

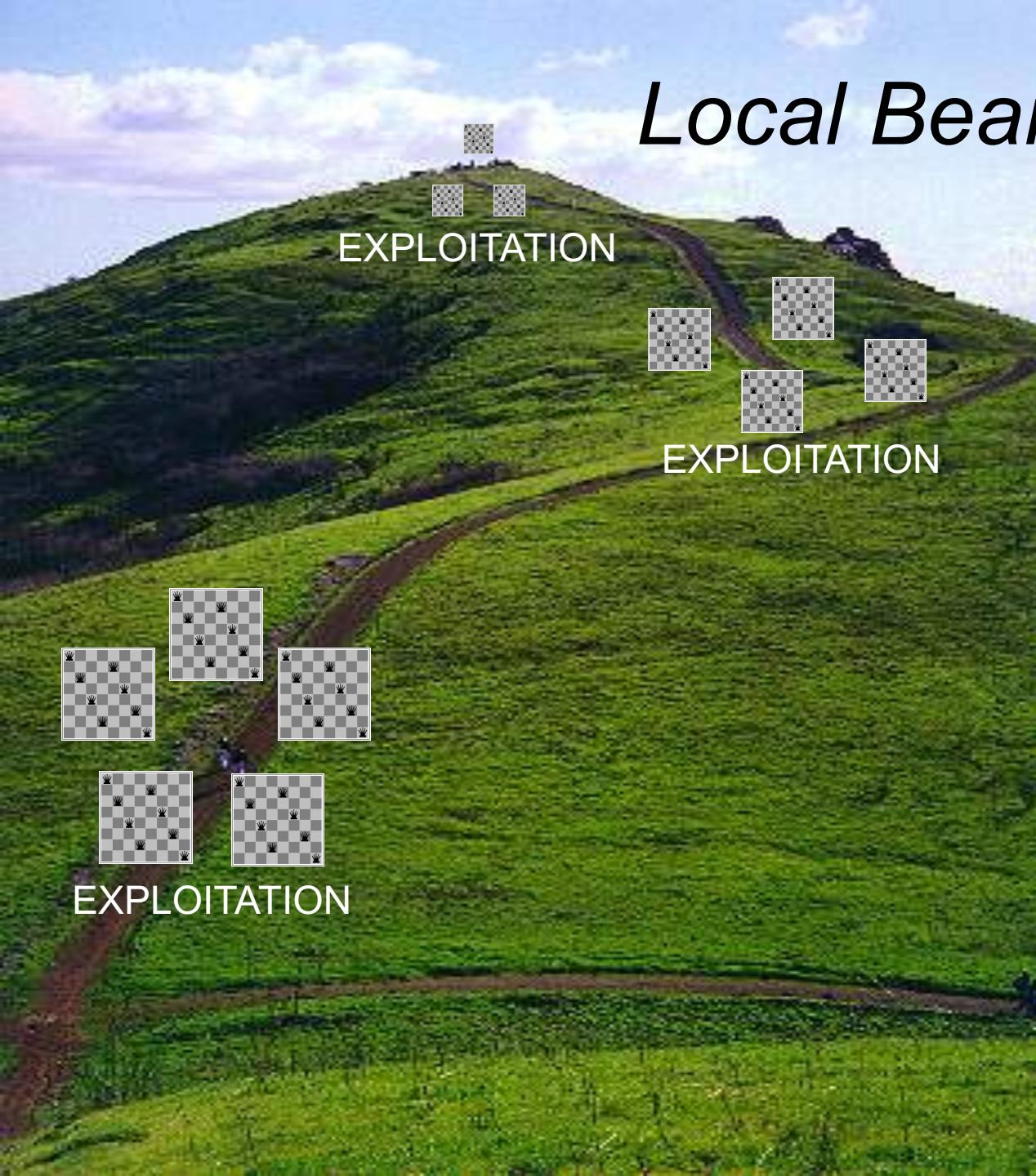
COMP0271: Inteligência Artificial

Busca local, parte 2

Professor: Hendrik Macedo

Universidade Federal de Sergipe, Brasil

Local Beam Search



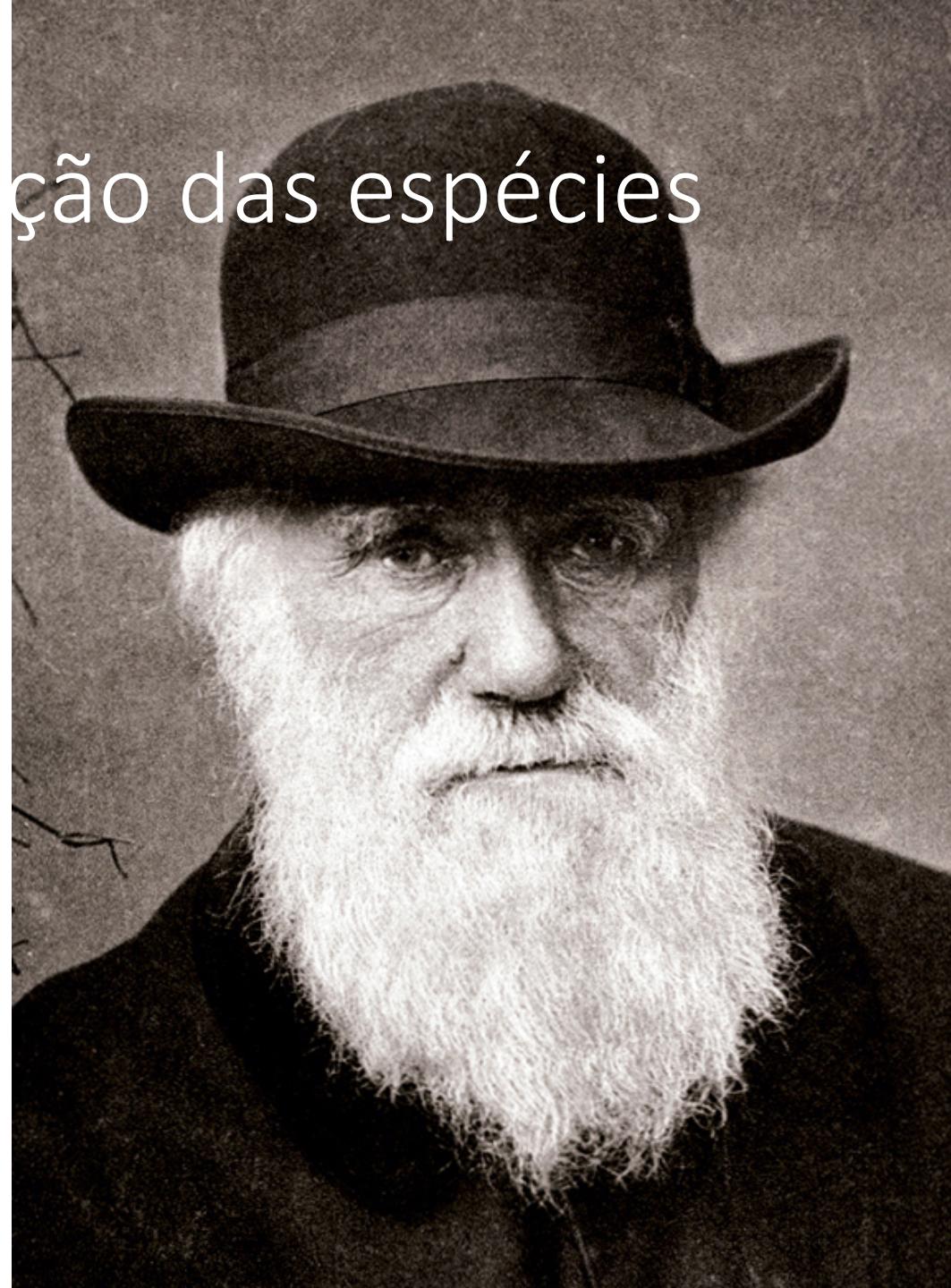
Inicia com k estados gerados aleatoriamente

A cada passo todos os sucessores de todos os k estados são gerados. Os melhores k estados são então escolhidos e o processo se repete

Ainda carece
de diversidade!

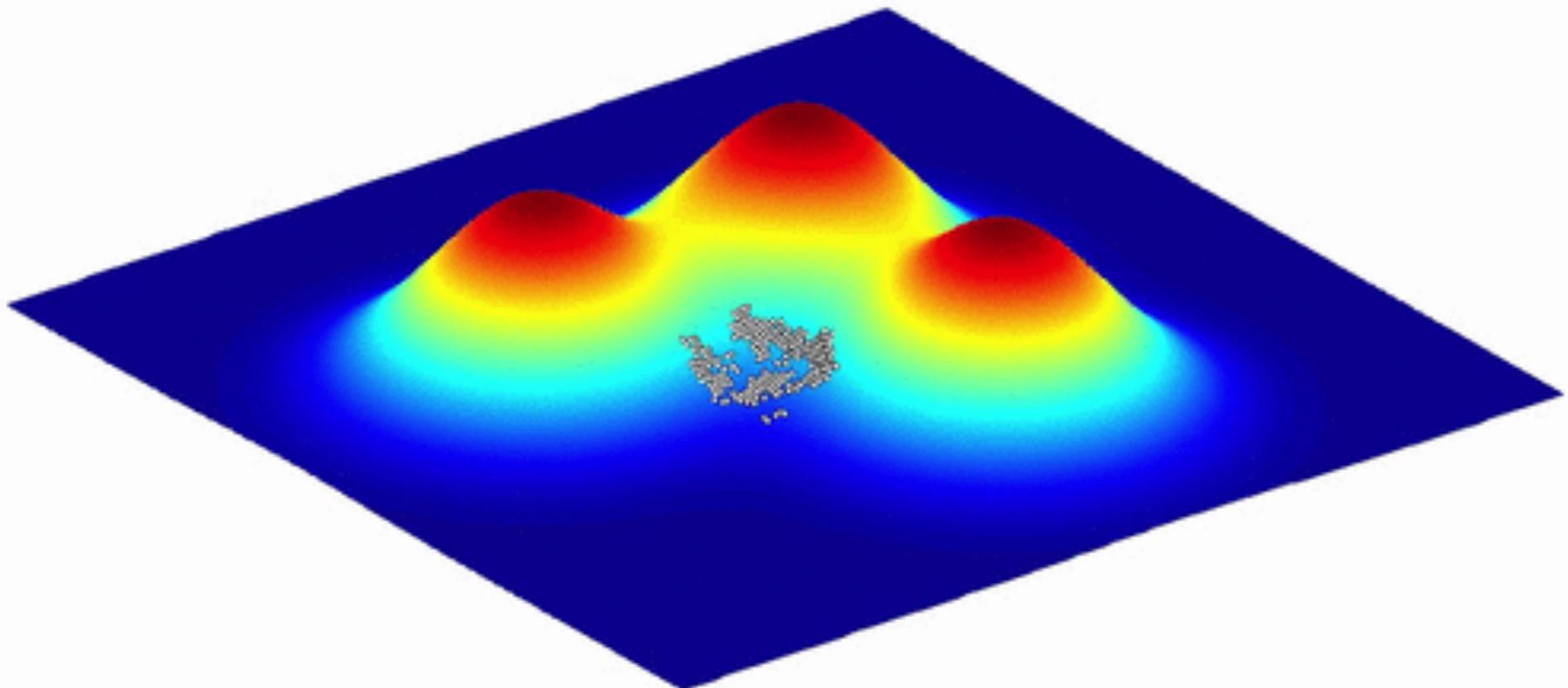
Metáfora da Evolução das espécies

Animais que vivem tempo suficiente para reproduzir são mais atrativos e, portanto, acasalam mais e geram uma quantidade maior e melhor de descendentes



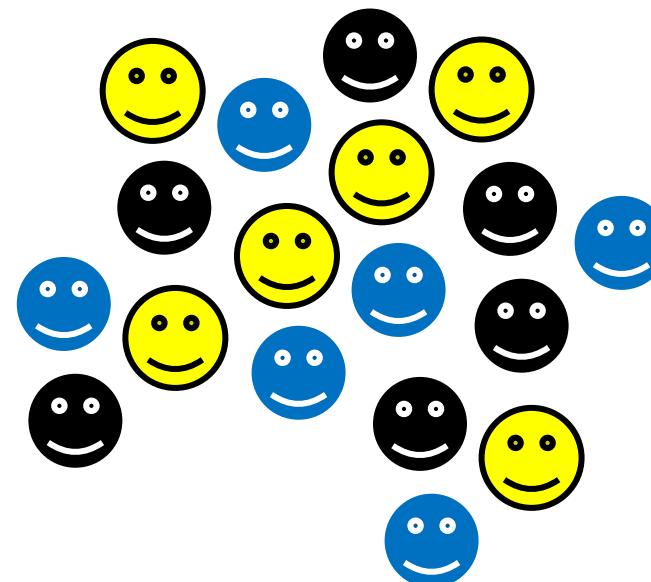
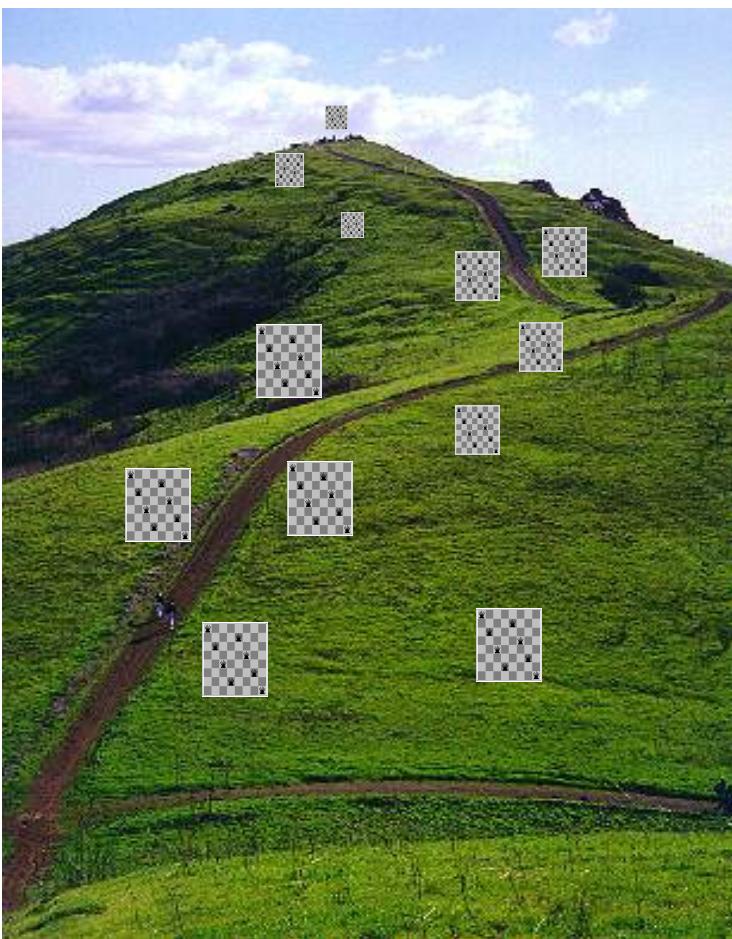
Algoritmos Genéticos (GA)

Aproximação computacional de como a evolução realiza busca na então denominada **Superfície de fitness**



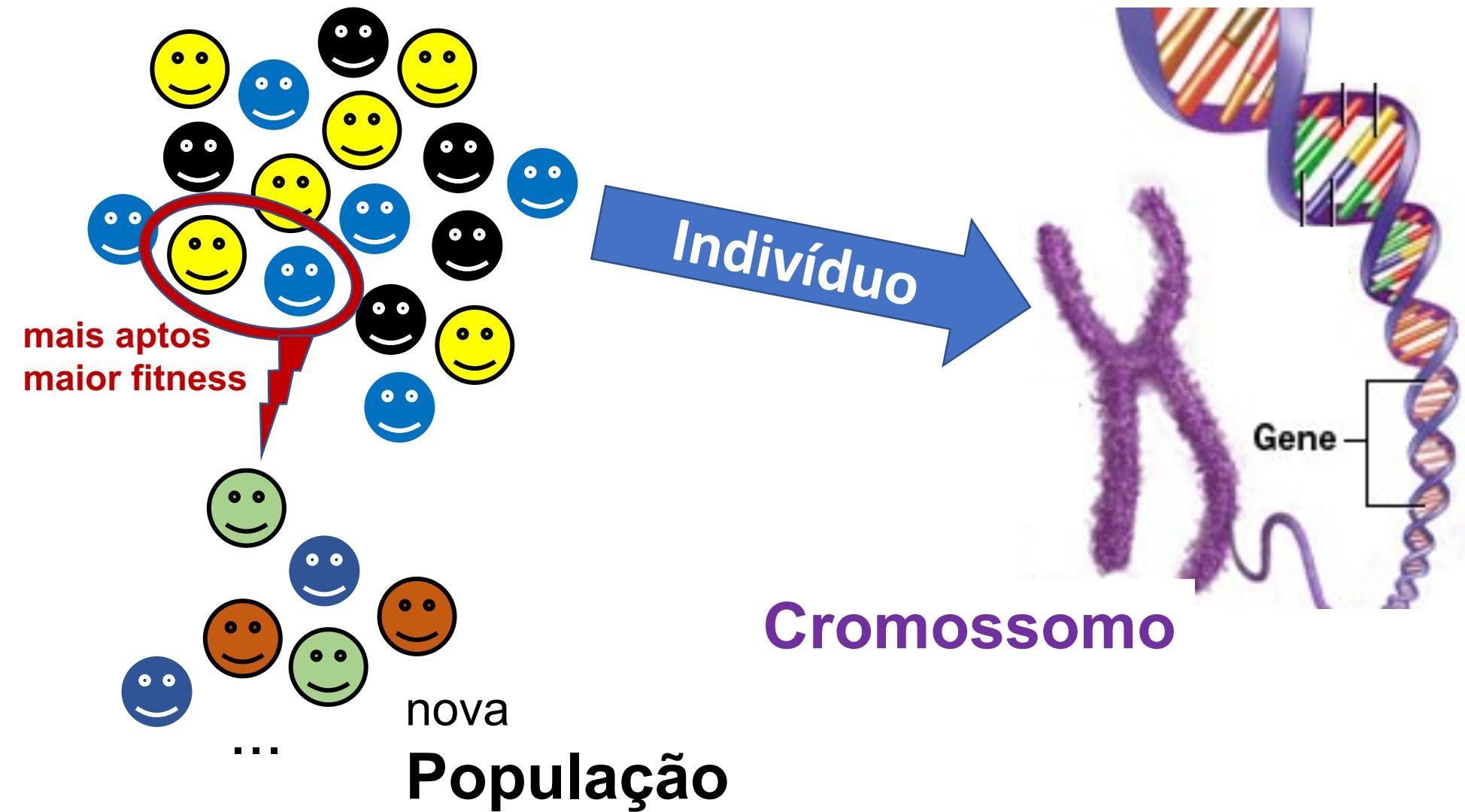
Algoritmos Genéticos (GA)

Variação do *Local Beam Search*



População
de soluções

Algoritmos Genéticos (GA)



Algoritmo Genéticos (GA)

- Precisamos modelar
 - como representar uma solução como um cromossomo;
 - uma forma de calcular o fitness desse cromossomo;
 - um método de seleção dos indivíduos "pais"
 - uma forma de gerar os descendentes



Camera
Weight: 1 kg
Value: 1000\$

①



②



Necklace
Weight: 4 kg
Value: 4000\$

③

Vase
Weight: 5 kg
Value: 4500\$



④



Knapsack
Capacity: 7 kg
Max value: ???

The Knapsack problem

GA: Representação do cromossomo

① ② ③ ④

0 1 1 0

Número L de genes ∴ valores escolhidos
a partir de um alfabeto

GA: Definir função *fitness*

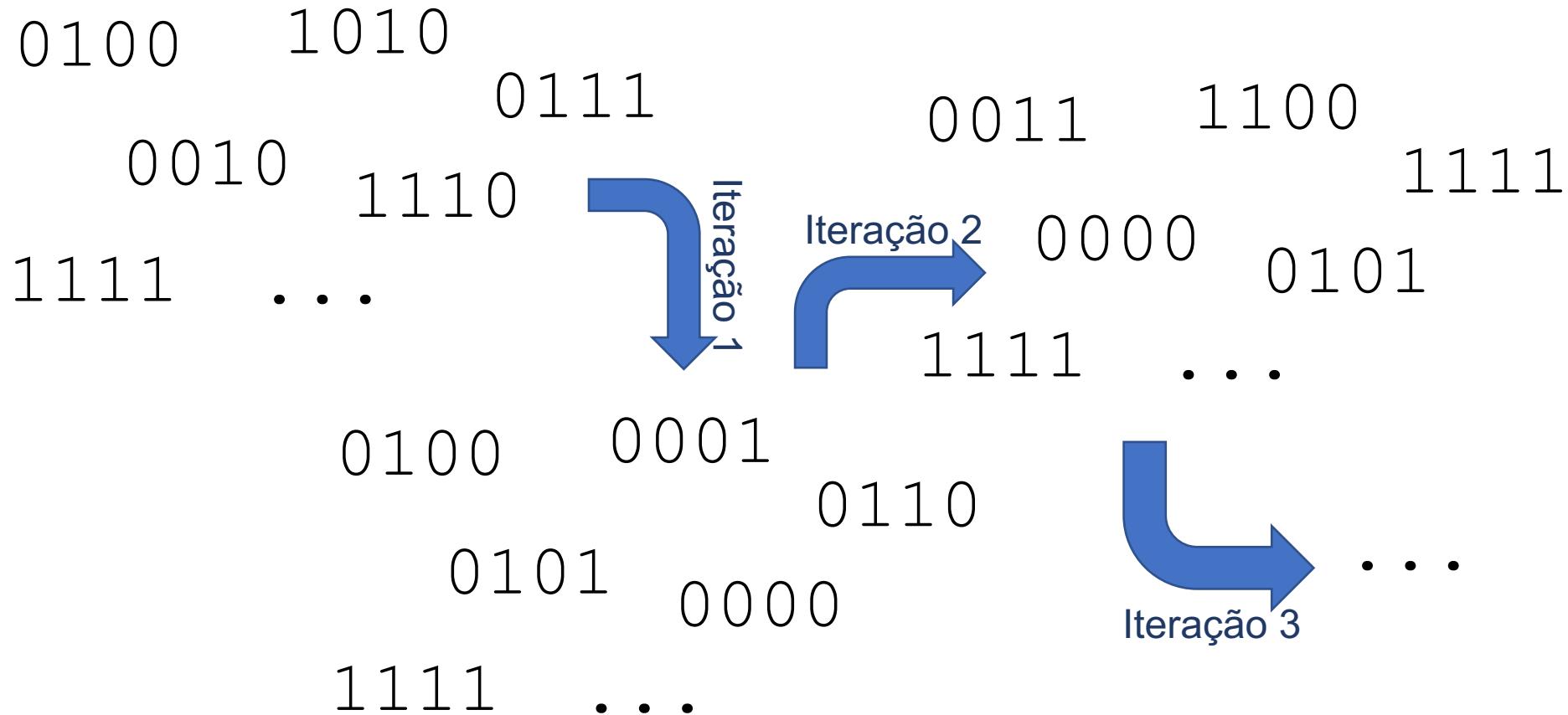
$$f(0110) = ?$$

$$f(g_1g_2g_3g_4) = \begin{cases} \sum_{i=1}^4 g_i * value_i , & \sum_{i=1}^4 g_i * weight_i \leq capacity \\ \left(\sum_{i=1}^4 g_i * value_i \right) - 2 * \left(capacity - \sum_{i=1}^4 g_i * weight_i \right), & \sum_{i=1}^4 g_i * weight_i > capacity \end{cases}$$

GA: População

População inicial

gerada aleatoriamente



GA: Seleção dos "pais"

Priorizar strings com maiores fitness,
mas sem ignorar as outras

0100 1010

0010 1110

1111 1100

0011

0111 0000

- Métodos
 - Tournament Selection
 - Seleciona 4 strings aleatoriamente e seleciona as duas mais aptas!
 - Fitness Proportional Selection (FPS)
 - Probabilidade de uma string ser selecionada é proporcional à sua aptidão dentro da população

$$p^\alpha = \frac{f^\alpha}{\sum_{\alpha'} f^{\alpha'}}$$

GA: como combinar as duas strings selecionadas?

0100 1010

0010 1110
1111 1100
0011

0111 0000

Operadores genéticos:

Crossover
Mutação

GA: Crossover (single point)

Posição aleatória



001101100100100110 **pai 1**

011100010101011010 **pai 2**

001101010101011010 **filho 1**

011100100100100110 **filho 2**

GA: Crossover (multi point)

Posição aleatória Posição aleatória

001101100100100110 **pai 1**

011100010101011010 **pai 2**

001101010101011110 **filho 1**

011100100100100010 **filho 2**

GA: Mutation

Posição aleatória com
probabilidade de ocorrência p



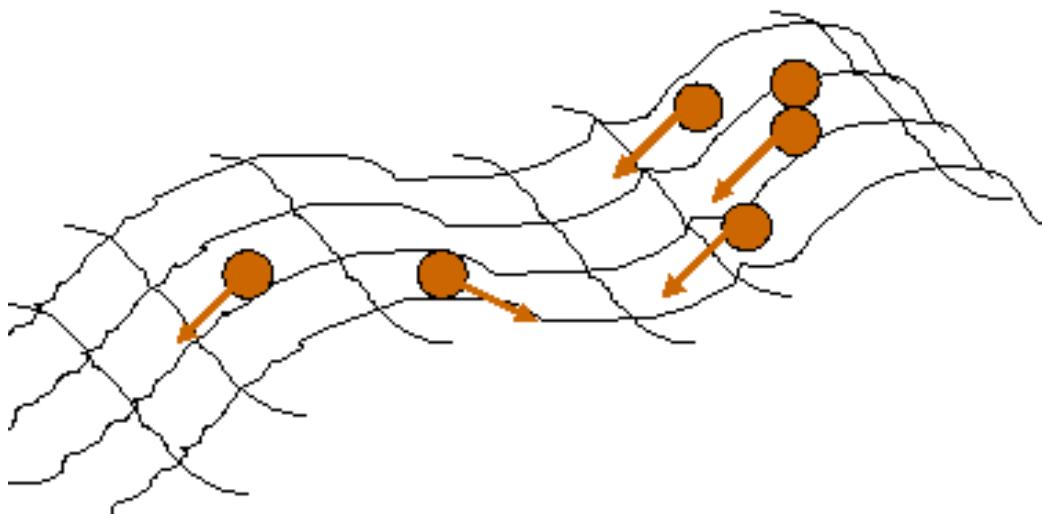
0011010101**1**1011110 **filho 1**

Posição aleatória com
probabilidade de ocorrência p



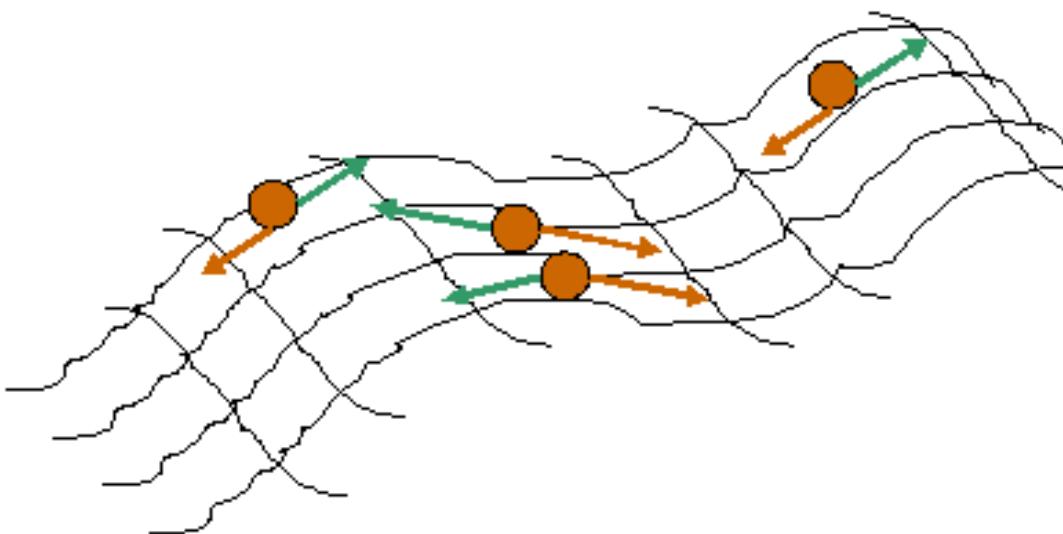
011**0**00100100100010 **filho 2**

Operadores Genéticos e Busca



Mutação cria variantes na vizinhança
Seleção aceita as melhores

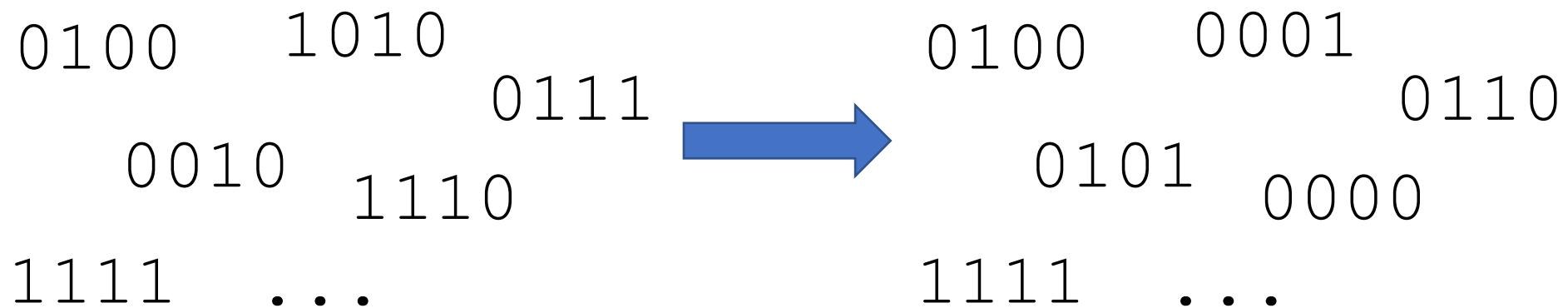
EXPLOITATION



Crossover combina variantes distantes para sair do min/max local

EXPLORATION

GA: Substituição da população



Métodos:

Complete substitution: Pais → Filhos

Elitism: Pais → Filhos seguido por N random filhos → N melhores pais

Tournament: 2 pais vs. 2 filhos → 2 melhores

Niching: subpopulações e imigrações

function GENETIC-ALGORITHM(*population*, FITNESS-FN) **returns** an individual

inputs: *population*, a set of individuals

FITNESS-FN, a function that measures the fitness of an individual

repeat

new-population \leftarrow empty set

for *i* = 1 **to** SIZE(*population*) **do**

$x \leftarrow$ RANDOM-SELECTION(*population*, FITNESS-FN)

$y \leftarrow$ RANDOM-SELECTION(*population*, FITNESS-FN)

$child \leftarrow$ REPRODUCE(x, y)

if (small random probability) **then** $child \leftarrow$ MUTATE($child$)

 add $child$ to *new-population*

$population \leftarrow new_population$

until some individual is fit enough, or enough time has elapsed

return the best individual in *population*, according to FITNESS-FN

function REPRODUCE(*x, y*) **returns** an individual

inputs: *x, y*, parent individuals

$n \leftarrow$ LENGTH(*x*); $c \leftarrow$ random number from 1 to *n*

return APPEND(SUBSTRING(*x*, 1, *c*), SUBSTRING(*y*, *c* + 1, *n*))

Ciclo evolutivo

(1) Definição de um indivíduo

(2) Geração aleatória
da população

*Função de
Aptidão*

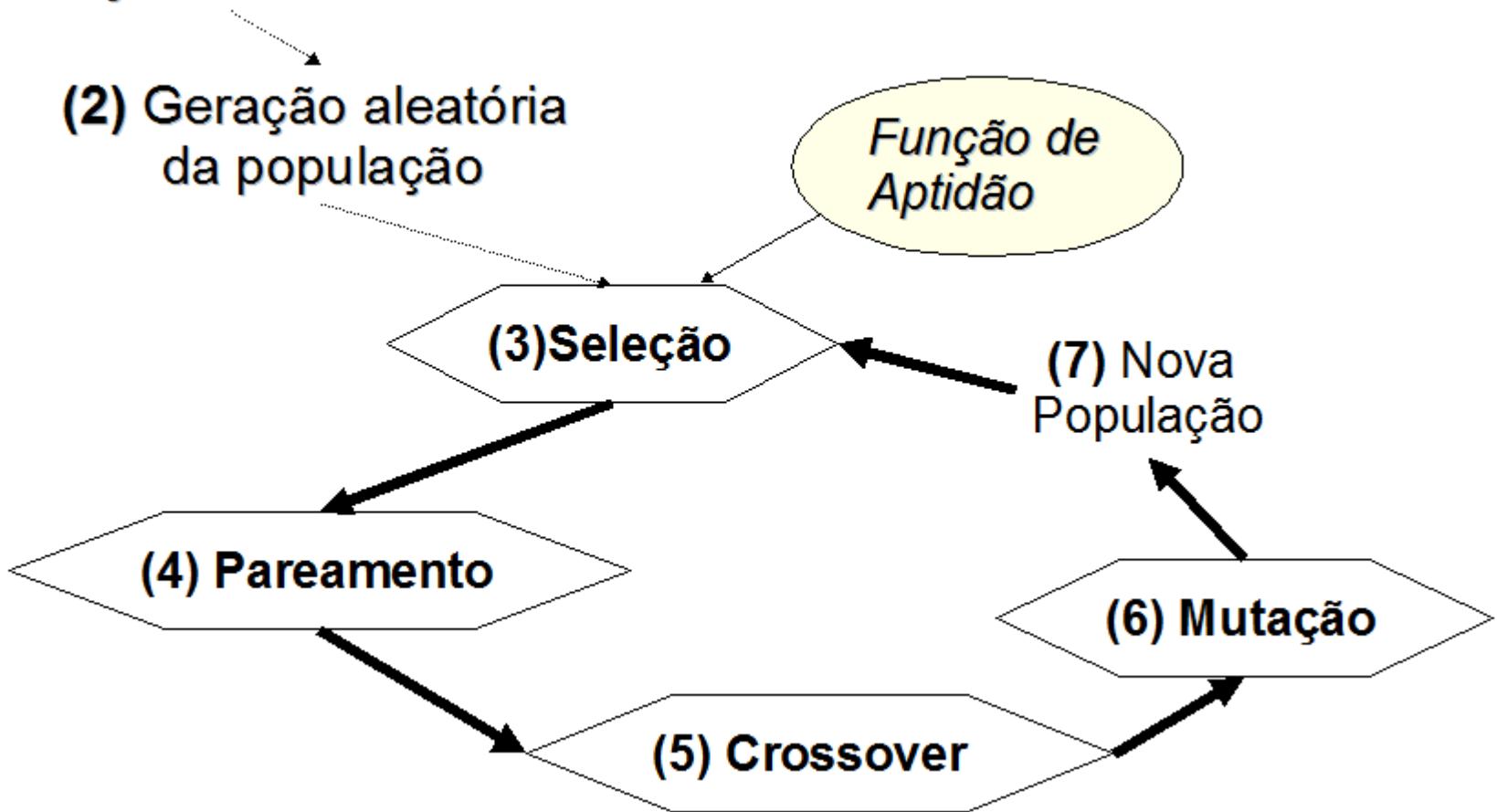
(3) Seleção

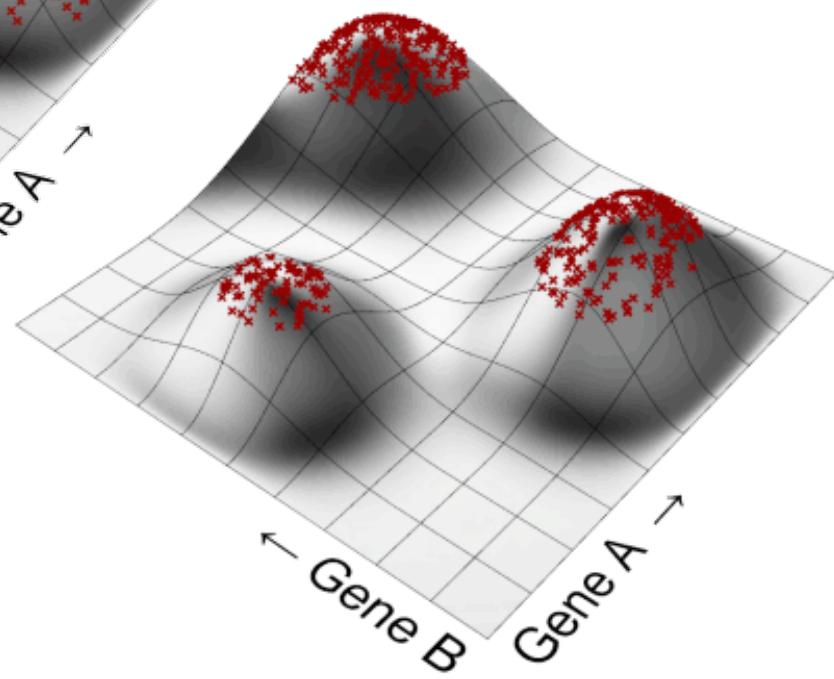
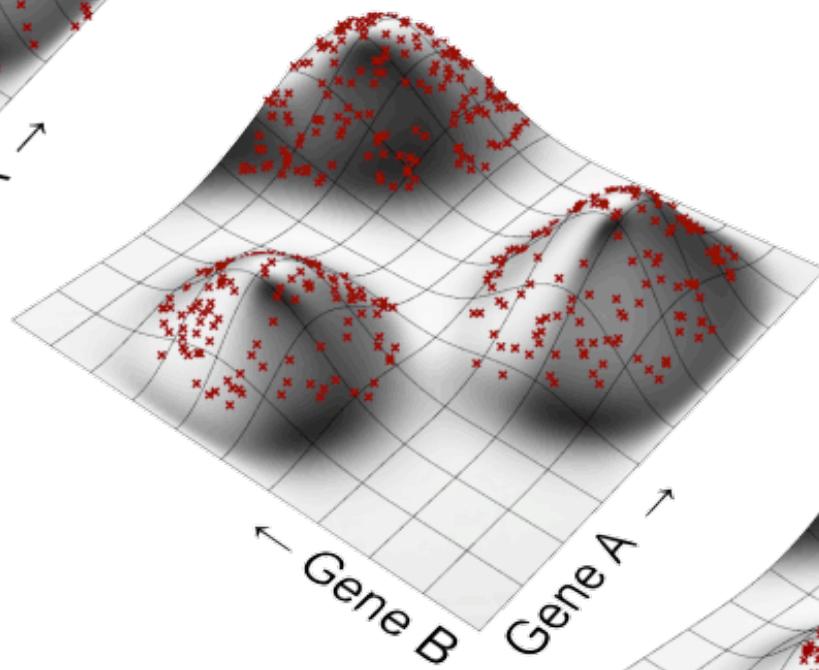
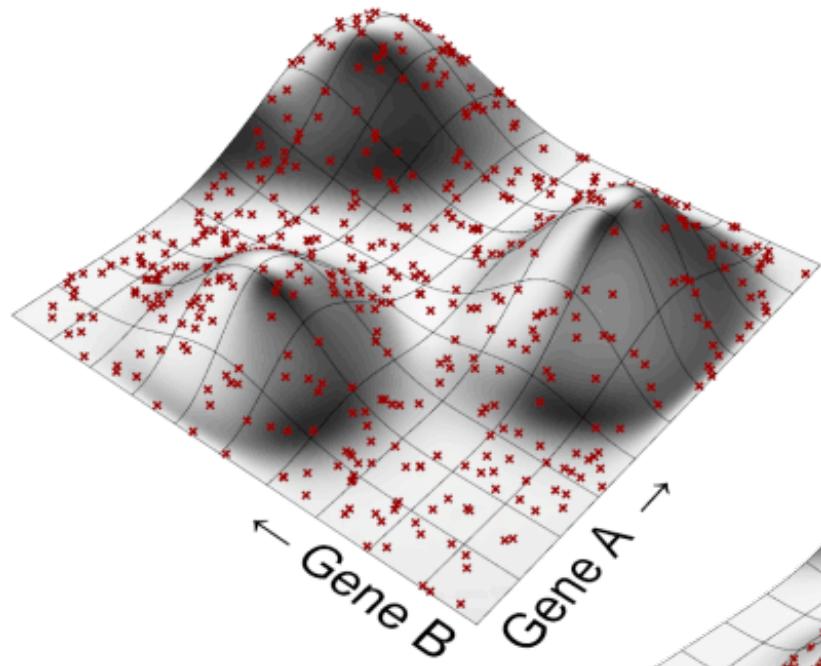
(7) Nova
População

(4) Pareamento

(6) Mutação

(5) Crossover





Experimento: the Knapsack problem

- Capacidade do Knapsak = 1000 (mil)
- 20 objetos diferentes a se carregar:

```
[ 183.42563994  89.16157914  29.83123647  214.17410172  262.24451376
 155.68178676  230.31329779  195.5483092   92.64810227  155.54096633
 123.30689605  58.42122925  11.30021587  25.19595481  270.55880368
 9.49158845   97.29361311  39.23812081  131.89304951  174.9339727 ]
```

Experimentos Knapsack ...: Exaustiva

Tempo de execução: 7.44638991355896

Maior valor encontrado: 999.999939891

Solução correspondente (itens selecionados):

```
[ 29.83123647  
 214.17410172  
 195.5483092  
 123.30689605  
 25.19595481  
 270.55880368  
 9.49158845  
 131.89304951]
```

Experimentos Mochila ...: Greedy

Tempo de execução: 0.00019121170043945312

Maior valor encontrado: = 988.59093282

Solução correspondente (itens selecionados):

```
[ 270.558803677  
 262.244513764  
 230.313297788  
 214.174101716  
 11.3002158747 ]
```

Experimentos Mochila ...: GA

Tempo de execução: 0.00047397613525390625

Maior valor encontrado: = 999.967995329

Solução correspondente (itens selecionados):

[0 996.190416609

...

2 997.468414987

3 999.578924798

4 999.59479577

...

12 999.692129618

...

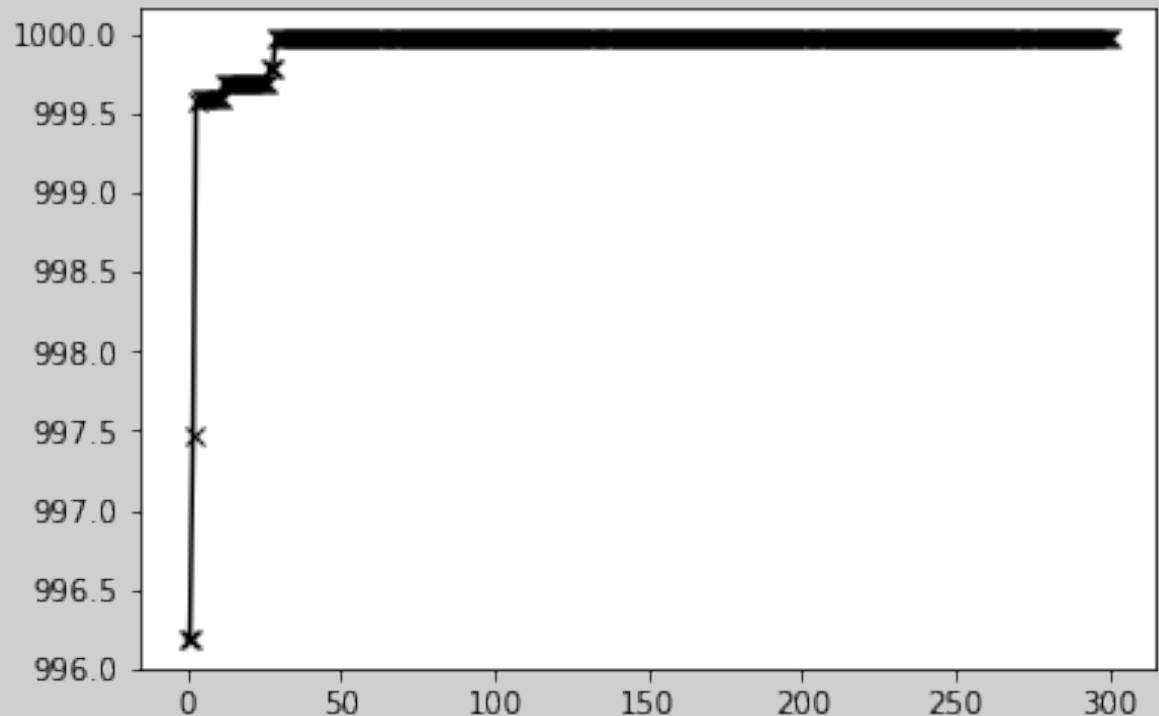
27 999.790038005

...

29 999.967995329

...

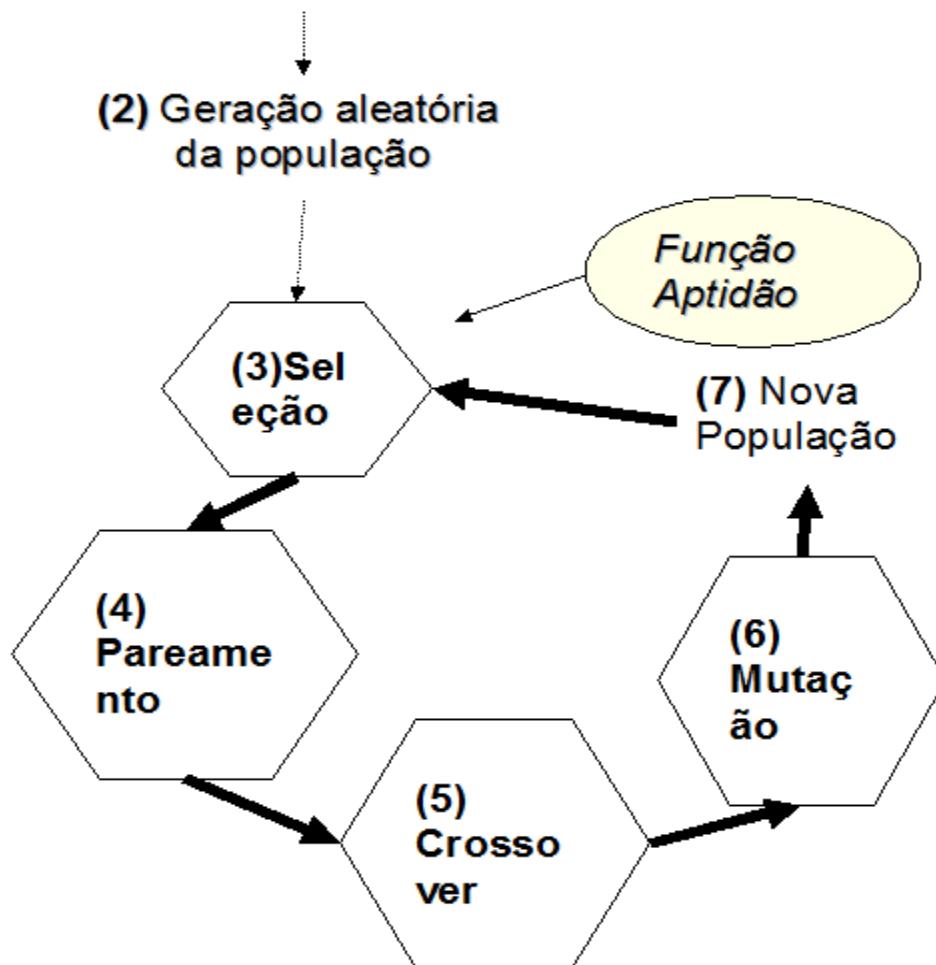
300 999.967995329]



Problema das Oito Rainhas

Simulação do Algoritmo Genético

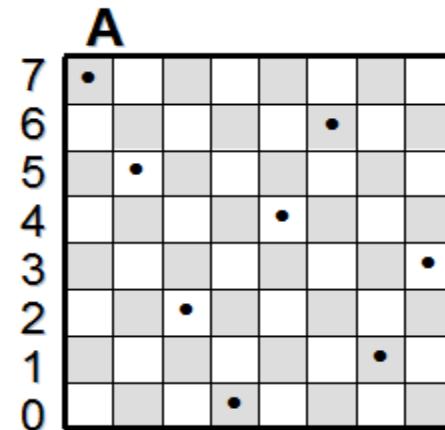
(1) Definição de um indivíduo



(1)

Definição de um indivíduo

O indivíduo (ou cromossomo) deve ser codificado em uma string de tamanho fixo e um conjunto finito de gene



Ex₁:

Gene={0,...,7}
Tam=8

A = 7 5 2 0
4 6 1 3

Ex₂:

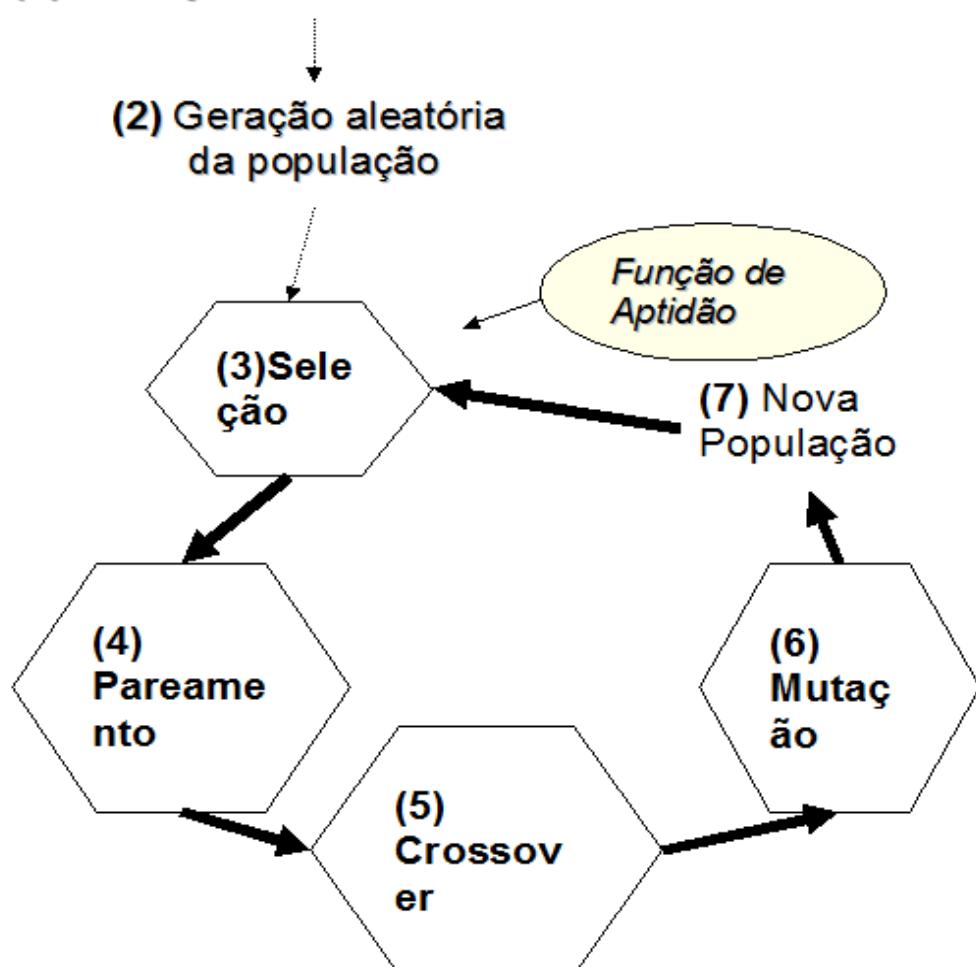
Gene={0,1}
Tam=8

A = 111 101 010 000
100 110 001 011

Problema das Oito Rainhas

Simulação do Algoritmo Genético

(1) Definição de um indivíduo



(2)

Geração da População

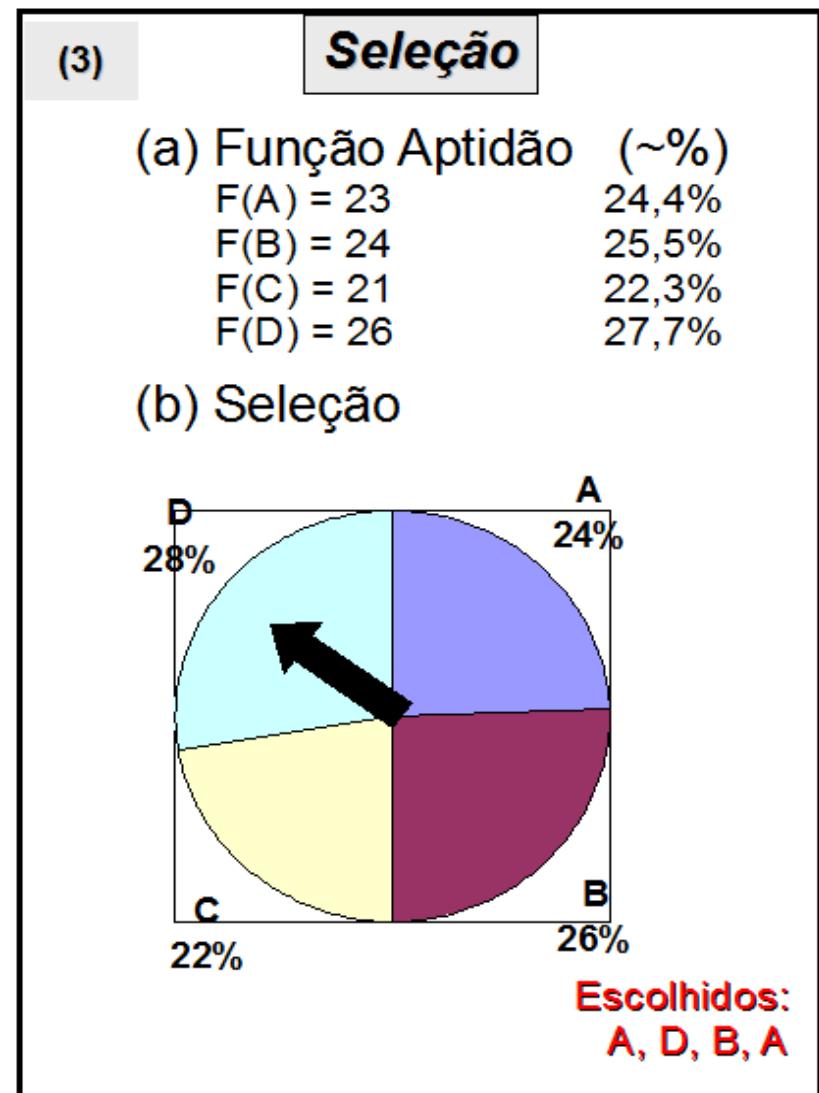
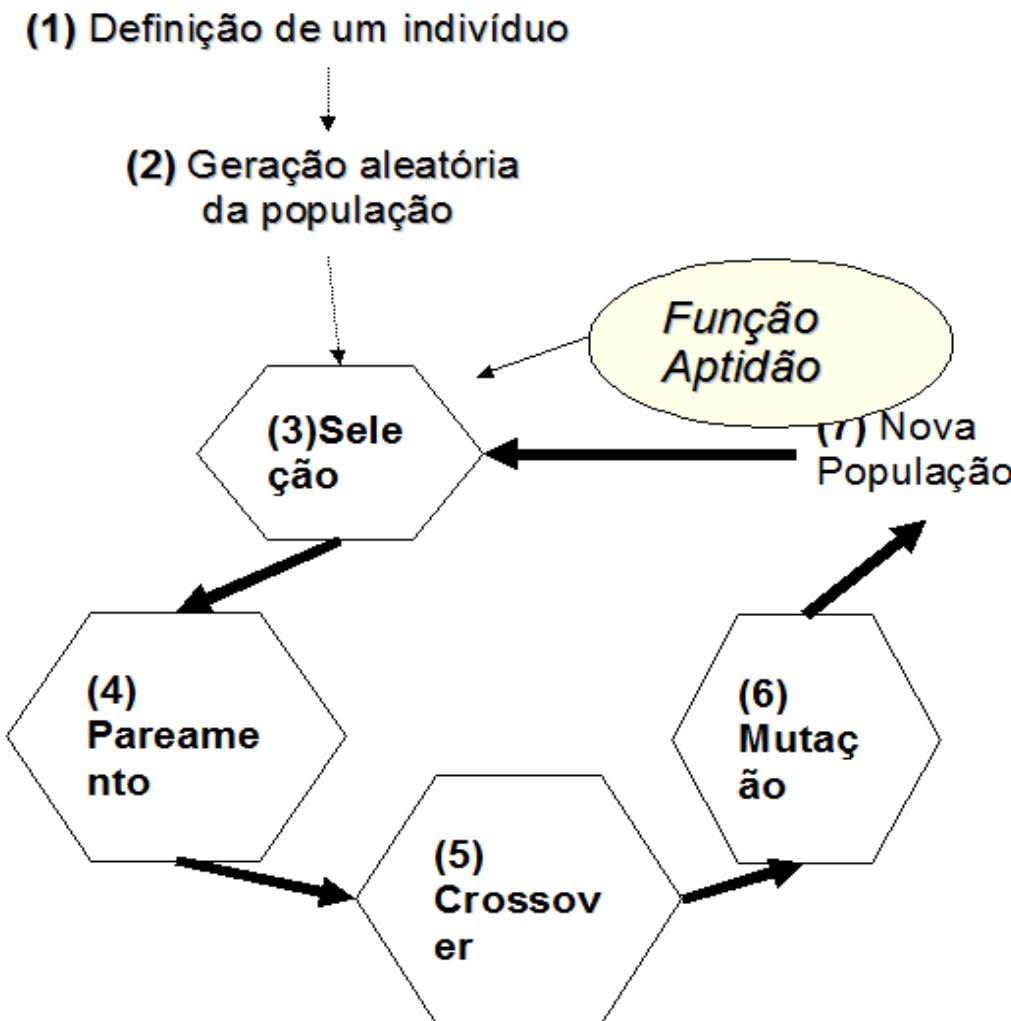
Devemos produzir um conjunto de indivíduos de forma aleatória

A =	7	2	0	5	3	1	5	0
B =	6	4	1	3	5	7	0	3
C =	6	4	0	6	1	7	0	5
D =	7	5	2	0	4	6	1	3

A =	111	010	000	101	011	001	101	000
B =	110	100	001	011	101	111	000	011
C =	110	100	000	110	001	111	000	101
D =	111	101	010	000	100	110	001	011

Problema das Oito Rainhas

Simulação do Algoritmo Genético



Problema das Oito Rainhas

Simulação do Algoritmo Genético

(1) Definição de um indivíduo

(2) Geração aleatória da população

(3) Seleção

Função Aptidão

(7) Nova População

(4)
Pareamento

(5)
Crossover

(6)
Mutação

(4)

Pareamento

Os indivíduos escolhidos são dispostos aleatoriamente dois a dois, para se reproduzirem

A =

111	010	000	101	011	001	101	000
-----	-----	-----	-----	-----	-----	-----	-----

B =

110	100	001	011	101	111	000	011
-----	-----	-----	-----	-----	-----	-----	-----

A =

111	010	000	101	011	001	101	000
-----	-----	-----	-----	-----	-----	-----	-----

D =

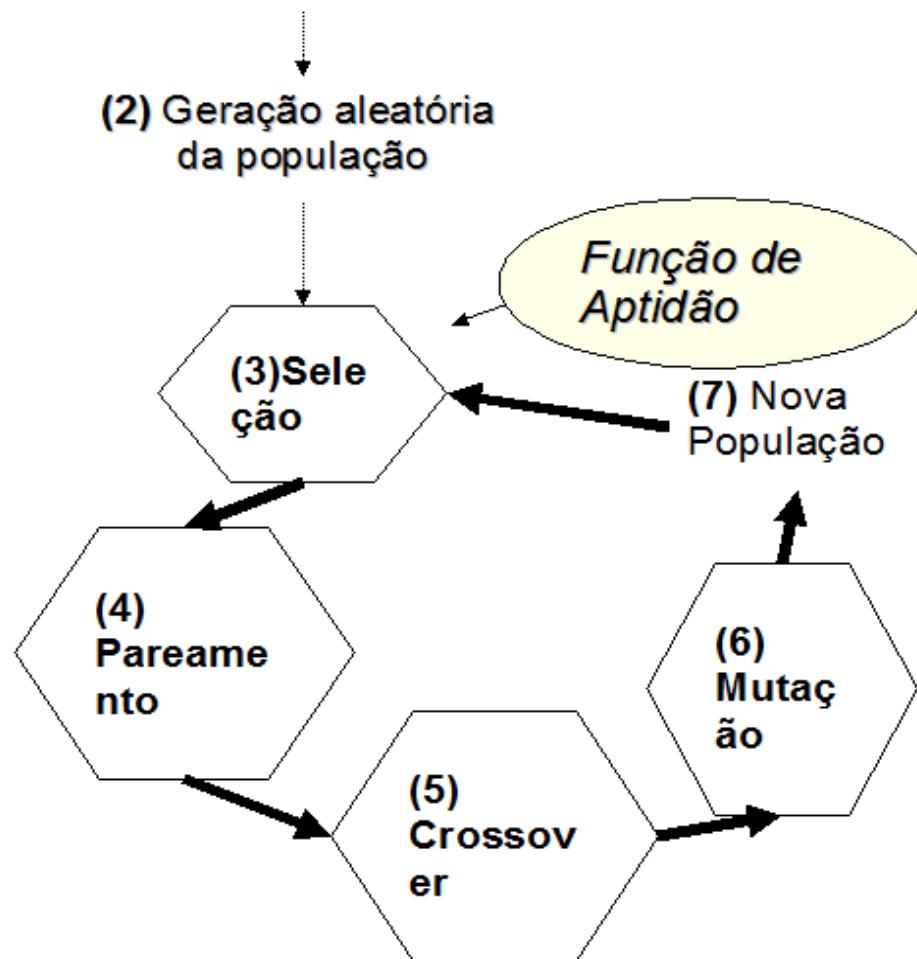
111	101	010	000	100	110	001	011
-----	-----	-----	-----	-----	-----	-----	-----

Observe que o indivíduo C não foi escolhido no processo de SELEÇÃO anterior

Problema das Oito Rainhas

Simulação do Algoritmo Genético

(1) Definição de um indivíduo



(5)

Crossover

Processo pelo qual os filhos gerados possuem partes do cromossomo dos pais

$$A = \boxed{111 \quad 010 \quad 000 \quad 101 \quad 011 \quad 001 \quad 101 \quad 000}$$

$$B = \boxed{110 \quad 100 \quad 001 \quad 011 \quad 101 \quad 111 \quad 000 \quad 011}$$

$$AB_1 = \boxed{111 \quad 010 \quad 000 \quad 101 \quad \textcolor{yellow}{101} \quad 111 \quad 000 \quad 011}$$

$$AB_2 = \boxed{110 \quad 100 \quad 001 \quad \textcolor{yellow}{011} \quad 011 \quad 001 \quad 101 \quad 000}$$

$$A = \boxed{111 \quad 010 \quad 000 \quad 101 \quad 011 \quad 001 \quad 101 \quad 000}$$

$$D = \boxed{\textcolor{yellow}{111} \quad 101 \quad 010 \quad 000 \quad 100 \quad 110 \quad 001 \quad 011}$$

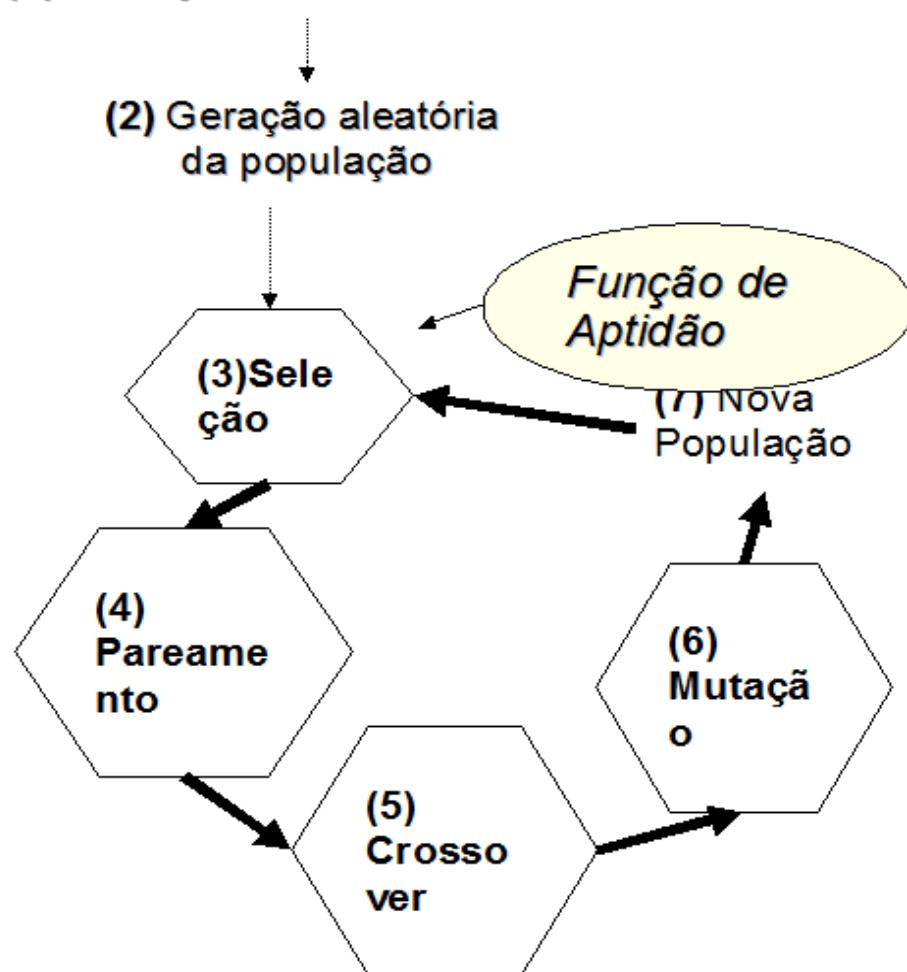
$$AD_1 = \boxed{111 \quad 010 \quad 000 \quad 101 \quad \textcolor{yellow}{100} \quad 110 \quad 001 \quad 011}$$

$$AD_2 = \boxed{111 \quad 101 \quad 010 \quad 000 \quad 011 \quad 001 \quad 101 \quad 000}$$

Problema das Oito Rainhas

Simulação do Algoritmo Genético

(1) Definição de um indivíduo



(6)

Mutação

Processo pelo qual os bits dos filhos (em pequena probabilidade) são invertidos

$$AB_1 = \boxed{111 \ 010 \ 000 \ 101 \ 101 \ 111 \ 000 \ 011}$$

$$AB_2 = \boxed{110 \ 100 \ 001 \ 011 \ 011 \ 001 \ 101 \ 000}$$

$$AB_2 = \boxed{110 \ 100 \ 001 \ 011 \ 111 \ 001 \ 101 \ 000}$$

$$AD_1 = \boxed{111 \ 010 \ 000 \ 101 \ 100 \ 110 \ 001 \ 011}$$

$$AD_2 = \boxed{111 \ 101 \ 010 \ 000 \ 011 \ 001 \ 101 \ 000}$$

Problema das Oito Rainhas

Simulação do Algoritmo Genético

(1) Definição de um indivíduo

(2) Geração aleatória da população

Função Aptidão

(3) Seleção

(7) Nova População

(4) Pareamento

(6) Mutação

(5) Crossover

(7)

Nova População

Uma nova população é gerada

$AB_1 =$

111	010	000	101	101	111	000	011
-----	-----	-----	-----	-----	-----	-----	-----

$AB_2 =$

110	100	001	011	111	001	101	000
-----	-----	-----	-----	-----	-----	-----	-----

$AD_1 =$

111	010	000	101	100	110	001	011
-----	-----	-----	-----	-----	-----	-----	-----

$AD_2 =$

111	101	010	000	011	001	101	000
-----	-----	-----	-----	-----	-----	-----	-----

Verifica-se se o indivíduo de maior adaptabilidade possível se encontra, ou seja, que possua a Função Aptidão máxima

Caso contrário, continua-se executando o ciclo até um determinado número de voltas



Hendrik Macedo

Escreve sobre *Inteligência Artificial* no Saense.

<http://www.saense.com.br/autores/artigos-publicados-por-hendrik-macedo/>