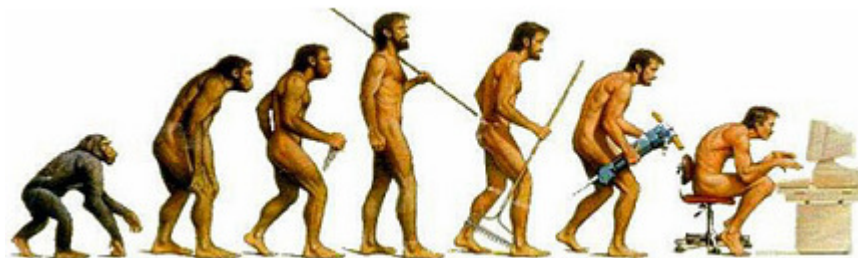


# Algoritmos Bio-inspirados



## Conceitos e Aplicações em Aprendizado de Máquina

Gisele L. Pappa

Departamento de Ciência da Computação

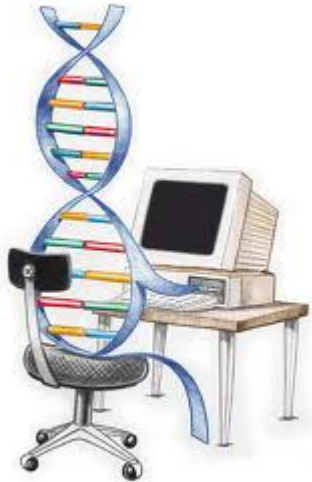
Universidade Federal de Minas Gerais

[glpappa@dcc.ufmg.br](mailto:glpappa@dcc.ufmg.br)

# Organização do Curso

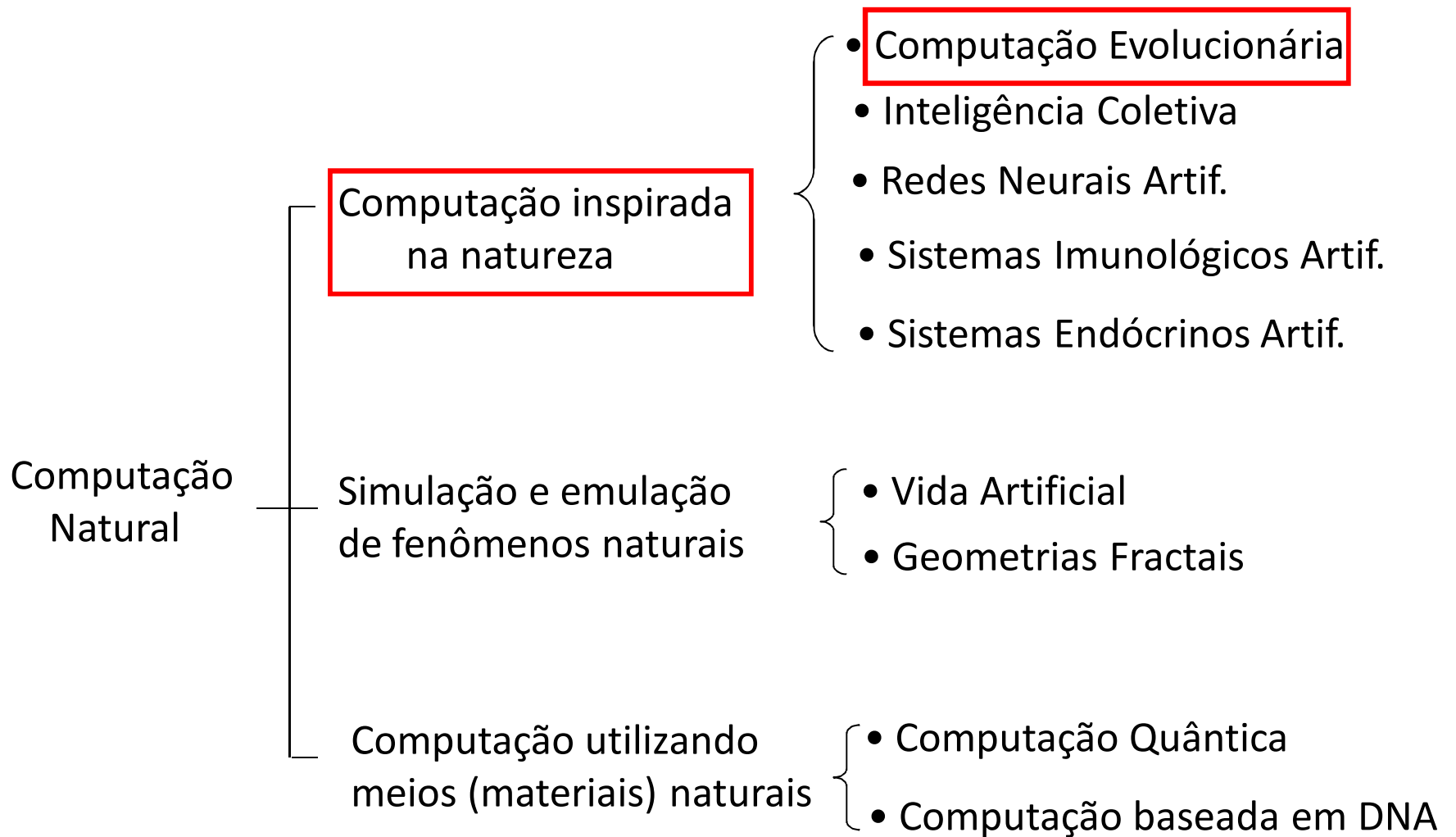
- Parte 1: Introdução a Algoritmos Bio-inspirados
- Parte 2: Introdução a Aprendizado de Máquina
- Parte 3: Aplicações de Algoritmos Evolucionários em Aprendizado de Máquina

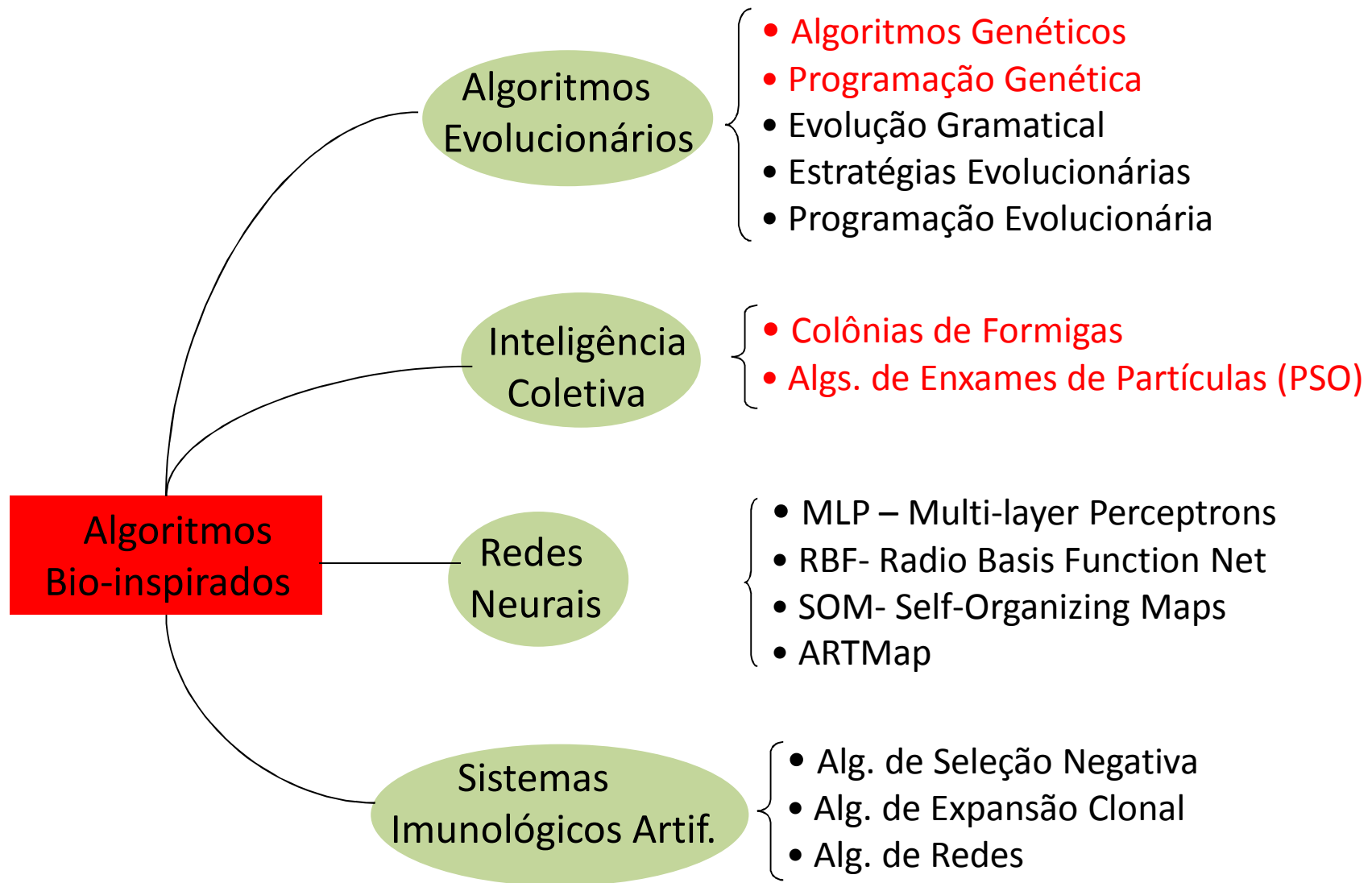
# Parte 1: Introdução a Algoritmos Bio-inspirados (ou Computação Natural)



# O que é Computação Natural?

- Natureza utilizada como fonte de inspiração ou metáfora para desenvolvimento de novas técnicas computacionais utilizadas para resolver problemas complexos
- Metáforas
  - Não necessariamente incluem todos os detalhes do sistema natural
  - Simplificações são necessárias

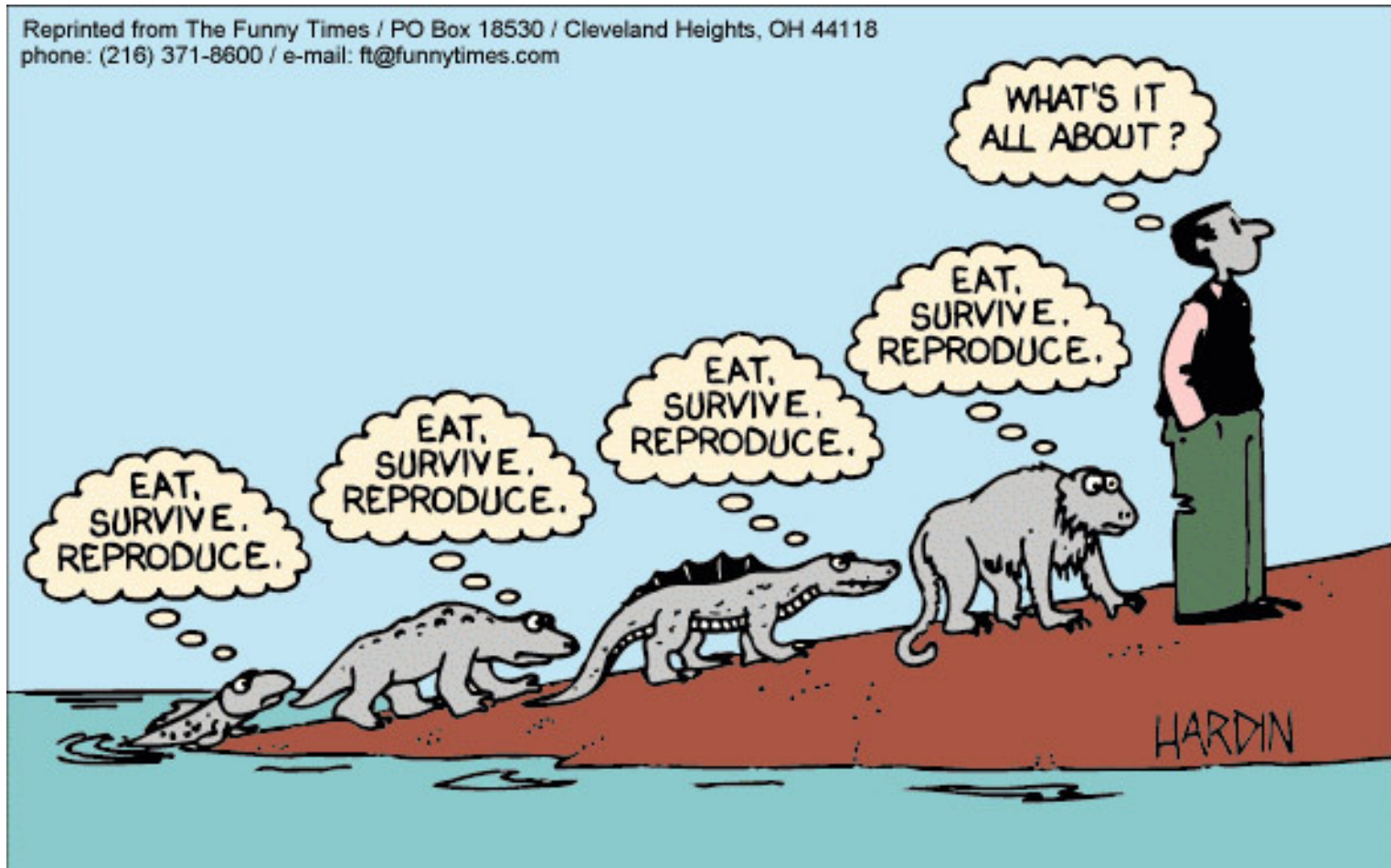




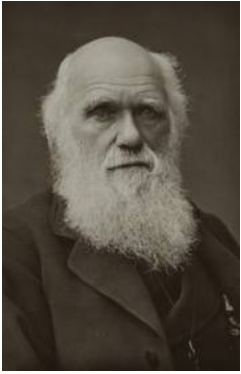
# O que é Computação Natural?

- Métodos estocásticos (não-determinístico)
- Não garantem que a solução ótima será encontrada, mas sim a quase-ótima.
- Grande maioria dos métodos apresenta uma maneira declarativa de resolver um problema (o quê fazer), em contraste com métodos procedurais (como fazer)

# Computação Evolucionária







# Computação Evolucionária (CE)

- Métodos inspirados na teoria da evolução de Darwin, propostos pela primeira vez em 1958
- Anos 60
  - Algoritmos Genéticos (e Programação Evolucionária) vs Estratégias Evolutivas
- 1992
  - Programação Genética
- Anos 90 essas técnicas foram combinadas com o nome Computação Evolucionária

# Onde se usa CE?

- Muitas aplicações para listar
  - Engenharia
  - Design de circuitos
  - Modelos financeiros
  - Jogos
  - Bioinformática
  - Modelagem
- Capazes de gerar resultados **competitivos com aqueles encontrados por humanos.**

# Independência de Aplicação

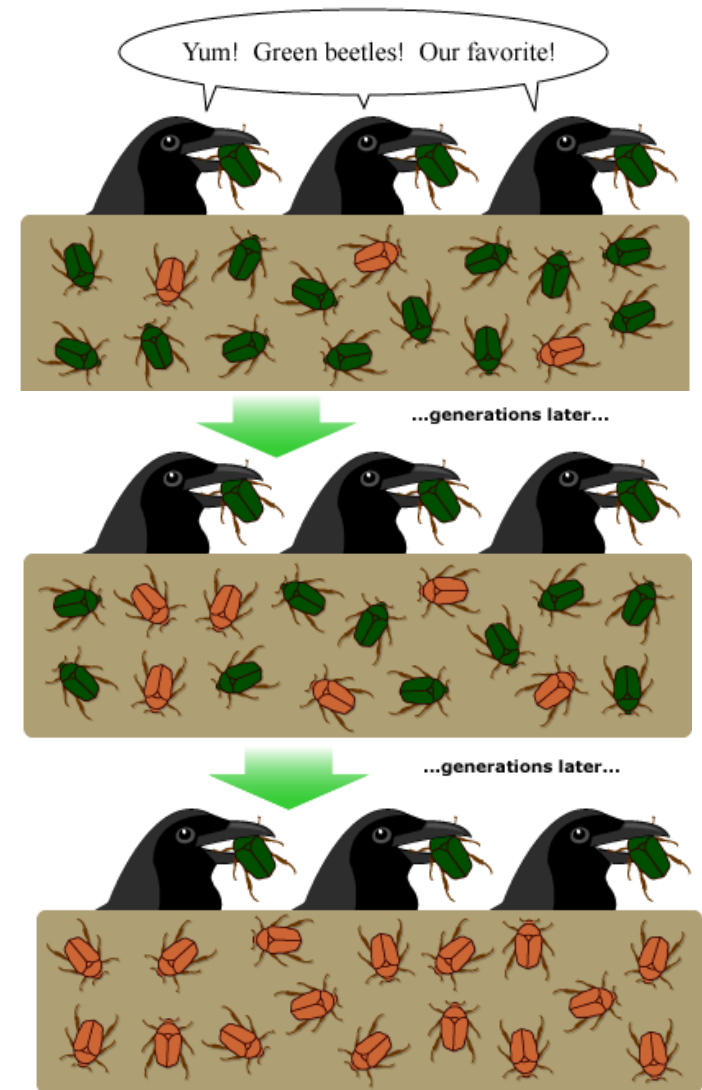
- O que faz EAs serem robustos para os mais diversos tipos de aplicações?
  - O algoritmo em si é o mesmo para qualquer problema
  - Existem 3 componentes importantes que devem ser definidos de acordo com o problema em mãos:
    - Representação dos indivíduos
    - Função de aptidão (fitness)
    - Operadores genéticos – novos operadores podem ser definidos especificamente para o problema

# Motivação para utilizar CE?

- Oferecem soluções robustas e adaptativas
- Processamento paralelo implícito
- Inteligência de máquinas
  - Possibilita incorporar conhecimento ao método sem explicitamente programá-lo
- Necessidade de validar teorias e conceitos da biologia evolucionária
- Busca global

# O que é evolução?

- Mudança das características (genéticas) de uma população de uma geração para a próxima
  - Mutação dos genes
  - Recombinação dos genes dos pais
- Seleção natural é seu principal agente causador



# O que é evolução?

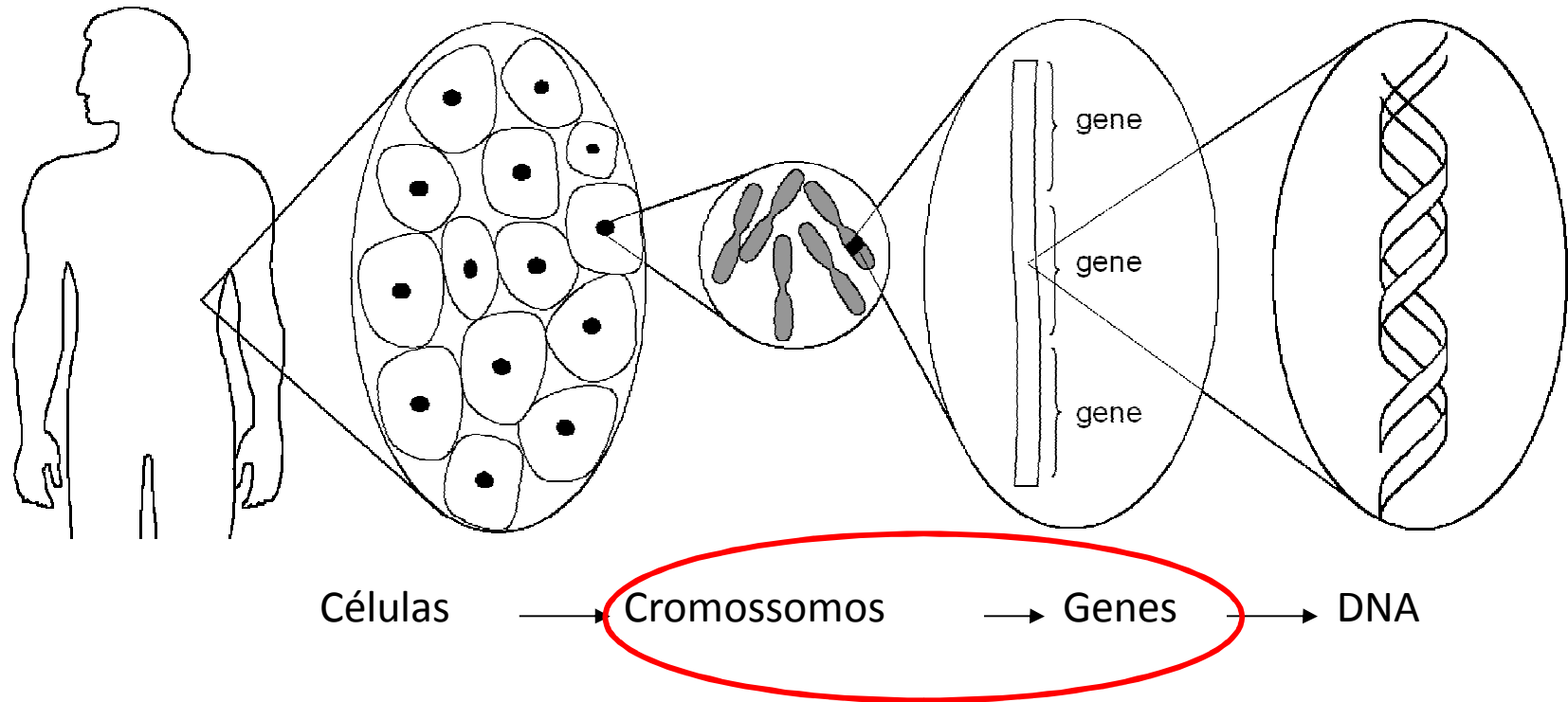
- *Good things come for those who wait?*



<http://www.youtube.com/watch?v=1t4sdgvy-pk>

# Biologia Evolucionária (1)

- Estruturas básicas:

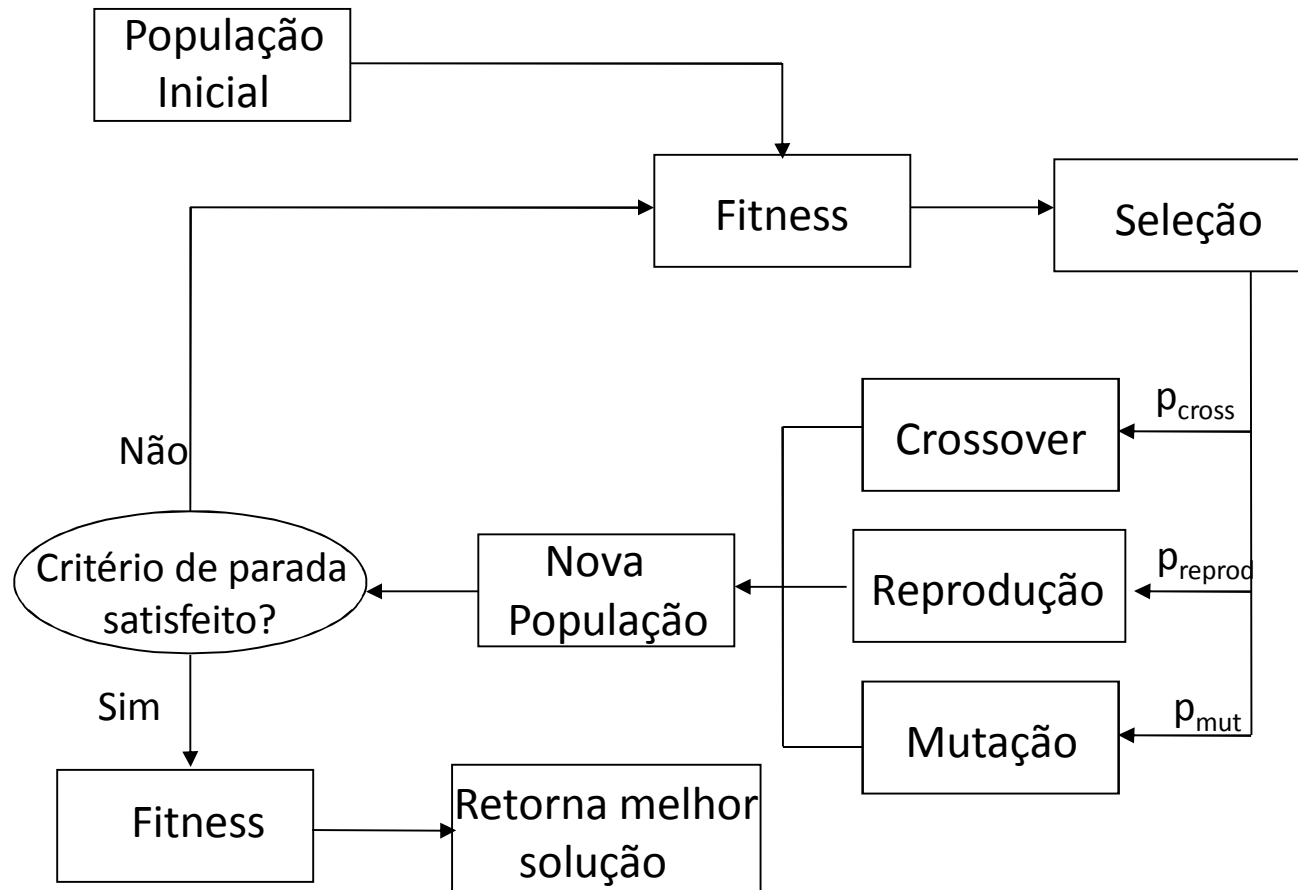


# Biologia Evolucionária (2)

- **Cromossomos**
  - Carregam informações hereditárias de um organismo
  - Podem ser dividido em genes
    - Um gene é uma região do DNA que controla uma característica hereditária
- **Genótipo**
  - Material genético contido em uma célula ou organismo
- **Fenótipo**
  - Características físicas ou bioquímicas de um organismo que podem ser observadas, e que são determinadas tanto pelo genótipo quanto por influências do meio
- **Fitness**
  - Probabilidade de reprodução de um indivíduo

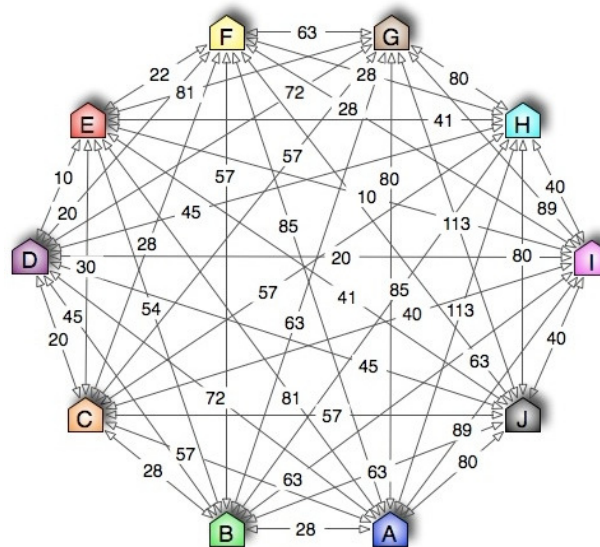


# Da biologia para computação



# Idéias Básicas

- AE é um procedimento iterativo que evolui uma população de indivíduos
- Cada indivíduo representa uma **solução candidata** para um dado problema
- Exemplo: **Problema do Caxeiro Viajante**



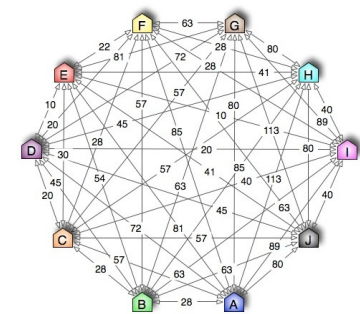
# Computação Evolucionária

- Historicamente, diferentes tipos de AEs têm sido associadas a **diferentes tipos de representação**
  - Vetores de bits : Algoritmos Genéticos
  - Vetores de números reais : Estratégias Evolucionárias
  - Máquinas de estado finito: Programação Evolucionária
  - Árvores: Programação Genética
- Foco do curso: **algoritmos genéticos e programação genética**

# Algoritmos Genéticos

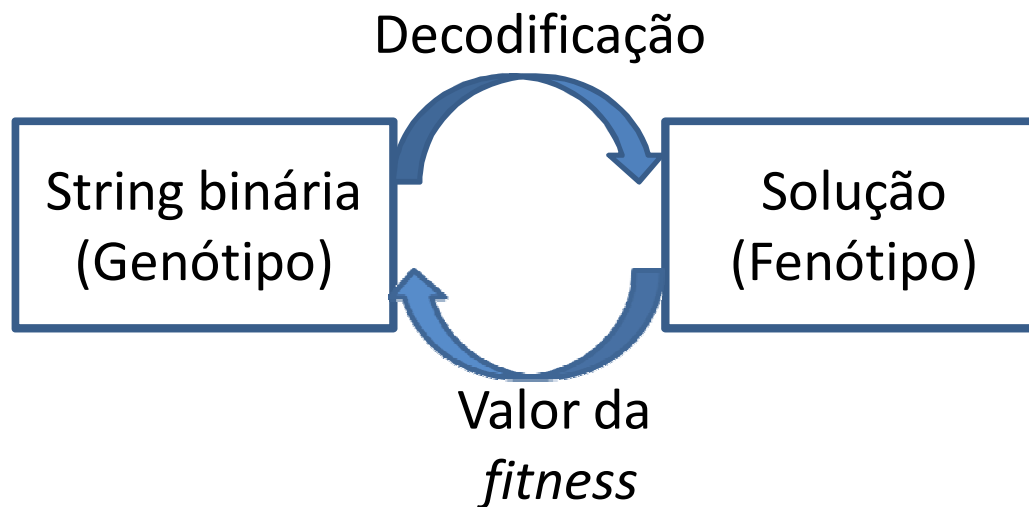
# Algoritmos Genéticos

- Técnica mais dissiminada em EA
- Introduzida por Holland em 1975, e desenvolvida por um de seus estudantes, Goldberg
- Indivíduos são strings binárias
- Qual seria a representação para o problema do caixeiro?

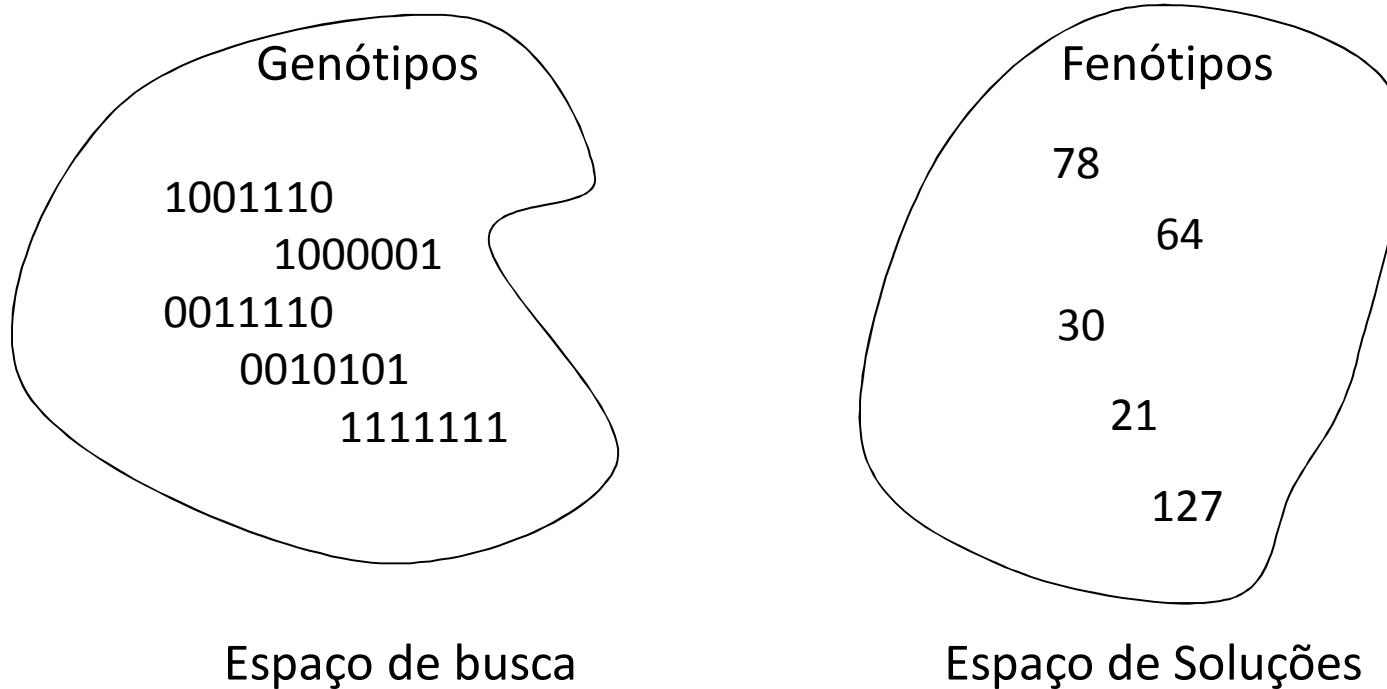


# Algoritmos Genéticos

- Cromossomo (indivíduo) tem **tamanho fixo**
  - Genes normalmente tem tamanho fixo
- Existe um mapeamento do genótipo para o fenótipo

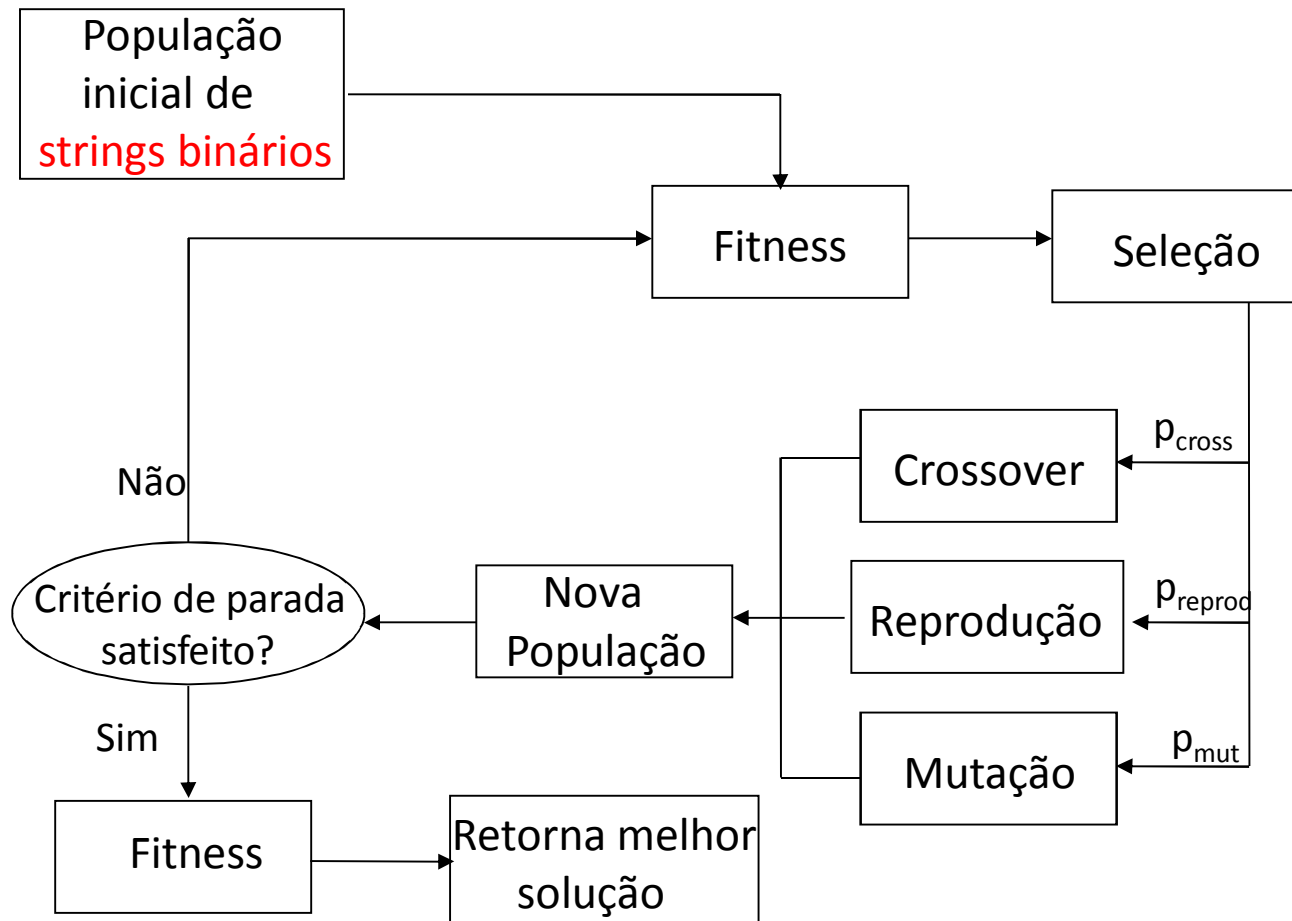


# Genótipo versus Fenótipo



- Em alguns algoritmos evolucionários não existe distinção entre genótipo e fenótipo

# Algoritmo Genético





# Exemplo – *OneMax*

- Objetivo: Maximizar o número de 1s em um *string* de bits de tamanho  $n$ 
  - Definição de parâmetros
    - $N = 8$  e tamanho da população = 4

Indivíduo

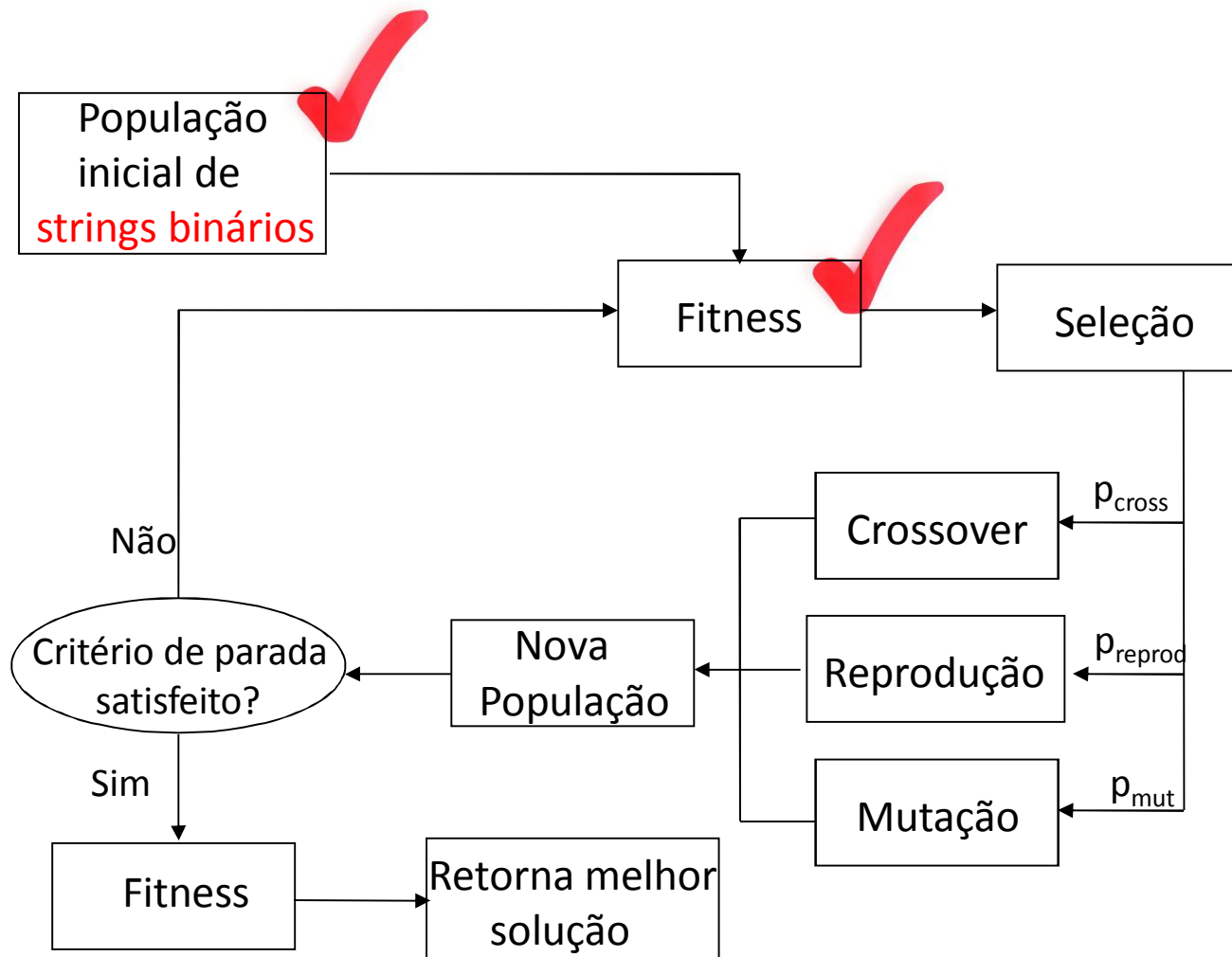
A

B

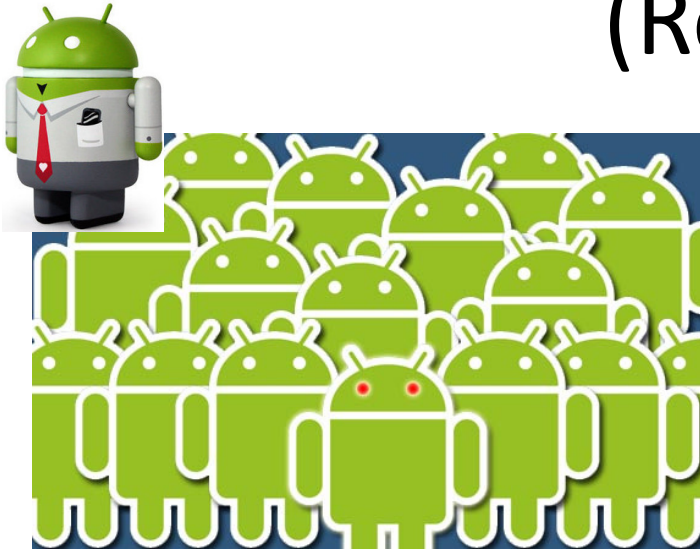
C

D

# Algoritmo Genético



# Seleção Proporcional a Fitness (Roleta Russa)



$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f_i$$



*Fitness  $f_i$*

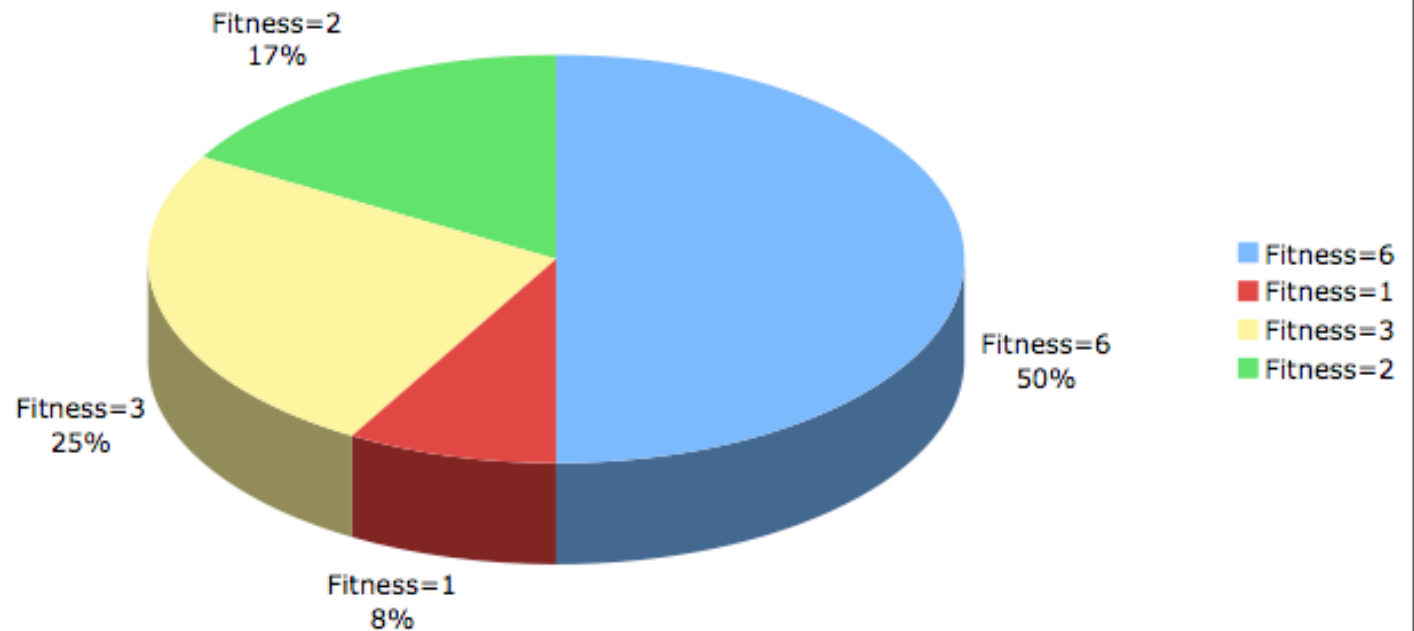
- Indivíduo  $j$  pode ser selecionado com a probabilidade

$$p_j = \frac{f_j}{\sum_{i=1}^N f_i}$$

# Seleção Proporcional a Fitness (Roleta)

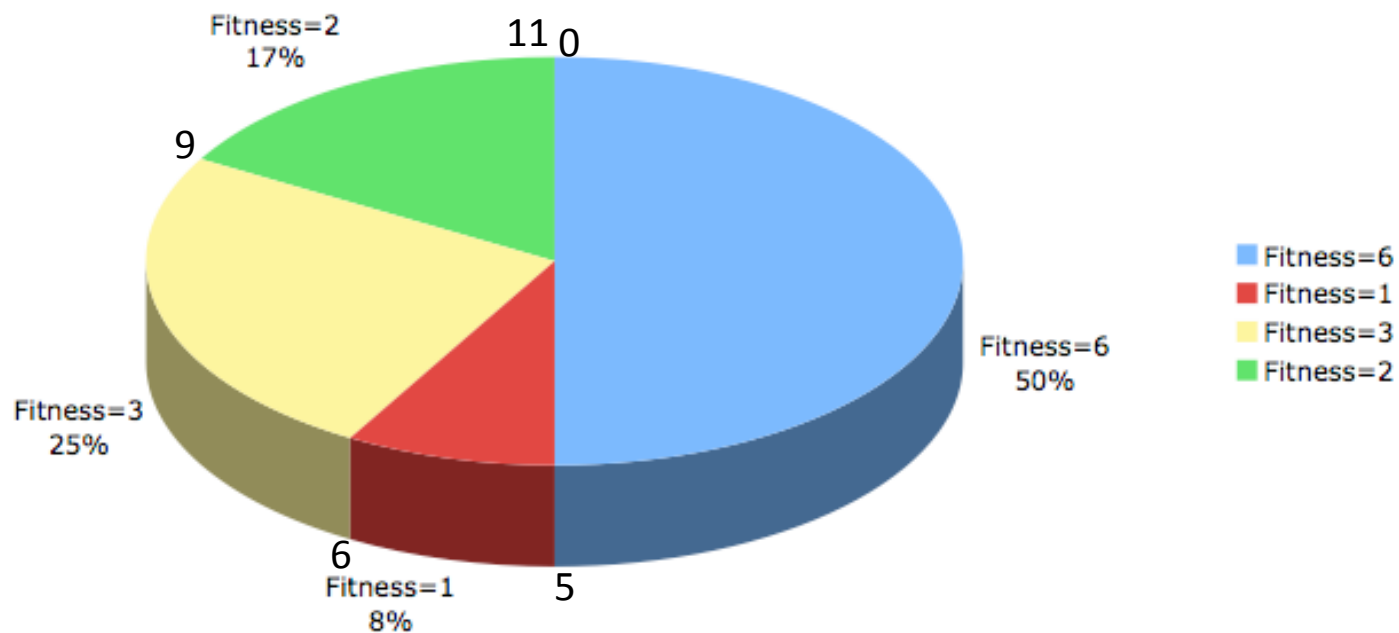
Indivíduo	String	Fitness
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

12



# Seleção Proporcional a Fitness (Roleta)

- Rodo a roleta  $r \in \left[ 0, \sum_{j=1}^N f_j \right)$



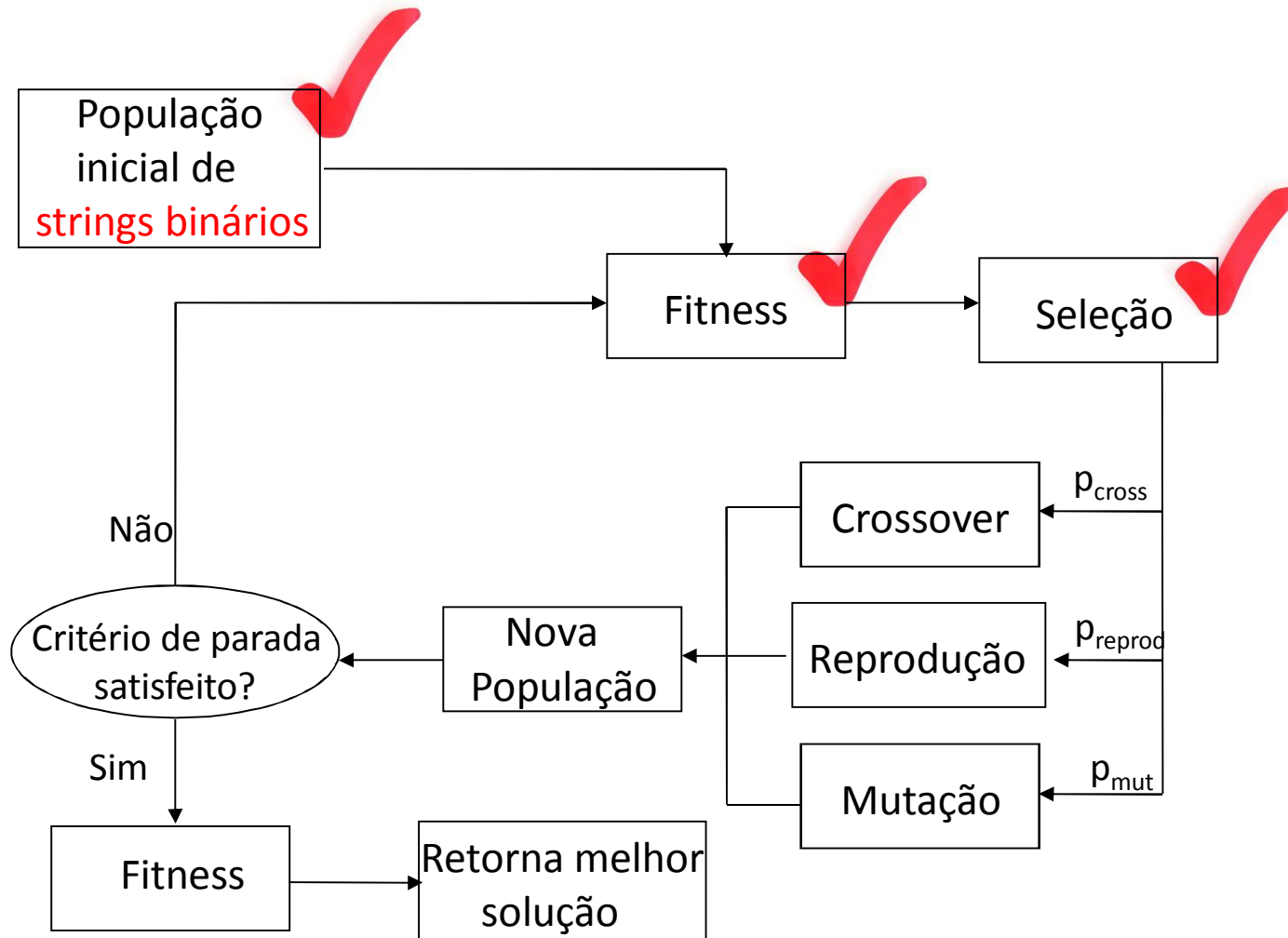
# Seleção por Torneio

- Um subconjunto de  $k$  indivíduos é retirado aleatoriamente da população
- O melhor indivíduo desse subconjunto de acordo com a fitness (o vencedor do torneio) é selecionado
  - $k$  = tamanho do torneio



- Quanto maior o valor de  $k$ , maior a pressão seletiva
- Tornou-se um dos métodos mais comuns, com  $k = 2$

# Algoritmo Genético





# Operadores Genéticos

- Cruzamento de um ponto (de acordo com probabilidade definidas pelo usuário)
  - Padrão para GAs
  - Probabilidades altas (70-99%)
  - **Ponto de cruzamento** é escolhido aleatoriamente

Pais		Filhos	
B	1 1101110	E	01101110
C	0 0100000	F	10100000





# Operadores Genéticos

- Outro tipo de crossover: Crossover Uniforme
  - Cada gene é trocado de acordo com uma probabilidade  $p_c$

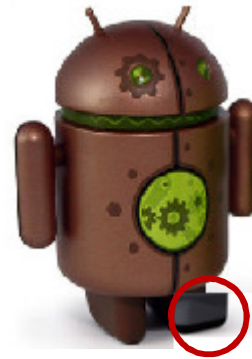
Pais		Filhos	
B	1 1 1 0 1 1 1 0	E	0 1 1 0 1 1 1 0
C	0 0 1 0 0 0 0 0	F	1 0 1 0 0 0 0 0



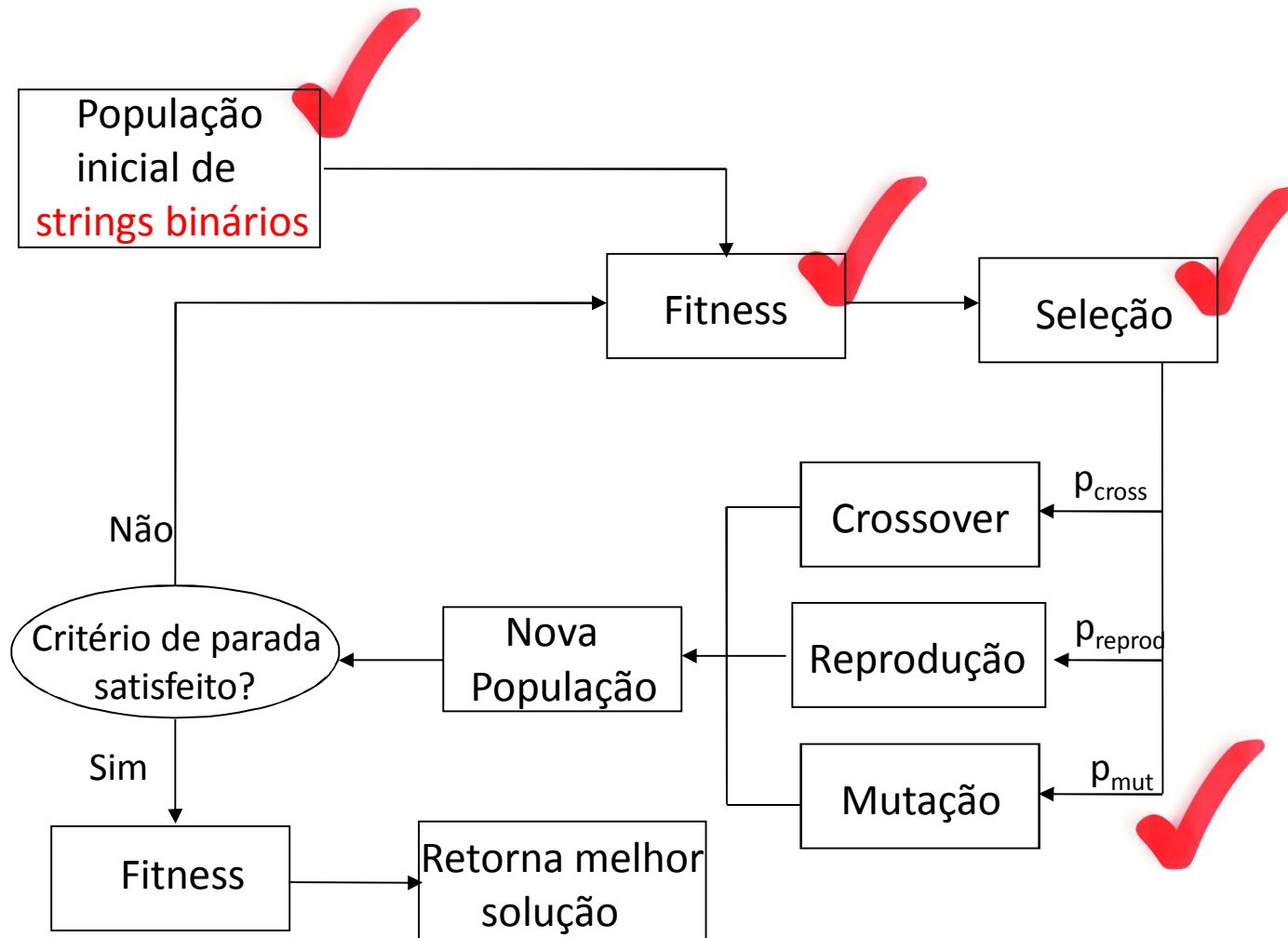
# Operadores Genéticos

- Mutação
  - Baixa probabilidade –em sistemas naturais os efeitos da mutação podem ser destrutivos
  - Calcula a probabilidade de trocar cada um dos genes (bits) do cromossomo **ou** escolhe um ponto de troca. A troca ocorre com um novo elemento aleatório

Pai	1 110 <b>1</b> 110
Filho	1110 <b>0</b> 110



# Algoritmo Genético



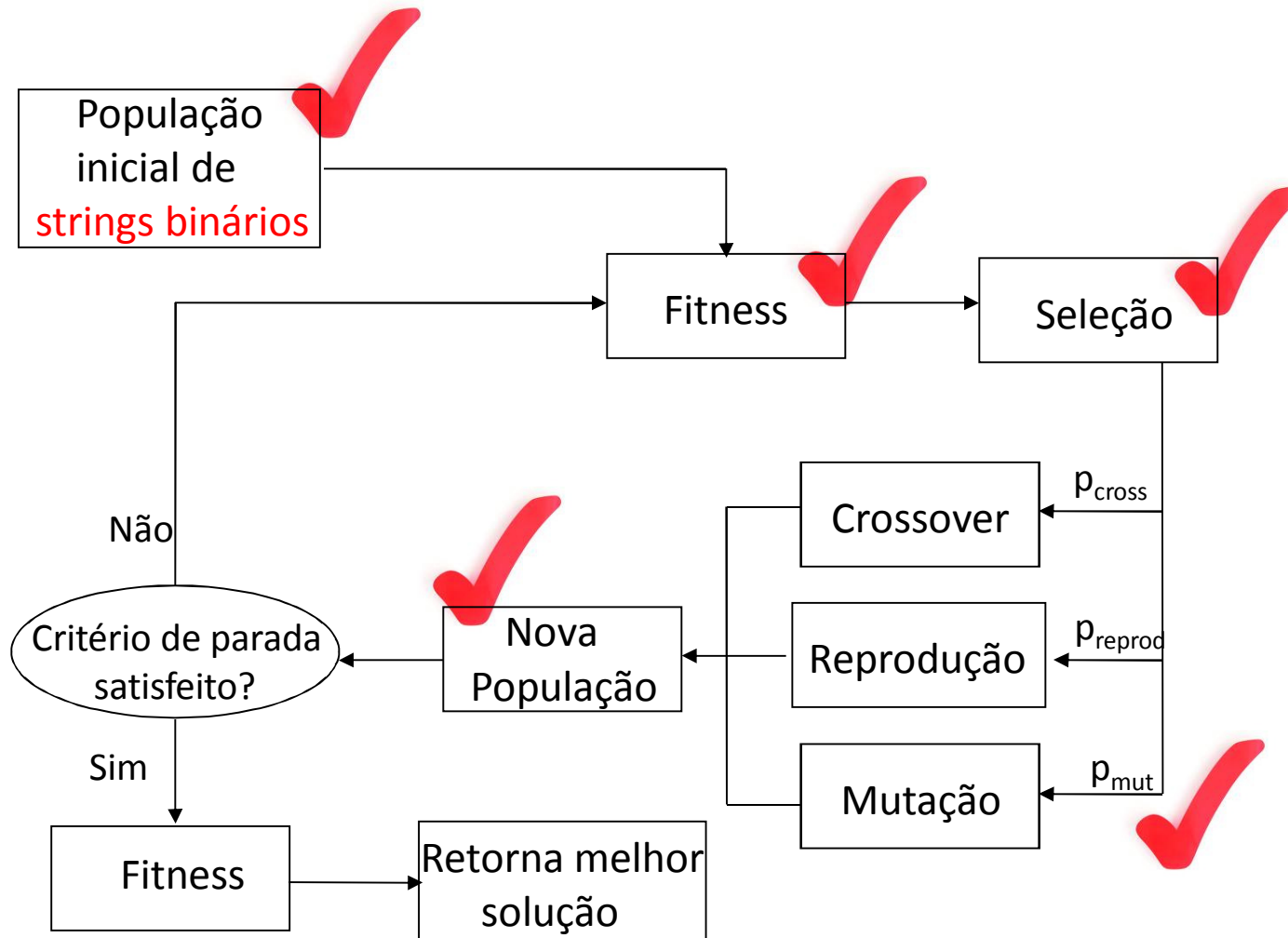
# Substituição da população atual pela nova

População atual	Indivíduo	String	Fitness
	A	00000110	2
	B	11101110	6
	C	00100000	1
	D	00110100	3



Nova população	Indivíduo	String	Fitness
	E	01001110	4
	F	10100000	2
	G	11101110	6
	H	00110100	3

# Algoritmo Genético

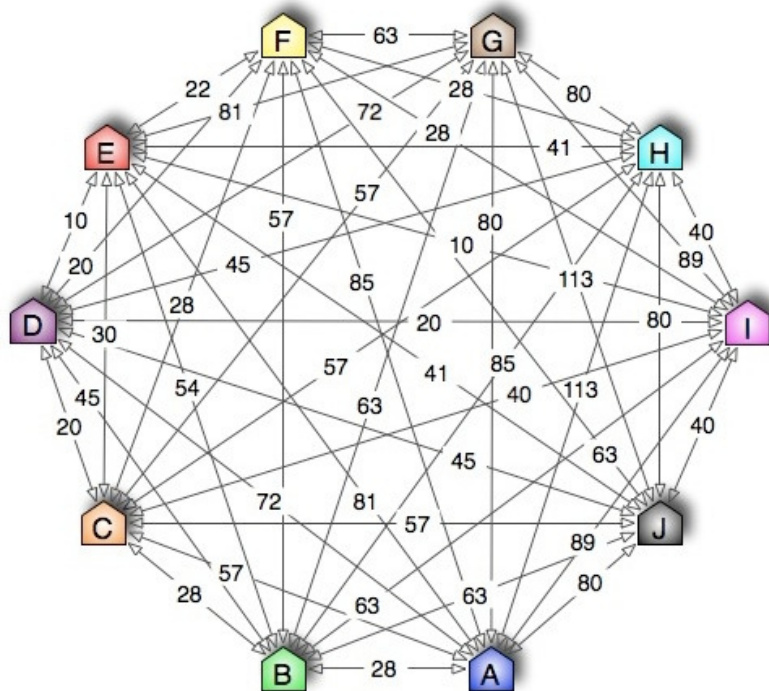


# Papel dos operadores na Evolução

- Seleção
  - Guia o algoritmo para áreas promissoras do espaço de busca
- Cruzamento
  - Muda o contexto de informação útil já disponível
- Mutação
  - Introduz inovação
- Conflito entre o papel da seleção e do cruzamento e mutação????
  - Buscar encontrar o melhor equilíbrio

# Exercício 1

- Como você modelaria um GA para resolver o problema do Caxeiro Viajante?



## Pontos Importantes:

- Representação dos indivíduos
- *Fitness*
- Operadores

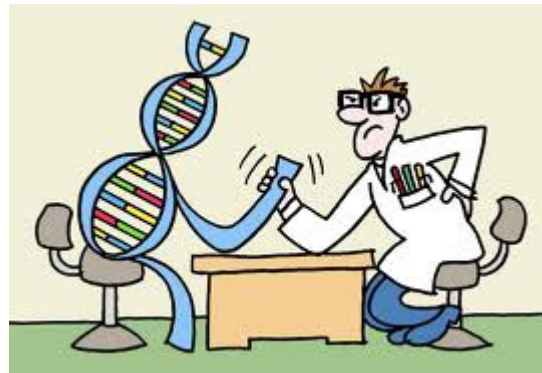
# Programação Genética



# Programação Genética

- Criada com o objetivo de evoluir programas
- Lista de 36 (re-)invenções que são competitivas com as soluções propostas por humanos
  - 2 geraram patentes

[www.genetic-programming.org](http://www.genetic-programming.org)



# Programação Genética

- Principais características
  - Um indivíduo é uma solução candidata **contendo funções e operadores**, e não apenas “dados” (variáveis /constantes)
  - Normalmente indivíduos tem tamanhos e formas variadas
  - Na teoria, um indivíduo é uma “receita” para resolver um dado problema, ao invés de uma solução para uma dada instância de um problema



# GA vs. GP

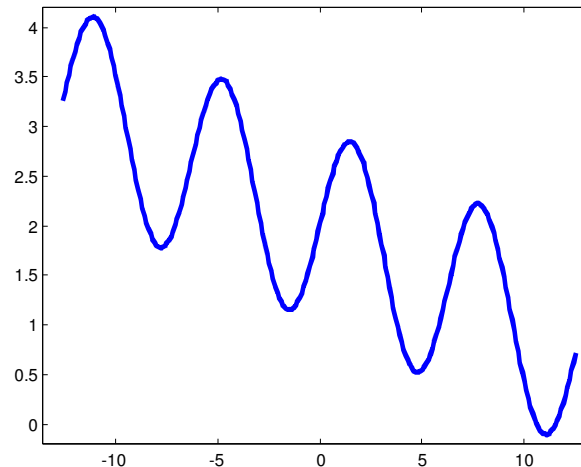
- Ex: **otimização** de funções vs. **aproximação** de funções
  - Dada uma função complexa, por exemplo  **$\sin(x) - 0.1x + 2$** , podemos usar um GA para encontrar o valor ótimo da função
  - Dado um conjunto de dados, contendo pares  $\langle x, f(x) \rangle$ , podemos utilizar GP para encontrar uma função  $g(x)$  que se aproxime da função desconhecida  $f(x)$  (regressão simbólica)
- Considere o problema do caixeiro viajante
  - Em um GA um indivíduo é uma sequência de cidades
  - Em GP um algoritmo deveria ser um “algoritmo” para resolver qualquer instância do problema, dado qualquer conjunto de cidades

# Otimização *versus* Aproximação de funções

**Exemplo de aplicação de um GA:**

**Encontre os máximos ou mínimos da função**

$$f(x) = \sin(x) - 0.1x + 2$$

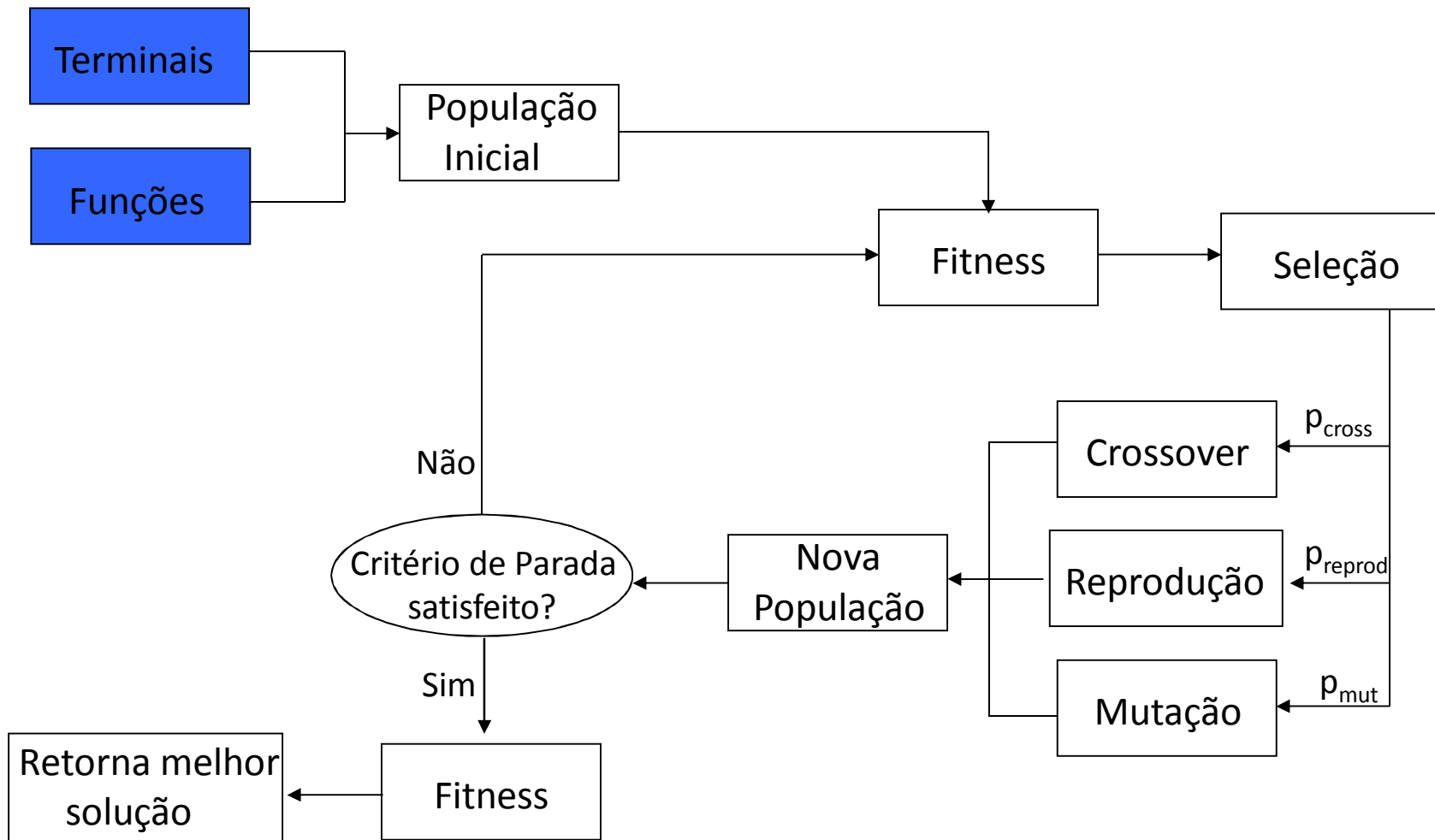


**Exemplo de aplicação de um GP:**

**Encontre a função (programa) que produz os pontos dados na tabela a seguir:**

x	saída
-10	3.6
-8	2.1
-6	2.7
-4	2.6
.	.
.	.

# Programação Genética



# Programação Genética (GP)

- 2 componentes básicos:
  - Conjunto de terminais – variáveis e constantes
  - Conjunto de funções – funções apropriadas para resolver o problema em questão
- Tipos de representação:
  - Representação linear
  - Representação por árvores (mais comum)
  - Representação por grafos

# Conjunto de Funções

- Propriedades desejadas
  - *Suficiência*: o poder de expressão é suficiente para representar uma solução candidata para o problema em questão
  - *Fechamento (Closure)*: uma função deve aceitar como entrada qualquer saída produzida por qualquer outro elemento do conjunto de funções ou do conjunto de terminais
  - *Parcimônia*: idealmente, conter apenas funções necessárias para resolver o problema em questão (propriedade não necessária, mas desejada)

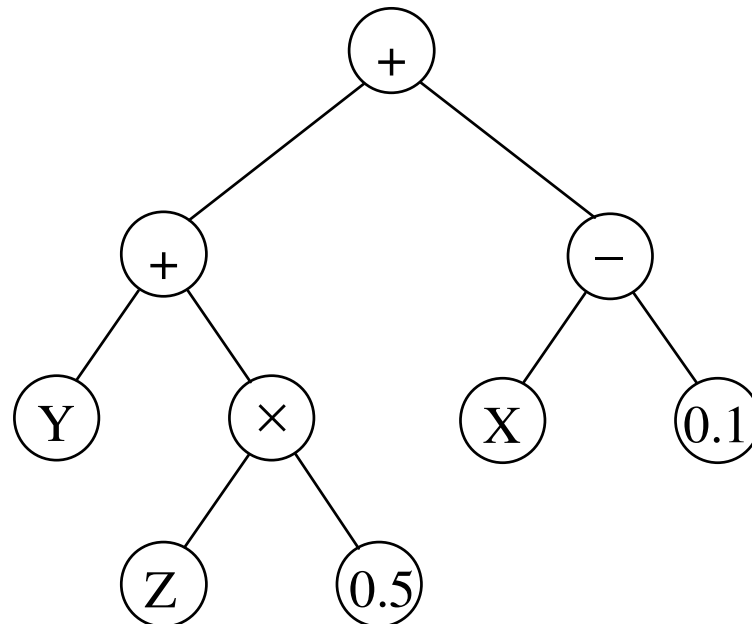
# Conjunto de Funções

- Como encontrar um equilíbrio entre poder de expressão e parcimônia?
  - Sugestão [Banzhaf et al. 1998, p. 111]:
$$FS = \{+, -, \times, /, \text{OR}, \text{AND}, \text{XOR}\}$$



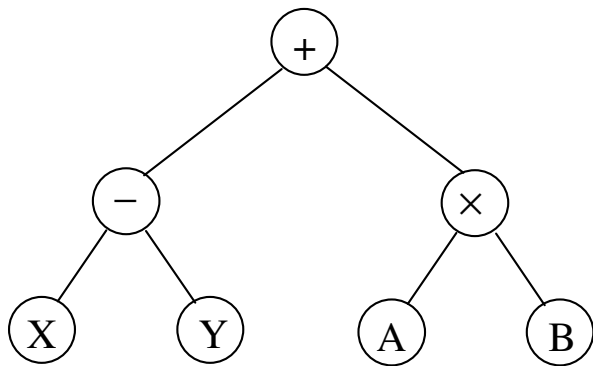
# Componentes básicos de um GP

- Focaremos na representação por árvore
  - Nós internos: funções ou operadores
  - Nós folhas: variáveis ou constantes

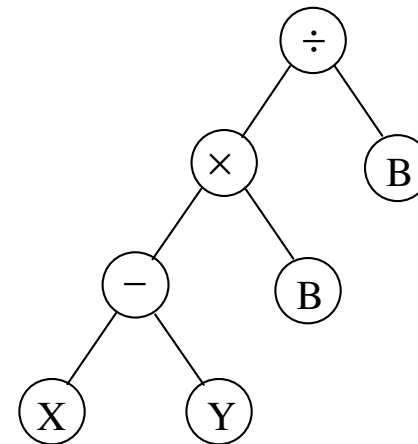


# Exemplos de Indivíduos

- Conjunto de funções:  $F = \{+, -, \times, \div\}$
- Conjunto de terminais:  $T = \{A, B, X, Y\}$



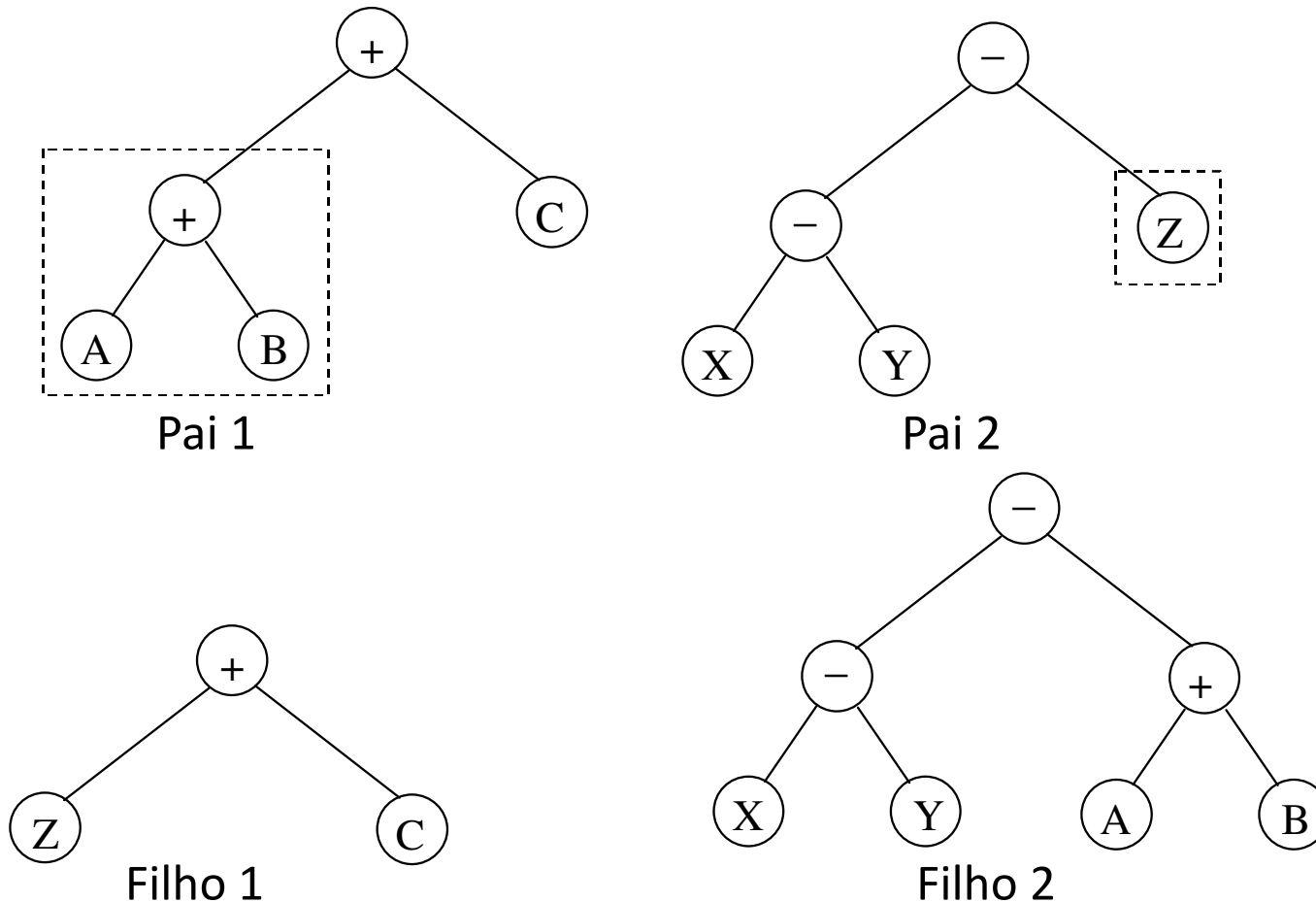
Indivíduo 1:  $(+ (- X Y) (\times A B))$



Indivíduo 2:  $(\div (\times (- X Y) B) B)$

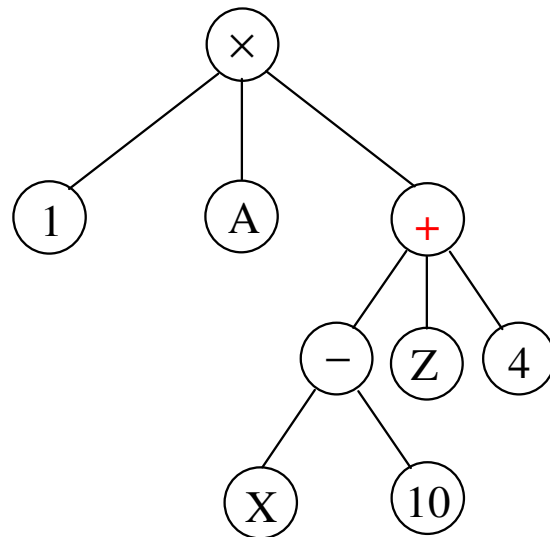
# Operadores Genéticos

- Crossover: troca sub-árvores

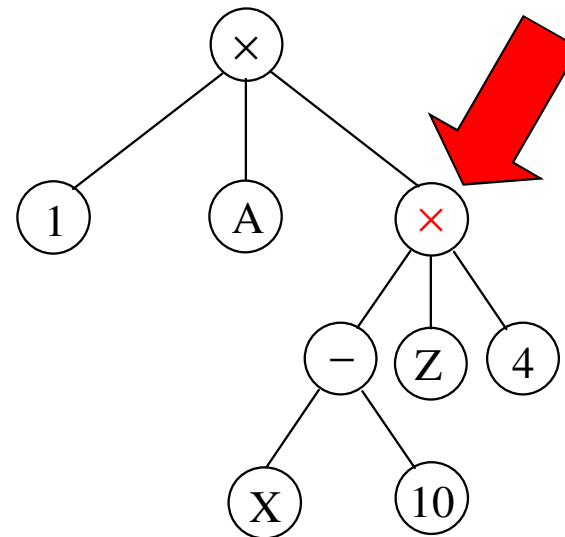


# Operadores Genéticos

- Mutação de um ponto



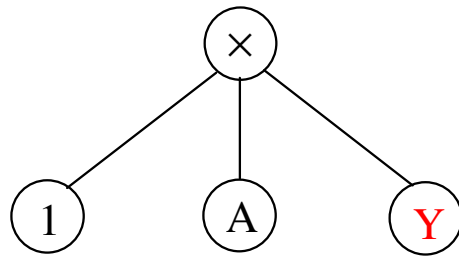
Pai



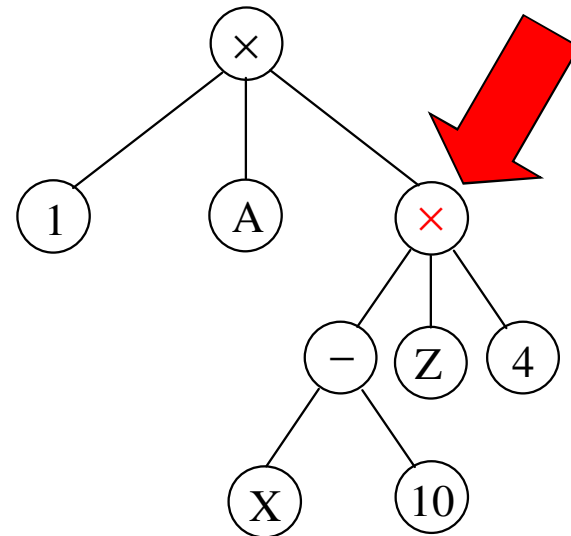
Filho

# Operadores Genéticos

- Mutação de expansão



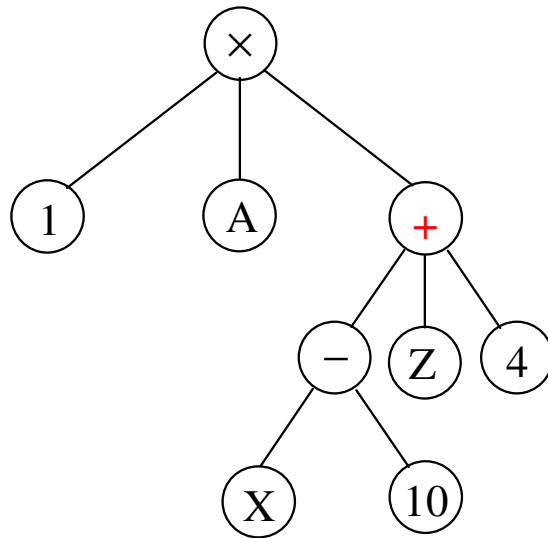
Pai



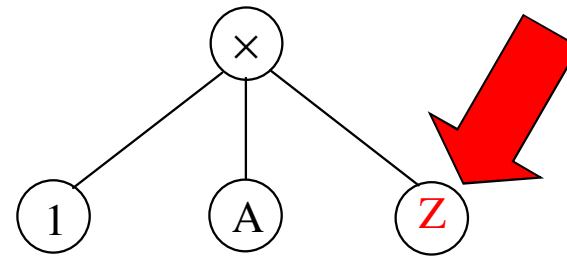
Filho

# Operadores Genéticos

- Mutação de redução



**Pai**



**Filho**

# Diferenças “Padrão” entre GA e GP

- **Algoritmos Genéticos (GA)**
  - Representação do indivíduo é, originalmente, um vetor binário. Atualmente vários tipos de representação são permitidas
  - Principal operador: cruzamento (altas probabilidades)
  - Operador secundário: mutação (baixa probabilidade)
- **Programação Genética (GP)**
  - Representação: utilizada não apenas dados, mas também funções
  - Objetivo original é evoluir programas ao invés de soluções para uma instância particular do problema
  - Acredita-se que cruzamento pode ter efeito destrutivo

# Parâmetros Necessários

- Tamanho da população
- Número de gerações
- Probabilidades de cruzamento
  - Se uniforme, probabilidade de trocar gene (0.5)
- Probabilidades de mutação
- Se seleção por torneio –  $k$
- Número de indivíduos do elitismo



# Avaliando a *Performance* de um EA

- Nunca tire conclusões a partir de uma execução
  - Rode o GA várias vezes
    - Para problemas simples idealmente pelo menos 30
- Utilize medidas estatística (médias, medianas, desvio padrão, etc)
- Salve o maior número de informações possíveis sobre sua população
  - Média, Melhor e Pior *fitness* a cada geração
  - Diversidade, etc
- Desenhe gráficos para acompanhar o progresso dessas variáveis
- Compare com um algoritmo de busca aleatória

# Design de antenas para missões da NASA [Hornby et al 2006]

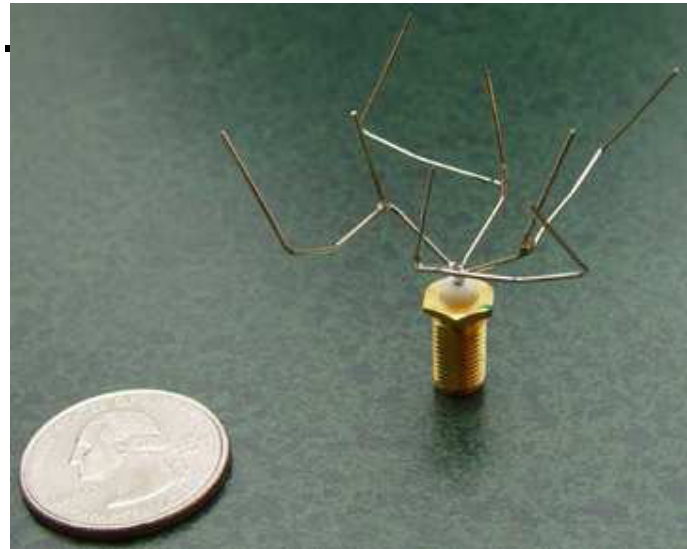
- NASA vem utilizando AEs para evoluir antenas (design) com as mais diversas finalidades desde 1990.
- Exemplo: antenas para o programa ST5, que visa lançar 3 micro-satélites (53cm x 28cm e 25kg)
- Cada satélite possui 2 antenas, centralizadas nas partes de cima e de baixo
- Desenvolveram um EA para evoluir antena considerando especificações da órbita dos micro-satélites



# Design de antenas para missões da NASA

- Objetivo:
  - Desenhar uma antena de 4 braços, separados por um ângulo de 90 graus
  - A antena deve ter um *voltage standing wave ratio* (VSWR) menor que 1.2 na frequência transmissora (8470 MHz) e menor que 1.5 na frequência receptora (7209).

Exemplo de antena desenvolvida



# Design de antenas para missões da NASA

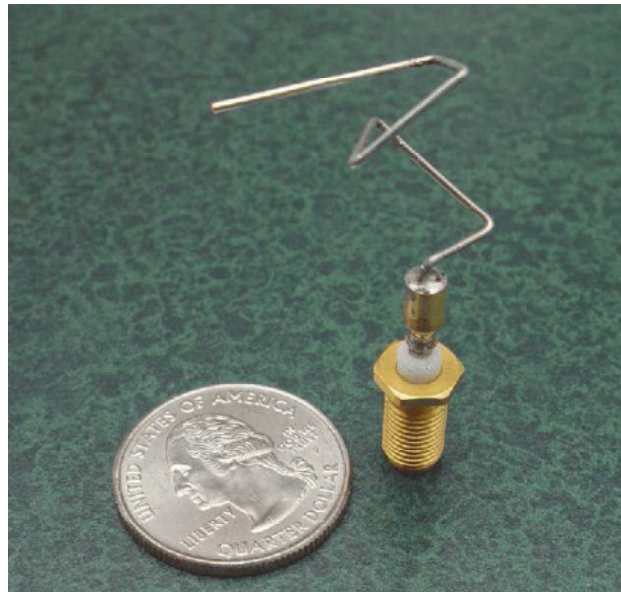
- Representação do Indivíduo
  - Representa um braço, e na hora da avaliação o AE cria uma antena completa com 4 braços idênticos
  - Representado por uma árvore, onde cada nó corresponde a um **operador construtor de antena**. A antena é criada executando os operadores em cada nó da árvore, começando pelo nó raiz
- Operadores construtores de antena :
  - forward(length, radius)
  - rotate-x(angle) / rotate-y(angle) /rotate-z(angle)

# Design de antenas para missões da NASA

- Fitness
  - Fitness calculada em função do VSWR
- Órbita inicial foi modificada, e uma nova antena com apenas um braço requerida
- Objetivo:
  - Desenhar uma antena de 1 braço

# Design de antenas para missões da NASA

- Apenas modificando a fitness do EA inicial e as restrições de design da antena, cientistas conseguiram desenvolver uma nova antena



Nova Antena

# Design de antenas para missões da NASA

- Em comparação com técnicas tradicionais de design de antenas, a antena evoluída apresenta vantagens em termos de
  - gasto de energia,
  - tempo de fabricação
  - complexidade
  - performance
- Comparação com uma antena especialmente fabricada para missão por humanos (QHA):
  - 2 QHAs: 38% de eficiência
  - 1 QHA com uma antena evoluída: 80% de eficiência
  - 2 antenas evoluídas: 93% de eficiência

Parte 2:

Aprendizado de Máquina e  
Mineração de Dados



Qual a diferença entre aprendizado de máquina e mineração de dados?

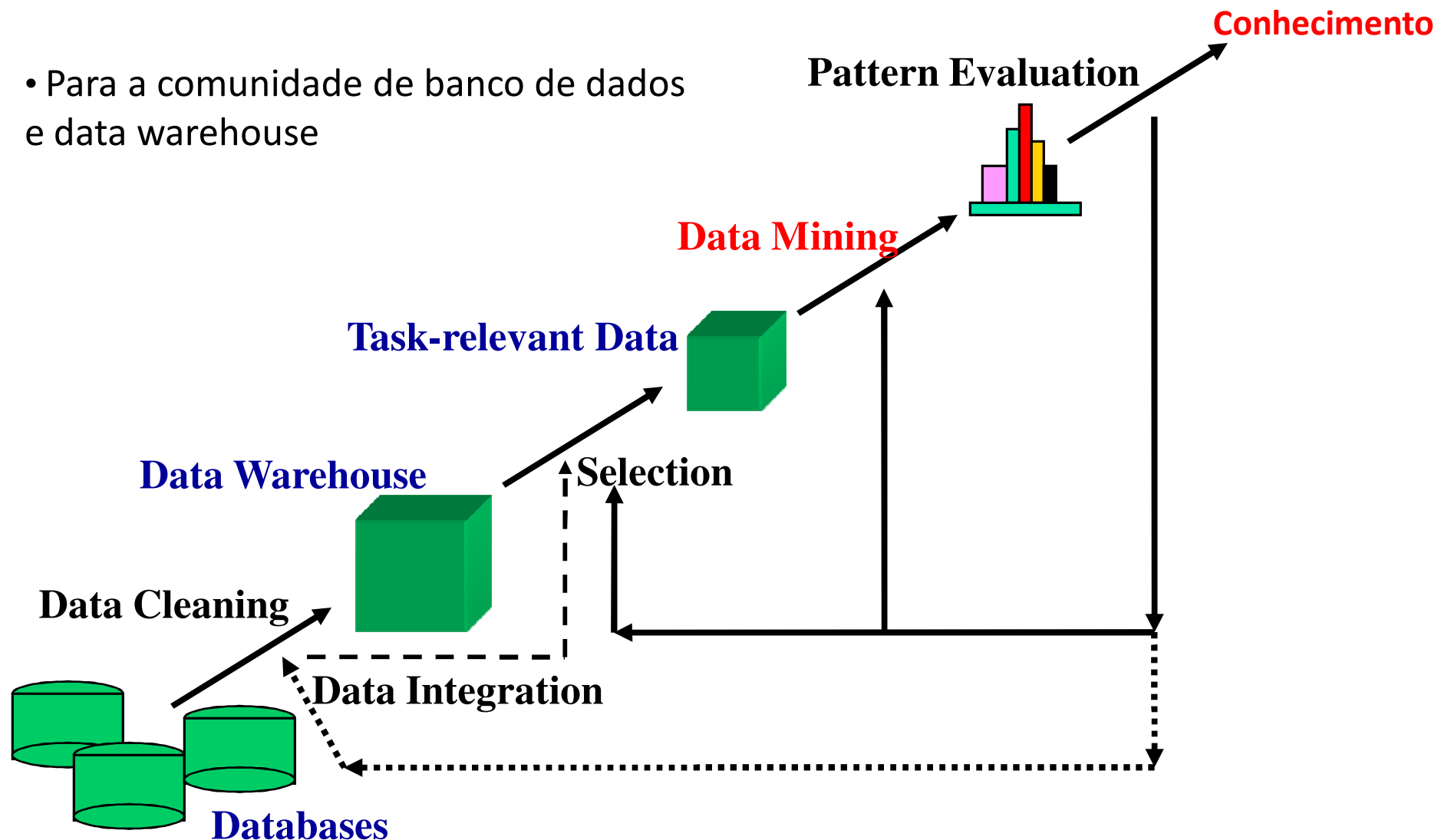
# O que é Mineração de Dados?

- Extração de padrões **interessantes** ou conhecimento de um grande volume de dados
- Também conhecido como KDD (*Knowledge Discovery in Databases*)
- O que é um padrão interessante?
  - Não-trivial
  - Implícito
  - Anteriormente desconhecido
  - Útil

Se ( <u>sexo</u> paciente == feminino) então grávida
---

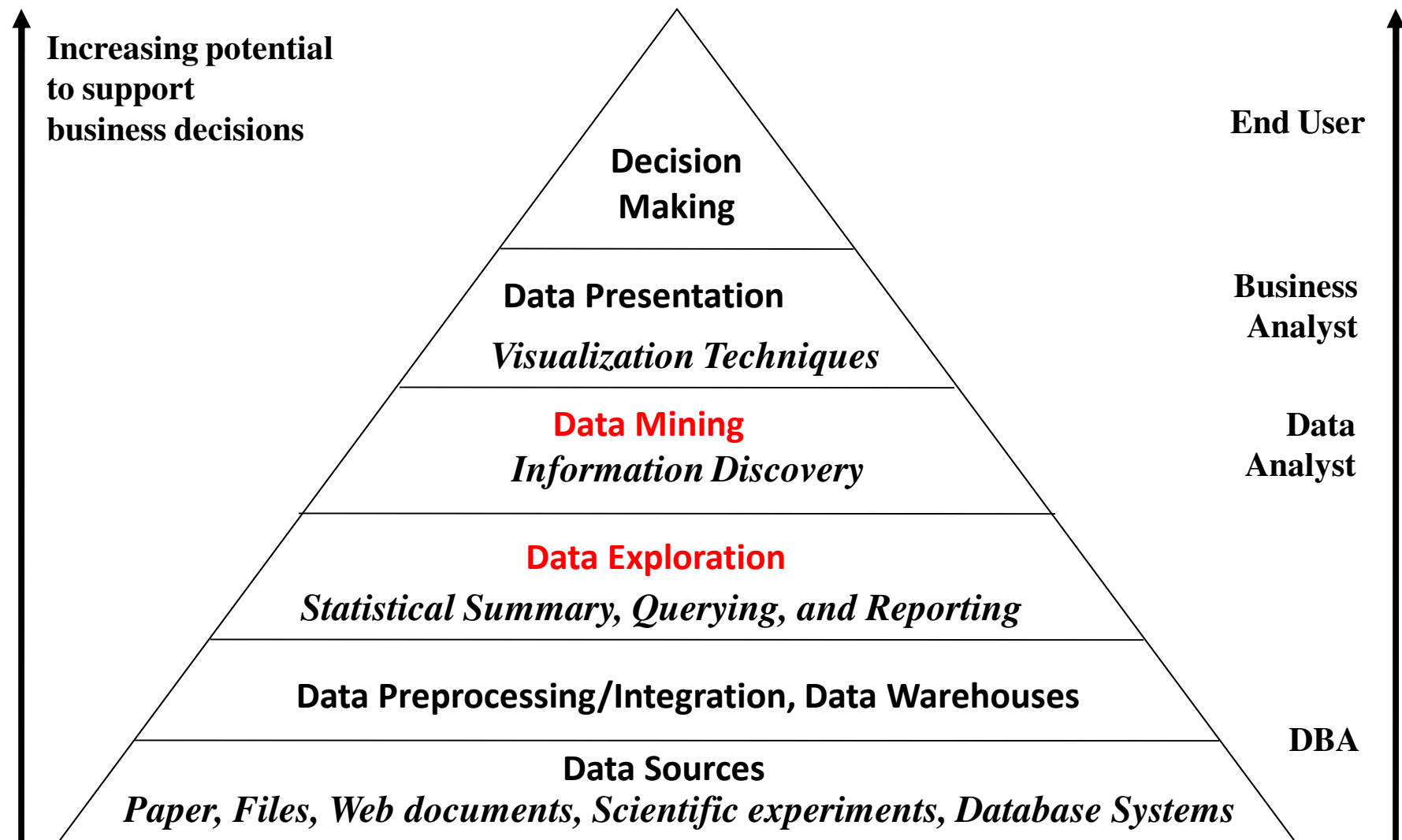
# O que é Mineração de Dados?

- Para a comunidade de banco de dados e data warehouse



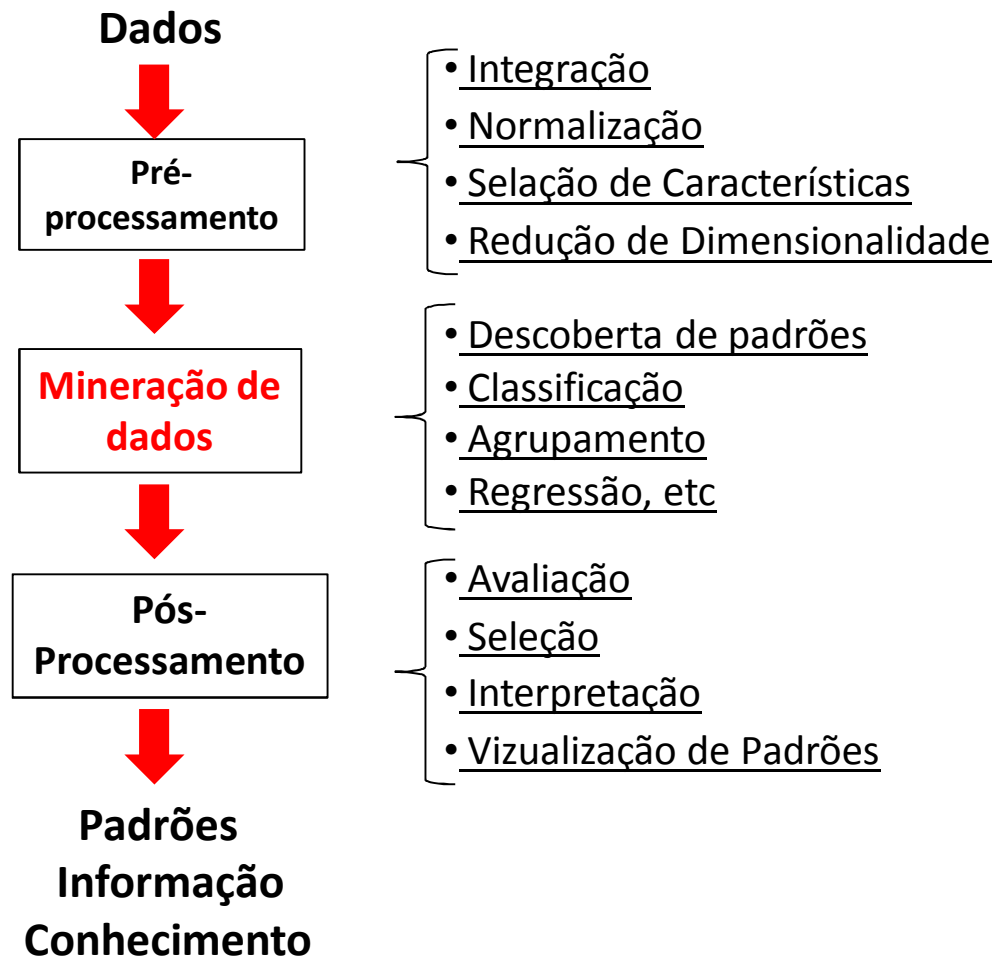
# O que é Mineração de Dados?

- Para a comunidade *business intelligence*



# O que é Mineração de Dados?

- Para as comunidades de aprendizado de máquina e estatística



# Mineração de Dados

- Por quê estudar?
  - Crescimento explosivo na quantidade de dados coletados e disponíveis
- De onde vêm esses dados?
  - Negócios: Web, comércio eletrônico, mercado de ações
  - Ciência: Bioinformática, sensoramento remoto, simulações
  - Sociedade: redes sociais, câmeras digitais, notícias



Search Trends

Tip: Use commas to compare multiple search terms.

Examples

[red hat, debian, gentoo, slackware](#)

[comic books, graphic novels](#)

[sudoku](#)

[mariners.com, seahawks.com, sonics.com](#)

[bbc.co.uk](#)

[weddingchannel.com](#)

### Hot Searches (USA)

1. [state of the union](#) 6. [maurice sendak](#)

2. [prince field](#)

3. [jorge posad](#)

4. [sotu](#)

5. [state of the](#)



Search Trends

Tip: Use commas to compare multiple search terms.

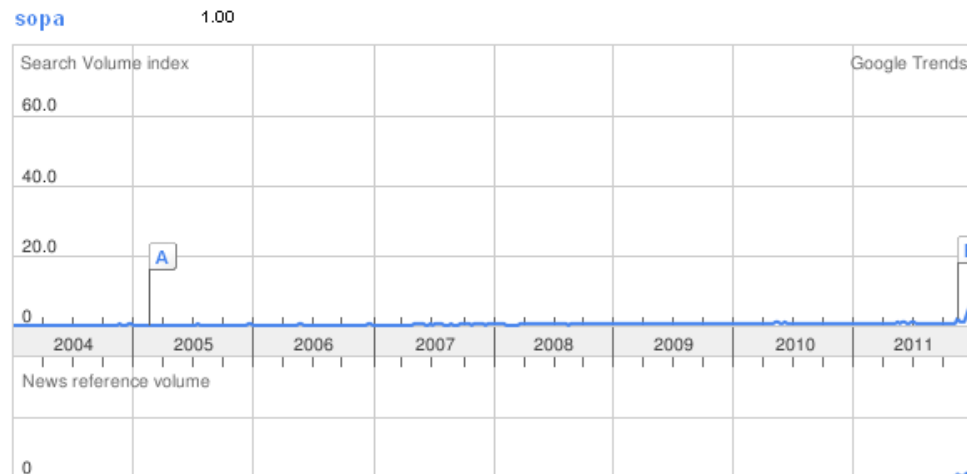
Searches Websites

All regions

All years

- Scale is based on the average worldwide traffic of [sopa](#) in all years. [Learn more](#)

- An improvement to our geographical assignment was applied retroactively from 1/1/2011. [Learn more](#)



**A** [SOPA DE FIDEO CON POLLO](#)

Kansas.com - Feb 23 2005

**B** [SOPA Drama: Google, Facebook Oppose Stop Online Piracy Act in Congress](#)

Slate - Nov 17 2011

**C** [SOPA and PIPA](#)

YouTube - Jan 18 2012

[More news results >](#)



A dengue é um dos principais problemas de saúde pública no mundo. A Organização Mundial da Saúde (OMS) estima que entre 50 a 100 milhões de pessoas se infectem anualmente, em mais de 100 países, de todos os continentes, exceto a Europa. Cerca de 550 mil doentes necessitam de hospitalização e 20 mil morrem em consequência da dengue.

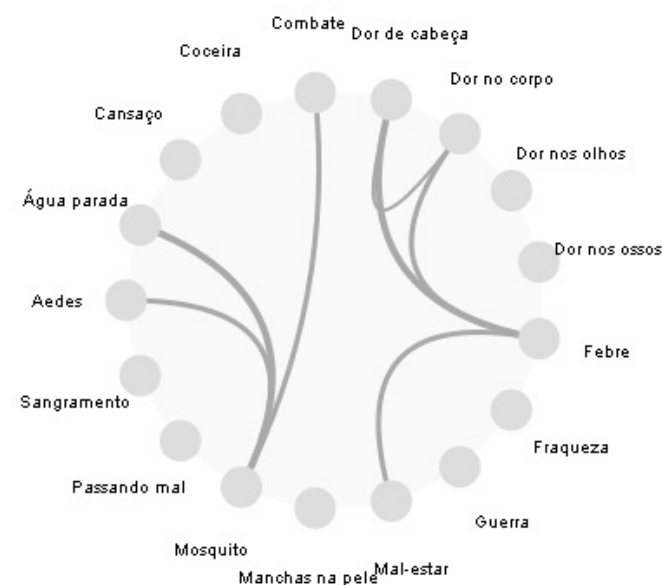
## Notícias do Observatório

### Nova versão do Observatório da Web - Monitoração da Dengue

O Observatório da Web - Monitoração da Dengue passou por algumas mudanças no seu layout para padronização com novos observatórios que serão lançados no futuro. **Leia mais**



Como esta análise é construída?



Como esta análise é construída?

## Destaques no Twitter (hoje)

### tweets veja mais



“ [Contigo!] Restart participa de campanha contra dengue em Rio Das Ostras:  
*Santa Maria, Rio Grande do Sul, BR*  
”



“ [Contigo!] Restart participa de campanha contra dengue em Rio Das Ostras: <http://t.co/jwMvQ48a>  
*Tatui, Sao Paulo, BR*  
”



# Sumário

- O que é mineração de dados?
- Visão Geral de Mineração de Dados
  - Tipo de conhecimento a ser extraído
  - Tipo dos dados a serem minerados
  - Tipos das técnicas a serem utilizadas
  - Aplicações consideradas
- Conhecendo seus dados
- Pré-processamento
- Classificação
- Avaliação dos Algoritmos de Classificação

# Visão Geral de Mineração de Dados

## 1. Tipo de conhecimento a ser minerado

- Mineração de padrões frequentes, Classificação, Agrupamento, etc
- Preditivo ou descritivo

## 2. Tipo dos dados a serem minerados

- Relacionais, sequências, temporais, espaciais, *streams*, textos, grafos, etc

## 3. Técnicas a serem utilizadas

- Estatística, aprendizado de máquina, visualização, reconhecimento de padrões, etc

## 4. Aplicações consideradas

- *Web mining*, bio-data mining, análise de fraudes, análise de tendências no mercado financeiro, etc

# Tipo de conhecimento a ser minerado

- **Mineração de padrões ou itens frequentes**
  - Que itens são frequentemente comprados juntos no Walmart?
- Tarefa de Associação
  - Fraldas -> cerveja [0.5%, 75%] (suporte, confiança)
- Problemas:
  - Como encontrar padrões de forma eficiente em grandes bases?
  - Como usar esses padrões nas tarefas de classificação e agrupamento?

# Tipo de conhecimento a ser minerado

- Aprendizado supervisionado
  - Rótulo das classes conhecido no conjunto de treinamento
- Classificação e Previsão
  - Construir modelos baseados em um conjunto de treinamento
  - Descrever ou distinguir classes para previsões futuras
- Métodos comuns
  - Árvores de decisão, regras de associação, redes Bayesianas, SVM, regressão logística, redes neurais, algoritmos evolucionários, etc

# Tipo de conhecimento a ser minerado

- Aprendizado não-supervisionado
  - Agrupamento
- Agrupar dados similares criando categorias (ou grupos)
- Princípio: maximizar a similaridade inter-categoria e minimizar a similaridade intra-categoria

# Tipo dos dados a serem minerados

- Bases de dados relacionais ou transacionais
- Bases de dados avançadas:
  - Multimídia - Imagens ou vídeos
  - Séries temporais
  - Dados espaciais ou espaço-temporais
  - Texto
  - Web
- Bases de dados representadas por grafos
  - Compostos químicos e redes sociais

# Maiores desafios da área

- Mineração dados heterogêneos
- Lidar com dados de alta dimensão
- Lidar com dados incompletos, incertos e com ruído
- Incorporar exceções e conhecimento a priori sobre os problemas sendo resolvidos
- Eficiência e escalabilidade dos algoritmos

# Conhecendo seus Dados: Foco em Dados Estruturados



# Instâncias

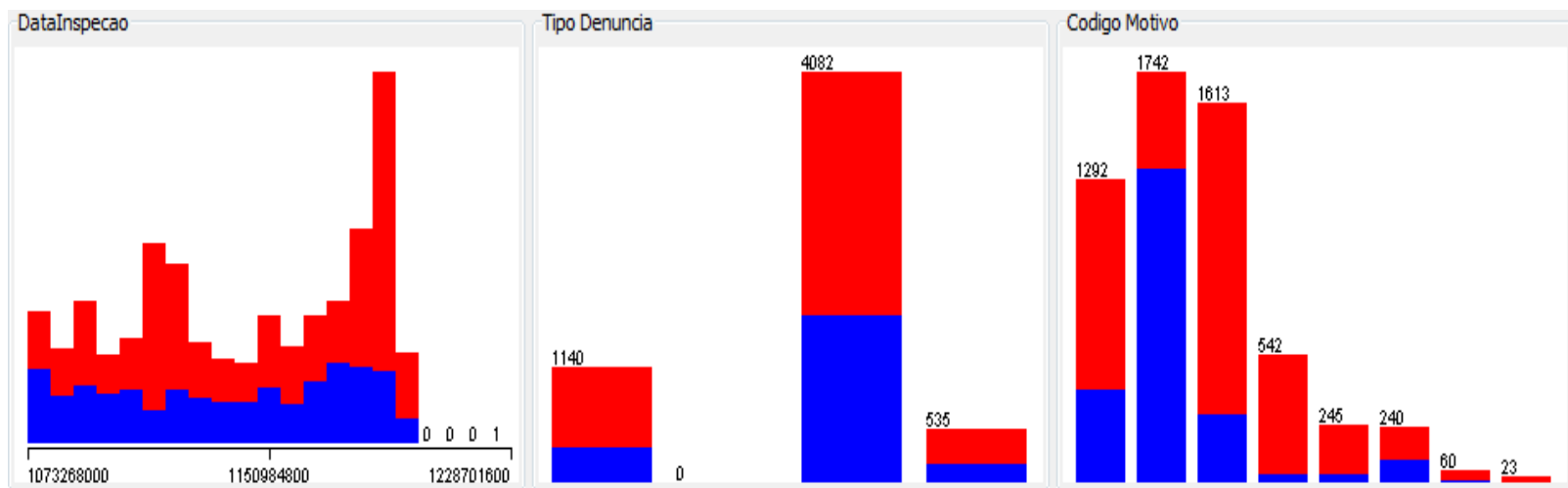
- Bases de dados são compostas por instâncias
- Uma instância representa uma entidade no mundo real
  - Ex: consumidores, pacientes, proteínas
- Instâncias são descritas por um conjunto de atributos
- Numa base de dados, linhas representam instâncias e colunas atributos

# Atributos

- Também conhecidos como dimensões, características, variáveis
  - Ex: nome, endereço, telefone
- Tipos de atributos:
  - **Nominais**: categorias, estados
    - Ex: cor do cabelo: {loiro, ruivo, preto, branco, castanho}
  - **Ordinais** : a ordem importa, mas o valor que representa cada categoria não é conhecido
    - Ex: altura: {baixo, médio, alto, muito alto}
  - **Numéricos**
    - Ex: salário, temperatura
- Atributo discreto (número finito de estados) versus contínuo (normalmente representado por um número real)

# Estatísticas básicas sobre os dados

- Tendência central, variação e espalhamento (*spread*)
- Dispersão dos dados
  - Média, mínimo e máximo
  - Exemplo: problema de 2 classes



Pré-Processamento:  
Visa aumentar a qualidade dos  
dados

# Principais tarefas de pré-processamento

- **Limpeza** de dados
  - Tratar dados faltantes (*missing values*), resolver inconsistências, identificar e remover *outliers*
- **Integração** de dados
- **Redução** de dados
  - Redução de dimensionalidade, numerosidade e compressão de dados
- **Transformação e discretização** de dados
  - Normalização de dados

# Redução de Dados

- Obter uma representação reduzida dos dados que, quando analisada, leve aos mesmos resultados obtidos com os dados completos
- Por quê?
  - Bases de dados são normalmente imensas, acarretando alto custo computacional
- Estratégias:
  - Redução de dimensionalidade
  - Redução de dados (numerosidade)
  - Compressão de dados

# Redução de Dimensionalidade

- Evita a maldição da dimensionalidade
  - Quanto mais dados, mais esparsa a base de dados e mais difícil de aprender
- Ajuda a reduzir o número de atributos irrelevantes e remover ruído
- Reduz o tempo necessário para a mineração
- Facilita a visualização dos dados

# Redução de Dimensionalidade

- Técnicas:
  - Transformadas *Wavelet*
  - PCA (*Principal Component Analysis*)
  - Métodos supervisionados e não lineares, como seleção de atributos
- Seleção de atributos
  - Dados normalmente não são criados para serem minerados
  - Remove atributos irrelevantes para o processo, tais como identificadores



# Seleção de Atributos

- Dados  $K$  atributos, existem  $2^K$  combinações possíveis
- Utilização de heurísticas para seleção:
  - O melhor atributo é selecionado através de testes de significância assumindo independência entre eles
  - Seleção *Greedy*
    - Seleciona o melhor atributo
    - Seleciona o segundo melhor condicionado ao primeiro, e assim por diante
  - Eliminação *Greedy*
    - Elimina o pior atributo a cada iteração
  - Seleção baseada em busca em largura
    - Seleciona sempre os  $n$  melhores, ao invés do melhor
  - Algoritmos evolucionários

# Referências e Links

- Hornby et al, Automated antenna design with evolutionary algorithms, AIAA Space , 2006
- M. Mitchell, Genetic Algorithms: An Overview, Complexity, 1 (1) 31-39, 1995.
- A Field Guide to Genetic Programming, Livro online, <http://www.gp-field-guide.org.uk/>

<http://www.aridolan.com/ga/gaa/SingleVarMin.html> –  
minimizar a função

<http://alphard.ethz.ch/gerber/approx/default.html>

<http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>