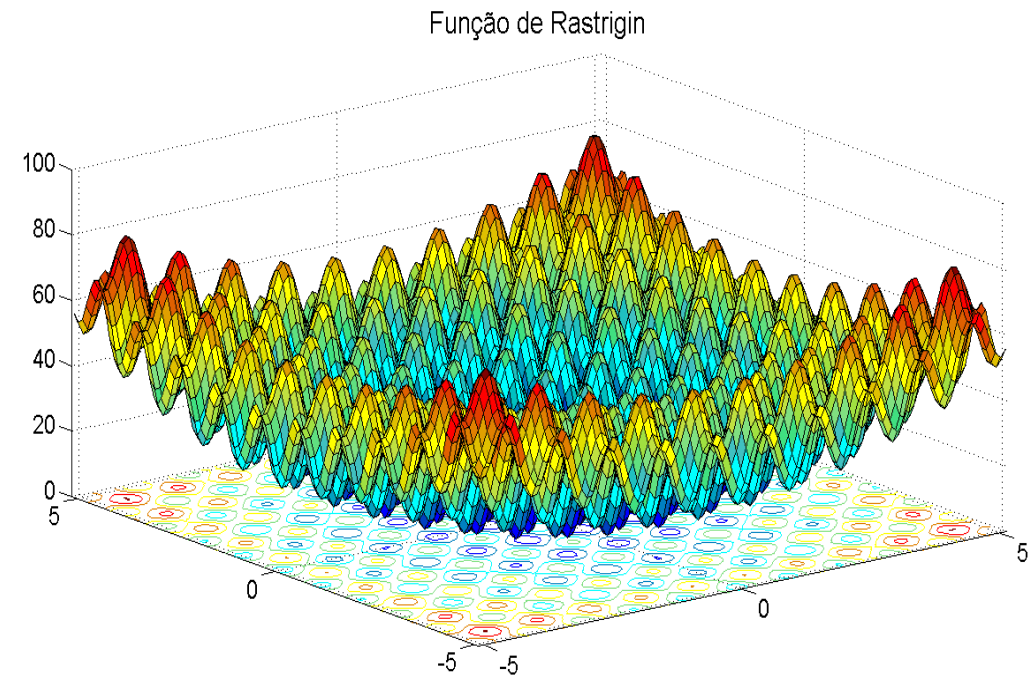
# *FUNÇÃO DE RASTRIGIN*

Possui vários ótimos locais.

$$f_{Rastrigin} = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$$

$[-5.12, 5.12]$

$o = (0, \dots, 0) = 0$



Mapa 3D da função de Rastrigin bidimensional

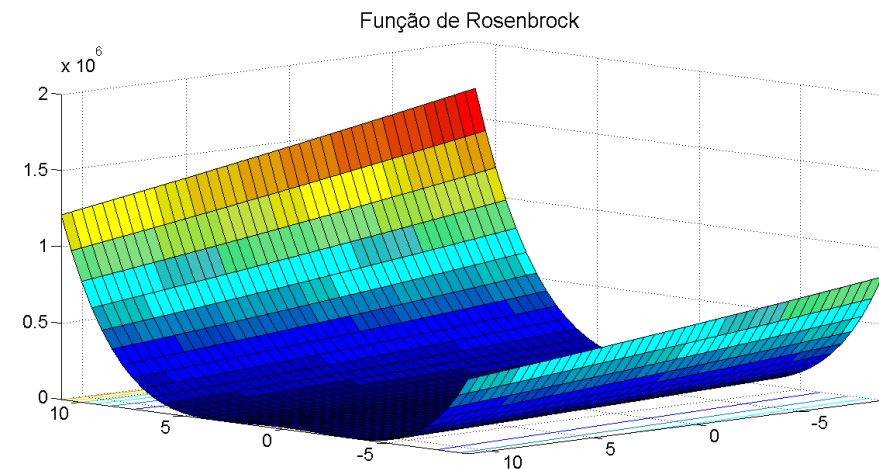
# *FUNÇÃO DE ROSENBROCK*

O mínimo global está dentro de um vale plano, longo, estreito e em formato parabólico. Chegar ao vale é uma tarefa fácil; a dificuldade está em convergir para o mínimo global, localizado no ponto  $o = (1, \dots, 1)$ .

$$f_{Rosenbrock} = \sum_{i=1}^{n-1} \left( 100(x_i^2 - x_{1+i})^2 + (x_i - 1)^2 \right)$$

$$[-9, 11]$$

$$o = (1, \dots, 1) = 0$$



Mapa 3D da função Rosenbrock bidimensional



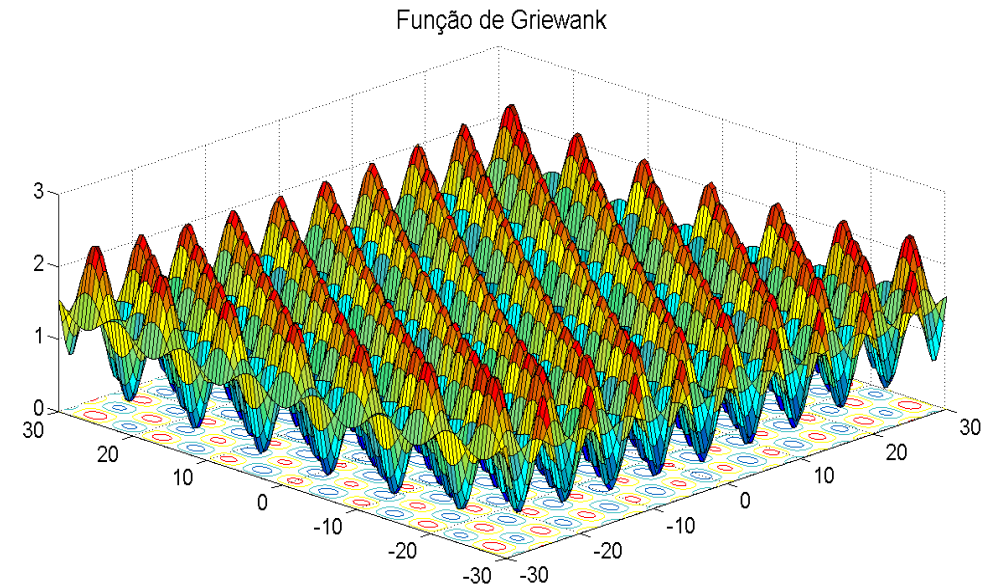
# *FUNÇÃO GRIEWANK*

É semelhante à função de Rastrigin, mas possui maior número de ótimos locais.

$$f_{Griewank} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$[-30, 30]$

$\mathbf{o} = (0, \dots, 0) = 0$



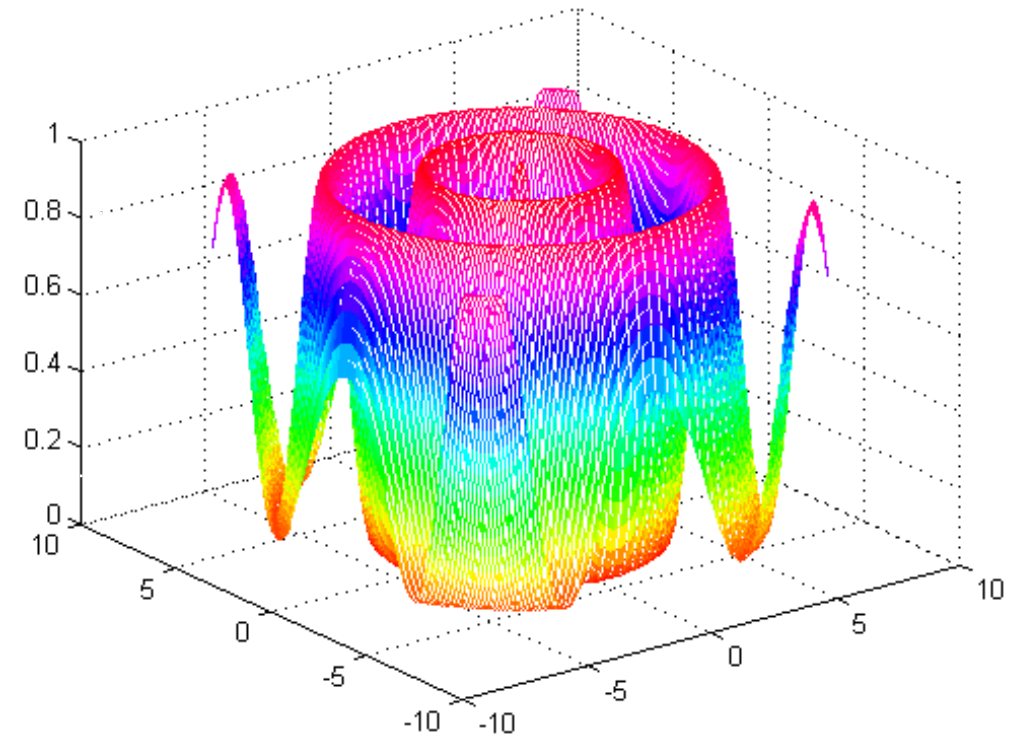
Mapa 3D da função de Griewank bidimensional

# SCHAFFER F6

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

$[-100, 100]$

$$o_{max} = (0, \dots, 0) = 1$$



# SOLVER EVOLUCIONÁRIO



Excel

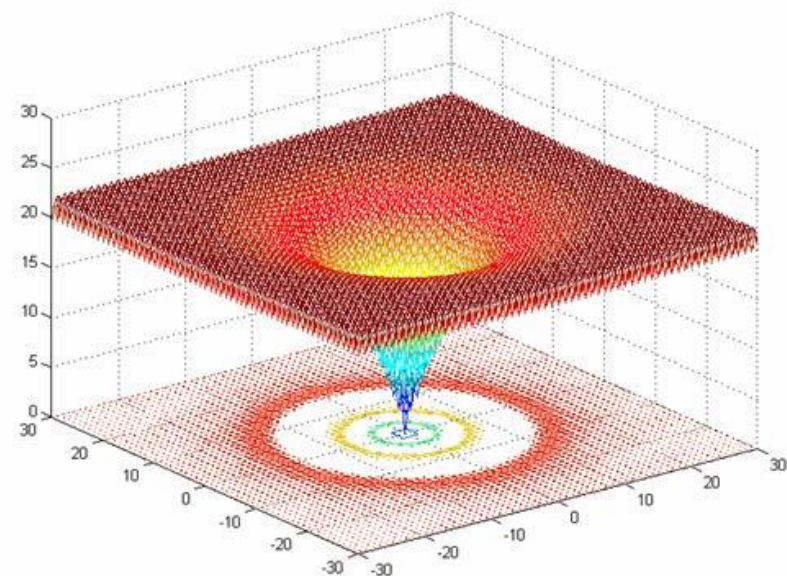
# SOLVER - EXERCÍCIOS

Utilizem o solver para minimizar a função de Ackley (duas variáveis com representação decimal)

$$\begin{aligned} f_{Ackley} &= -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) \\ &\quad - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1) \end{aligned}$$

$$[-30, 30]$$

$$o = (0, \dots, 0) = 0$$



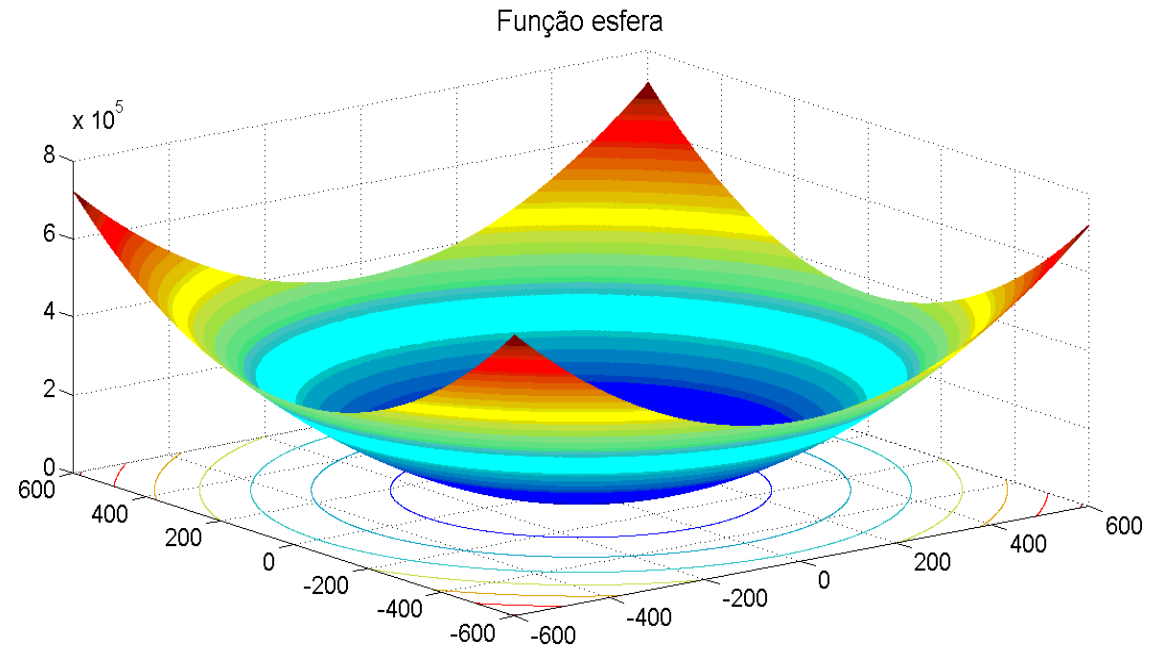
# SOLVER - EXERCÍCIOS

Utilizem o solver para minimizar a função Esfera (duas variáveis com representação **decimal** e **binária**)

$$f_{sphere} = \sum_{i=1}^n x_i^2$$

$$[-600, 600]$$

$$o = (0, \dots, 0) = 0$$



A função BIN2DEC converte números em representação binária, com no máximo 9 bits, para decimal. O décimo bit indica o sinal (- ou +). No nosso caso, precisamos de 10 bits, então proponham a decodificação.

# Computation Intelligence Car Evolution Using Box2D Physics (v3.2)

52 fps average

Physics step: 2 ms (435 fps)

16 MB used

Hide

Input Seed / Choose Terrain

623

Generation: 3 Max Score: 623.3

Copy All

Copy Selected

Car	Score	Time
0	623.3	1:07

311

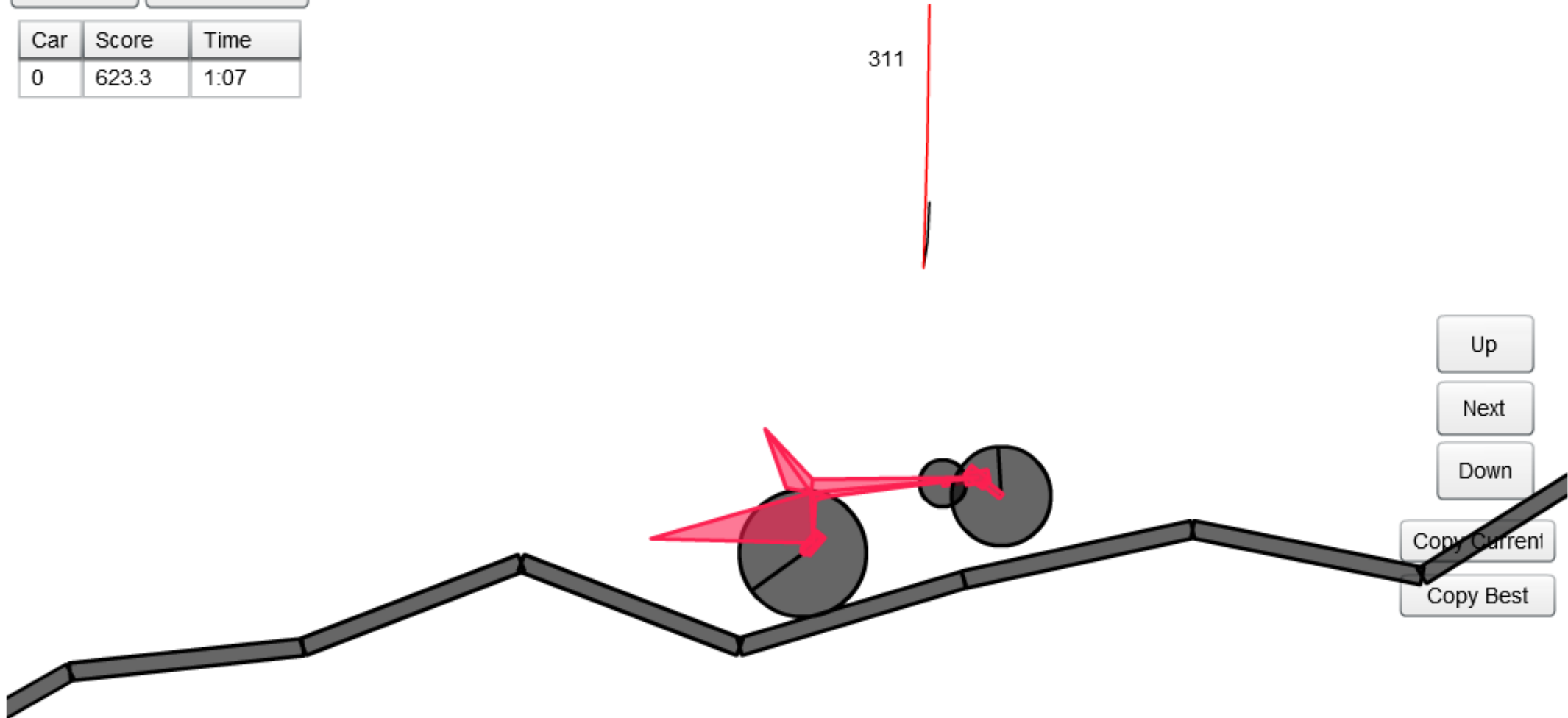
Up

Next

Down

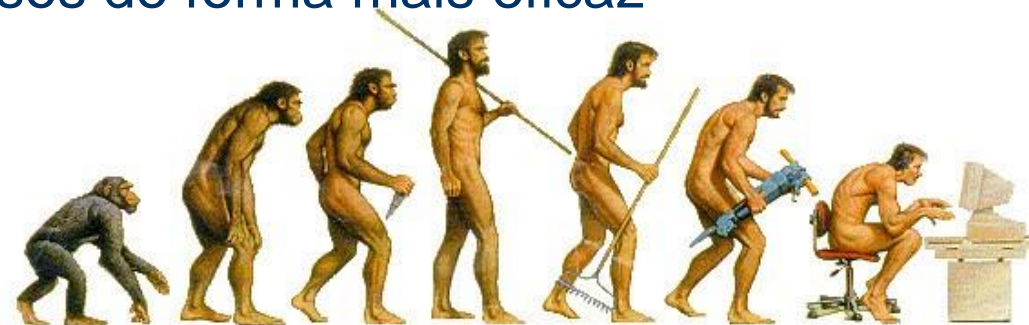
Copy Current

Copy Best



# TEORIA DA EVOLUÇÃO DE DARWIN: SELEÇÃO NATURAL

- Todos os ambientes possuem recursos finitos  
(isto é, só pode suportar um número limitado de indivíduos);
- As formas de vida têm um instinto básico voltado para a reprodução e sobrevivência;
- Portanto, algum tipo de seleção é inevitável;
- Aqueles indivíduos que competem pelos recursos de forma mais eficaz têm maior chance de reprodução.



# CONCEITOS BÁSICOS

Os Algoritmos Genéticos são algoritmos baseados nos mecanismos de seleção natural e genética.

São inspirados no Princípio da Evolução das Espécies proposto por Darwin:

*“quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.”*



# CONCEITOS BÁSICOS

Nos Algoritmos Genéticos cada indivíduo **representa** uma possível **solução** para um problema;

Uma população de indivíduos é criada e submetidas a **seleção, cruzamento e mutação**. A qualidade de cada indivíduo é determinada por sua **avaliação**;

É gerado um processo de **evolução natural** desses indivíduos, que eventualmente deverá gerar um indivíduo que caracterizará uma boa solução (talvez até a melhor) para o problema.

# CONCEITOS BÁSICOS

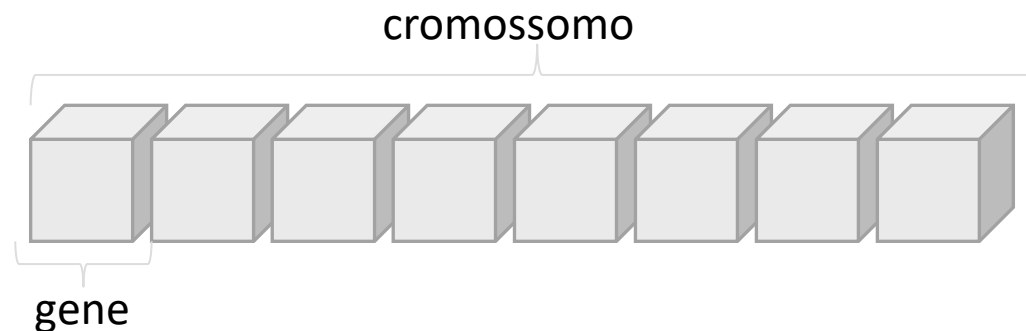
Os Algoritmos Genéticos são flexíveis e permitem a fácil inclusão de instruções específicas para o problema de interesse;

A qualidade dos resultados depende diretamente da qualidade da modelagem do problema:

- representação cromossômica e decodificação;
- função de avaliação;
- operadores genéticos.

# REPRESENTAÇÃO

- Consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser tratada pelo computador;
- Quanto mais ela for adequada ao problema, maior a qualidade dos resultados obtidos.



# OPERAÇÕES

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

# OPERAÇÕES

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos são reproduzidos com base na aptidão

# OPERAÇÕES

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos são reproduzidos com base na aptidão
- **Cruzamento:** troca de genes

# OPERAÇÕES

- **Seleção:** privilegia os indivíduos mais aptos
- **Reprodução:** indivíduos são reproduzidos com base na aptidão
- **Cruzamento:** troca de genes
- **Mutação:** troca aleatória de um gene

# OPERADORES GENÉTICOS

- Os operadores genéticos têm a função de modificar os indivíduos e consequentemente gerar novos indivíduos;
- Para isso são utilizados dois tipos de operadores distintos:
  - cruzamento
  - mutação

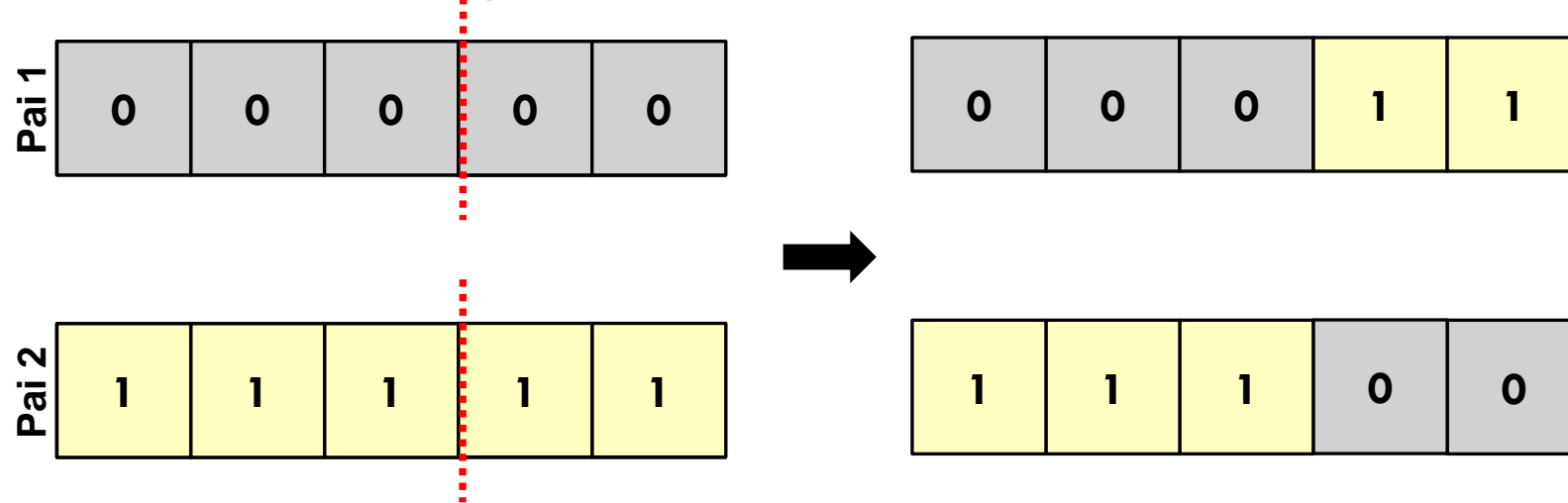


# OPERADORES GENÉTICOS

- Cruzamento
  - consiste em recombinar o material genético de dois indivíduos a fim de criar dois novos indivíduos;
  - esse operador tem a função de extrair genes de diferentes indivíduos, e recombina-los para formar novos indivíduos.

# OPERADORES GENÉTICOS

- Cruzamento – 1 ponto de corte



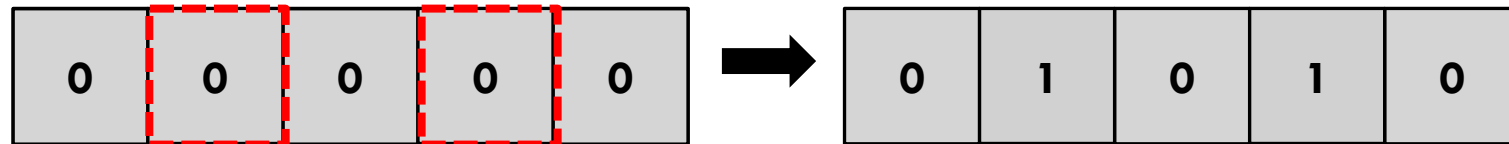
- Existem outros operadores de cruzamento, como por exemplo, de 2 pontos de cortes, uniforme, baseado em maioria, etc.

# OPERADORES GENÉTICOS

- **Mutação**
  - **introduz diversidade em uma população, ou seja, é responsável pela variação dos indivíduos;**
  - **consiste em aplicar modificações aleatórias em uma ou mais características de um indivíduo para criar um novo.**

# OPERADORES GENÉTICOS

- Mutação



# OPERADORES GENÉTICOS

Utilizando os operadores genéticos de cruzamento e de mutação, os Algoritmos Genéticos conseguem um equilíbrio entre:

- a capacidade de exploração do espaço de soluções; e
- o aproveitamento das melhores soluções ao longo da evolução;

Com isso, se mostram interessantes para a resolução de problemas complexos de otimização.

# PRÓS E CONTRAS

## Prós

- Mais eficiente do que procurar um espaço de busca muito grande.
- Fácil, na medida em que, se a representação e a avaliação de indivíduos estiver correta, uma solução pode ser encontrada sem qualquer trabalho analítico explícito.

## Contras

- Pode ficar preso em um ótimo local, embora o crossover possa ajudar a mitigar isso.
- Pode ser difícil descobrir qual a melhor forma de representar um indivíduo.

# Componentes de um Algoritmo Genético

# COMPONENTES DE UM ALGORITMO GENÉTICO

- 1. Problema**
- 2. Representação**
- 3. Decodificação**
- 4. Avaliação**
- 5. Operadores**
- 6. Técnicas**
- 7. 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

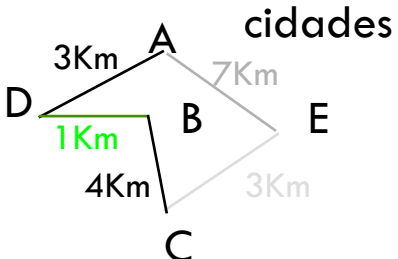
Tipo de Problema  $\leftrightarrow$  Representação

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

# 3. DECODIFICAÇÃO

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

**Cromossomas “representam” soluções.**

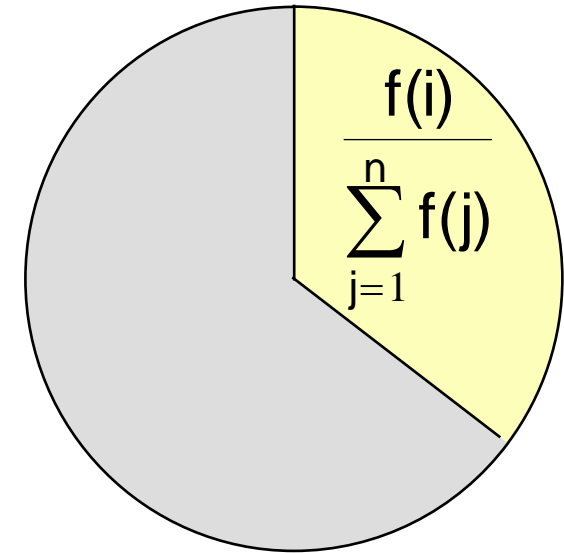
<u>Cromossoma</u>	<u>DECODIFICAÇÃO</u>	<u>Solução</u>
0011011	bin ➡ inteiro	$x=27$ $\in$
ADBCE		rota $A \rightarrow D \rightarrow B \rightarrow C \rightarrow E$

## 4. AVALIAÇÃO

Elo entre o algoritmo genético e o problema.

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

Chances de seleção são proporcionais à aptidão.



## 5. OPERADORES

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

**1. Crossover**

**2. Mutação**

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

## 6. TÉCNICAS

- Técnicas de Representação
- Técnicas de Inicialização da População
- Técnicas de Eliminação da População Antiga
- Técnicas de Reprodução
- Técnicas de Seleção de Genitores
- Técnicas de Aptidão
- Técnicas de Parametrização
- Técnicas de Elitismo
- Técnicas de Seleção de Operadores

# 7. PARÂMETROS

- Tamanho da população (total de indivíduos)
- Número de gerações
- Probabilidade de crossover
- Probabilidade de mutação
- etc.



# LISTAS

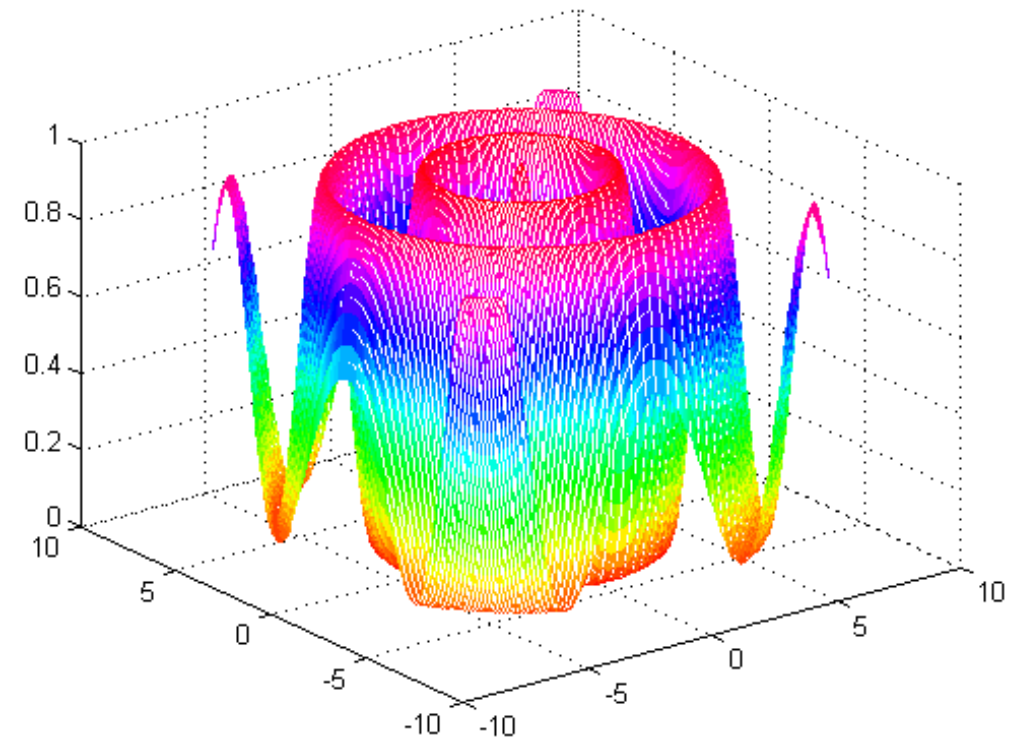
02/09 - 1ª Lista – ICA-CASES: Aplicações de Algoritmos Genéticos

09/09 - 2ª Lista – GADEMO (doc)

# FUNÇÃO F6

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

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



# CURVA MÉDIA DE EXPERIMENTOS PARA $F6(X,Y)$

- Usamos o número de dígitos 9 consecutivos após o ponto decimal para distinguir avaliações muito próximas de 1,00.

Exemplo:

Avaliação	dígitos 9
0,99873578	2
0,82435787	0
0,99995432	4

# ICADEMO

ICADEMO

Arquivo Ajuda

Escolha da Função

☒ F6 ☐ F6 Elevada

Características da Execução

Avaliações: 4000 (Max:15000)

Tamanho da População: 100 (Max:200)

Taxa de Crossover (%): 65 (Max:100%)

Taxa de Mutação (%): 0.8 (Max:100%)

Total de rodadas: 5 (Max:100)

Gerar gráfico de busca aleatória

Características do Algoritmo

☒ Normalização Linear ☐ Elitismo ☐ Steady State ☐ Steady State s/ duplicados ☐ nenhuma das características acima

Min 1 Max 100

Gap (%) 80

Tipo de Crossover

☒ One Point ☐ Two Point ☐ Uniform

Melhor: X= Y= Rodada:

Algoritmo Genético

Número de 9's

Avaliações

Rodada Anterior

Indivíduo

Aval:

X:

Y:

Controles

Iniciar

Salvar imagem

Encerrar

☒ Manter gráfico

# GUIDELINES

# CROSSOVER

Partes de dois cromossomas são trocadas a partir de posição escolhida aleatoriamente

Probabilidade de Crossover : **65%**

- Teste Verdadeiro    ☐ **Efetua Cruzamento**
- Teste Falso        ☐ **Copia os Genitores**

*Valores ideais são obtidos experimentalmente*

$P_1$	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

Troca cada gene de um cromossoma se o teste de probabilidade for verdadeiro

Probabilidade de Mutação: **0,8% (0,008)**

- Teste Verdadeiro    ☐ **troca bit**
- Teste Falso        ☐ **mantém bit**



*Valores ideais são obtidos experimentalmente*

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	0.005	0.84		1	1	1	0
0	0	1	0		0.768	0.473	0.894	0.001		0	0	1	1

# Evolução X Convergência

## □ Crossover:

- acelerador do processo de busca
- **tira proveito das soluções** mais promissoras

## □ Mutação

- operador **exploratório**
- **dispersa a população** pelo espaço de busca

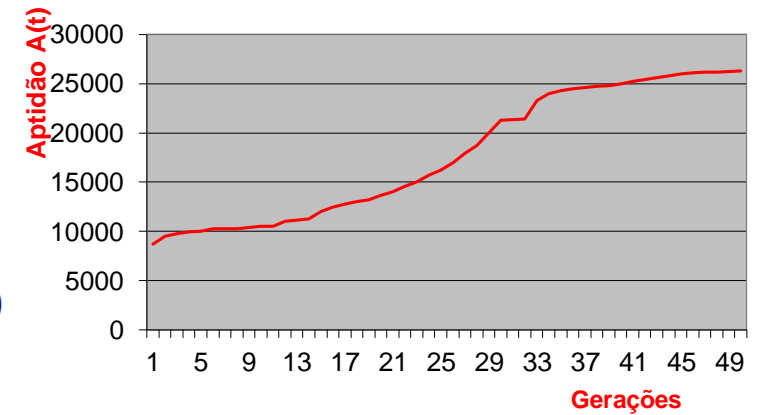
## □ Problemas de Convergência:

- população com **indivíduos muito similares**
- não há mais evolução:
  - **ótimo encontrado** ou **convergência prematura (mínimo local)**
- para continuar a evoluir é preciso **introduzir mais diversidade** na população

□ bom desempenho no início da evolução

□ pouco ou nenhum desempenho no final

Curva da Média de Experimentos





# Análise do Desempenho de um GA

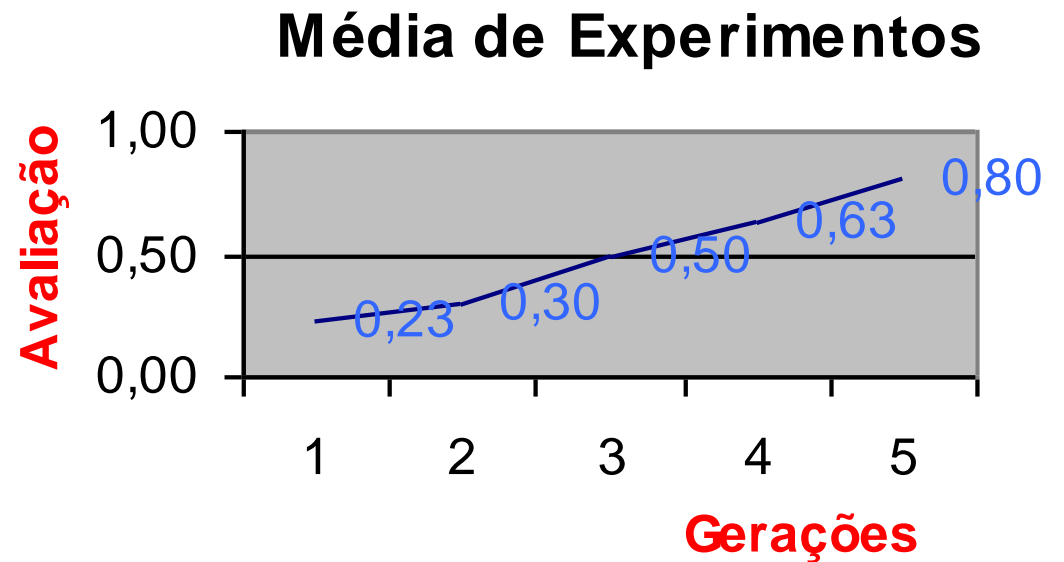
- ❑ **Melhor de um Experimento (valor)**
- ❑ **Melhores por Geração**
- ❑ **Média de Melhores por Geração em vários Experimentos**

# ANÁLISE DO DESEMPENHO DE UM GA

- GAs são estocásticos: desempenho varia a cada experimento.
- São necessários muitos experimentos para se conhecer o desempenho médio do GA.

# ANÁLISE DO DESEMPENHO DE UM GA

	Melhores por geração				
	Experimentos				
	1o.	2o.	3o.	4o.	Média
ger 1	0.2	0.3	0.1	0.3	0.23
ger 2	0.3	0.3	0.2	0.4	0.30
ger 3	0.4	0.6	0.4	0.6	0.50
ger 4	0.7	0.6	0.6	0.6	0.63



# Novas Técnicas e Parâmetros

- Técnica de Aptidão: Normalização Linear
- Elitismo
- Reprodução Steady State
- Ajuste dos Parâmetros

# Novas Técnicas e Parâmetros

- Técnica de Aptidão: Normalização Linear
- Elitismo
- Reprodução Steady State
- Ajuste dos Parâmetros

# Exemplo Comparativo

Rank dos cromossomas

Avaliação original

Normalização Linear (min=0, max = 1)

6	5	4	3	2	1
200	9	8	7	4	1
1	0.0402	0.0352	0.0302	0.0151	0

$$z = \frac{x - \min(x)}{[\max(x) - \min(x)]}$$

- **SUPER INDIVÍDUO:** cromossoma 6
  - poucas chance de recombinação com outros indivíduos; elimina competidores em poucas gerações;
  - rápida convergência.
- **COMPETIÇÃO PRÓXIMA:** entre cromossomas 3, 4 e 5
  - é preciso aumentar a pressão seletiva sobre os melhores

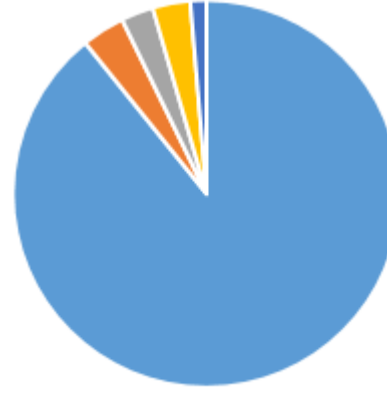
# Exemplo Comparativo

Avaliação Original



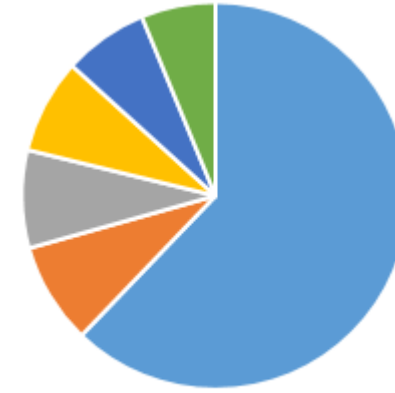
■ 1 ■ 2 ■ 3 ■ 4 ■ 5 ■ 6

Normalização Linear  
(min=0, max = 1)



■ 1 ■ 2 ■ 3 ■ 4 ■ 5 ■ 6

Normalização Linear  
(min=1, max = 10)



■ 1 ■ 2 ■ 3 ■ 4 ■ 5 ■ 6

# Novas Técnicas e Parâmetros

- Técnica de Aptidão: Normalização Linear
- **Elitismo**
- Reprodução Steady State
- Ajuste dos Parâmetros



# Elitismo

- ❑ Melhor cromossoma de  $P(t)$  é copiado em  $P(t+1)$ , após o mutação e crossover.
- ❑ Reduz o efeito aleatório do processo seletivo.
- ❑ Garante que o melhor indivíduo da próxima geração é melhor ou igual ao da geração anterior.

# Novas Técnicas e Parâmetros

- Técnica de Aptidão: Normalização Linear
- Elitismo
- Reprodução Steady State
- Ajuste dos Parâmetros

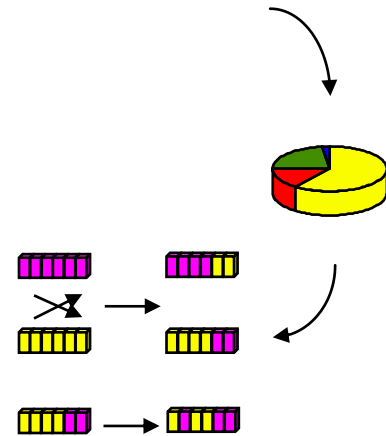
# Reprodução Steady State

- ❑ Substituição parcial de indivíduos a cada geração
- ❑ Bons indivíduos (material genético) são preservados, garantindo mais chances de reprodução
- ❑ Indivíduos mantidos não precisam ser reavaliados
- ❑ Método:
  - Crie  $n$  filhos (seleção+crossover+mutação), onde  $n <$  tamanho da população
  - Elimine os  $n$  piores membros da população
  - Avalie e introduza os filhos na população
- ❑ **GAP** = fração da população que é **trocada**

# Exemplo de Steady State

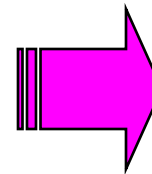
C19	120
C18	110
C17	100
C16	99
C15	95
C14	81
C13	76
C12	67
C11	58
C10	44
C9	42
C8	36
C7	22
C6	20
C5	19
C4	17
C3	10
C2	8
C1	5

avaliações  
de  $P(t)$



38
6
121
88
58
17

crie  $n$   
novos



120
110
100
99
95
81
76
67
58
44
42
36
22
38
6
121
88
58
17

substitua os  $n$   
piores

ordena



121
120
110
100
99
95
88
81
76
67
58
58
44
42
38
36
22
17
6

avaliações  
de  $P(t+1)$

# Steady State sem Duplicados

- ❑ Substituição parcial de indivíduos com **exclusão de duplicados**
- ❑ Evita os **duplicados que são mais frequentes** com steady state
- ❑ Maior **eficiência do paralelismo** de busca, garantindo pop\_size indivíduos diferentes
- ❑ Descendentes duplicados são desprezados
- ❑ Maior **overhead para teste** de igualdade

# Novas Técnicas e Parâmetros

- Técnica de Aptidão: Normalização Linear
- Elitismo
- Reprodução Steady State
- Ajuste dos Parâmetros

# Operadores

- ❑ Crossover de 2 pontos
- ❑ Crossover Uniforme
- ❑ Operadores Independentes e Seleção de Operadores

# Crossover de 2 Pontos

- Semelhante ao crossover de 1 ponto
- 2 pontos são escolhidos aleatoriamente
- Crossover de 1 ponto não consegue combinar todos os padrões de dois genitores

$P_1$	1	1	0	1	1	0	0	1	0	1	1	0	1	1
$P_2$	0	0	0	1	0	1	1	0	1	1	1	1	0	0
				↓	<i>pontos de corte</i>						↓			
$P_1$	1	1	0	1	1	0	0	1	0	1	1	0	1	1
$P_2$	0	0	0	1	0	1	1	0	1	1	1	1	0	0
$F_1$	1	1	0	1	0	1	1	0	1	1	1	0	1	1
$F_2$	0	0	0	1	1	0	0	1	0	1	1	1	0	0



# Operadores

- Crossover de 2 pontos
- **Crossover Uniforme**
- Operadores Independentes e Seleção de Operadores

# Crossover Uniforme

- A contribuição de cada genitor é decidida aleatoriamente por um padrão (máscara de bits)
- $F1 = P1$  onde padrão é '1' e  $= P2$  onde padrão é '0'
- $F2 = P2$  onde padrão é '1' e  $= P1$  onde padrão é '0'
- Capacidade de combinar quaisquer padrões

<i>P1</i>	1	0	0	1	0	1	1
<i>P2</i>	0	1	0	1	1	0	1
<i>Padrão</i>	1	1	0	1	0	0	1
<i>F1</i>	1	0	0	1	1	0	1
<i>F2</i>	0	1	0	1	0	1	1

# Operadores

- Crossover de 2 pontos
- Crossover Uniforme
- Operadores Independentes e Seleção de Operadores

# Operadores Independentes

- ❑ GAs podem incorporar diversos operadores genéticos.
- ❑ **Qual operador usar** a cada instante?
- ❑ Operadores não devem ser usados com a mesma intensidade a cada fase da evolução
  - *mais crossover no início e mais mutação no final da evolução.*
- ❑ **Solução:** uma **roleta** sorteia **um operador** a cada reprodução.
- ❑ **Pesos** (chances) dos operadores são parâmetros do algoritmo.

