

Inteligência Artificial

Outras estratégias de busca e Computação Evolutiva

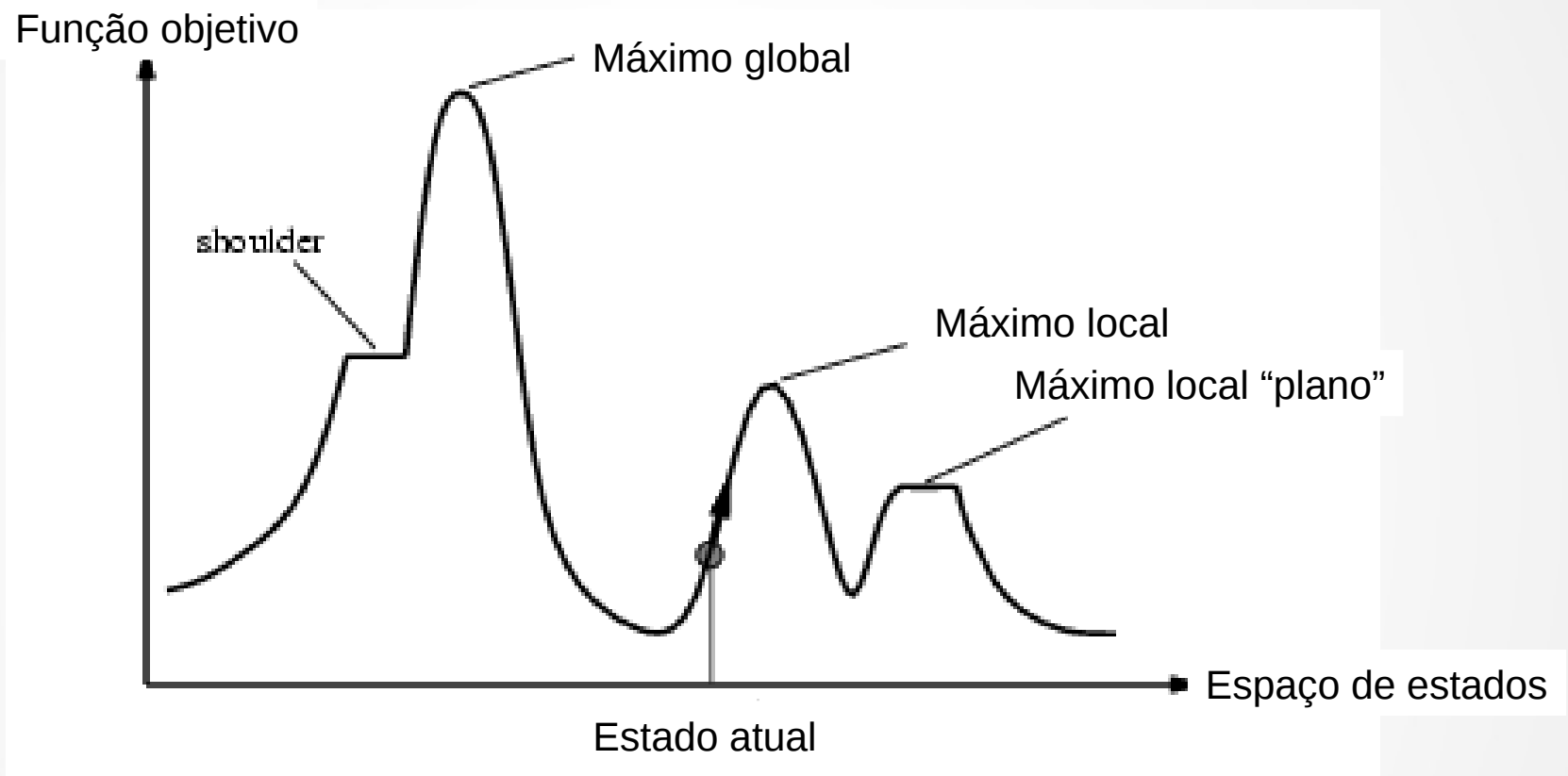
Prof. Fabio Augusto Faria

Material adaptado de Profa. Ana Carolina Lorena e livro
“Inteligência Artificial, S. Russell e P. Norving”

1º semestre 2017



Espaço de busca



Algoritmo **completo** sempre encontra uma solução, caso ela exista

Algoritmo **ótimo** sempre encontra mínimo/máximo global

Busca subida em encosta

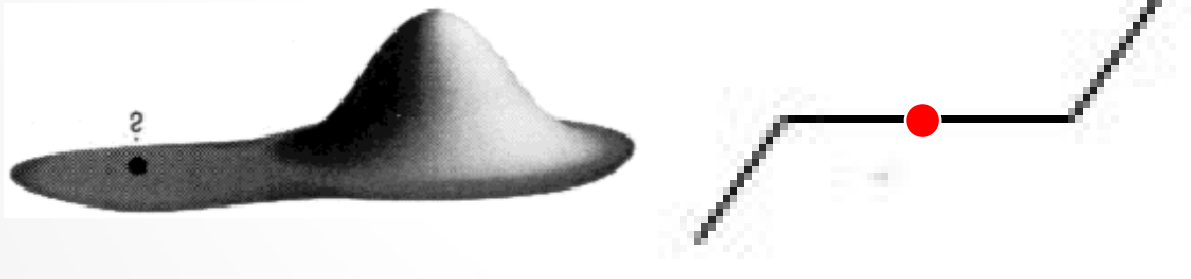
- **Máximos Locais:**

- em contraste com *máximos globais*, são picos mais baixos do que o pico mais alto no espaço de estados (solução ótima)
- a função de avaliação leva a um valor máximo para o caminho sendo percorrido: essa função utiliza informação local
- porém, o nó final está em outro ponto mais alto
- isto é uma consequência das decisões irrevogáveis do método
 - e.g., xadrez: eliminar a Rainha do adversário pode levar o jogador a perder



Busca subida em encosta

- **Platôs:**
 - uma região do espaço de estados onde a função de avaliação dá o mesmo resultado.

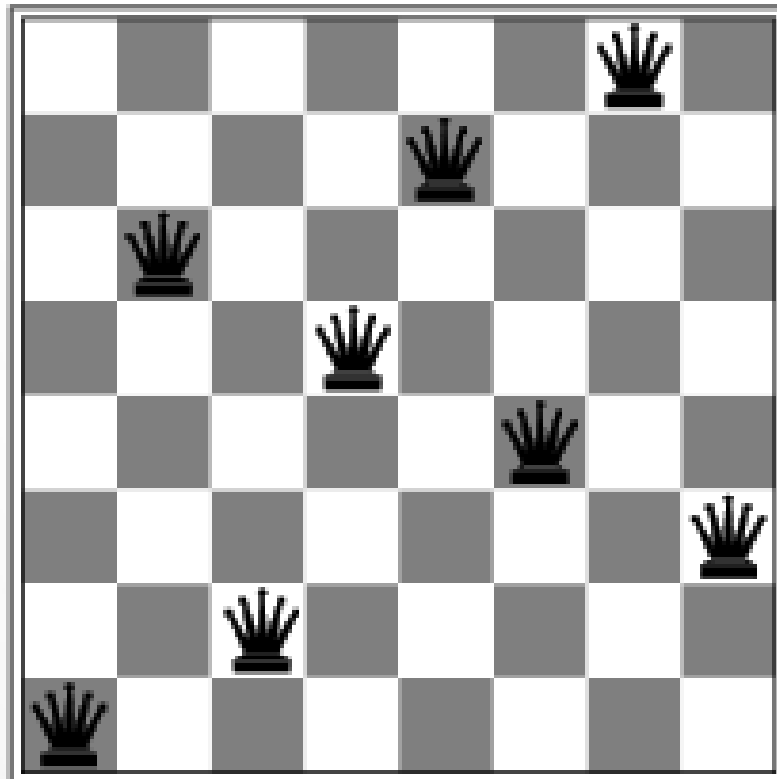


Busca subida em encosta

Mínimo local com $h = 1$

Todo sucessor tem custo mais alto

Alcançada em 5 passos a partir do estado inicial



Busca subida em encosta

Problema das 8 rainhas

Estados iniciais aleatórios

- 86% das vezes busca fica paralisada
 - Resolve apenas 14% das instâncias do problema
- Mas é rápida
 - 4 passos em média quando tem sucesso
 - 3 passos em média quando fica paralisada
 - Em espaço que tem cerca de 17 milhões de estados

Busca subida em encosta

Mudar de estado em platôs

- Quando estados têm avaliações iguais
- Colocando um limite de vezes

Ex. **Problema das 8 rainhas**

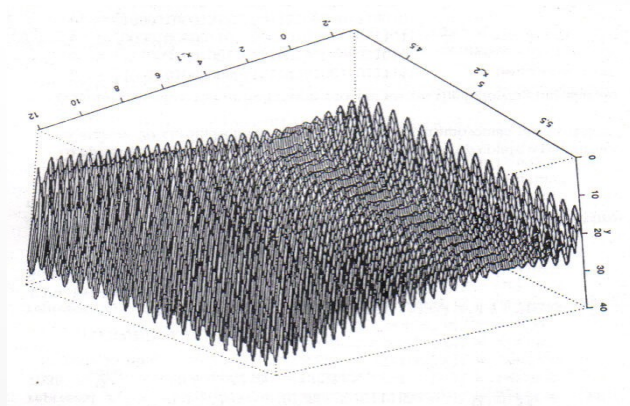
- Com estados iniciais aleatórios, usando essa estratégia
 - Passa a resolver 94% das instâncias do problema
 - Mas demora mais
 - » 21 passos em média quando tem sucesso
 - » 64 passos em média quando falha

Busca de subida em encosta

Sucesso depende da topologia do espaço de estados

Se houver poucos máximos locais e platôs

- Subida de encosta com reinício aleatório encontrará boa solução com rapidez
- Mesmo para mais complexos, pode encontrar máximo local razoavelmente bom
 - Com poucos reinícios



Têmpera Simulada

- Têmpera: esquentar o metal/vidro e depois esfriá-lo gradualmente
- Mudar a idéia de subida em encosta para descida de gradiente
- Uma bola rolando uma superfície acidentada e esta parará num **mínimo local**
- Para evitar, deve-se **agitar** a superfície para que a bola **saia** desse mínimo local

Têmpera Simulada

- A idéia desta estratégia é agitar com força no início e diminuir gradualmente a intensidade da agitação
agitação == aleatoriedade
- Diferente da **subida em encosta** é que esta estratégia não escolhe o melhor estado sucessor, mas escolhe o melhor movimento **aleatório** (“agito”)
- Caso esse movimento aleatório não seja melhor, aceita-se um movimento com alguma probabilidade < 1 (vide Fig. 4.5)
- Exemplos: layout VLSI e escalonamentos.

Busca em Feixe Local

- Manter apenas 1 estado na memória é uma abordagem extrema para resolver problema de limitação de memória
- Esta estratégia mantém o controle de K estados, por vez
 - 1- Inicia com K estados gerados aleatoriamente
 - 2- Expande os sucessor de cada um dos K
 - 3- Se objetivo, sucesso
 - 4- Se não, seleciona os K melhores dentre todos os estados e volta para 1

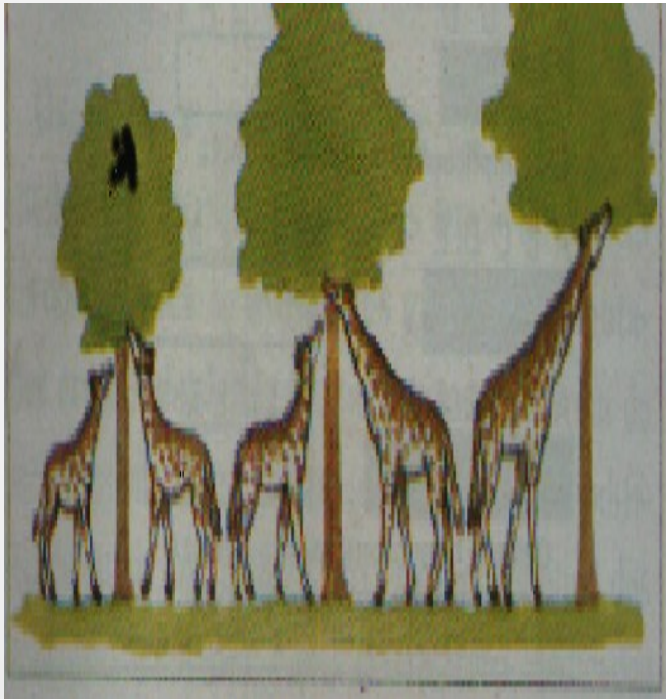
Busca em Feixe Local

- Esses K reinícios aleatórios paralelos, os estados **se conversam** e tendem a abandonar caminhos infrutíferos
- Deslocando os recursos para o processo de **maior progresso**
- Problema está na **falta de diversidade** entre os K estados, se concentrando em uma pequena região do espaço de estados (Busca de subida de encosta)
- Solução é **Feixe Local Estocástico** que escolhe os K estados sucessores de forma **aleatória** considerando uma **probabilidade** maior que a atual (**seleção natural**)

Algoritmos Genéticos

Engloba métodos e técnicas computacionais inspirados:

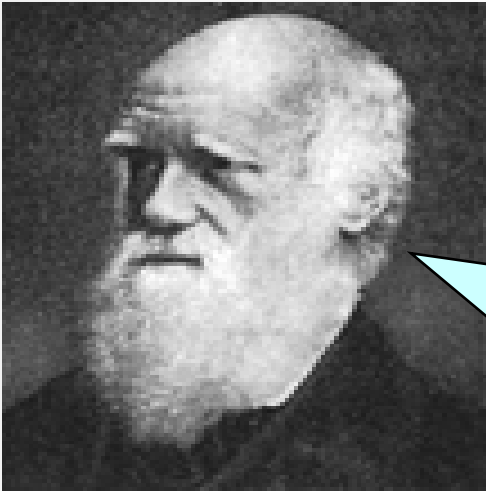
- na teoria da evolução das espécies, de seleção natural (Darwin)
 - na Genética iniciada por Mendel
-



Bases da evolução:

- diversidade é gerada por cruzamento e mutações
- os seres mais adaptados ao seus ambientes sobrevivem
- as características genéticas de tais seres são herdadas pelas próximas gerações

Algoritmos Genéticos



1859 - Charles Darwin publica o livro “A Origem das Espécies”

As espécies evoluem pelo princípio da seleção natural e sobrevivência do mais apto

Charles Darwin



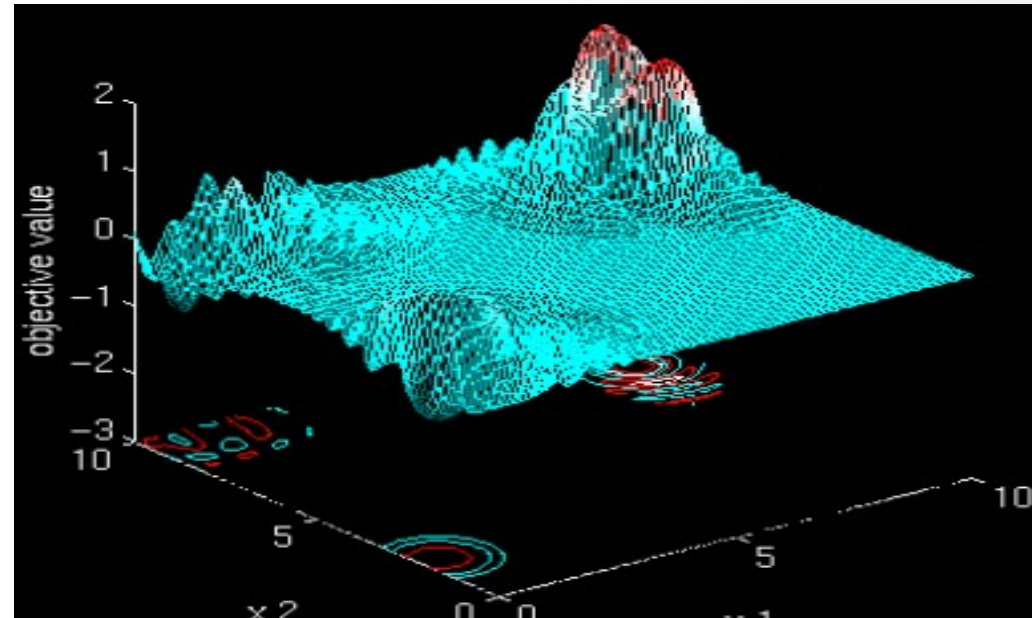
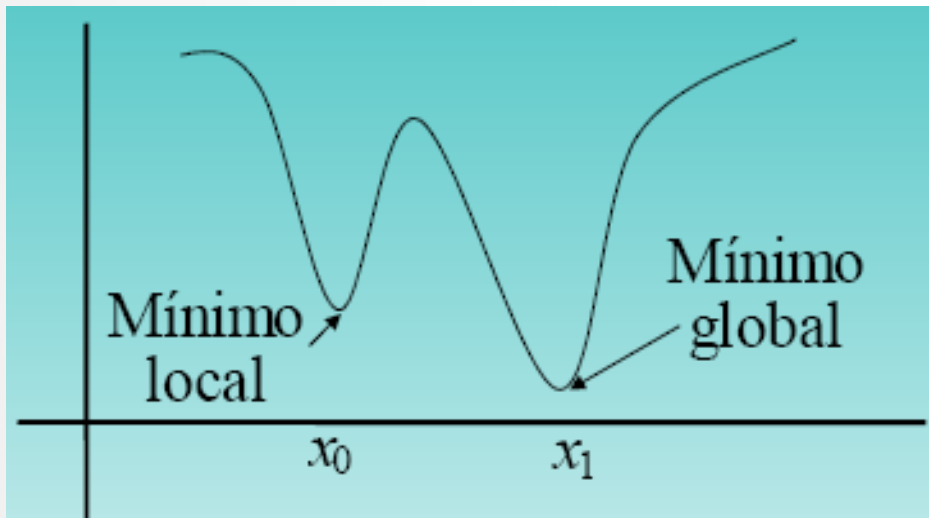
1865 - Gregor Mendel apresenta experimentos do cruzamento genético de ervilhas

Pai da Genética

Gregor Mendel

Algoritmos Genéticos

Nos anos 1960 John Holland e seus alunos propuseram a construção de um algoritmo de busca e otimização: os algoritmos genéticos

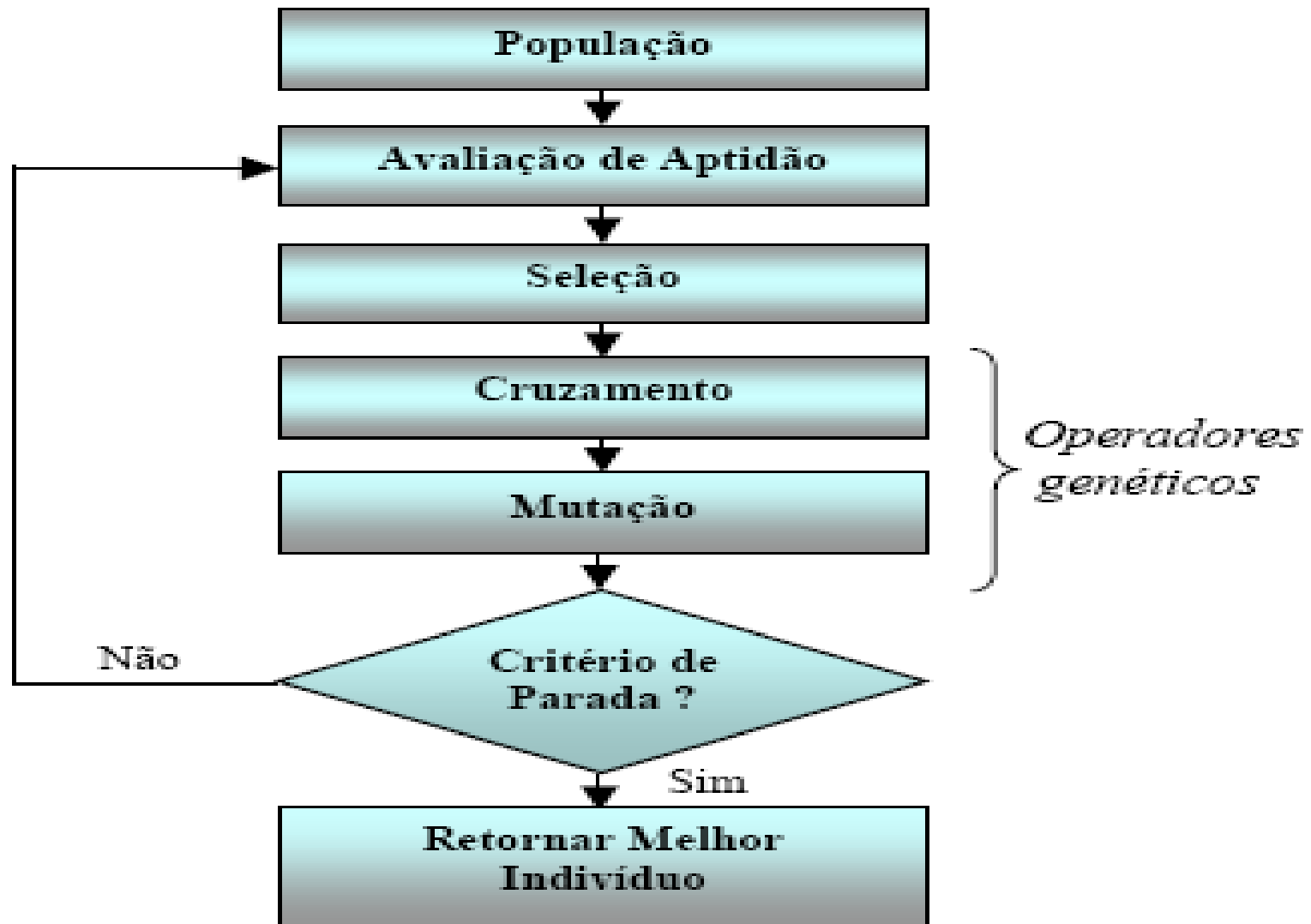


Algoritmos Genéticos

Os algoritmos genéticos usam como base, e procuram combinar:

- A teoria da *evolução das espécies* - a sobrevivência das estruturas/soluções mais adaptadas a um ambiente/problema
- Estruturas *genéticas* - utiliza a conceitos de hereditariedade e variabilidade genética para troca de informações entre as estruturas, visando a melhoria das mesmas

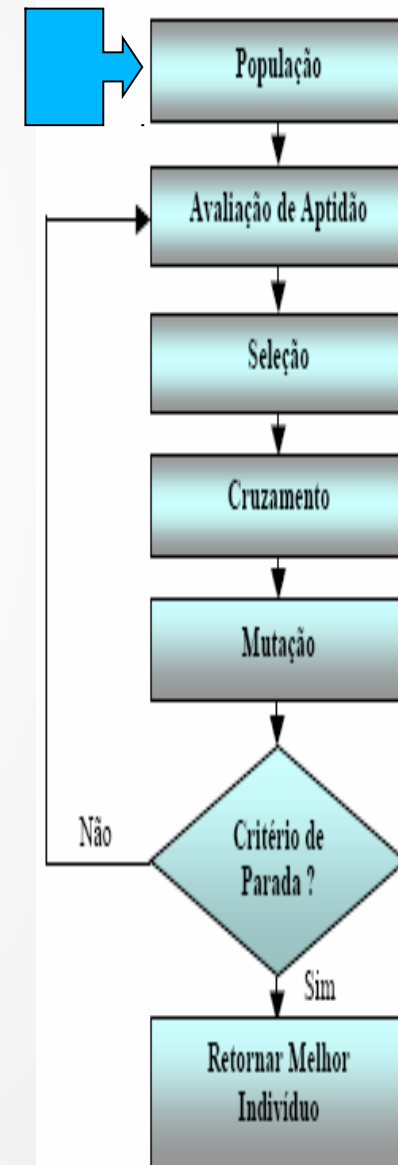
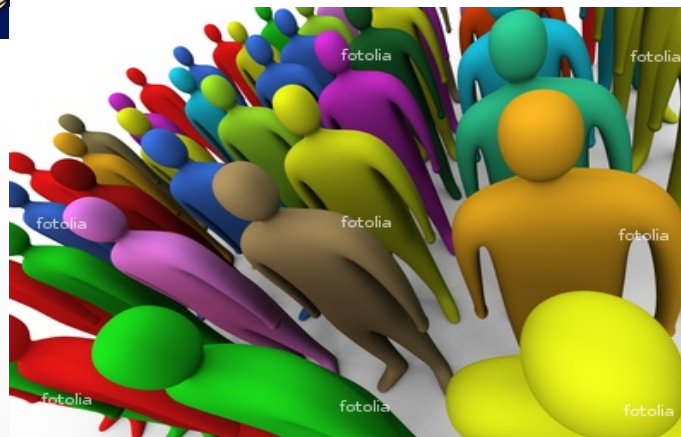
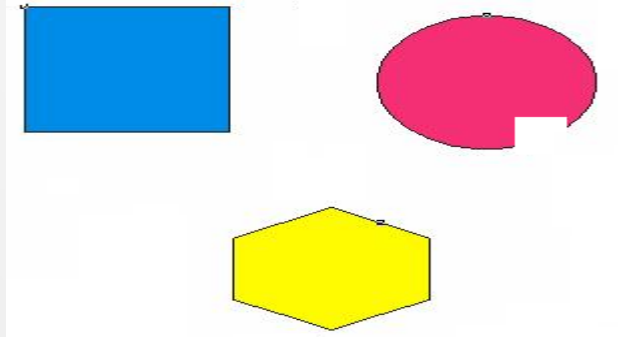
Algoritmos Genéticos



População

A população de um algoritmo genético é o conjunto de indivíduos que estão sendo cogitados como solução

Cada indivíduo é uma possível solução do problema



Indivíduo

Um indivíduo no AG é um **cromossomo**

Ou seja, um indivíduo é um conjunto de atributos da solução

Geralmente é uma cadeia de bits que representa uma solução possível para o problema

Outras representações são possíveis

Boa representação depende do problema

Exemplo: população de tamanho N=5

Geração de indivíduos, com seus cromossomos

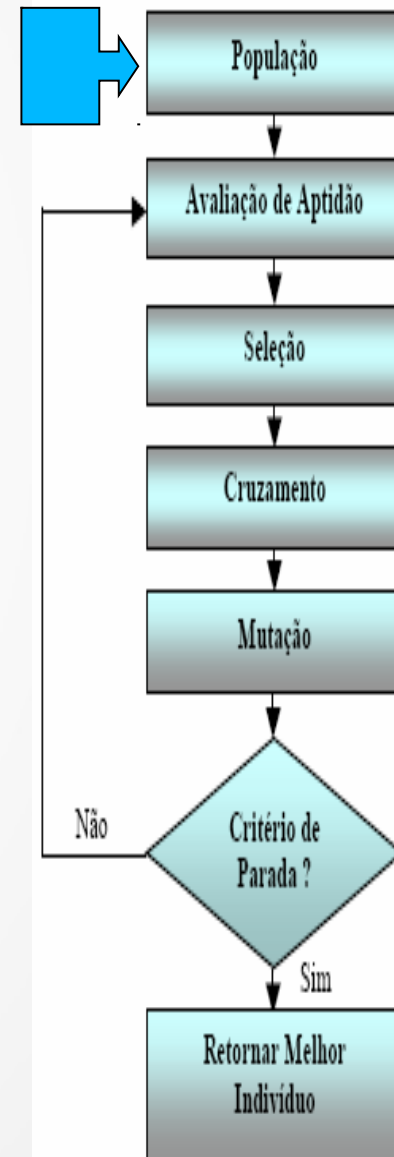
Cada elemento do vetor é um gene, um atributo da solução

Indivíduo 1 = [1 1 1 0 1]

Indivíduo 2 = [0 1 1 0 1]

Indivíduo 3 = [0 0 1 1 0]

Indivíduo 4 = [1 0 0 1 1]



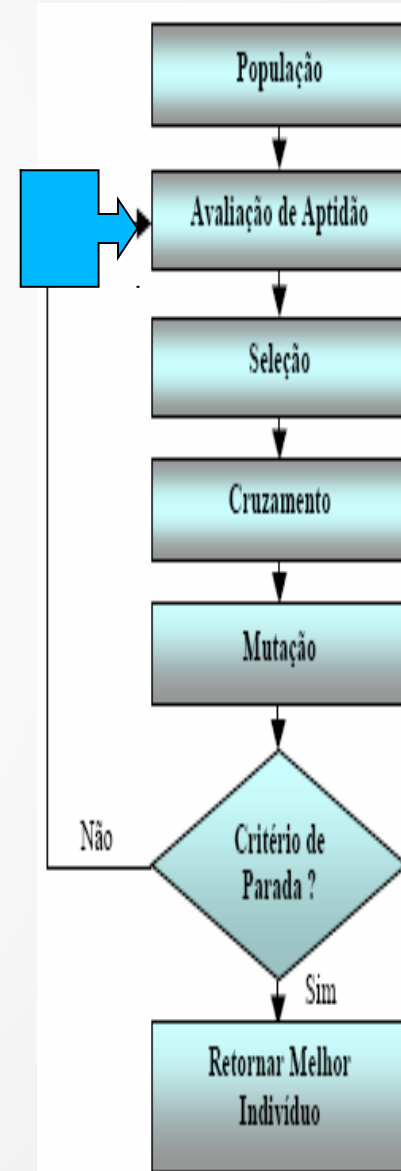
Função de aptidão

A função de avaliação, função de *fitness*, determina uma nota a cada indivíduo

Esta nota avalia quão boa é a solução que este indivíduo representa

Por exemplo, o objetivo de um AG pode ser maximizar o número de 1s

Indivíduos	Função de aptidão (<i>fitness</i>)
[11101]	4
[01101]	3
[00110]	2
[10011]	3
Aptidão média	3



Seleção

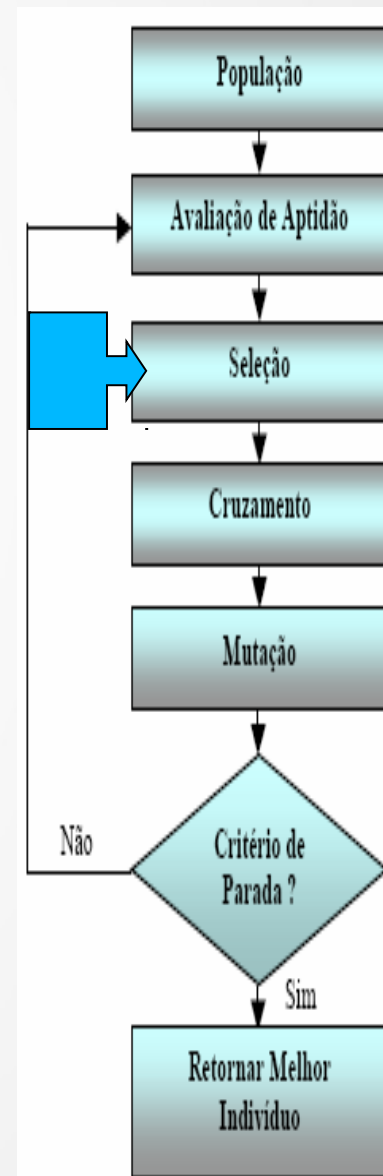
De acordo com a teoria de Darwin, o melhor sobrevivente para criar a descendência é selecionado

Há muitos métodos para selecionar o melhor cromossomo

Dentre eles:

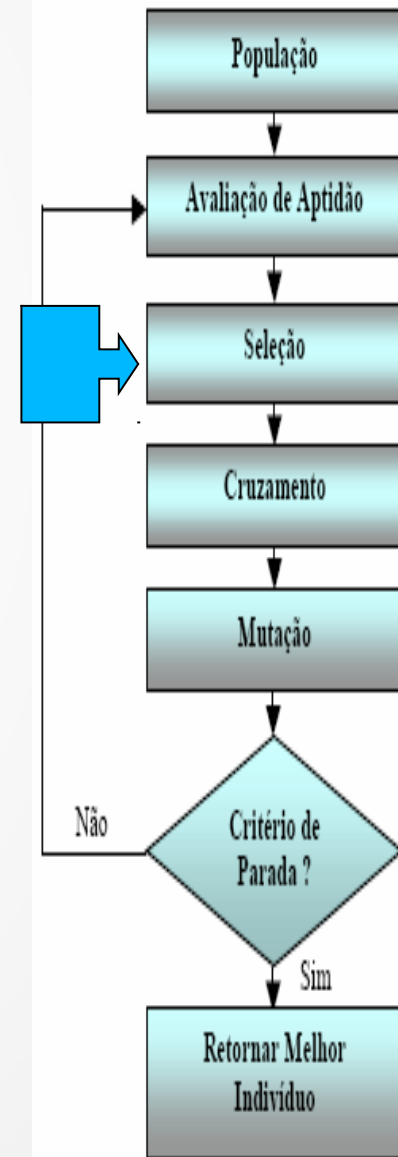
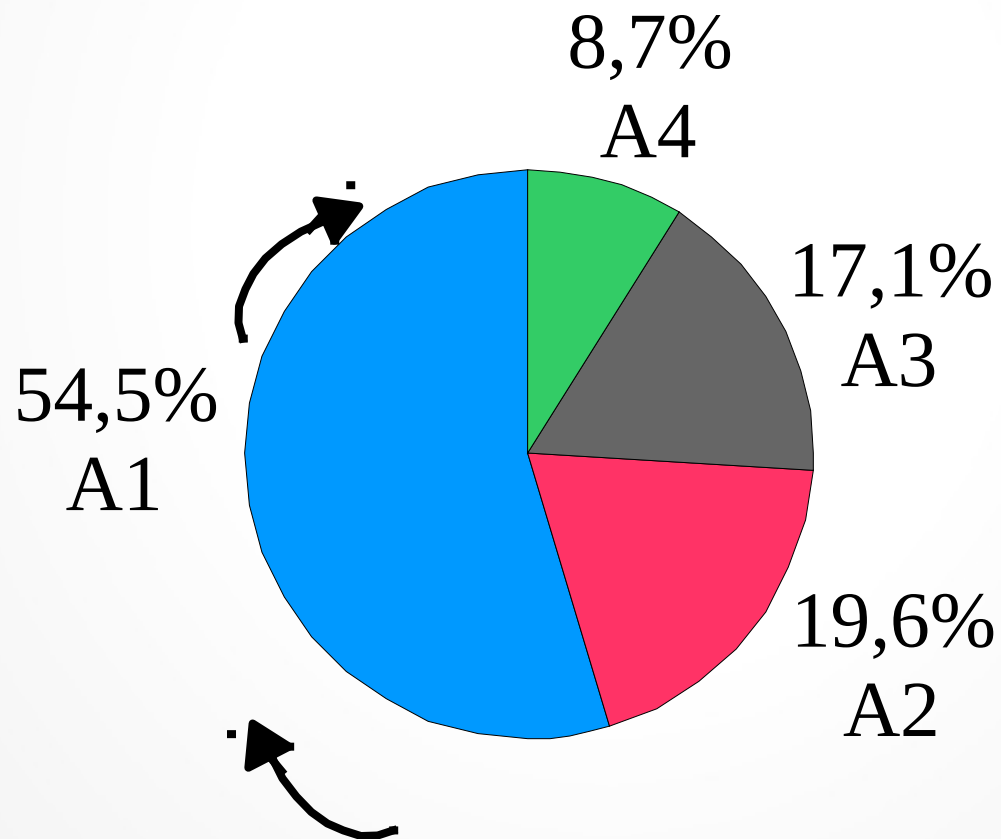
- Seleção por roleta
- Seleção por torneio

A seleção dirige o AG para as melhores regiões do espaço de busca



Seleção por roleta

Para visualizar este método considere um círculo dividido em N regiões (tamanho da população), onde a área de cada região é proporcional à aptidão do indivíduo

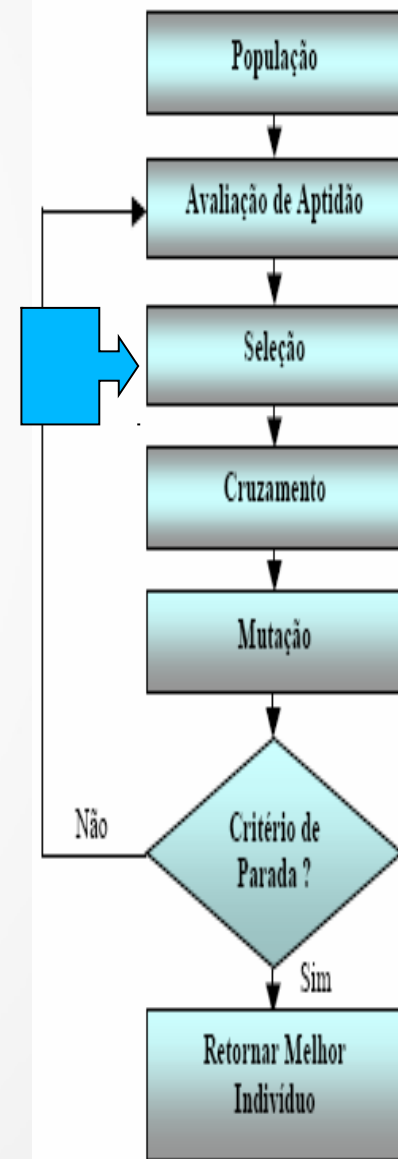
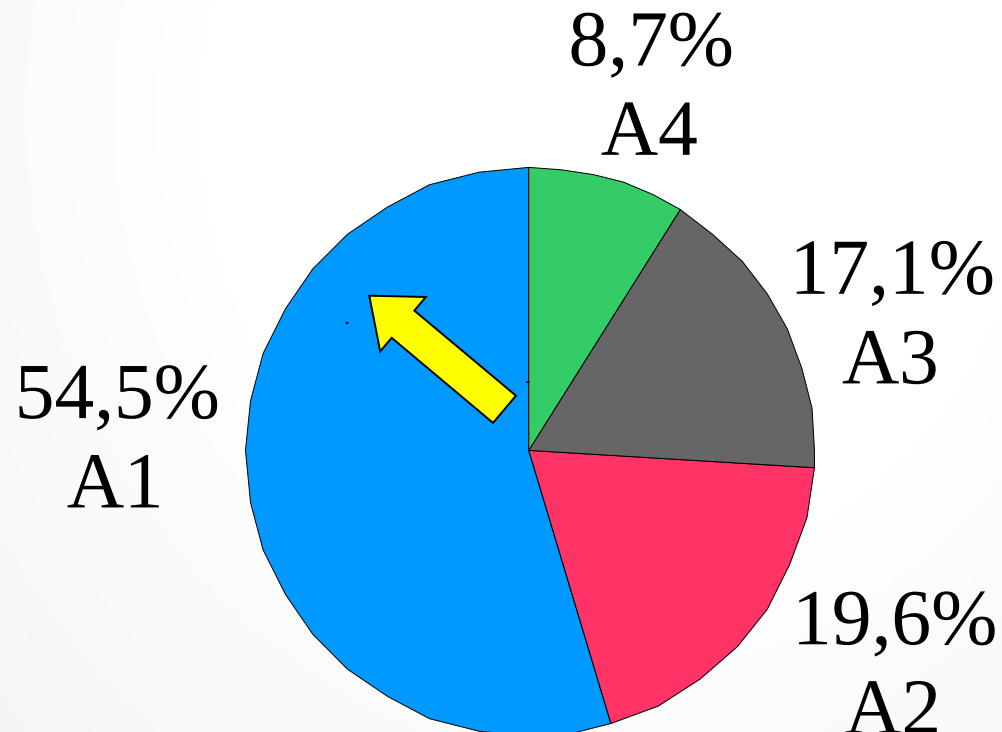


Seleção por roleta

Coloca-se sobre este círculo uma “roleta”

A roleta é girada um determinado número de vezes, dependendo do tamanho da população

São escolhidos como indivíduos que participarão da próxima geração, aqueles sorteados na roleta



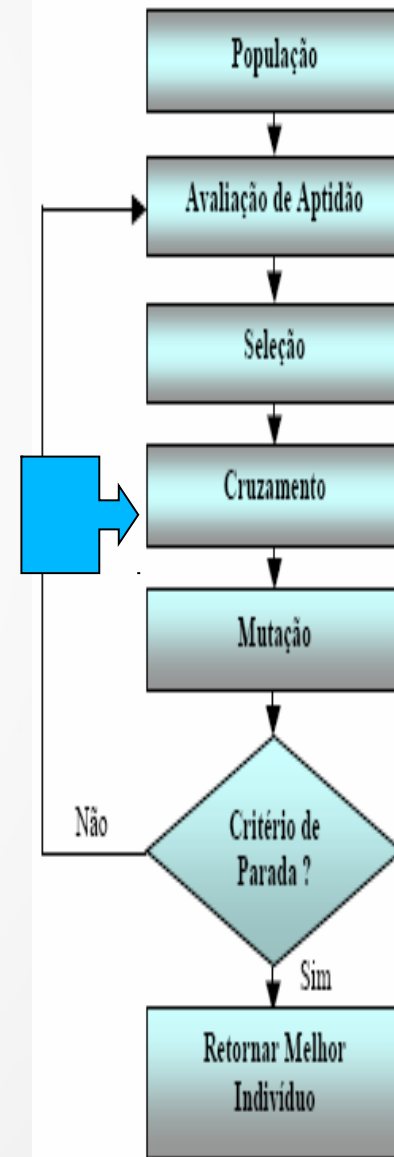
Operadores genéticos

Um conjunto de operações é necessário para que, dada uma população, se consiga gerar populações sucessivas que (espera-se) melhorem sua aptidão com o tempo

Estas operações são os **operadores genéticos**.
São eles:

- **Cruzamento**
- **Mutação**

Os operadores genéticos permitem explorar áreas desconhecidas do espaço de busca



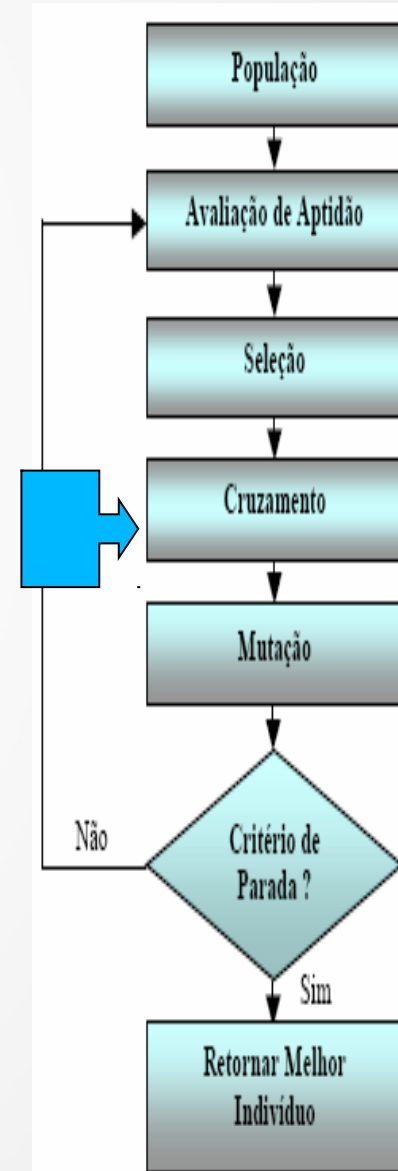
Cruzamento

O operador *crossover* (**cruzamento**) cria novos indivíduos, misturando características de **dois indivíduos pais**

O resultado desta operação é **um indivíduo** que potencialmente combine as melhores características dos indivíduos usados como base

Alguns tipos de cruzamento são:

- **Cruzamento em um ponto**
- **Cruzamento em dois pontos**



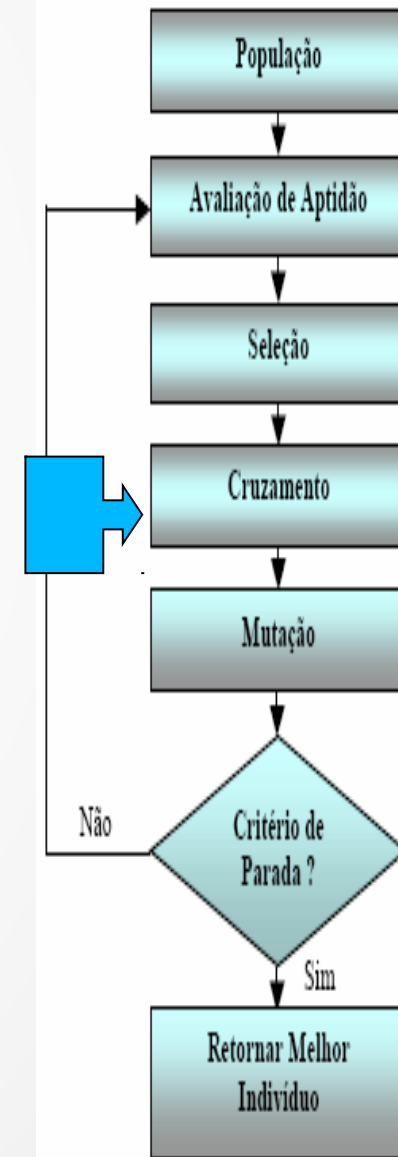
Cruzamento de um ponto

No cruzamento de um ponto divide-se cada progenitor em duas partes, em uma localidade k (escolhida aleatoriamente)

Indivíduo 1



Indivíduo 2



Cruzamento de um ponto

O descendente 1 consiste em genes 1 a $k-1$ do progenitor 1, e genes k a n do progenitor 2

O descendente 2 é “reverso”

Indivíduo 1



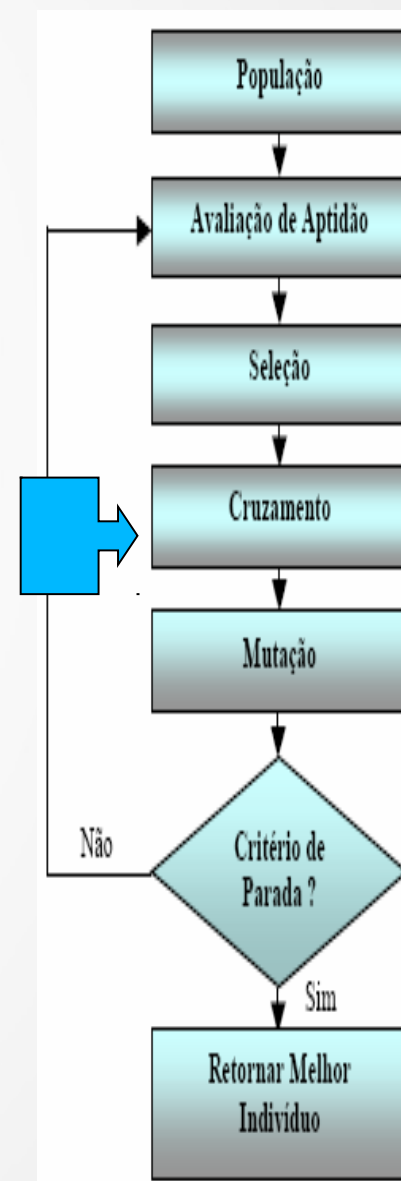
Indivíduo 2



Descendente 1

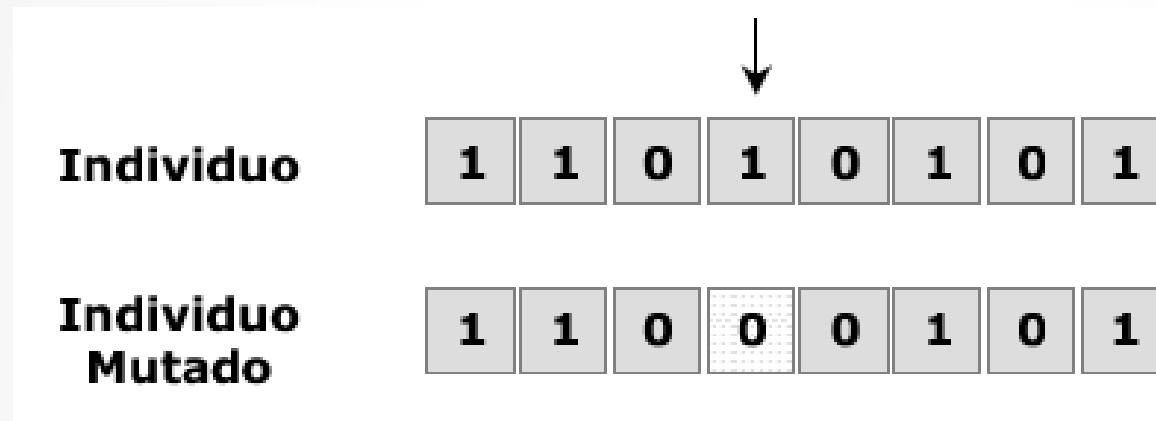


Descendente 2



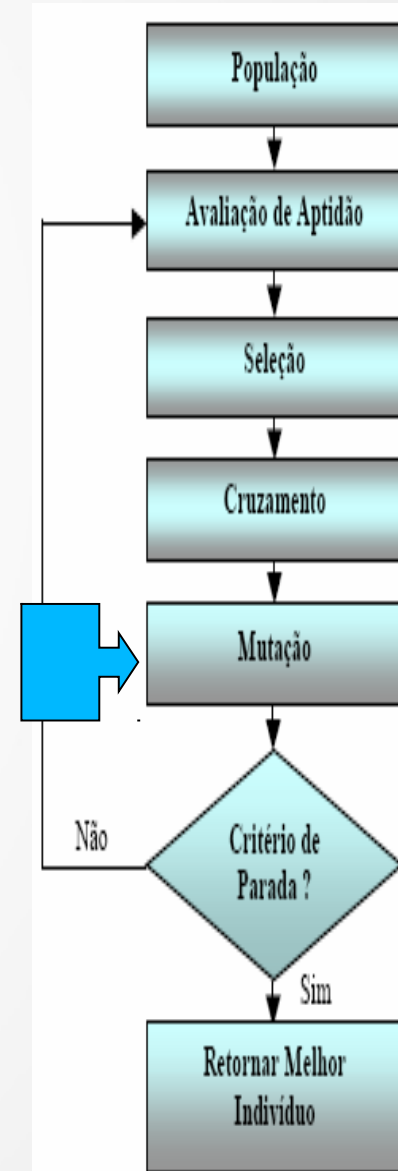
Mutação

A **mutação** modifica aleatoriamente alguma característica do indivíduo, sobre o qual é aplicada



O operador de mutação é necessário para a introdução e manutenção da diversidade genética da população

Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca, possivelmente, não será zero



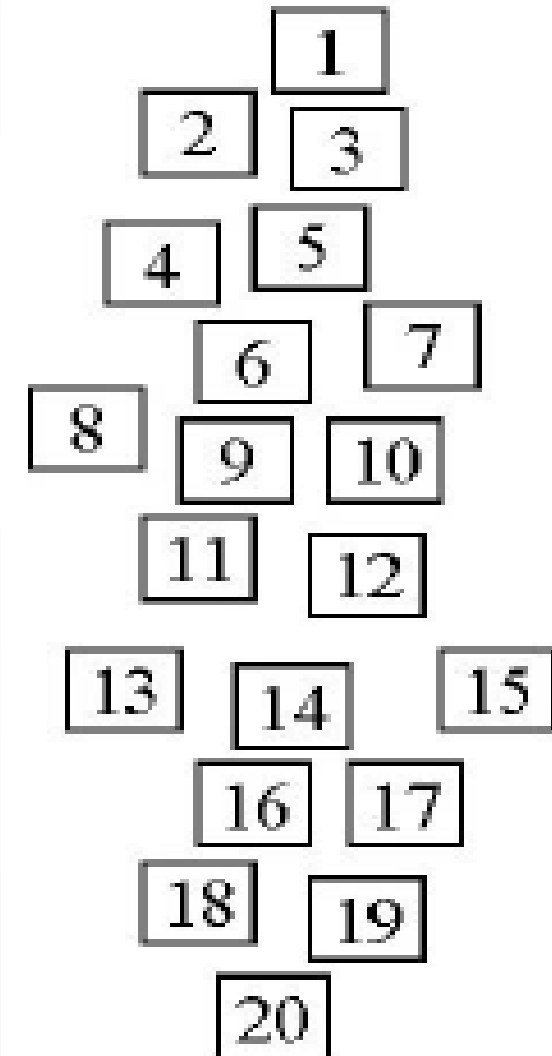
Gerações

Algoritmo é iterado até algum critério de parada

A cada passo, um novo conjunto de indivíduos é gerado a partir da população anterior

A este novo conjunto dá-se o nome de **geração**

Com a criação de uma grande quantidade de gerações que é possível obter resultados dos AGs



Algoritmo

Algoritmo_genético

- p = tamanho da população
- r = taxa de cruzamento
- m = taxa de mutação

Codificação e avaliação de aptidão são pontos chave

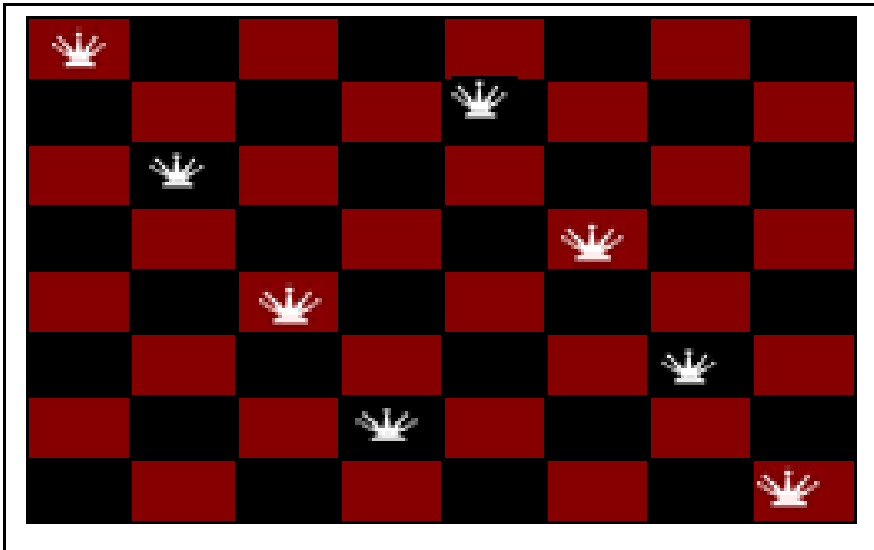
1. $P \leftarrow$ gerar aleatoriamente p indivíduos
2. **Para cada** i em P , computar $Aptidão(i)$
3. **Enquanto** critério_parada não é atingido
 - 3.1 Selecionar p membros de P para reprodução
 - 3.2 Aplicar cruzamento a pares de indivíduos selecionados segundo taxa r , adicionando filhos em PS
 - 3.3 Realizar mutação em membros PS , segundo taxa m
 - 3.5 $P \leftarrow PS$
 - 3.6 **Para cada** i em P , computar $Aptidão(i)$
4. **Retornar** o indivíduo de P com maior aptidão

Exemplo

Codificando o problema

Ex.: problema 8 rainhas

- Cada estado deve especificar posição de 8 rainhas, em coluna com 8 quadrados
 - $8 \times \log_2 8 = 24$ bits se codificação binária



Binário = 111 101 011 001 110 100 010 000
Inteiro = 8 6 4 2 7 5 3 1

Exemplo

(a) Gerando população inicial

8 dígitos, com valores de 1 a 8

- Exemplo: população com 4 cadeias de 8 dígitos
 - Representam estados de 8 rainhas

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 5 4 3 2 1 3

Exemplo

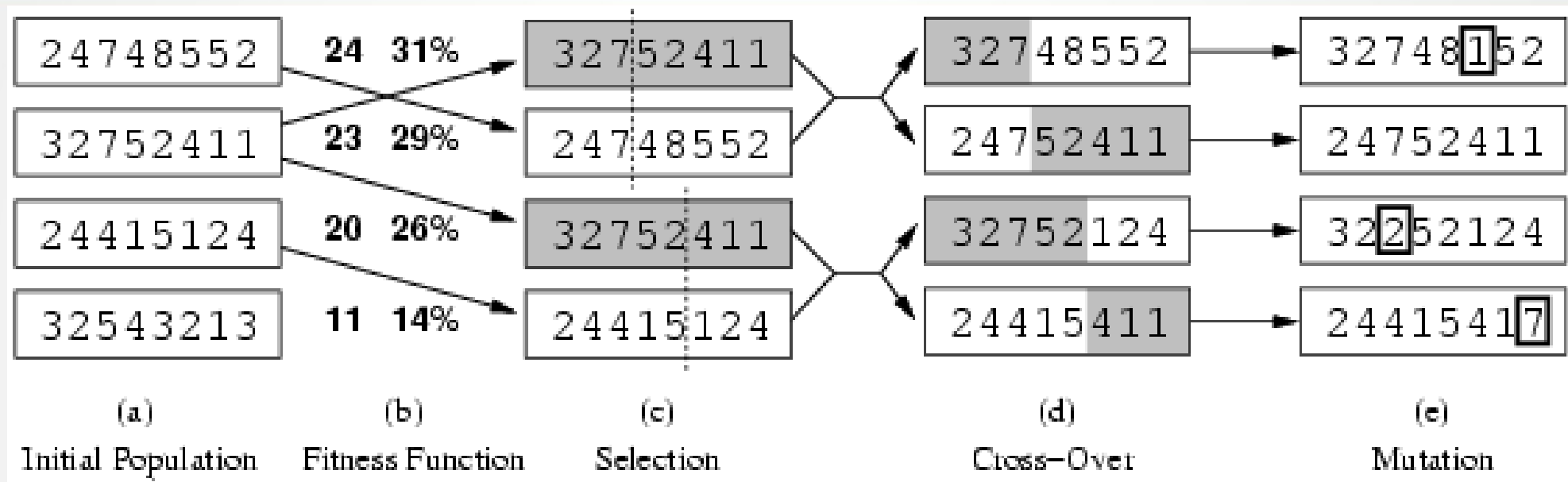
24748552	24
32752411	23
24415124	20
32543213	11

(b) *Avaliação*

Função de avaliação (aptidão)

- Deve retornar valores maiores para estados melhores
- Ex.: 8 rainhas: número de pares de rainhas não-atacantes
 - Valor 28 para uma solução
 - » $(\min = 0, \max = 8 \times 7/2 = 28)$

Exemplo

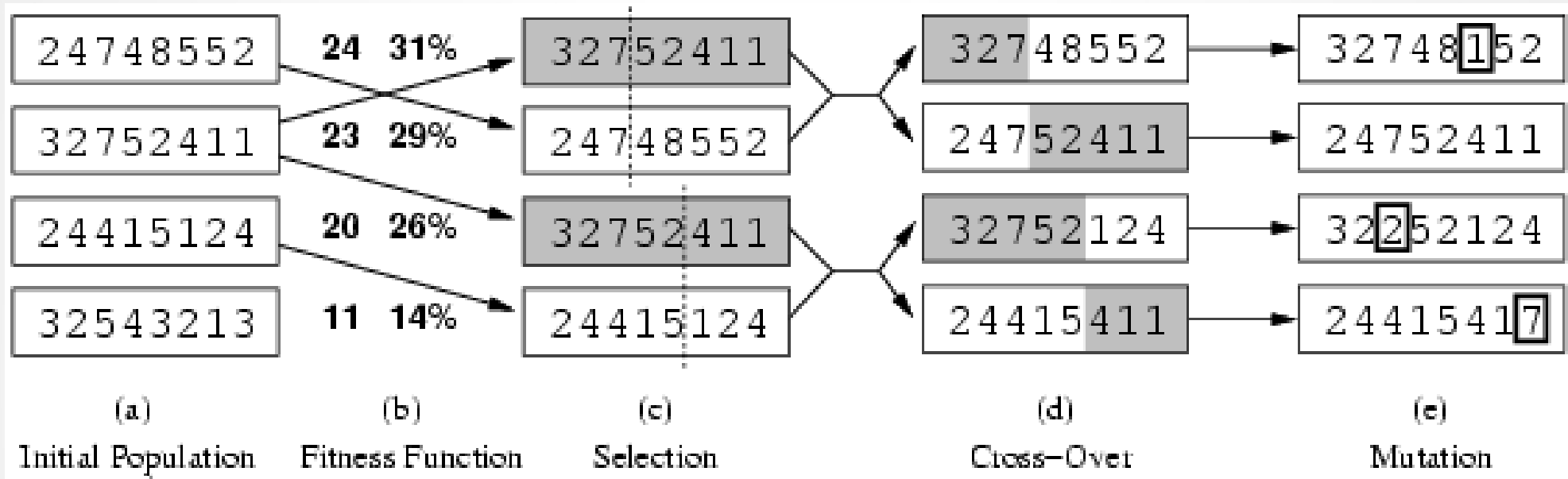


(c) *Seleção*

Proporcional à aptidão do indivíduo

- Vários métodos
- Todos tendem a privilegiar indivíduos mais aptos
 - $24/(24+23+20+11) = 31\%$ no exemplo
 - $23/(24+23+20+11) = 29\%$ etc

Cruzamento



(d) *Cruzamento*

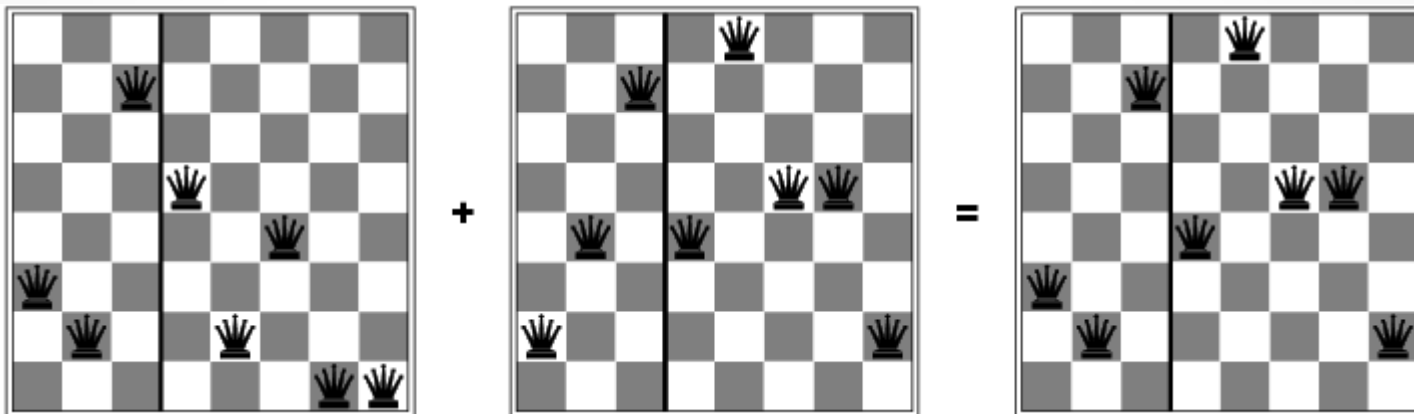
Indivíduos selecionados formam pares

Operador de cruzamento combina pares

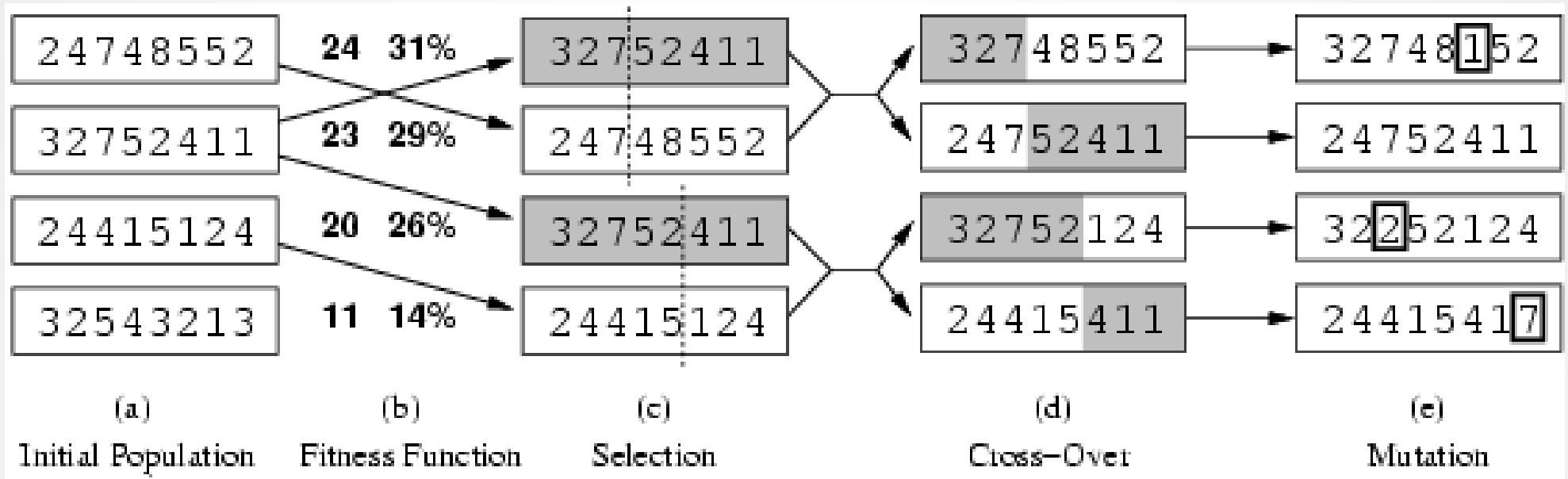
- Ponto de cruzamento gerado ao acaso

Algoritmos Genéticos

Cruzamento



Algoritmos Genéticos



(e) *Mutação*

Mudança aleatória do valor de um gene

- Com pequena probabilidade

Introduz variações aleatórias

- Permitindo soluções pularem para diferentes partes do espaço de busca

Observações

- Se o AG estiver corretamente implementado, a população evolui em gerações sucessivas
- Aptidão do melhor indivíduo e do indivíduo médio aumentam em direção a um ótimo global

Trabalho Individual – 24/04/2015

- Implemente dois programas para resolver o problema das 8-rainhas, utilizando conceitos de: (1) Subida de Encosta; (2) Algoritmos Genéticos;
- O que entregar?
 - 1) **Dois programas** (subida de encosta e algoritmo genético);
 - 2) **Relatório comparativo** (e.g., tempo gasto para encontrar solução, # de falhas, # de reinícios, gerações de indivíduos, configuração de cada parâmetro, critérios adotados para resolver falhas, etc) entre os programas implementados.

Animação buscas

AG

<http://math.hws.edu/xJava/GA/>

<http://www.obitko.com/tutorials/genetic-algorithms/example-function-minimum.php>

<http://www.obitko.com/tutorials/genetic-algorithms/example-3d-function.php>

<http://www.obitko.com/tutorials/genetic-algorithms/tsp-example.php>

Referências

- Livros:
 - Russel e Norvig: Inteligência Artificial, cap 4

Programação Genética

- Extensão de Algoritmos Genéticos
- Indivíduos são programas
- Palestra sobre PG no Mundo Visual