utad

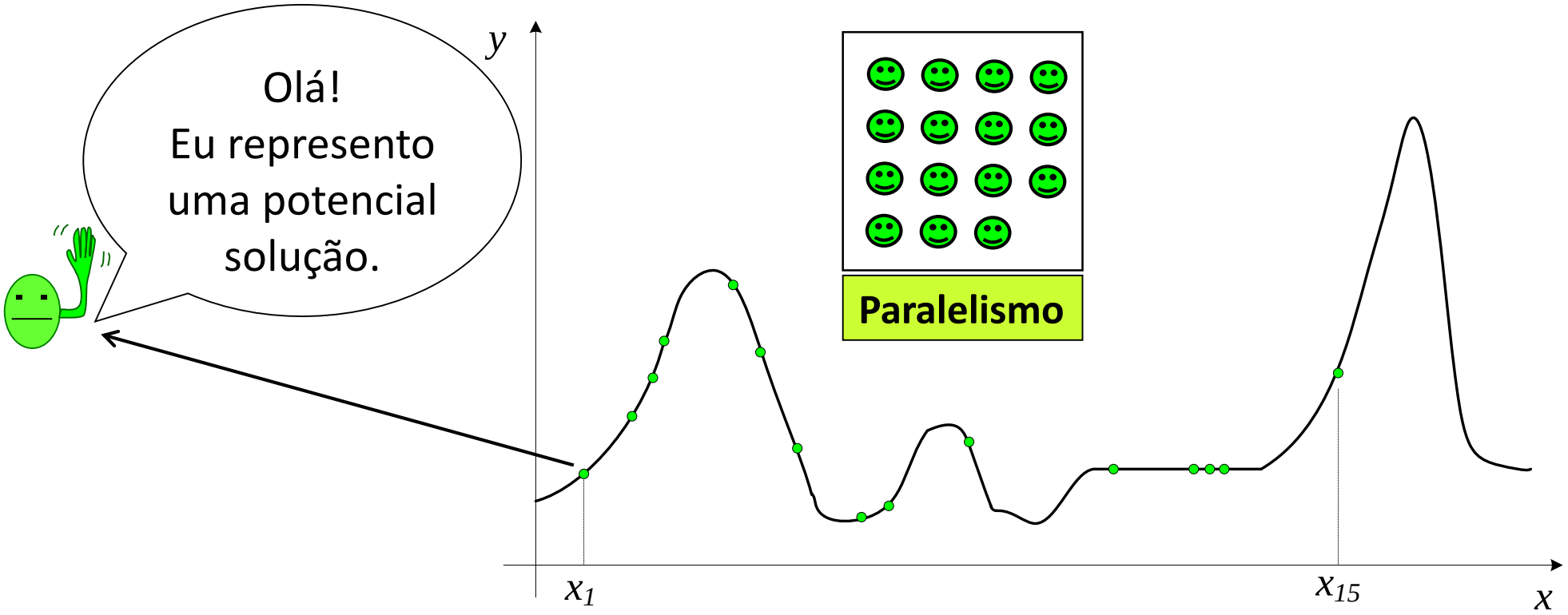
Inteligência Artificial

Métodos de Pesquisa (*Search*) – Parte III

Paulo Moura Oliveira
Departamento de Engenharias
Gabinete F2.15, ECT-1
UTAD
email: oliveira@utad.pt

O Apelo da Evolução

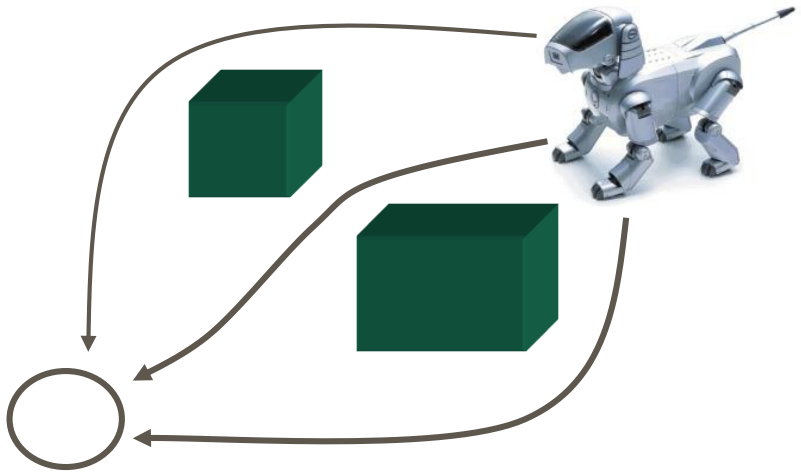
Porque não utilizar mais do que uma potencial solução para um problema de pesquisa?



Para problemas mais complexos a utilização de várias soluções em paralelo pode ser muito vantajosa.

O Apelo da Evolução

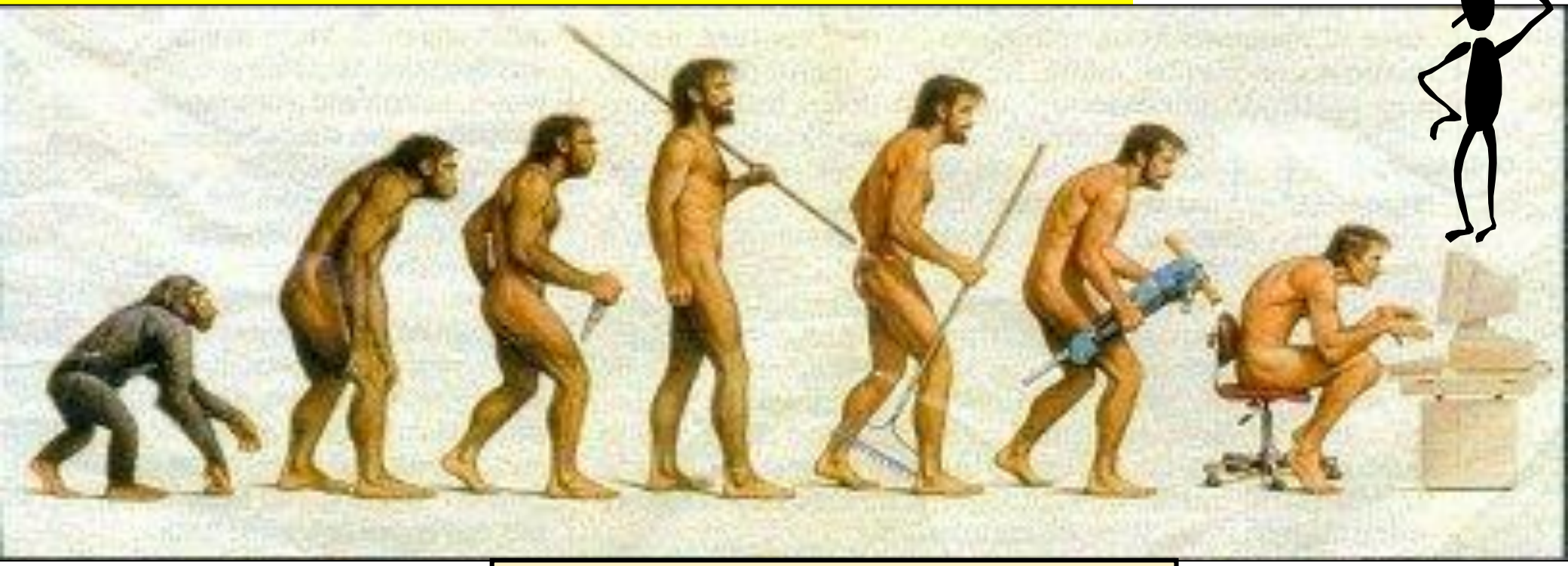
E se o espaço de pesquisa não for estático? Na maioria dos problemas práticos o espaço de pesquisa é dinâmico (e.g. Robótica, Condução Autónoma).



Em muitas aplicações o algoritmo de pesquisa deve ter poder de **adaptação** em ambientes dinâmicos.

O Apelo da Evolução

Porquê utilizar a evolução como uma inspiração para resolver problemas computacionais?

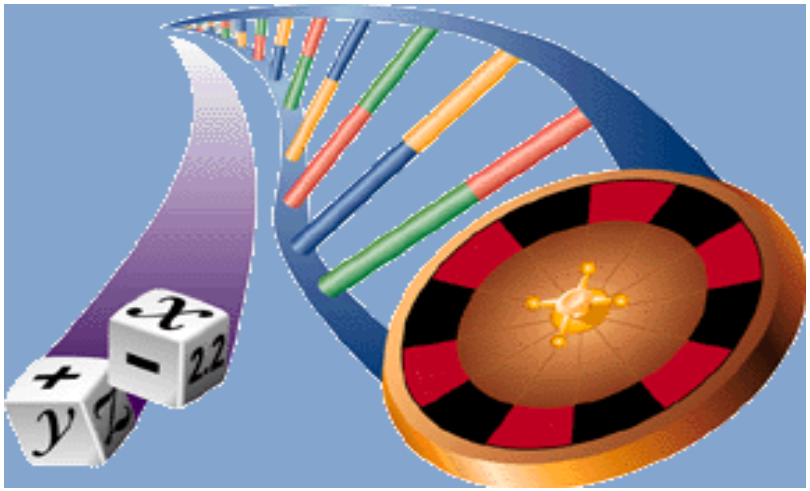


Os sistemas evolutivos:

1. são intrinsecamente paralelos,
2. têm capacidade de adaptação.

Computação Evolucionária

- ✓ Nas décadas de 1950 e 1960 os sistemas evolutivos foram estudados de forma independente por vários cientistas.



A ideia geral nestes sistemas é **evoluir** uma **população de soluções candidatas** a resolver um dado problema usando **operadores** inspirados na **variação genética** e na **seleção natural**.

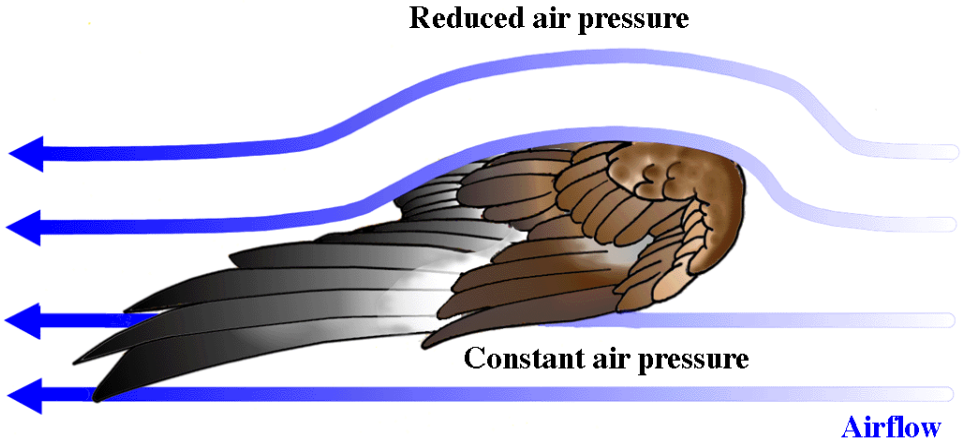
Computação Evolucionária

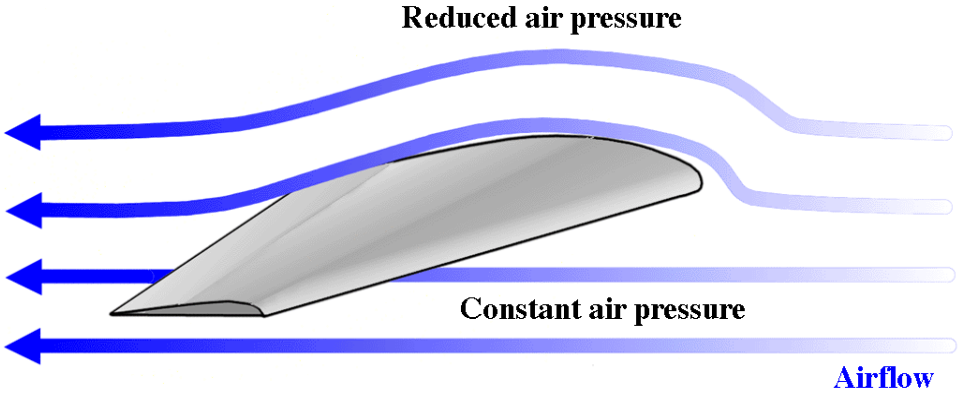
- ✓ Os Algoritmos Evolutivos baseiam-se (mas não só) num modelo da evolução biológica natural formulado inicialmente por Charles Darwin, conhecido como a Teoria da Evolução de Darwin.

Esta teoria explica as **mudanças de adaptação das espécies** pelo princípio da ***Seleção Natural***, no qual as espécies que melhor se **adaptam** às suas condições ambientais são as que melhores possibilidades têm de **sobreviver e evoluir**.

Computação Evolucionária (Origens)

- ✓ Na década de 1960, Rechenberg (1965,1973) propôs Estratégias Evolutivas, um método utilizado para otimizar parâmetros reais para dispositivos como asas de aviões (*airfoils*).





Ingo Rechenberg, Mais info em
https://en.wikipedia.org/wiki/Ingo_Rechenberg
Acedido em 28-7-2018

Computação Evolucionária (Origens)

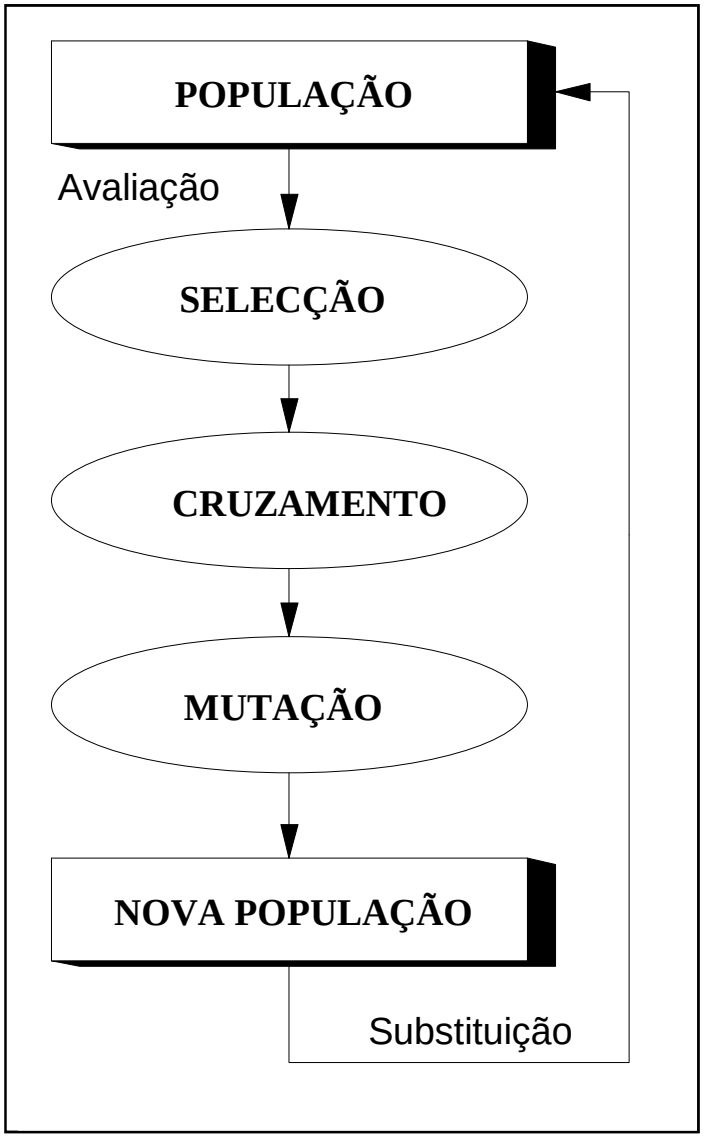
- ✓ Fogel, Owens e Walsh (1966) desenvolveram a **Programação Evolucionária**, uma técnica em que soluções candidatas a determinadas tarefas eram representadas por máquinas de espaços finitos.
- ✓ Os **Algoritmos Genéticos** foram desenvolvidos por John Holland na Universidade de Michigan nas décadas de 1960 e 1970.
- ✓ Desde então muitos outros algoritmos classificados como evolutivos, tais como:
  - ✓ Programação Genética
  - ✓ Evolução Diferencial
  - ✓ ……

Algoritmo Genético (*Genetic Algorithm*, GA)

O que é um Algoritmo Genético?

Um AG é um algoritmo de pesquisa baseado nos mecanismos da seleção natural e da genética.

- A ideia básica é manter uma população de estruturas de conhecimento que representam potenciais soluções para um dado problema.



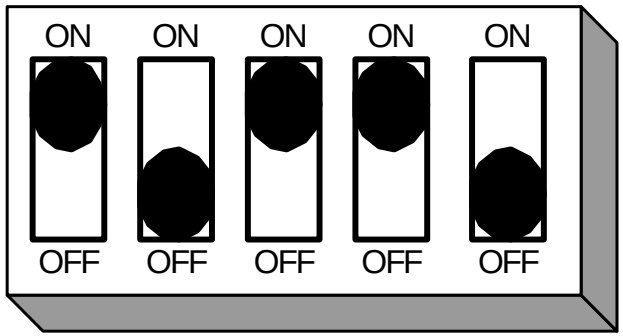
Algoritmo Genético (*Genetic Algorithm*, GA)

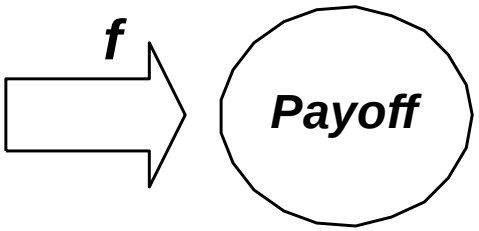
População: Conjunto de estruturas de conhecimento (cromossomas) que representam potenciais soluções para um dado problema

Algoritmo Genético (*Genetic Algorithm*, GA)

População:

✓ Consideremos como exemplo problema da caixa negra ilustrado na figura:





f- Função Objetivo

1 - ON
0 - OFF

1 0 1 1 0

Representação de **um elemento da população** usando uma **codificação binária**.

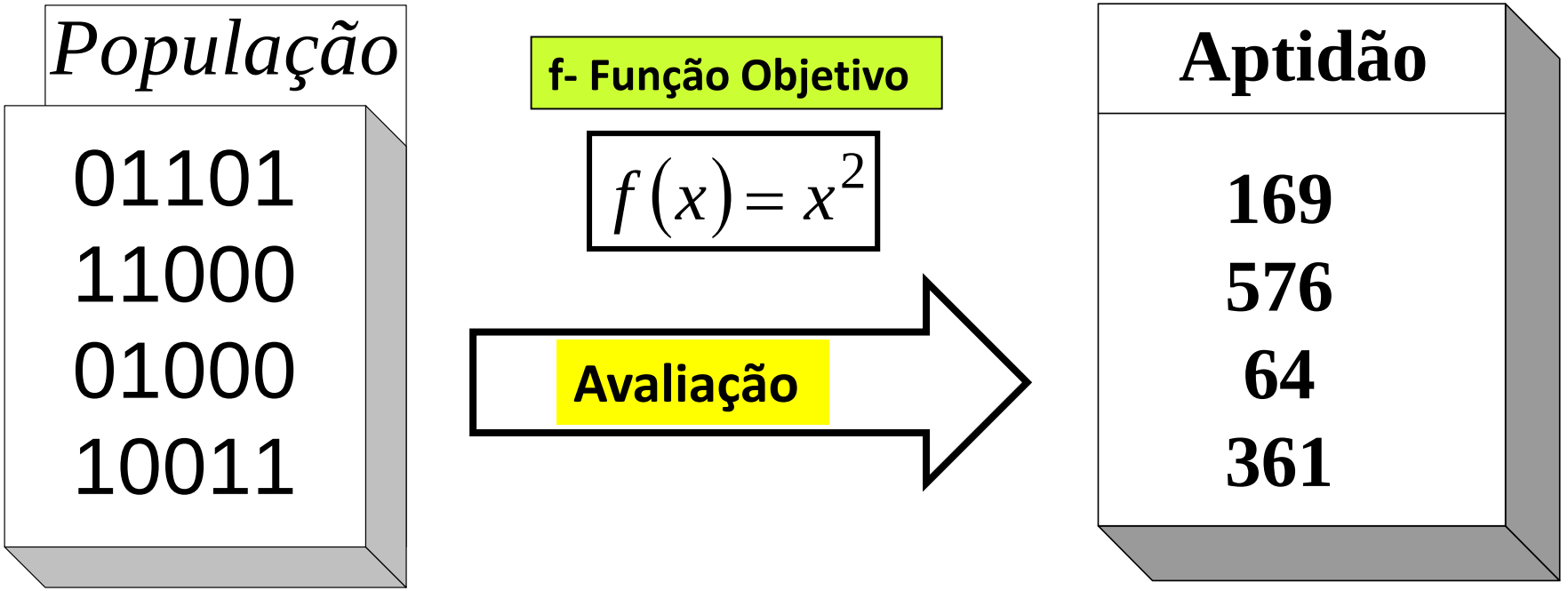
O AG usa codificações!

Algoritmo Genético (*Genetic Algorithm*, GA)

População:

✓ Consideremos que a ***população*** genética ***inicial*** é inicializada de uma forma ***aleatória*** usando o processo de atirar a moeda (20 vezes):

Esta população inicial vai dar origem a uma série de novas populações utilizando o AG.

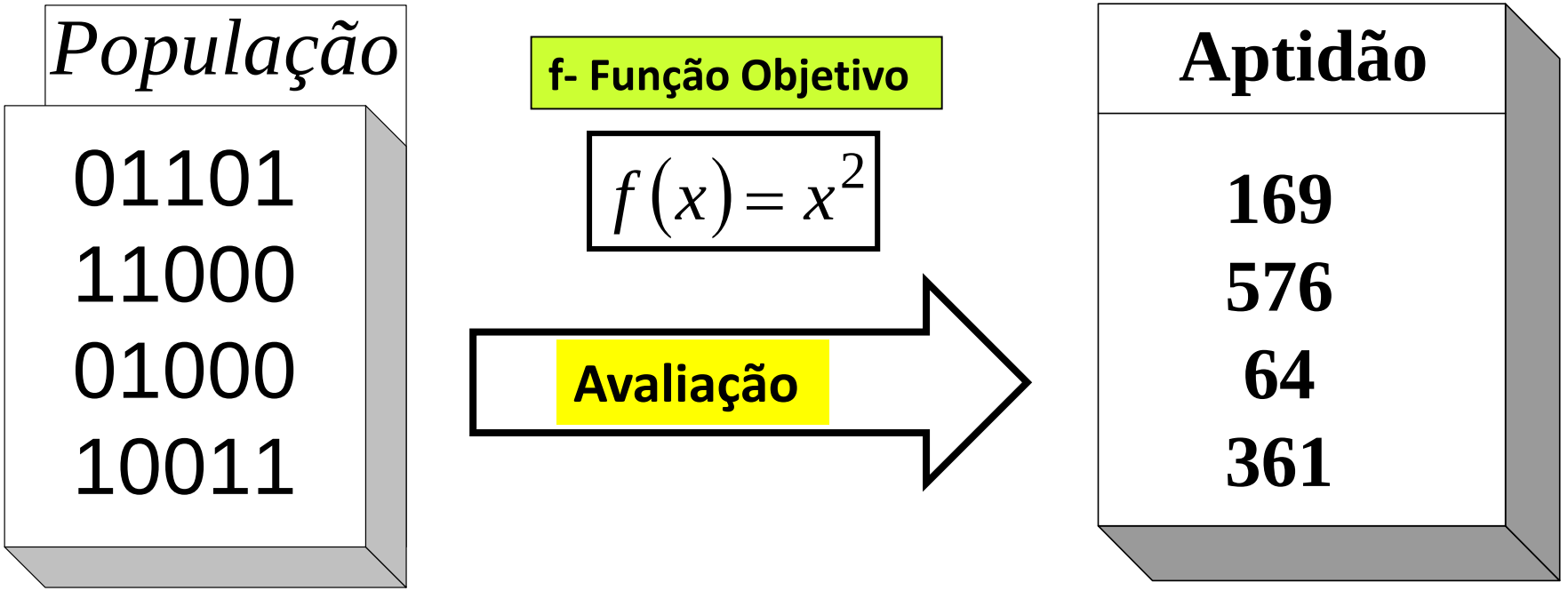


Algoritmo Genético (*Genetic Algorithm*, GA)

População:

✓ Consideremos que a ***população*** genética ***inicial*** é inicializada de uma forma ***aleatória*** usando o processo de atirar a moeda (20 vezes):

Esta população inicial vai dar origem a uma série de novas populações utilizando o AG.



Algoritmo Genético (*Genetic Algorithm*, GA)

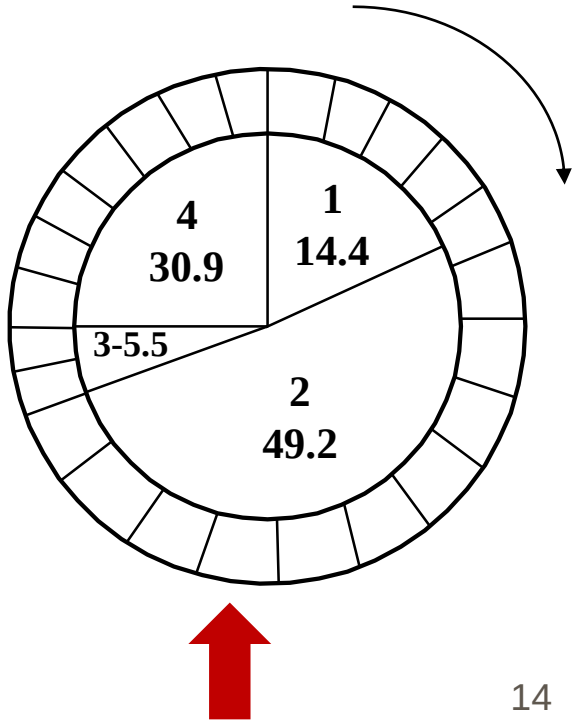
Seleção:

- ✓ Os indivíduos com maior aptidão devem ter maior probabilidade de contribuir com um ou mais descendentes na próxima geração.

Versão artificial da seleção natural e da sobrevivência do mais apto!

- ✓ Como implementar o operador da seleção? Há muitos métodos, sendo o da Roda da Roleta o mais conhecido.

Nº	String	Aptidão	% do Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

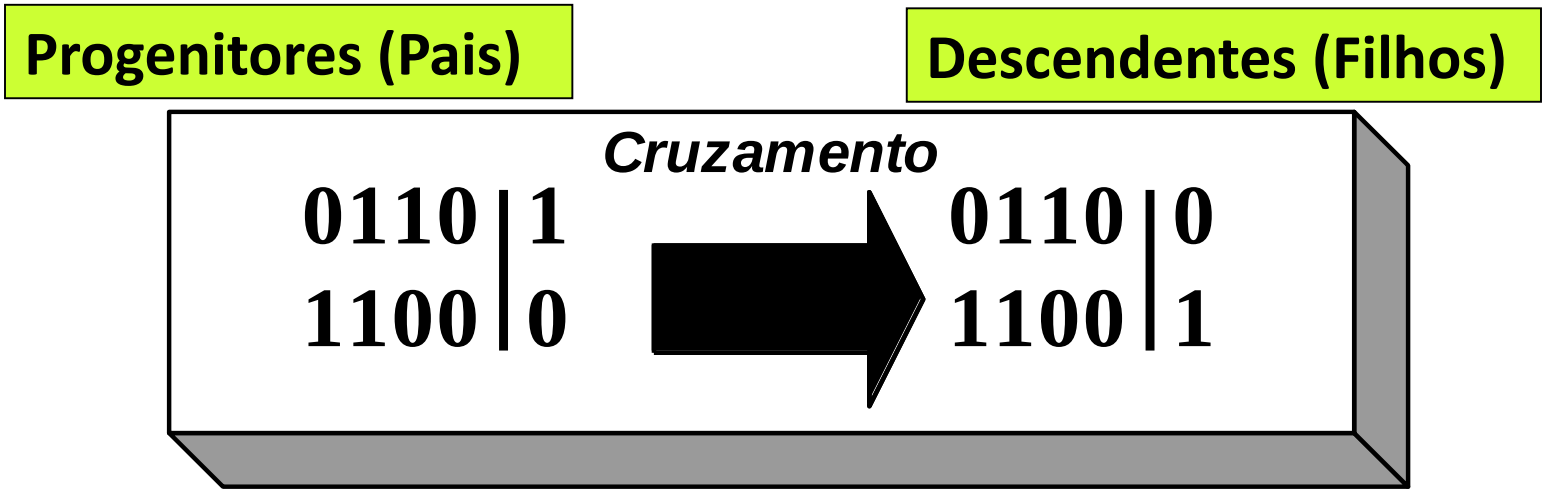


Algoritmo Genético (*Genetic Algorithm*, GA)

Cruzamento:

- ✓ Os indivíduos selecionados para gerar a nova população vão ter o seu material genético cruzado de forma a obter os descendentes.

- ✓ Como implementar o operador de cruzamento? Há muitos métodos, sendo um dos mais simples, o **cruzamento de ponto simples**.



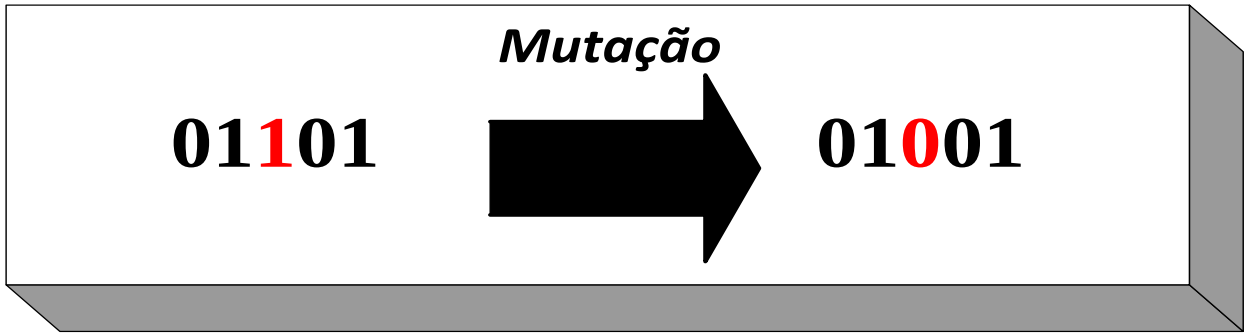
- ✓ O cruzamento é feito de acordo com uma determinada **probabilidade de cruzamento**, p_c (esta taxa usualmente é elevada).

Algoritmo Genético (*Genetic Algorithm*, GA)

Mutação:

- ✓ Embora a seleção e cruzamento pesquisem de forma eficaz podem ocasionalmente tornar-se zelosos demais e perder informação genética importante.

- ✓ Como implementar o operador de Mutação? Uma forma simples consiste em alterar um bit de acordo com uma determinada **probabilidade de mutação**, p_m (esta taxa usualmente é baixa).

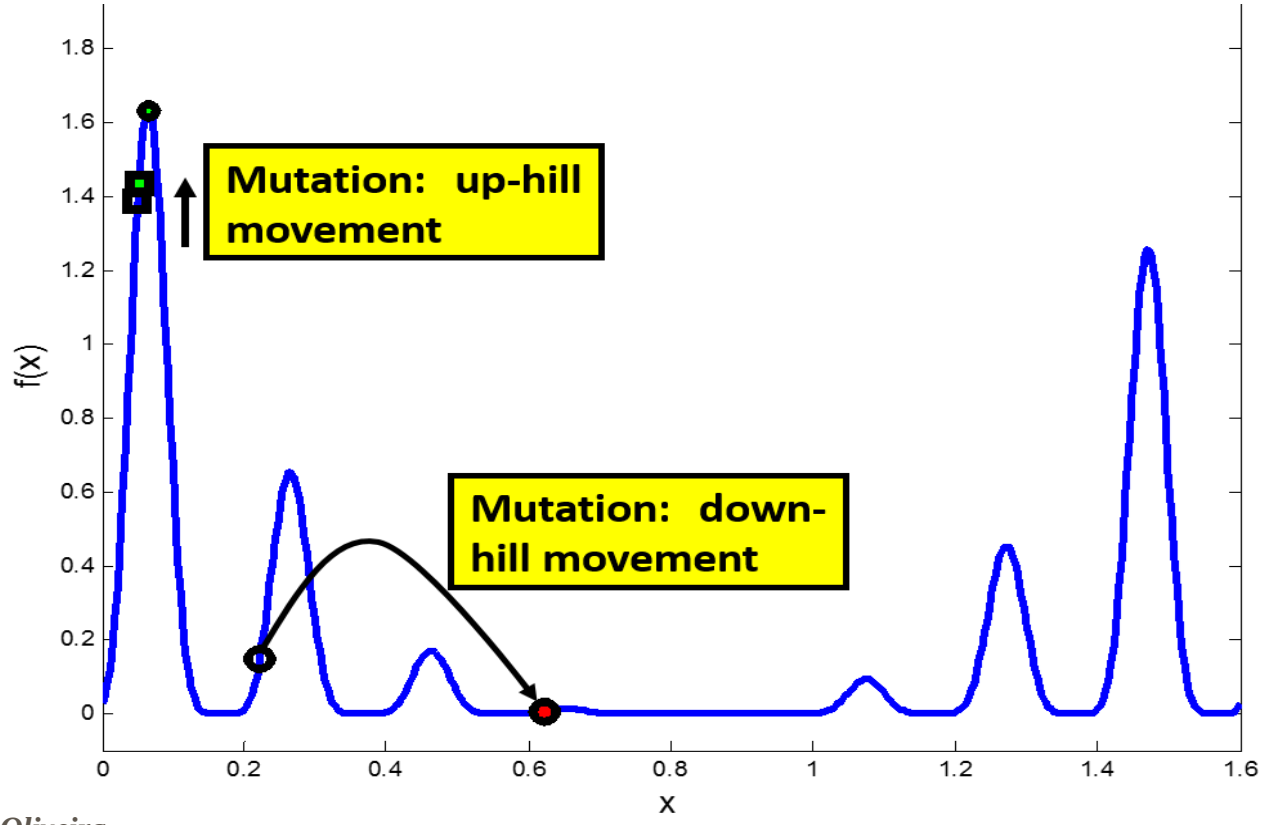


- ✓ Neste exemplo de população: assumindo p_m =0.001. Isto significa que num total de 5x4=20 bits, 0.001*20=0.02 bits são sujeitos a mutação numa dada geração. Ou seja nenhum bit é modificado.

Algoritmo Genético (*Genetic Algorithm*, GA)

Mutação:

- ✓ A mutação embora, de uma forma geral, ocorra com frequência muito baixa pode desempenhar uma papel muito importante, pois pela simples mutação de um bit a solução pode variar significativamente no espaço.



Algoritmo Genético (*Genetic Algorithm*, GA)

Resumo do Exemplo:

Nº	População Inicial	Valor de *x* *Inteiro*	$f(x)=x^2$	$pselect_i$ $\frac{f_i}{\sum f}$	nº de cópias esperado $\frac{f_i}{\bar{f}}$	nº de cópias obtido na RR
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
Total			1170	1.0	4.0	4.0
Média			293	0.25	1.0	1.0
Máximo			576	0.49	1.97	2.0

Algoritmo Genético (*Genetic Algorithm*, GA)

Resumo do Exemplo:

✓ Após um cruzamento:

Série de Acasalamento	Parceiro *(random)*	*Ponto de Cruzamento (random)*	População Nova	Valor de *x*	*f(x)*
0110 | 1	2	4	0110 0	12	144
1100 | 0	1	4	1100 1	25	625
11 | 000	4	2	11 011	27	729
10 | 011	3	2	10 000	16	256
Total					**1754**
Média					**439**
Máximo					**729**

✓ Comparação com os valores obtidos na geração anterior:

1170
293
576

Algoritmo Genético (*Genetic Algorithm*, GA)

Um Algoritmo Genético Simples:

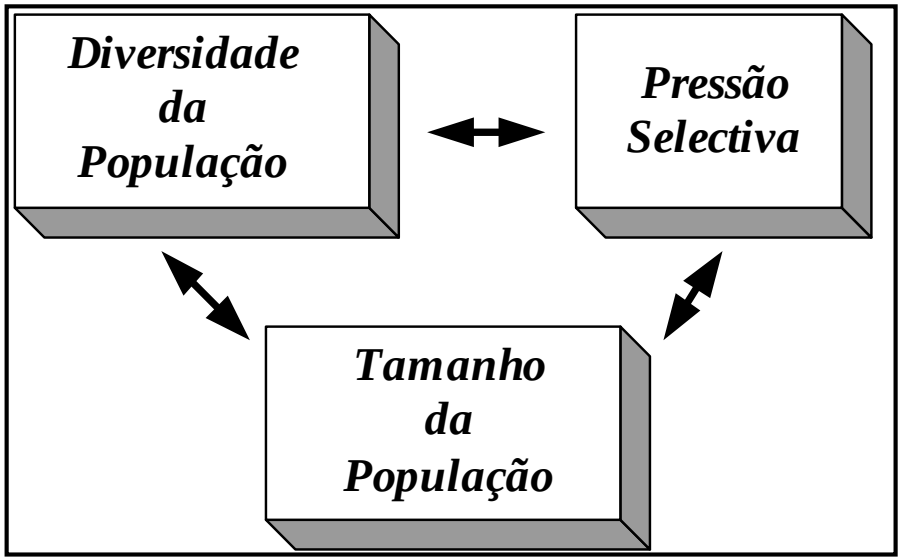
```
t = 0
inicializar população X(t)
while(!(critério de término))
   avaliar a população
   seleção
   cruzamento
   mutação
   substituição da população
   t=t+1
end while
```

- ✓ Vimos o princípio de funcionamento para cada um dos passos deste algoritmo. Assumindo uma codificação binária.
- ✓ Existe um manancial de tópicos associados aos GA, alguns dos quais extravasam o carácter deste curso introdutório à IA.

Algoritmo Genético (*Genetic Algorithm*, GA)

Tópicos Selecionados

1. Que método de codificação é o mais apropriado para um dado problema? (e.g. binário, real, inteiro, …)
2. Que tipo operador de **seleção**, **cruzamento e mutação** se deve utilizar?
3. Como garantir a **diversidade genética** da população ao longo da evolução do GA?

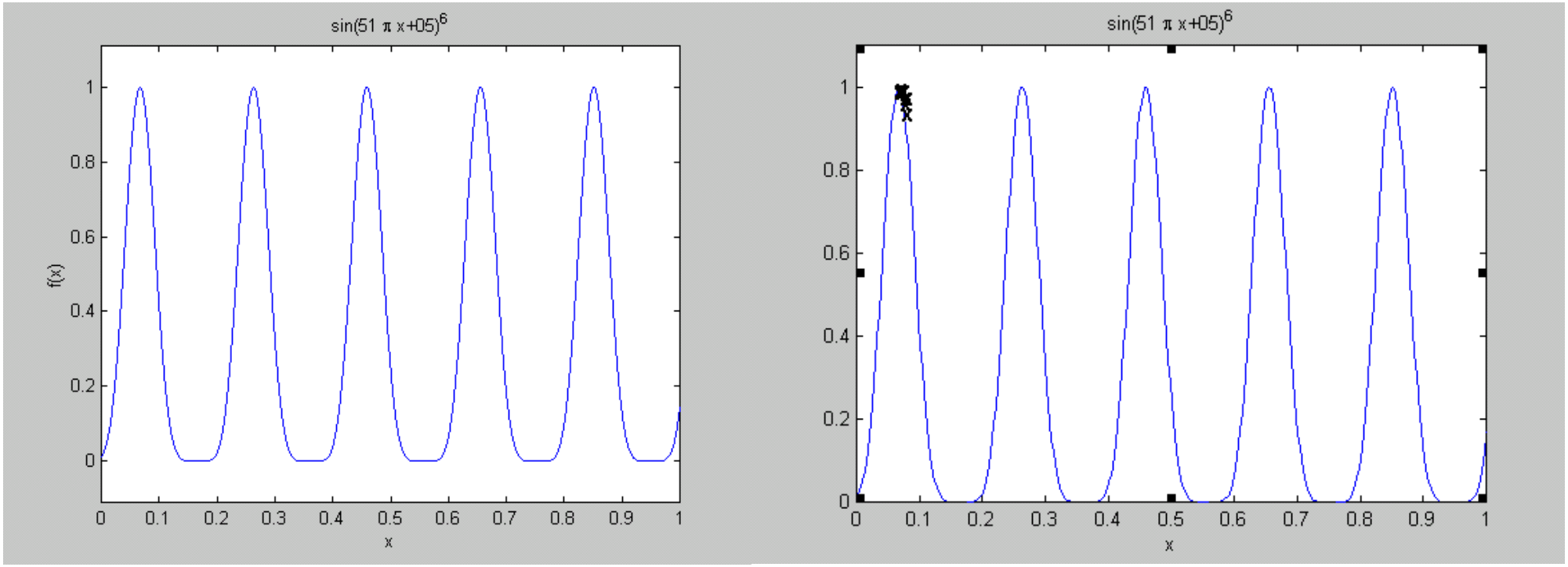


4. Como evitar a **convergência prematura** do algoritmo?

Algoritmo Genético (*Genetic Algorithm*, GA)

Tópicos Selecionados

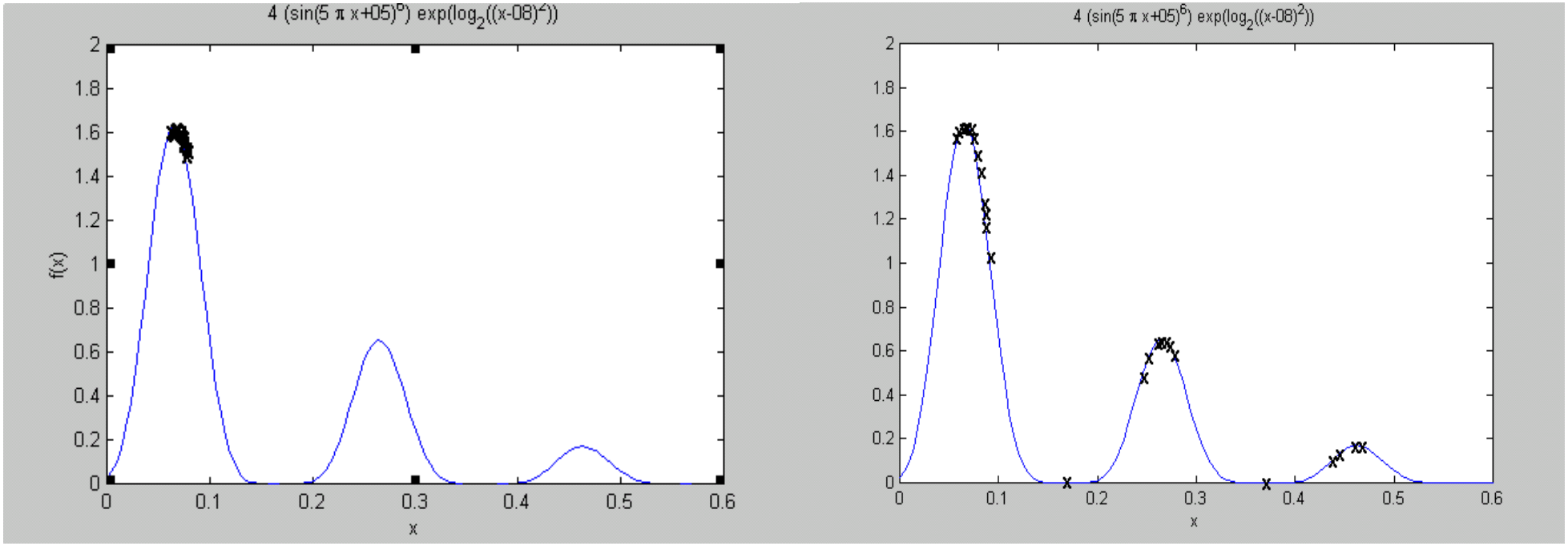
4. Em problemas multimodais (com mais de que um ótimo global) como fazer com que o GA explore e permaneça nos vários picos?



Algoritmo Genético (*Genetic Algorithm*, GA)

Tópicos Selecionados

4. Como manter ótimos locais na população?



- ✓ Em problemas com espaço de pesquisa dinâmico, isto pode ser importante.

Algoritmo Genético (*Genetic Algorithm*, GA)

Tópicos Selecionados

5. Como inicializar a população?

6. Quando e como se deve utilizar técnicas de restrição ao acasalamento? (e.g. *imbreeding*, *crossbreeding*, ….)

7. Como proceder à substituição da população?

8. Como garantir que a informação genética de um indivíduo fora de série não seja perdida na evolução?

- ✓ O ***Elitismo*** foi uma técnica proposta por De Jong (1975) que garante que o melhor individuo num geração está na nova população.

(nota: outros modelos de elitismo foram propostos na sua tese de doutoramento).

Programação Genética (*Genetic Programming*, GP)

- ✓ Enquanto num GA um elemento da população originalmente é uma *string* binária na programação genética um elemento é representado por uma árvore (programa).

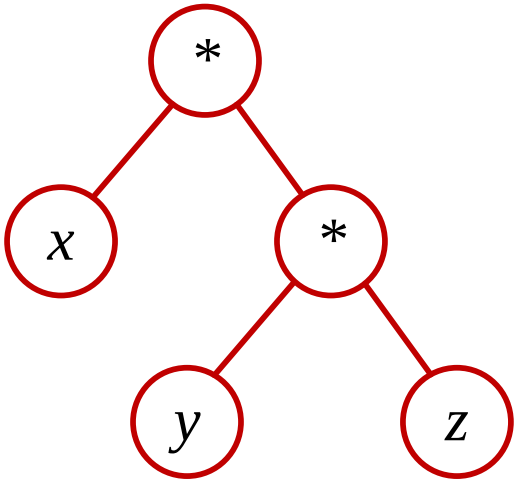
- ✓ GP adequa-se a ser implementada em linguagens do tipo LISP.

Exemplo (*) :

*(*x(*yz))

Computa:

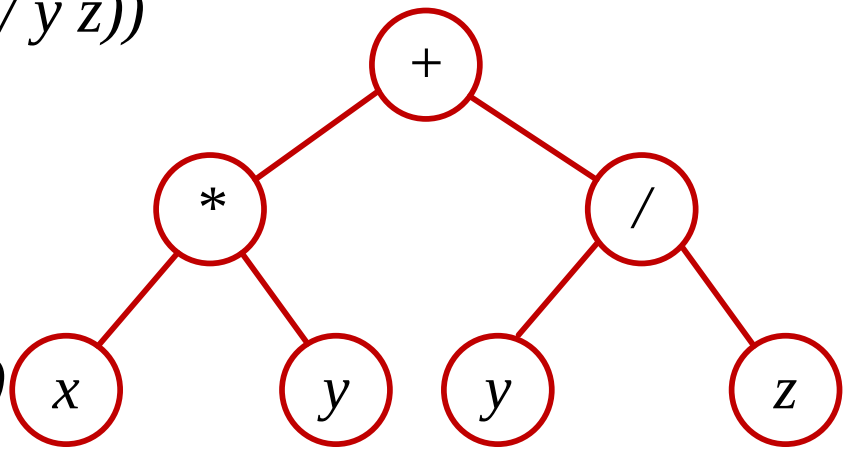
x*(y*z)



+ ((* x y)(/ y z))

Computa:

(x*y)+(y/z)



- ✓ Nota: adaptado de Lucci S. e Kopec D., ArtificiAl intelligence in the 21st century

Programação Genética (*Genetic Programming*, GP)

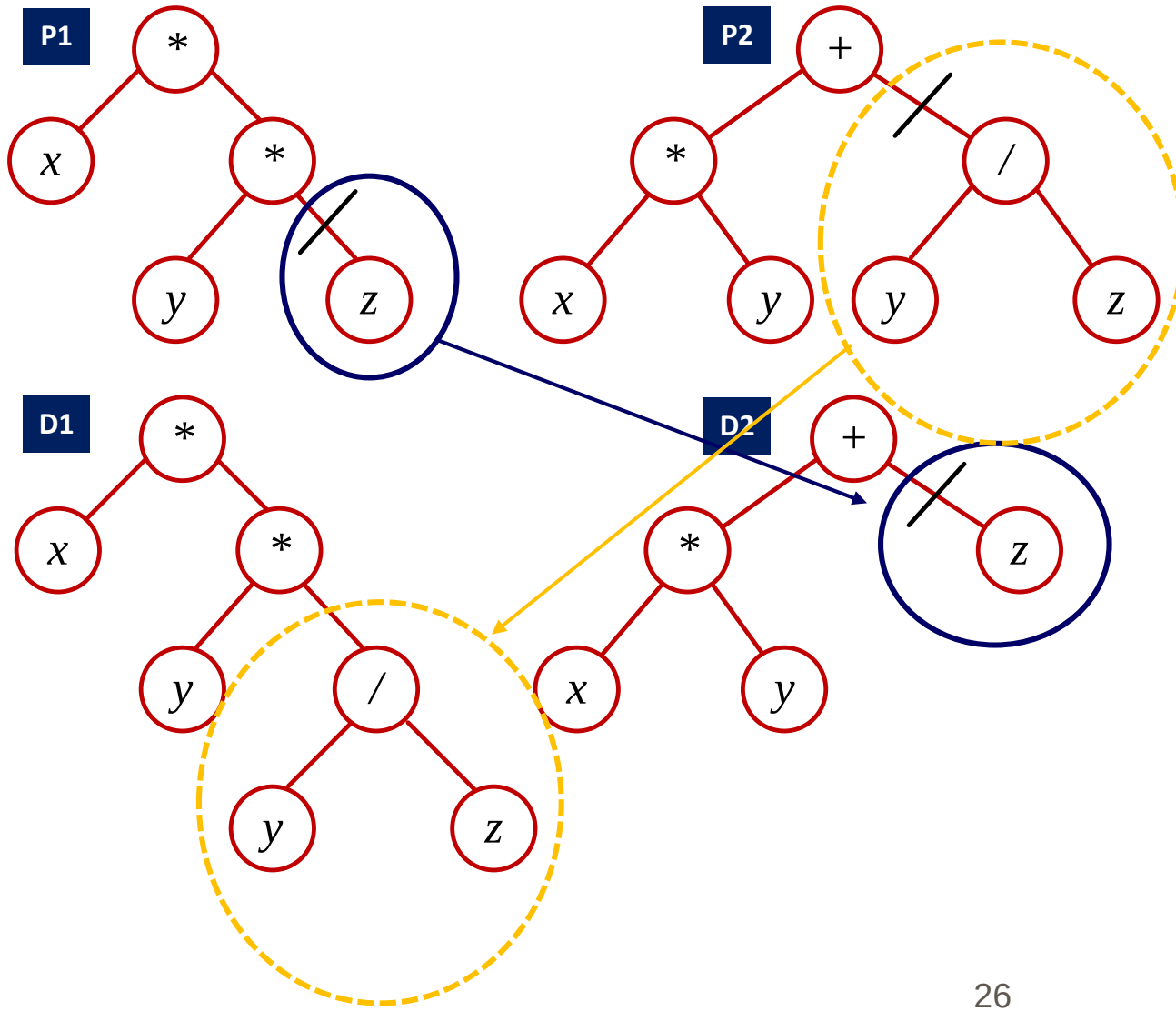
✓ Alguns operadores da GP são: **cruzamento**, **inversão** e **mutação**.

Cruzamento:

Selecionar dois progenitores (programas) aleatoriamente

Selecionar duas sublistas aleatoriamente

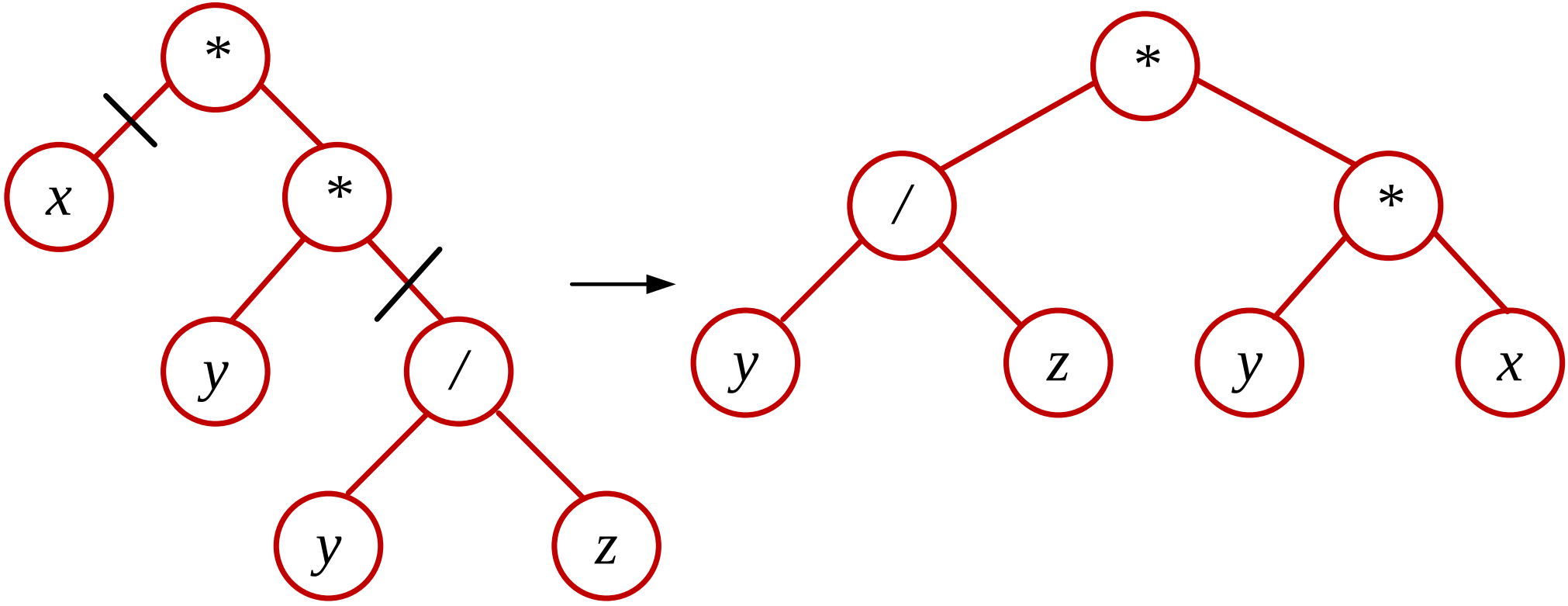
Trocar as sublistas para obter os dois descendentes



Programação Genética (*Genetic Programming*, GP)

Inversão:

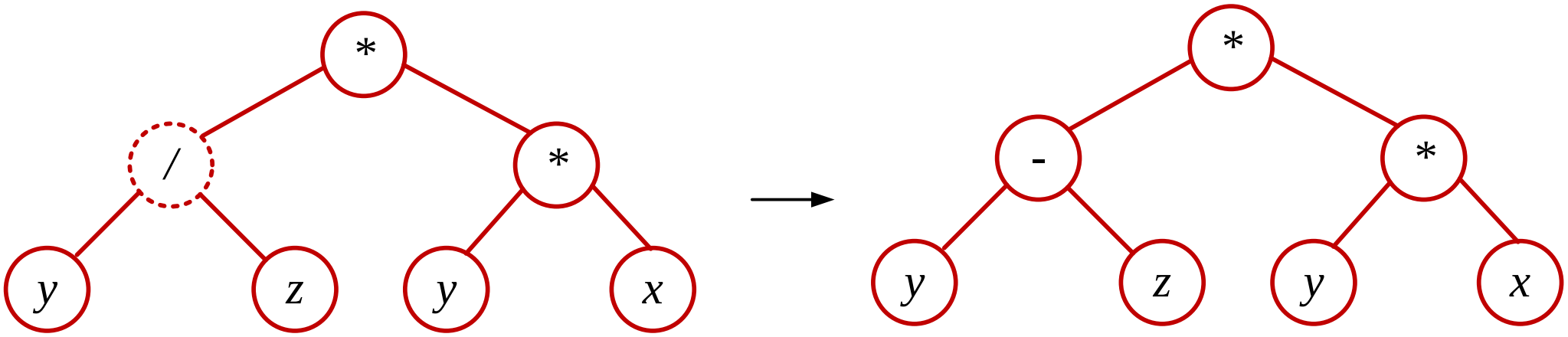
Selecionar um progenitor (programa) aleatoriamente
Selecionar dois pontos de fratura aleatoriamente
Trocar as árvores respetivas



Programação Genética (*Genetic Programming*, GP)

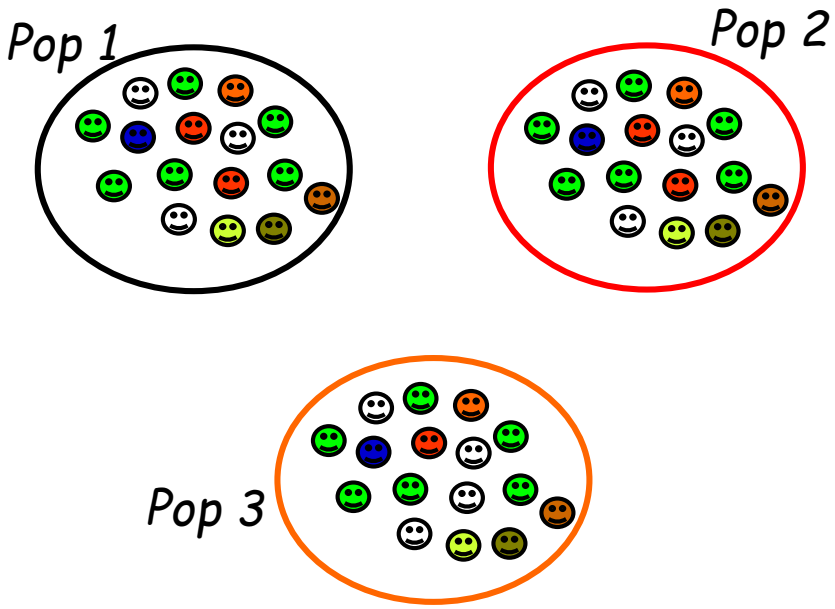
Mutação:

Selecionar um progenitor (programa) aleatoriamente
Substituir um símbolo escolhido aleatoriamente por outro



Algoritmos Evolutivos em Paralelo

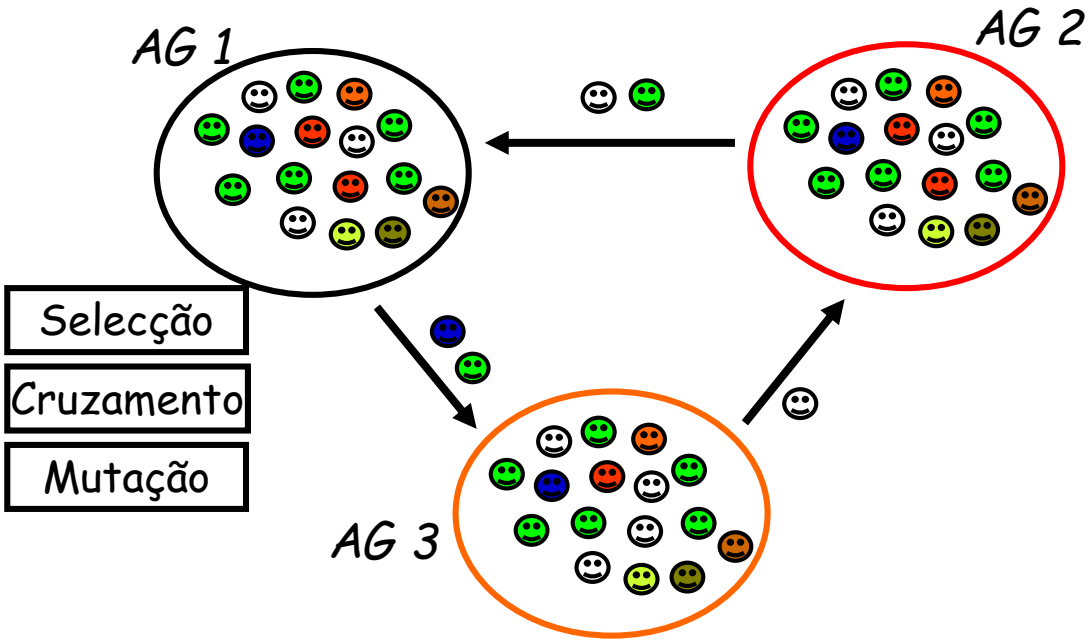
- ✓ Problemas mais complexos podem utilizar algoritmos com várias populações genética a evoluir em paralelo.



- ✓ Cada população pode evoluir isoladamente (modelo das ilhas) ou haver trocas entre as várias populações.

Algoritmos Evolutivos em Paralelo

- ✓ Uma variante segue o **modelo das ilhas (também chamado distribuído)**.



Um AE é executado para cada sub-população.

Há migração de indivíduos entre populações.

Cada ilha pode ser atribuída a um processador.

- ✓ Vantagens: a migração pode melhorar a diversidade melhora a capacidade de pesquisa e pode haver divisão de tarefas.

Coevolução Artificial de Espécies

Métodos Baseados na Coevolução de Espécies

Coevolução Competitiva

Coevolução Cooperativa

- ✓ Como pode a coevolução artificial ser utilizada para resolver problemas computacionais?