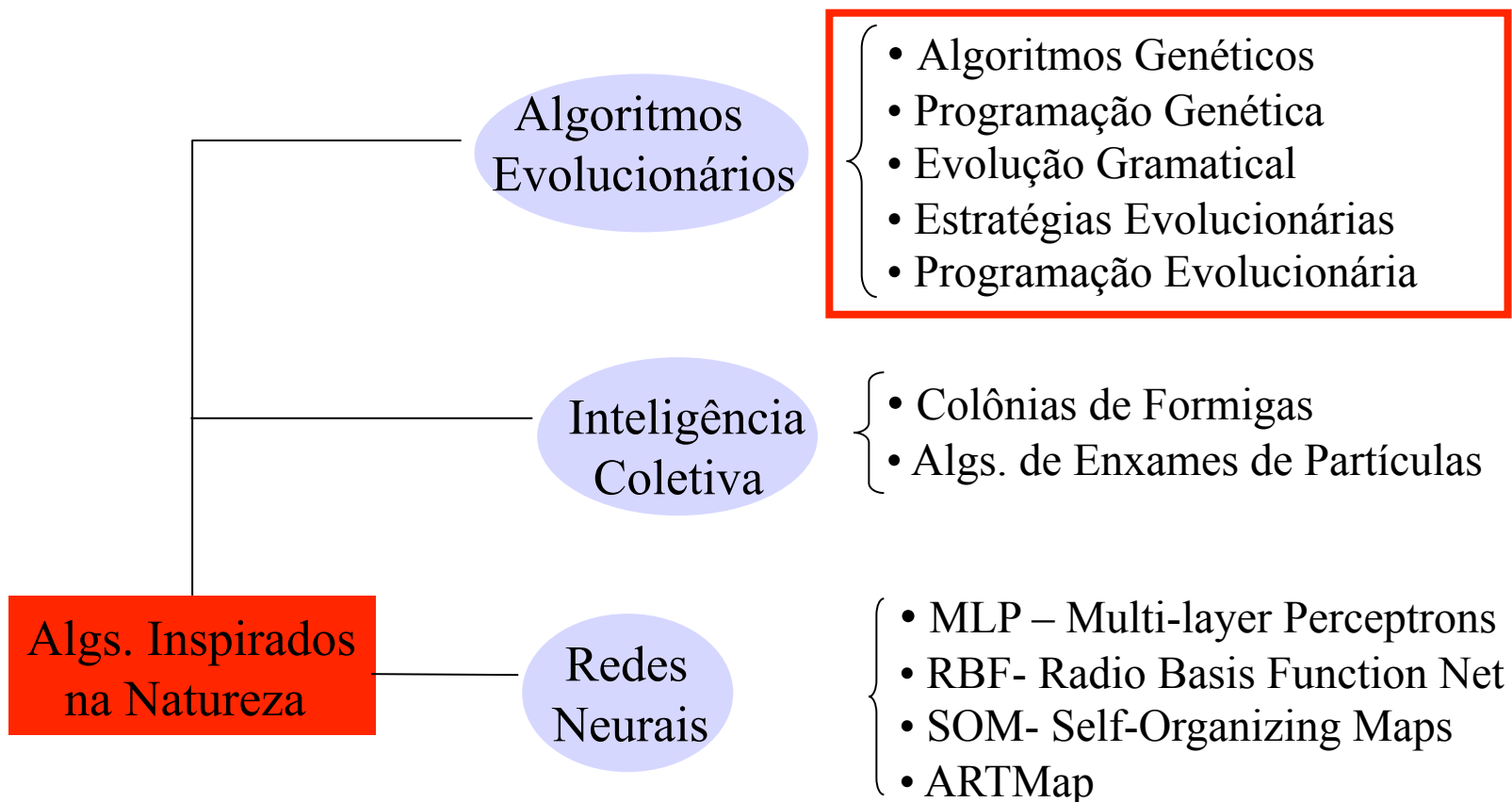
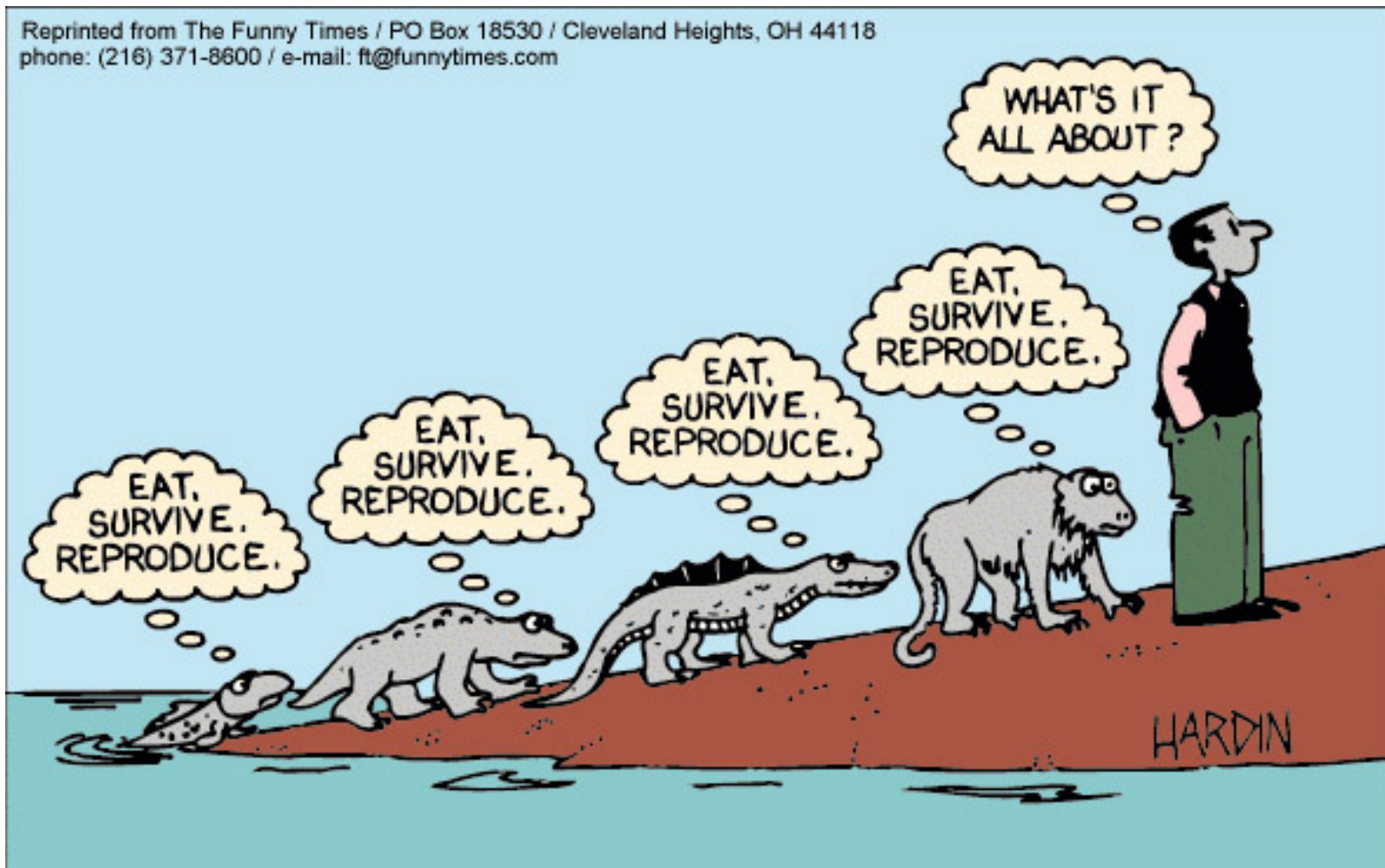


# Computação Evolucionária

Gisele L. Pappa



# Evolução



# Introdução

- 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

# Introdução

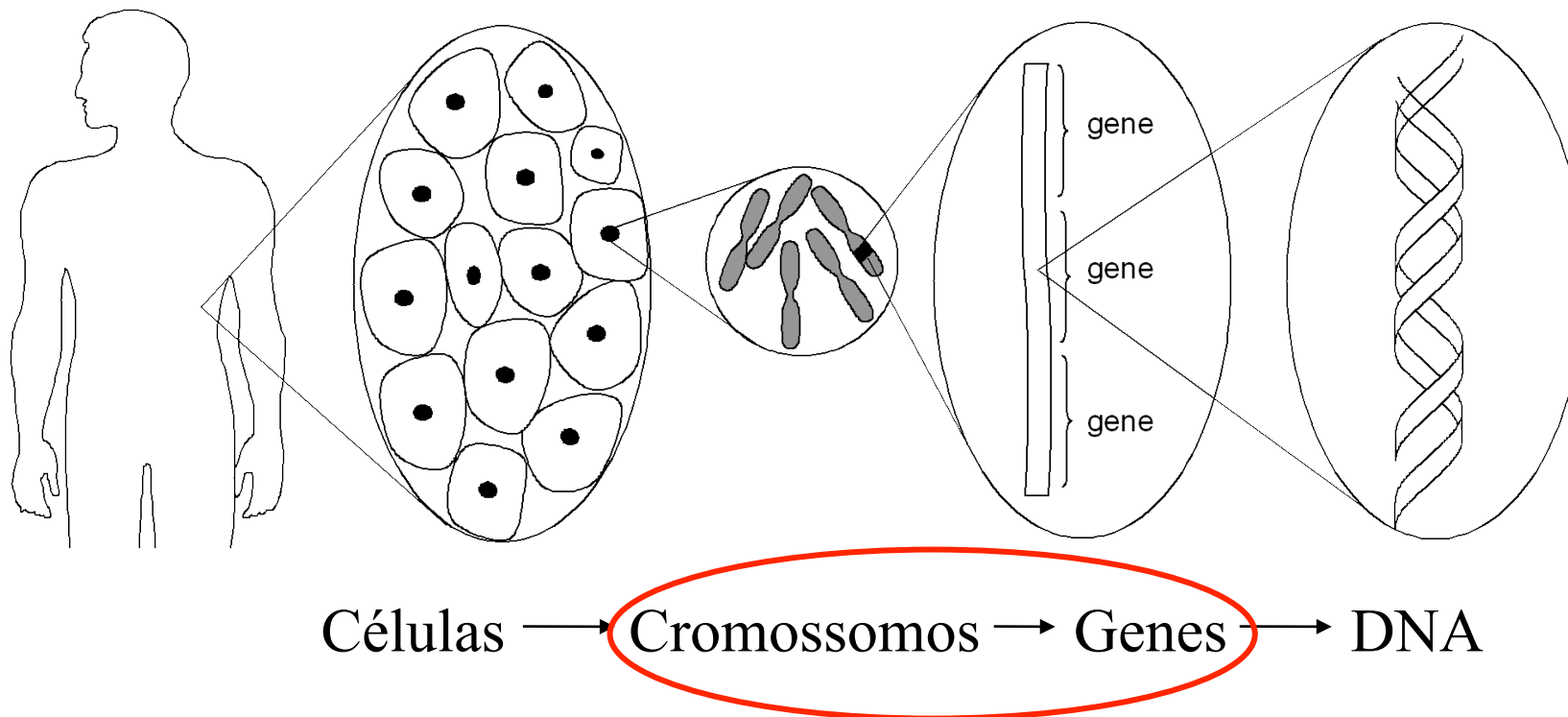
- Historicamente, diferentes tipos de AEs têm sido associadas a diferentes tipos de representação
  - Vetores binários : Algoritmos Genéticos
  - Vetores de números reais : Estratégias Evolucionárias
  - Máquinas de estado finito: Programação Evolucionária
  - Árvores: Programação Genética
- Essas diferenças são irrelevantes, sendo a melhor estratégia
  - Escolher uma representação adequada para o problema
  - Escolher operadores genéticos que sejam adequados a representação escolhida

# Biologia Evolucionária (1)

- 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

# Biologia Evolucionária (2)

- Estruturas básicas:



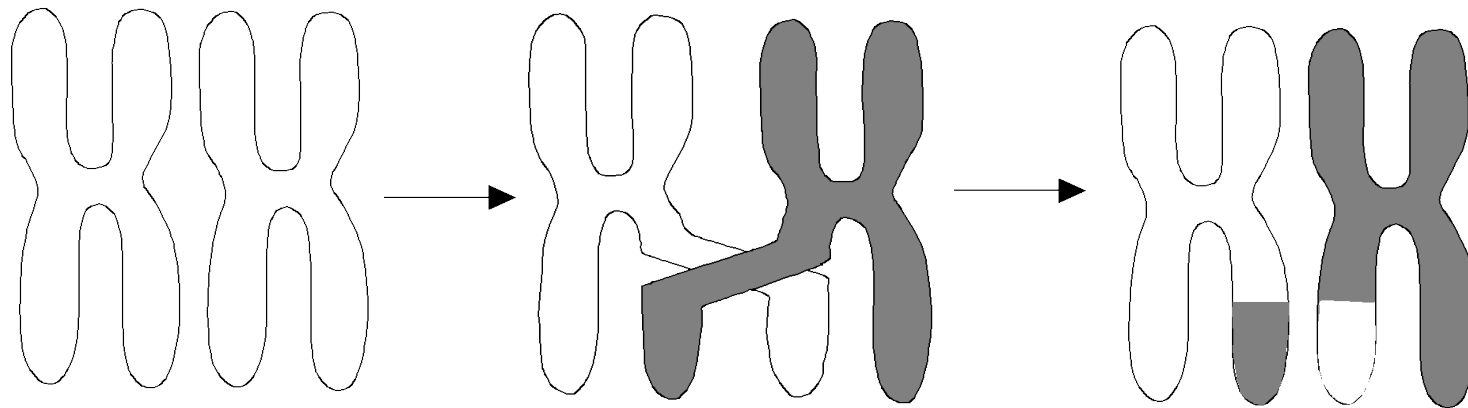
# Biologia Evolucionária (3)

- **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



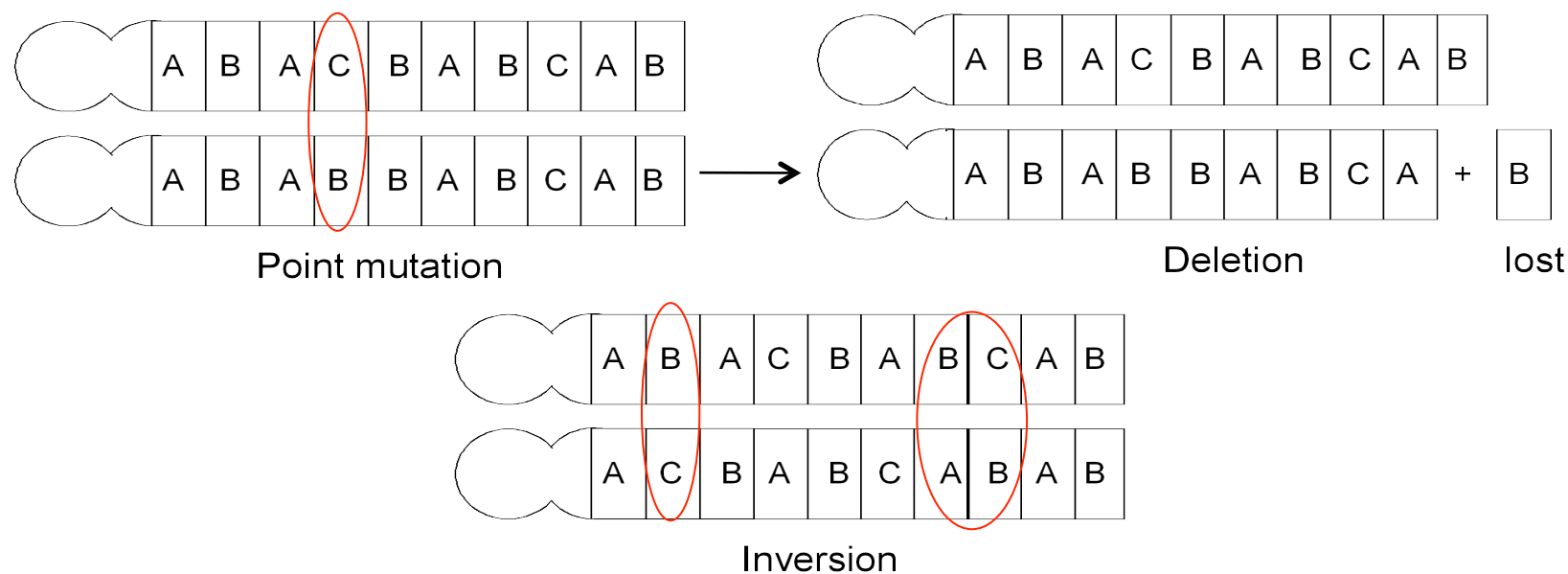
# Biologia Evolucionária (4)

- Variação genética
  - **Cruzamento**: troca de material genético entre dois cromossomos

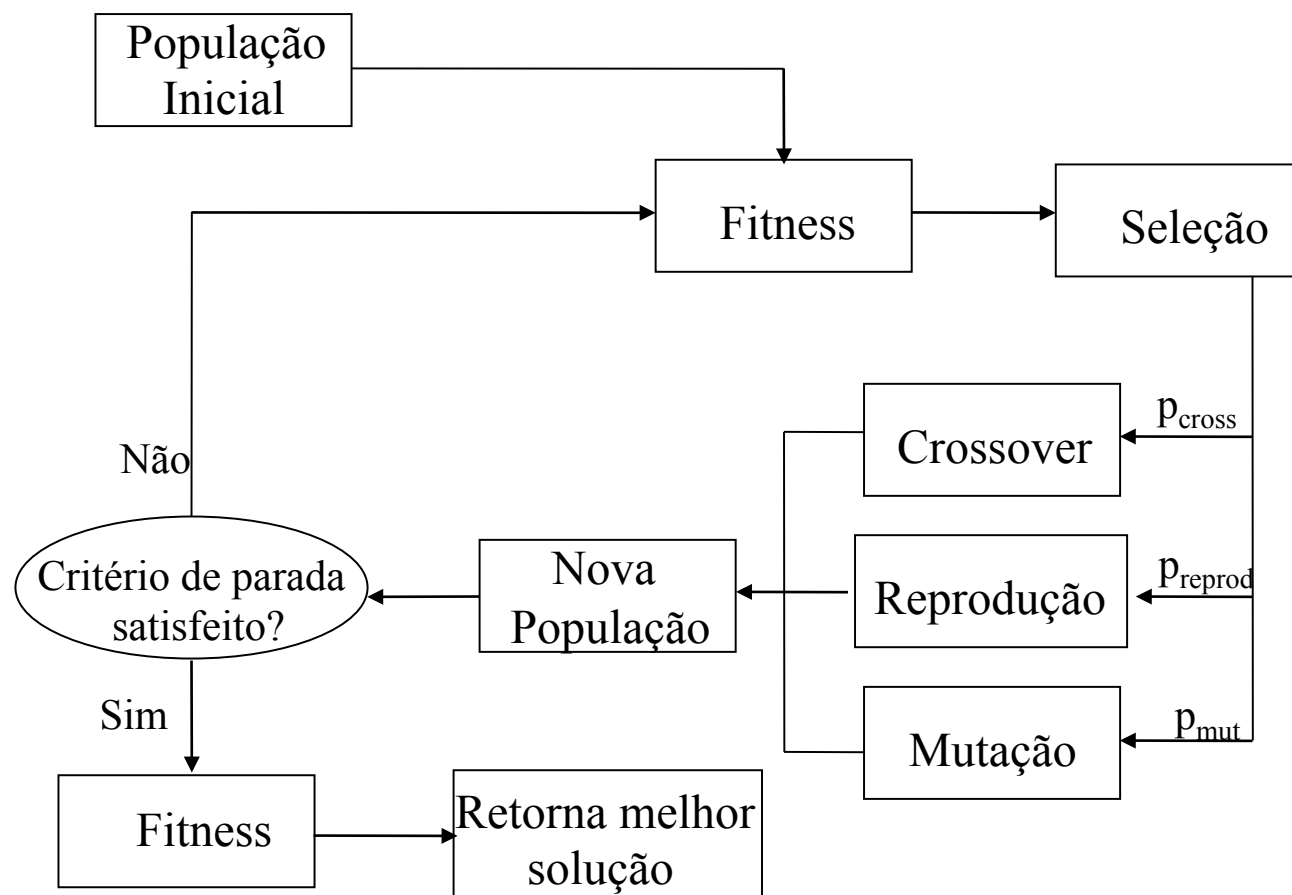


# Biologia Evolucionária (5)

- Variação genética
  - Mutação



# 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
- A cada iteração (geração):
  - Os melhores indivíduos são selecionados de acordo com uma função de aptidão (fitness)
  - Operadores genéticos são aplicados aos indivíduos selecionados, visando produzir novos indivíduos (“filhos”)
  - Novos indivíduos são avaliados utilizando a função de aptidão

# Exemplos de Aplicação

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

# Motivação

- Maneira declarativa de resolver um problema (o que fazer), em contraste com métodos procedurais (como fazer)
- Processamento paralelo
- Oferecem soluções robustas e adaptativas
- Necessidade de validar teorias e conceitos da biologia evolucionária
- Inteligência de máquinas
  - Possibilita incorporar conhecimento ao método sem explicitamente programá-lo
- Evolução automática de códigos de programas
- Busca global

# 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

# Algoritmos Genéticos

- Versão original têm 2 características principais:
  - usa uma representação binária e de tamanho fixo
  - Forte uso de cruzamento
- Desenvolveram-se muitas teoria a partir da versão original – teoria dos esquemas
  - Gera a noção de *building blocks*
- Com o passar to tempo, outras representações foram propostas.
  - Representação real é muito utilizada hoje em dia



# Estratégia Evolucionária

- Versão original:
  - Representada por vetores de números reais
  - Único indivíduo
  - Dirigida por mutações -> variações pequenas são melhores que variações grandes
- Durante seu processo de desenvolvimento:
  - Populacional – usa um esquema de *overproduction*
    - Seleção considera tanto os pais quanto os filhos
  - Reconheceu o uso de operadores de recombinação
  - Em várias versões, inclui valores de parâmetros nos indivíduos ex: taxa de mutação

# Programação Evolucionária

- Representado por máquinas de estado finito
- Principal operador é a mutação

# Programação Evolucionária

- Representado por máquinas de estado finito
- Principal operador é a mutação

# Programação Genética

- Representação original baseada em árvores e de tamanho variável
- Ideia era evoluir programas automaticamente
- Versões mais recentes utilizam genomas lineares e grafos

# Problema do Caxeiro Viajante (TSP)

- Problema de otimização combinatorial

# Espaço de Busca

- Definido pelo número de soluções que podem ser geradas
- No caso do TSP:
  - Número de permutações de  $n$  cidades
- Filtro:
  - Exclua delas as que não começam na cidade desejada
- Por onde começar?

# Problema do Caxeiro Viajante

- **Representação dos indivíduos**
  - Um vetor de inteiros ou caracteres, representando uma permutação entre cidades:  $I_1$  A F C E D B G
- **Função de fitness**
  - Tamanho do caminho percorrido ( função a ser minimizada)
  - Função objetiva
- **Operadores genéticos**
  - Deve-se considerar restrições do problema ?
  - Passar apenas uma vez por cada cidade?

# Problema do Caxeiro Viajante

- Pseudocódigo:

1. Gerar permutações de cidades (soluções) aleatoriamente
2. Avaliar cada uma delas, calculando o tamanho do caminho percorrido

REPEAT

3. Selecionar as melhores permutações para reprodução (aquelas com caminho menor)
  4. Criar novas permutações combinando partes das melhores soluções
  5. Avaliar novas permutações
- UNTIL (boa solução encontrada ou número máximo de gerações encontrado)

# Problema do Caxeiro Viajante

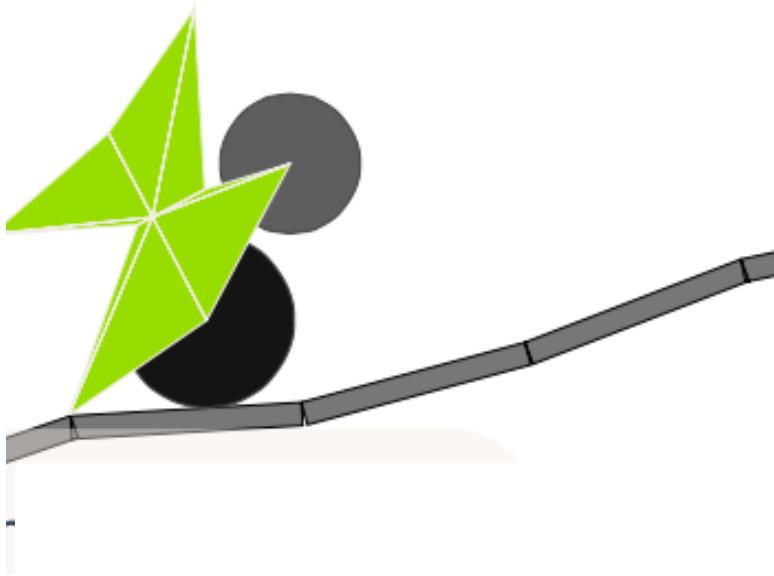
- Cruzamento

- Reconhecer uma solução inválida e substituir a cidade repetida pela ignorada

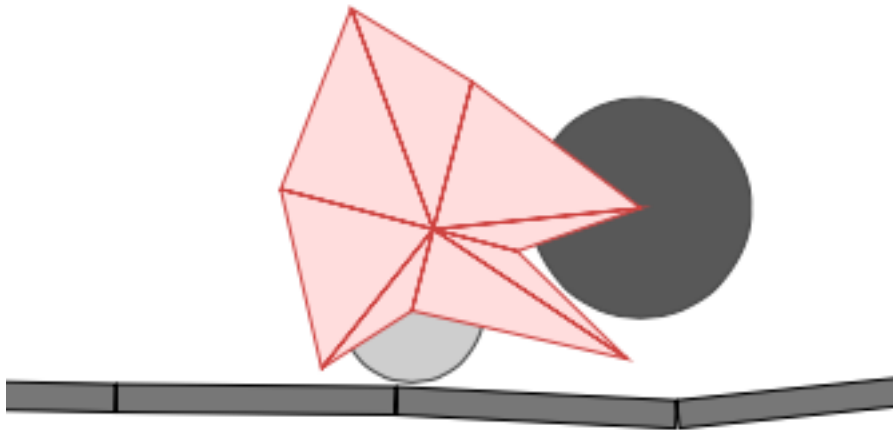
$I_1$  A B C D E F G     $C_1$  A B C D E G B  $\rightarrow$  A B C D E G F  
 $I_2$  C A D F E G B     $C_2$  C A D F E F G  $\rightarrow$  C A D F E B G

- Primeira população não tem que ser necessariamente criada aleatoriamente
  - Podemos usar um método guloso
- "A B C D E F G" é igual a "G F E D C B A"





<http://gencar.co>

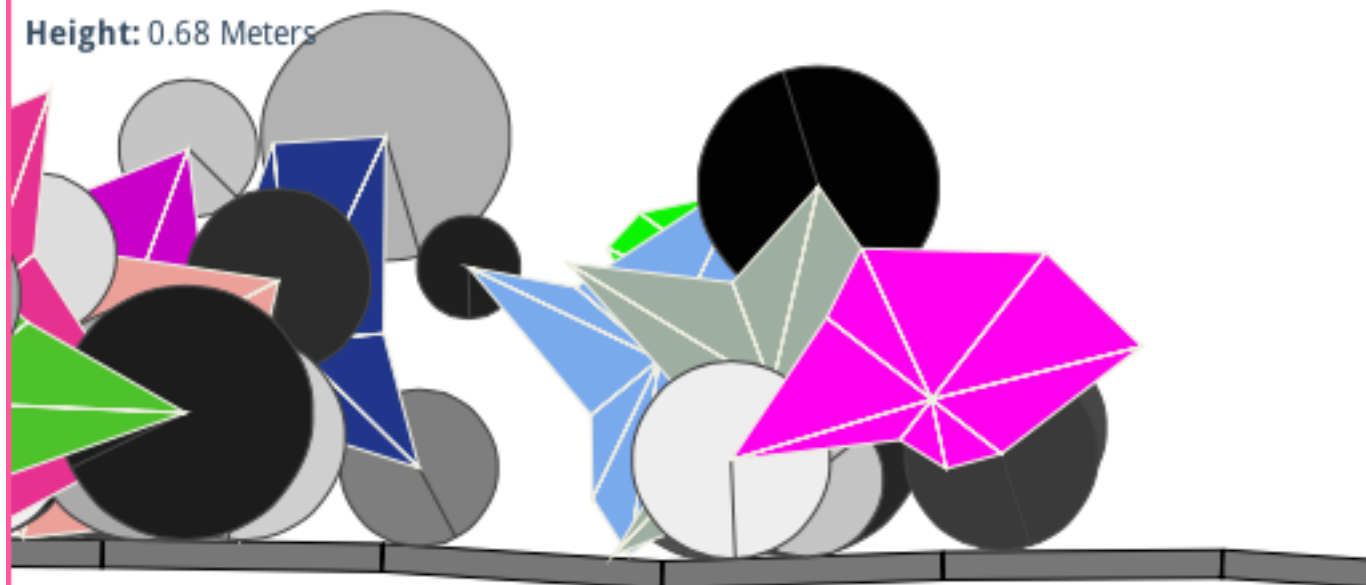


**Generations:** 0

**Cars Alive:** 20

**Distance:** 7.1 Meters

**Height:** 0.68 Meters



World Champion Cars: **0**

<http://gencar.co>

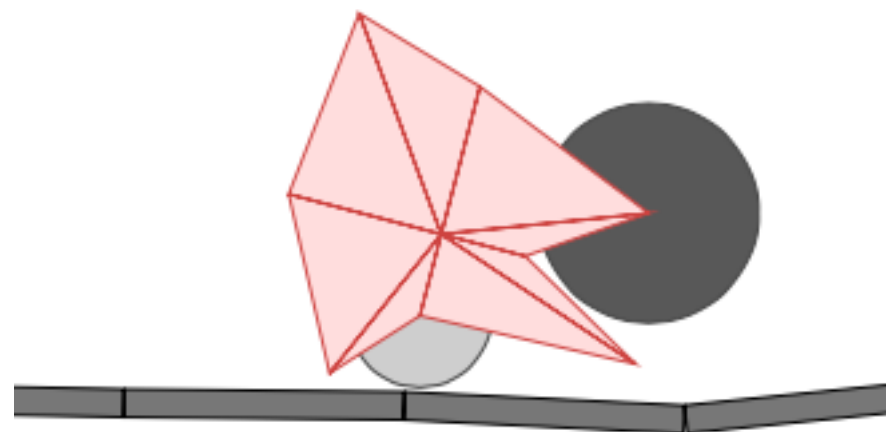
Cromossomo está dividido em 4 partes:

Forma ( 8 genes, um por vértice)

Tamanho da roda (2 genes)

Posição da roda (2 genes)

Densidade da roda (2 genes)



# Função de Fitness

Pode ser testado contra um grupo fixo de pistas  
(oponentes)

Difícil escolher o nível de dificuldade correto

Pode ser testado contra um grupo adaptativo de  
agentes

Co-evolução

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

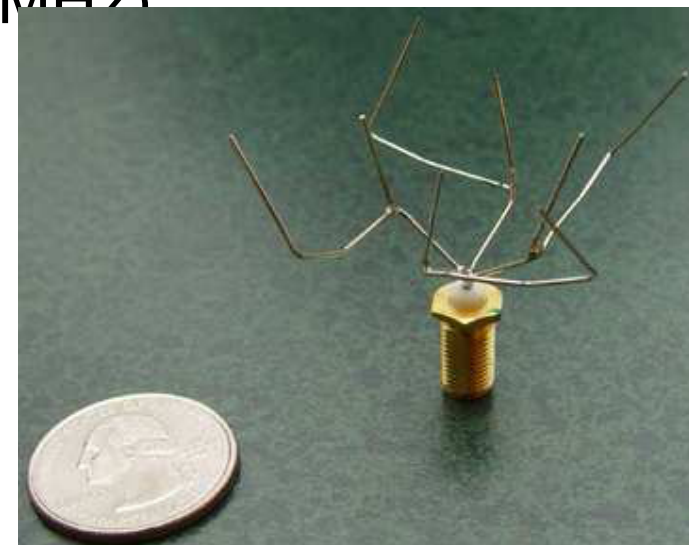
- NASA vem utilizando AEs para evoluir antenas (design) com as mais diversas finalidades desde 1990.
- Exemplo: antenas para o programa ST5, que visa lançar 3 micro-satélites (53cm x 28cm e 25kg)
- Cada satélite possui 2 antenas, centralizadas nas partes de cima e de baixo
- Desenvolveram um EA para 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 (7209.125 MHz)

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)

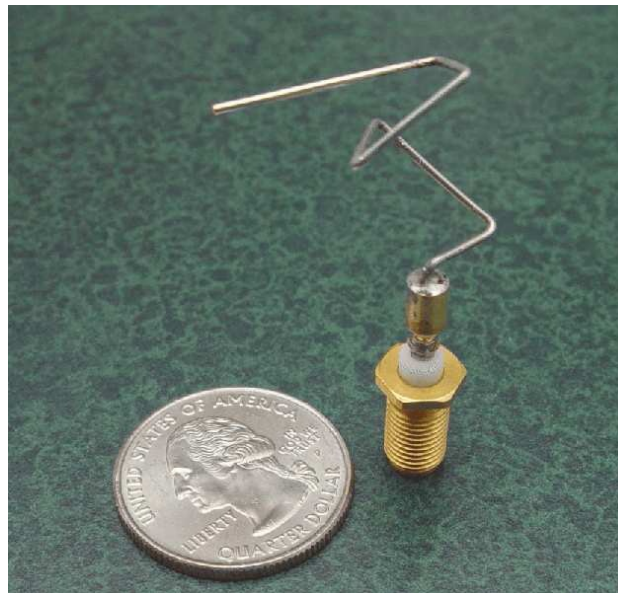
# Design de antenas para missões da NASA

- 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

# Agradecimento

- Alguns desses slides foram retirados do livro “Fundamentals of Natural Computing”, de Leandro Nunes de Castro e outros das aulas de computação natural de Alex A. Freitas

## Referências

- [Hornby et al 2006], Automated antenna design with evolutionary algorithms, AIAA Space 2006, disponível no LearnLoop

## Leitura Recomendada

- Genetic Algorithms: An Overview, M. Mitchell, Complexity, 1 (1) 31-39, 1995.

