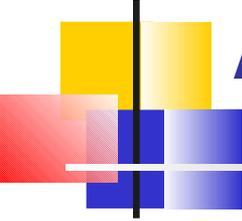


Algoritmos Genéticos

Texto base:

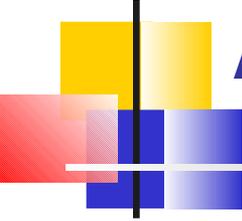
Stuart Russel e Peter Norving - "Inteligência Artificial"

junho/2007



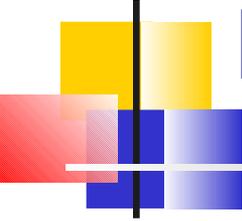
Algoritmo Genético

- Uma variante da busca em feixe estocástica
- Estado sucessor gerado pela combinação de dois estados pais
- Analogia com a seleção natural:
 - Busca em feixe estocástica – reprodução assexuada
 - Algoritmo genético – reprodução sexuada



Algoritmo Genético

- Começam com um conjunto de k estados gerados aleatoriamente chamado de **população**
- Um estado é chamado de **indivíduo**, ou **cromossomo**
 - Normalmente representado por uma cadeia de valores
 - Ex: Um estado das 8 rainhas deve especificar a posição das 8 rainhas, cada uma em uma coluna de 8 quadrados
 - Pode ser representado por 8 dígitos, variando de 1 a 8
 - Ou por uma cadeia de 24 bits = cada 3 bits = 1 posição

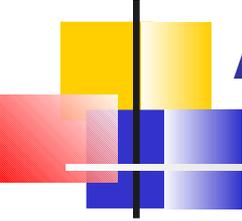


Exemplo indivíduo – 8 rainhas

- 2 4 7 4 8 5 5 2 = 001|011|110|011|111|100|100|001
- 3 2 7 5 2 4 1 1 = 010|001|110|100|001|011|000|000

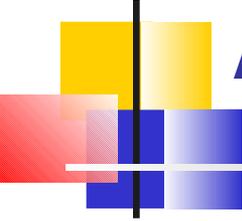
8				*				
7			*					
6								
5					*	*		
4		*		*				
3								
2	*							*
1								

8								
7			*					
6								
5				*				
4						*		
3	*							
2		*			*			
1							*	*



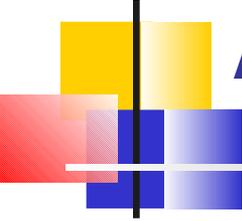
Algoritmo Genético

- Cada estado (ou indivíduo) é avaliado pela função de avaliação – chamada de **função de fitness**
- Quanto melhor o estado – maior é o valor da função fitness
 - Ex. das 8 rainhas: n^o de pares de rainhas não atacantes
 - 2 4 7 4 8 5 5 2 = 24
 - 3 2 7 5 2 4 1 1 = 23
 - 2 4 4 1 5 1 2 4 = 20
 - 3 2 5 4 3 2 1 3 = 11



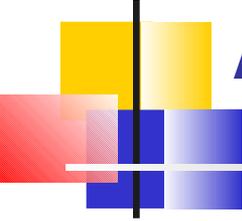
Algoritmo Genético

- Se o **método de seleção** dar maior probabilidade de um indivíduo com maior valor de fitness ser escolhido...
- Temos as seguinte probabilidades de escolha:
 - $2\ 4\ 7\ 4\ 8\ 5\ 5\ 2 = 24 \Rightarrow 31\%$
 - $3\ 2\ 7\ 5\ 2\ 4\ 1\ 1 = 23 \Rightarrow 29\%$
 - $2\ 4\ 4\ 1\ 5\ 1\ 2\ 4 = 20 \Rightarrow 26\%$
 - $3\ 2\ 5\ 4\ 3\ 2\ 1\ 3 = 11 \Rightarrow 14\%$



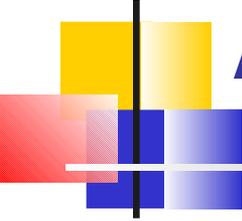
Algoritmo Genético

- Vamos supor que aleatoriamente (mas respeitando a probabilidade) foram selecionados os indivíduos:
 - 2 4 7 | 4 8 5 5 2
 - 3 2 7 | 5 2 4 1 1
 - 2 4 4 1 5 | 1 2 4
 - 3 2 7 5 2 | 4 1 1
- Normalmente, um ponto de ***crossover*** é escolhido ao acaso



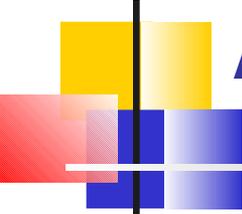
Algoritmo Genético

- E os filhos gerados por meio do *crossover* são:
 - 3 2 7 | 4 8 5 5 2
 - 2 4 7 | 5 2 4 1 1
 - 2 4 4 1 5 | 4 1 1
 - 3 2 7 5 2 | 1 2 4
- Este processo de reprodução faz com que o algoritmo genético explore estados longe dos estados pais, no começo da execução
- À medida em que os melhores indivíduos ficam na população, a probabilidade de gerar um filho longe dos pais, diminui



Algoritmo Genético

- Os indivíduos gerados podem sofrer **mutação** com uma pequena probabilidade
- A ideia é que quando os pais são muito parecidos, a mutação possa trazer alguma característica que ajude a escapar do ótimo local
 - 3 2 7 | 4 8 **3** 5 2
 - 2 4 7 | 5 2 4 1 1
 - 2 4 4 1 5 | 4 1 **6**
 - 3 2 **6** 5 2 | 1 2 4



Algoritmo Genético - Geral

Função ALGORITMO-GENÉTICO(*população*, FN-FITNESS) **retorna** um indivíduo

Entradas: *população*, um conjunto de indivíduos

FN_FITNESS, uma função que mede a adaptação de um indivíduo

Repita

```
nova_população <- conjunto vazio
```

```
para i<-1 até TAMANHO(população) faça
```

```
  x <- SELEÇÃO-ALEATÓRIA(população, FN-FITNESS)
```

```
  y <- SELEÇÃO-ALEATÓRIA(população, FN-FITNESS)
```

```
  filho <- REPRODUZ(x,y)
```

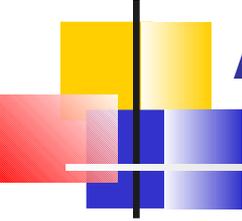
```
  se (pequena probabilidade aleatória)
```

```
    então filho <- MUTAÇÃO(filho)
```

```
  adicionar filho à nova população
```

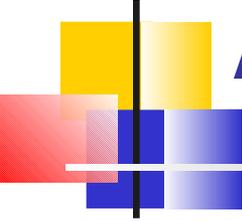
Até algum critério de parada

Retornar o melhor indivíduo da *população*, de acordo com FN-FITNESS



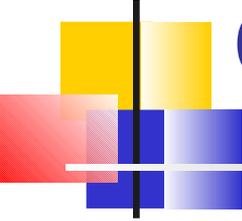
Algoritmo Genético

- Troca informações entre processos de busca paralelos
- A principal vantagem vem da operação de *crossover*.
 - Combinar grandes blocos de genes que evoluem de forma independente para executar funções úteis
 - Ex: a colocação da três primeiras rainhas nas posições 2, 4 e 6 (em que elas não se atacam as outras) constitui um bloco útil
 - Estes blocos podem ser combinados com outros, para formar uma solução



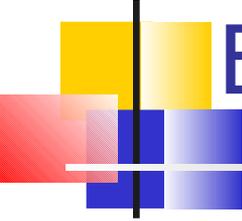
Algoritmo Genético

- A combinação de blocos úteis funciona usando a idéia de **esquema**
- Um esquema é uma sub-cadeia na qual algumas posições podem ser deixadas sem especificação
- Ex: 246*****
- Cadeias do tipo 24625176 são chamadas **instâncias** do problema



Questões centrais

- Como representar os indivíduos?
- Quem é a população inicial?
- Como definir a função objetivo?
- Quais são os critérios de seleção?
- Como aplicar/definir o operador de reprodução?
- Como aplicar/definir o operador de mutação?
- Como garantir a convergência e ao mesmo tempo a solução ótima?



Exemplo 1

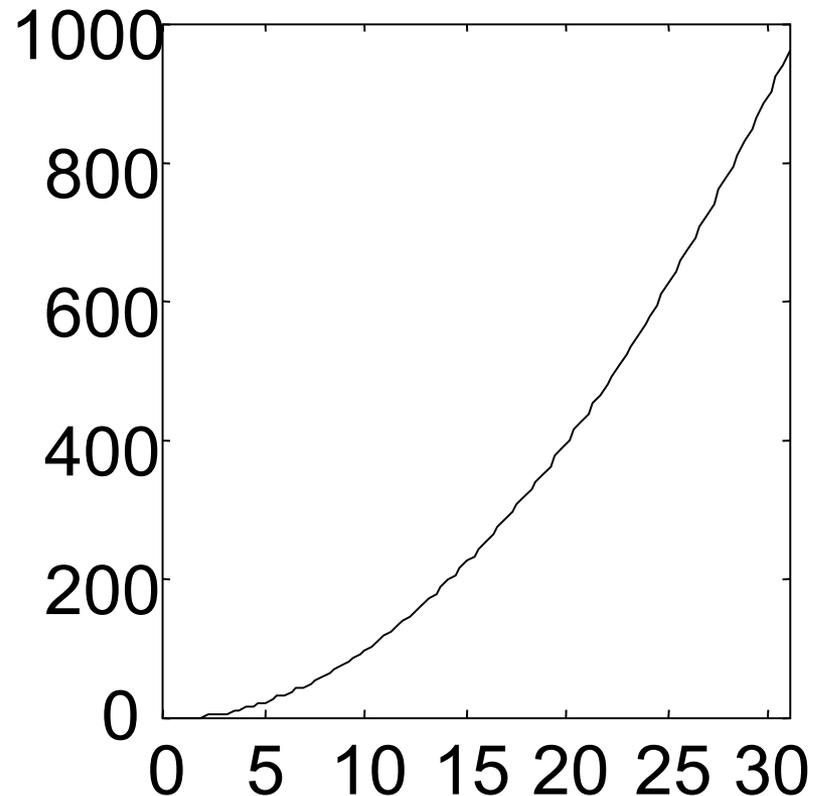
Problema: Use um AG para encontrar o ponto máximo da função:

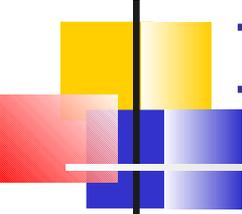
$$f(x) = x^2$$

com x sujeito as seguintes restrições:

$$0 \leq x \leq 31$$

x é inteiro

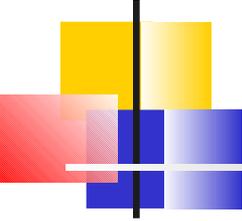




Indivíduo

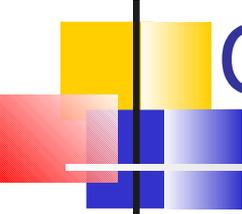
- Cromossomo

- Estrutura de dados que representa uma possível solução para o problema de forma não ambígua
- Os parâmetros do problema de otimização são representados por cadeias de valores.
- Exemplos:
 - Vetores de reais, (2.345, 4.3454, 5.1, 3.4)
 - Cadeias de bits, (111011011)
 - Vetores de inteiros, (1,4,2,5,2,8)
 - ou outra estrutura de dados.



Indivíduo

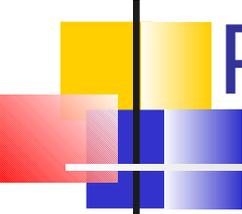
- Na implementação, cada indivíduo tem um valor de *fitness* associado a ele
- Aptidão pode ser:
 - Igual a função objetivo
 - Baseado no **ranking** do indivíduo da população



Cromossomo do Problema 1

- Cromossomos binários com 5 bits:
 - 0 = 00000
 - 31 = 11111
- Aptidão
 - Neste problema, a aptidão pode ser a própria função objetivo.
 - Exemplo:

$$\text{aptidão}(00011) = f(3) = 9$$



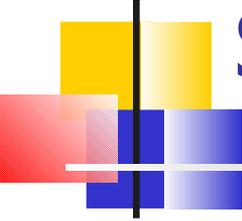
População Inicial do Problema 1

É aleatória (mas quando possível, o conhecimento da aplicação pode ser utilizado para definir população inicial)

Pop. inicial	cromossomos	x	$f(x)$	
	$A_1 = 1\ 1\ 0\ 0\ 1$	25	625	54,5%
	$A_2 = 0\ 1\ 1\ 1\ 1$	15	225	19,6%
	$A_3 = 0\ 1\ 1\ 1\ 0$	14	196	17,1%
	$A_4 = 0\ 1\ 0\ 1\ 0$	10	100	8,7%

Probabilidade de seleção
proporcional a aptidão

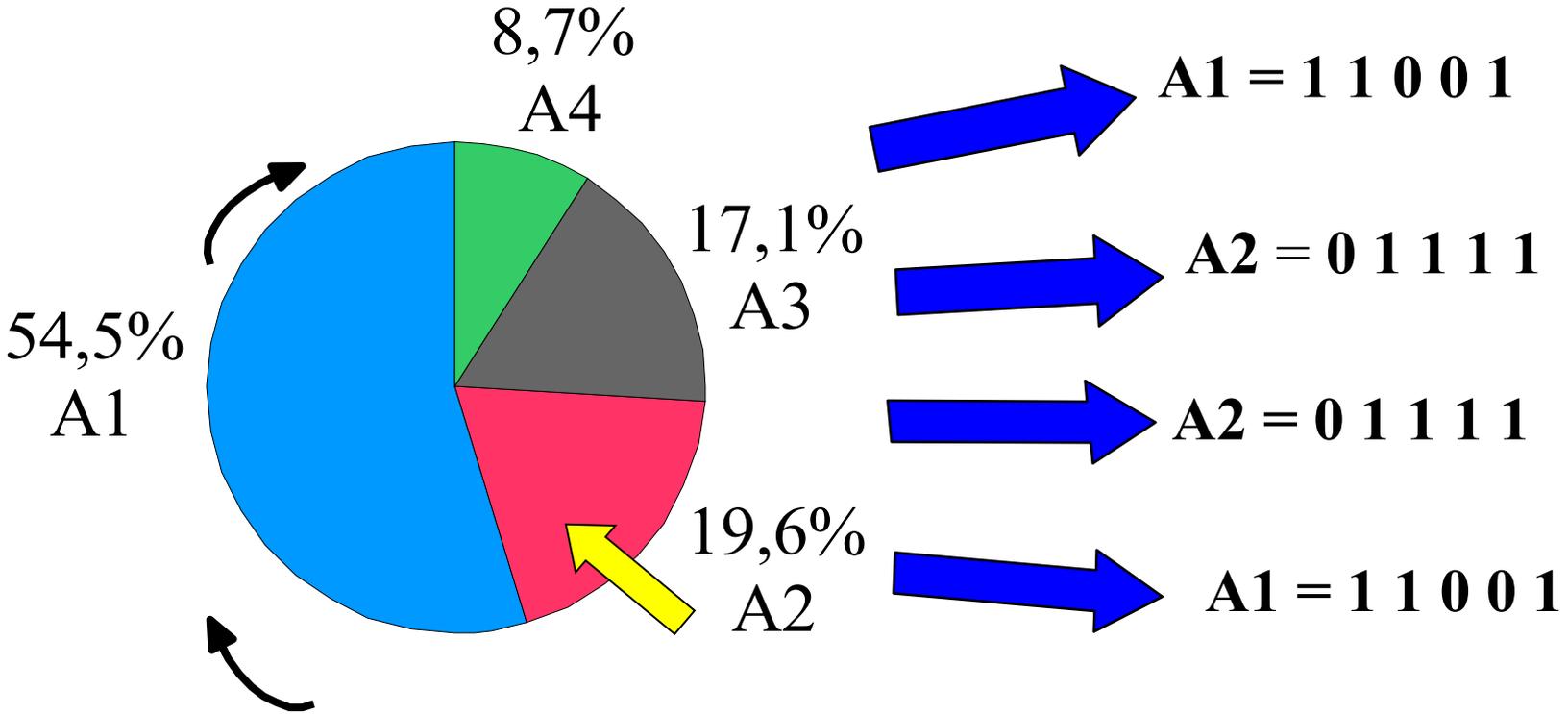
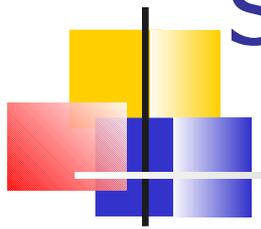
$$p_i = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)}$$



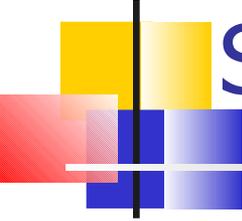
Seleção

- Seleção
 - Tem como objetivo propagar material genético dos indivíduos mais adaptados
 - Os melhores indivíduos (maior aptidão) são selecionados para gerar filhos
 - Dirige o AG para as melhores regiões do espaço de busca
- Tipos mais comuns de seleção
 - Proporcional a aptidão (roleta)
 - Torneio
 - Ranking (os n mais adaptados)

Seleção proporcional a aptidão (Roleta)

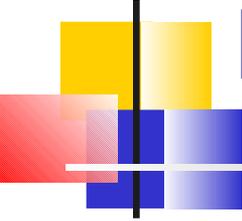


Problema: converge muito rápido por causa da variação pequena



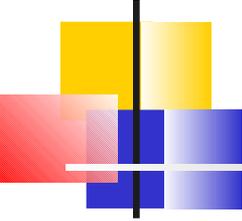
Seleção

- Torneio: escolhe-se n (tipicamente 2) indivíduos aleatoriamente da população e o melhor é selecionado.
- Ranking: seleciona-se os n indivíduos mais adaptados



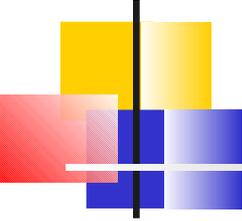
Reprodução - *Crossover*

- Função:
 - combinar e/ou perpetuar material genético dos indivíduos mais adaptados
 - Cria novos indivíduos misturando características de dois ou mais indivíduos pais (*crossover*) - variação
- Em termos de busca:
 - Principais mecanismos de busca do AG
 - Permite explorar áreas desconhecidas do espaço de busca



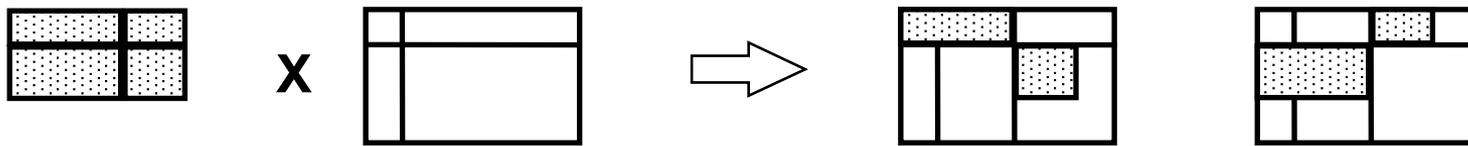
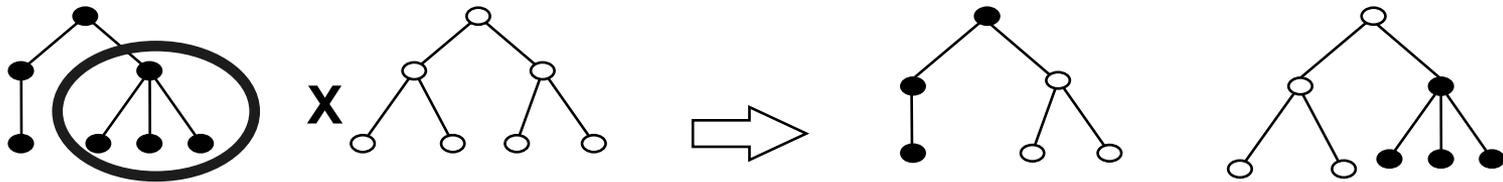
Crossover

- Os filhos são formados a partir dos bits dos pais
- Cruzamento em um ponto
 - Pai 1: 1010101011 | 0101010111
 - Pai 2: 0000100101 | 0101110010
 - Filho1: 10101010110101110010
 - Filho2: 00001001010101010111
- Cruzamento multi-ponto
 - Pai 1: 101010 | 101101 | 01010111
 - Pai 2: 000010 | 010101 | 01110010
 - Filho1: 000010 | 101101 | 01110010
 - Filho2: 101010 | 010101 | 01010111



Crossover

- Os pontos de corte dos cruzamentos em um ponto ou multi-ponto podem ser estáticos ou escolhidos aleatoriamente
- Quanto mais estruturada for a representação do cromossomo, mais difícil fica de se definir o cruzamento

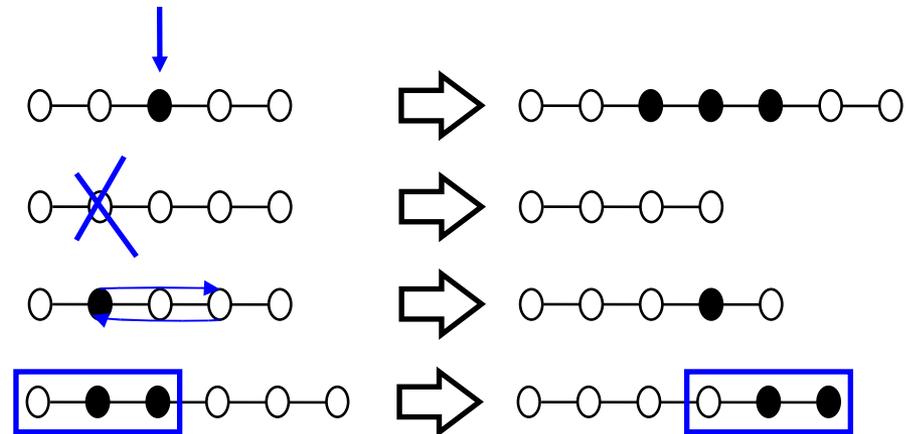


Mutação

- Objetivo:
 - gerar diversidade (p/ escapar de ótimos locais)

- Tipos:

- Gerativa
- Destrutiva
- Swap
- Swap de seqüência



- Obs: Existe uma "taxa de mutação" (ex. % da população selecionada) que pode diminuir com o tempo para garantir convergência

Crossover e mutação do Problema 1

Pais

Filhos

A1 = 1 1 0 | 0 1
 A2 = 0 1 1 | 1 1

Crossover
 (1 ponto)



1 1 0 1 1
 0 1 1 0 1

mutação



1 1 0 1 1
 0 0 1 0 1

A2 = 0 1 1 1 | 1
 A1 = 1 1 0 0 | 1

Crossover
 (1 ponto)

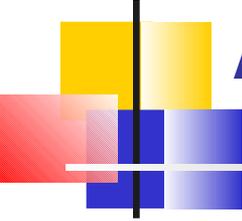


0 1 1 1 1
 1 1 0 0 1

mutação

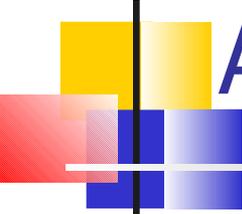


1 0 1 1 1
 1 1 0 0 1



Adição dos filhos à nova população

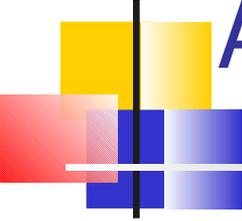
- Objetivo:
 - garantir uma convergência adequada
- Tipos:
 - simples: a nova geração **substitui** a antiga
 - elitista ou *steady-state*: a nova geração se **mistura** com a antiga
- Critérios de substituição no caso elitista:
 - os piores
 - os mais semelhantes
 - para evitar convergência prematura
 - os melhores
 - os pais
 - aleatoriamente, ...



A primeira geração do Problema 1

- Substituição simples

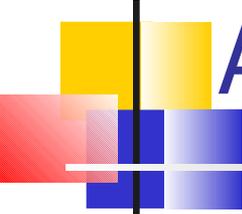
cromossomos	x	$f(x)$	prob. de seleção	
1	1 1 0 1 1	27	729	29,1%
2	1 1 0 0 1	25	625	24,9%
3	1 1 0 0 1	25	625	24,9%
4	1 0 1 1 1	23	529	21,1%



As demais gerações do Problema 1

					x	$f(x)$		
Segunda Geração	1	1	1	0	1	1	27	729
	2	1	1	0	0	0	24	576
	3	1	0	1	1	1	23	529
	4	1	0	1	0	1	21	441

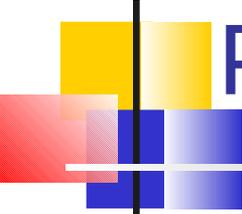
						x	$f(x)$	
Terceira Geração	1	1	1	0	1	1	27	729
	2	1	0	1	1	1	23	529
	3	0	1	1	1	1	15	225
	4	0	0	1	1	1	7	49



As demais gerações do Problema 1

					x	$f(x)$		
Quarta Geração	1	1	1	1	1	31	961	
	2	1	1	0	1	1	27	729
	3	1	0	1	1	1	23	529
	4	1	0	1	1	1	23	529

					x	$f(x)$	
Quinta Geração	1	1	1	1	1	31	961
	2	1	1	1	1	31	961
	3	1	1	1	1	31	961
	4	1	0	1	1	1	23



Problema 2

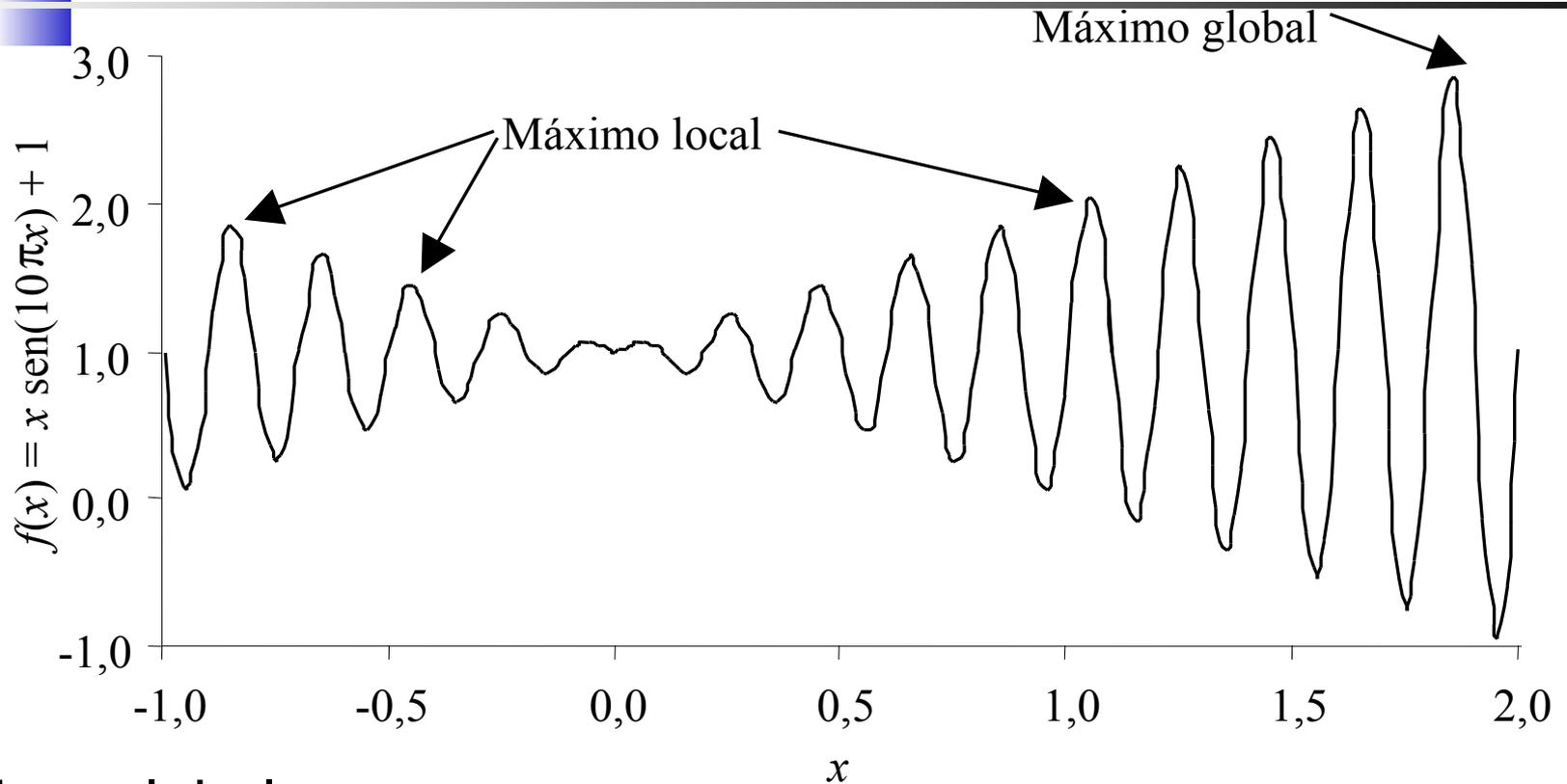
Achar o máximo da função utilizando Algoritmos Genéticos

$$f(x) = x \operatorname{seno}(10\pi x) + 1,0$$

Restrita ao intervalo:

$$- 1,0 \leq x \leq 2,0$$

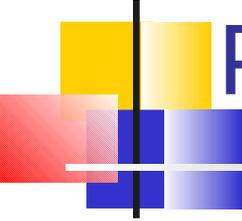
Problema 2



Máximo global:

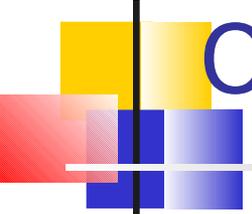
$$x = 1,85055$$

$$f(x) = 2,85027$$



Problema 2

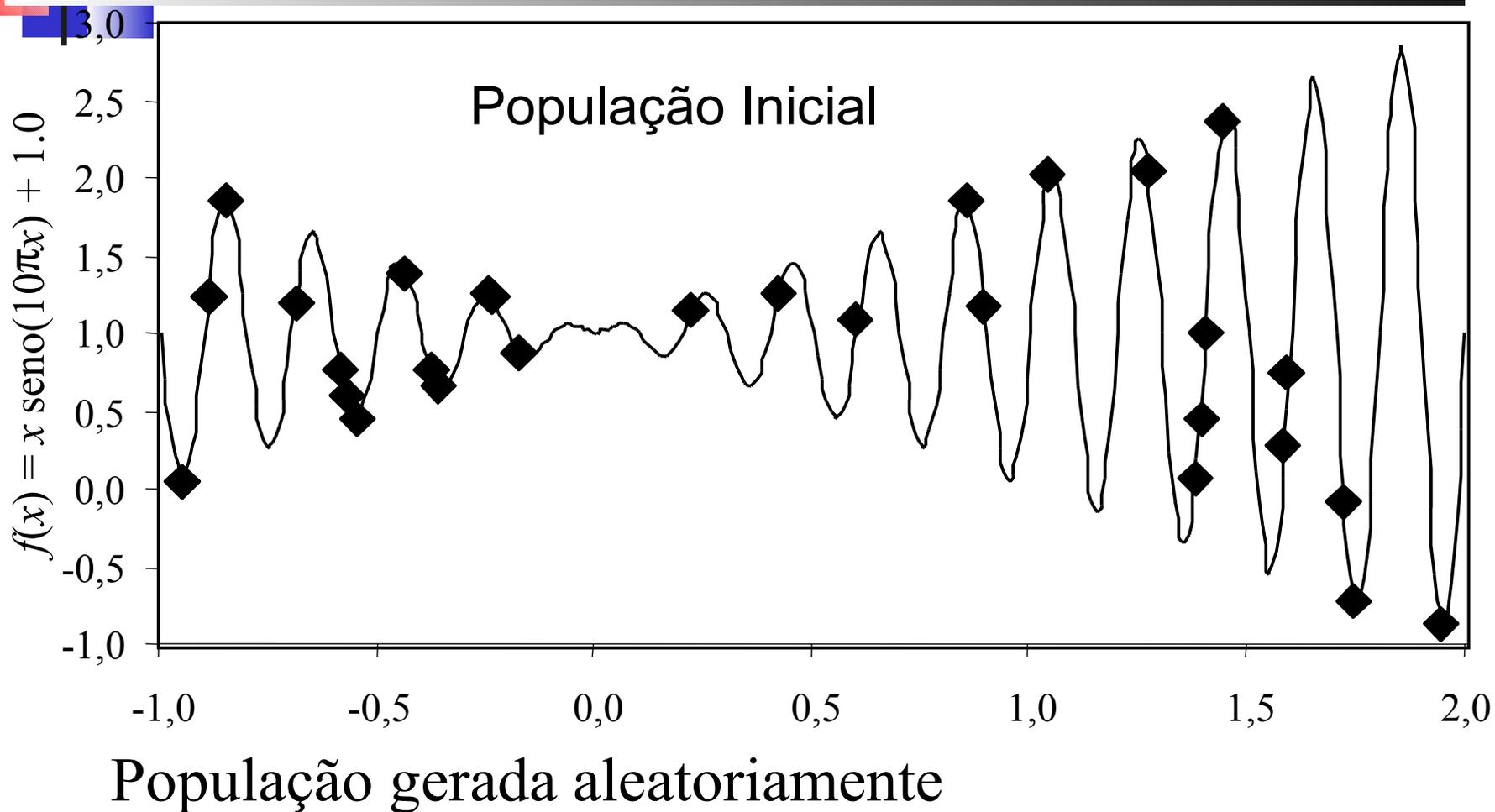
- Função multimodal com vários pontos de máximo.
- É um problema de otimização global (encontrar o máximo global)
- Não pode ser resolvido pela grande maioria dos métodos de otimização convencional.
- Há muitos métodos de otimização local, mas para otimização global são poucos.



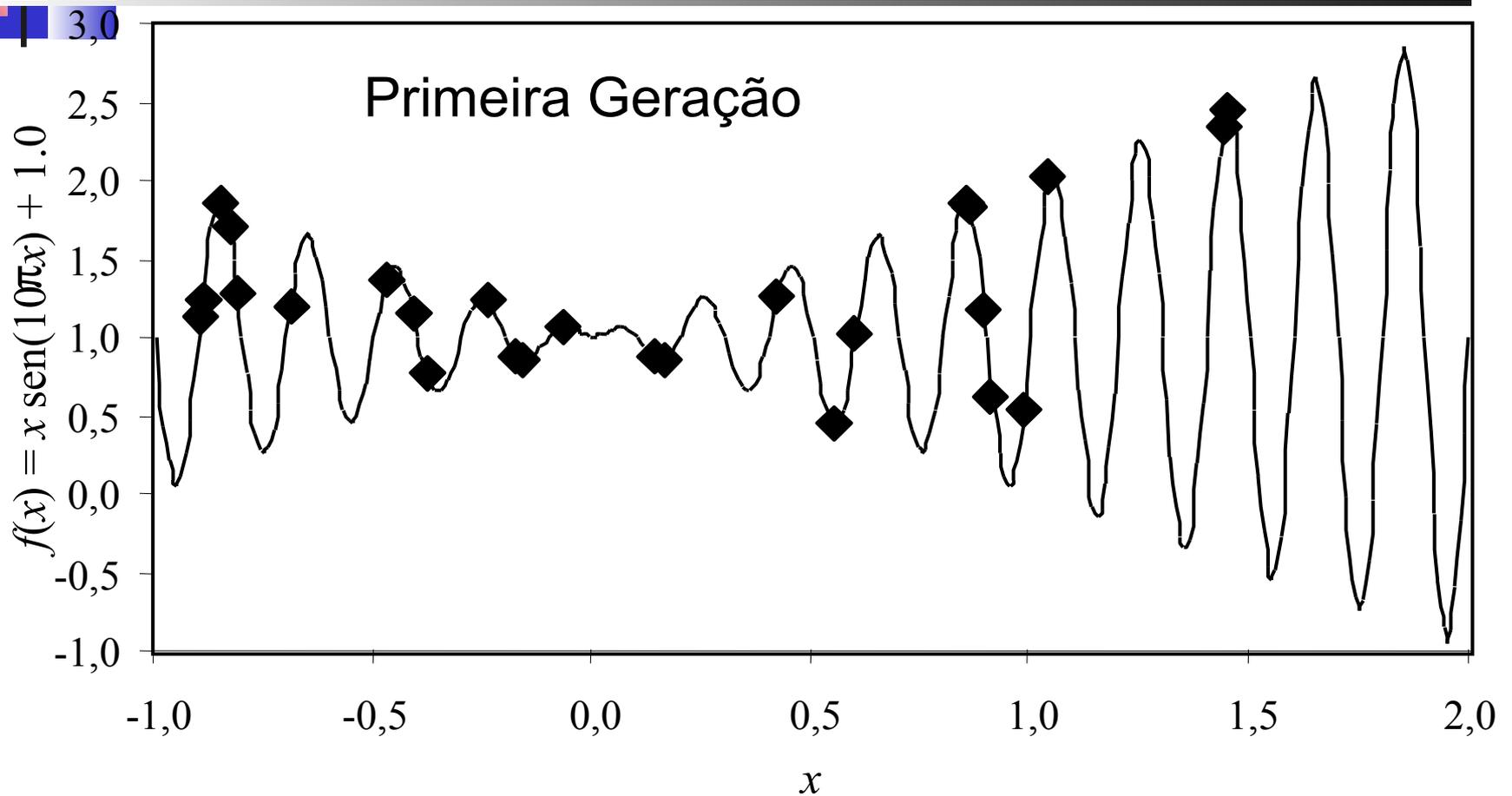
O Cromossomo Problema 2

- Representar o único parâmetro deste problema (a variável x) na forma de um cromossomo:
 - Quantos bits deverá ter o cromossomo?
 - Quanto mais bits melhor precisão numérica
 - Longos cromossomos são difíceis de manipular
- Cromossomo com 22 bits
 - **1000101110110101000111**

As Gerações do Problema 2

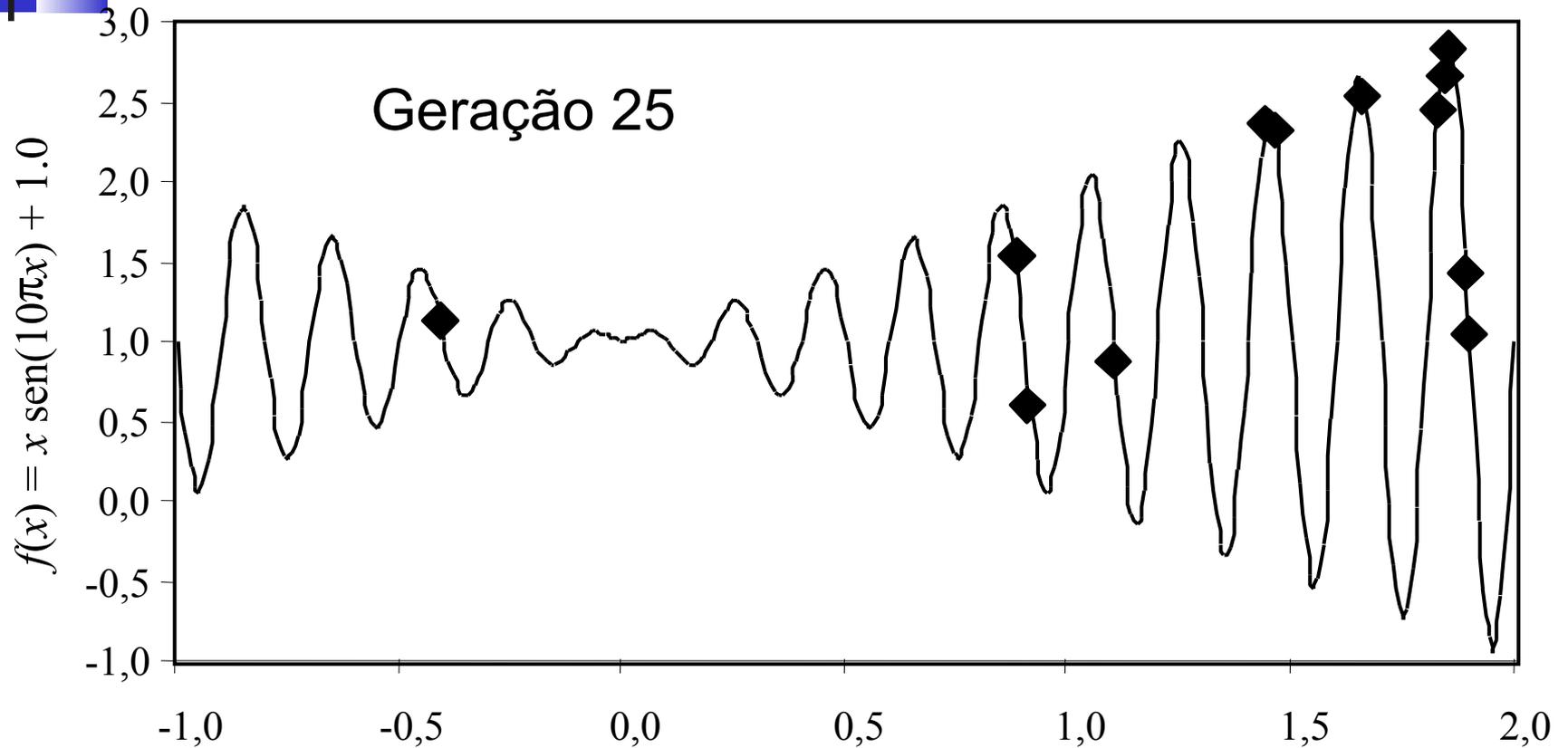


As Gerações do Problema 2



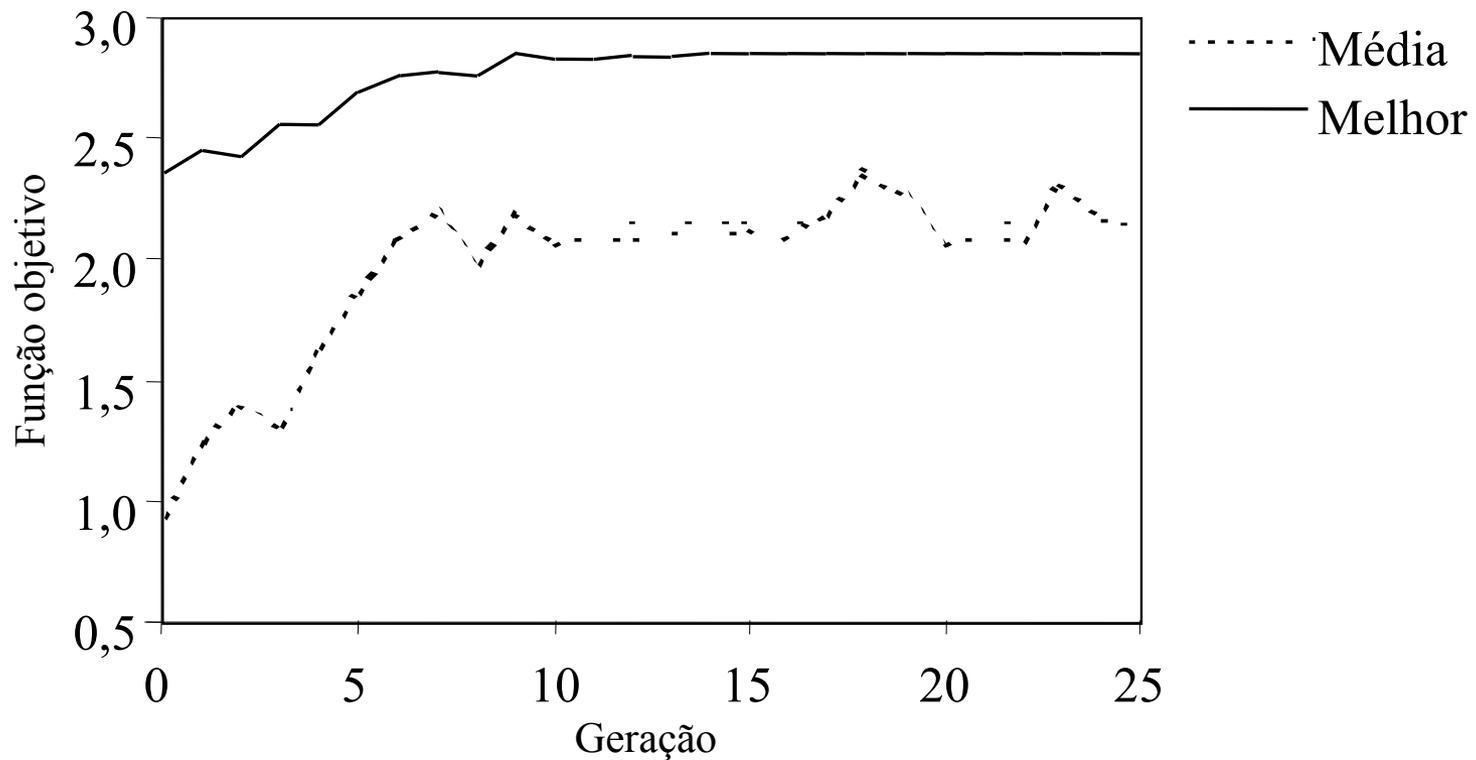
Pouca melhoria

As Gerações do Problema 2

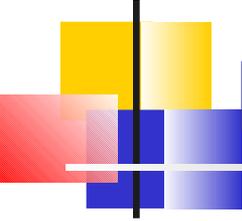


A maioria dos indivíduos encontraram o máximo global

As Gerações do Problema 2



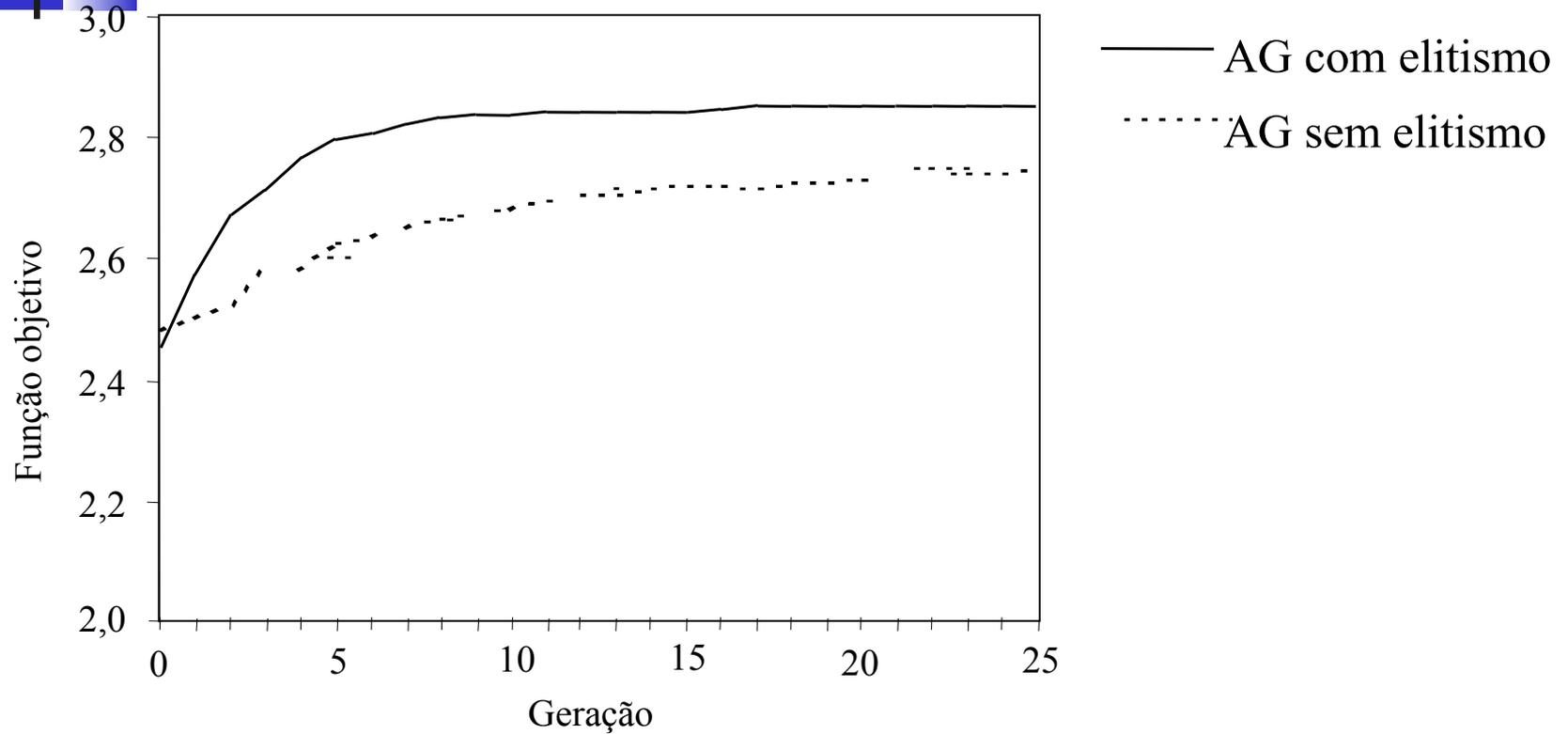
Na geração 15 o AG já encontrou o ponto máximo

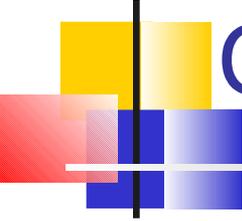


Elitismo

- A substituição simples da geração antiga pela nova podem destruir a melhor indivíduo
- Por que perder a melhor solução encontrada?
- Elitismo transfere cópias dos melhores indivíduos para a geração seguinte

Elitismo no Problema 2





Critérios de Parada

- Número de gerações
- Encontrou a solução (quando esta é conhecida)
- Perda de diversidade (estagnação)
- Convergência
 - nas últimas k gerações não houve melhora na aptidão