Computação Evolucionária e Algoritmos Genéticos

Sabedoria da natureza aplicada à computação

Eduardo J. Spinosa

Adaptado por Huei Diana Lee e Newton Spolaôr

Motivação

- "...Se **variações úteis** para qualquer organismo devem ocorrer para que ele venha a existir, certamente indivíduos assim caracterizados terão a **melhor chance** de serem preservados na luta por sobrevivência; e do forte princípio de **hereditariedade**, eles tenderão a produzir gerações com características similares. Este princípio de preservação, eu batizei, para ser sucinto, de **Seleção Natural**."

Charles Darwin, 1859

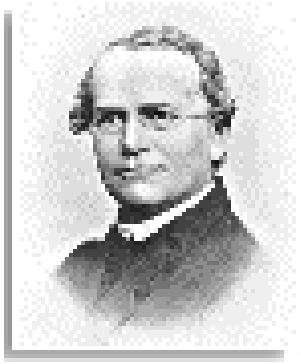
Teoria da Evolução

1859 - Charles Darwin publica o livro "*A Origem das Espécies*":

Charles Darwin

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

Teoria da Evolução



Gregor Mendel

- 1865- Gregor Mendel, "pai da genética", apresenta experimentos do cruzamento genético de ervilhas
  - Leis da hereditariedade

- A Teoria da Evolução começou a partir da conceituação integrada da Seleção Natural com a Genética

Algoritmos Genéticos – AG

- São técnicas de busca e otimização
- É a metáfora da Teoria da Evolução das espécies iniciada pelo Fisiologista e Naturalista inglês Charles Darwin
- Desenvolvido por John Holland (1975) e seus alunos (Livro: *Adaptation in Natural and Artificial Systems*)
- Popularizado por David Goldberg (1989)

Otimização

- É a busca da melhor solução para um dado problema
  - Consiste em tentar várias soluções e usar a informação obtida para conseguir soluções cada vez melhores
- Exemplo de otimização:
  - Parâmetro que deve ser maximizado, buscando várias soluções alternativas até alcançar uma satisfatória

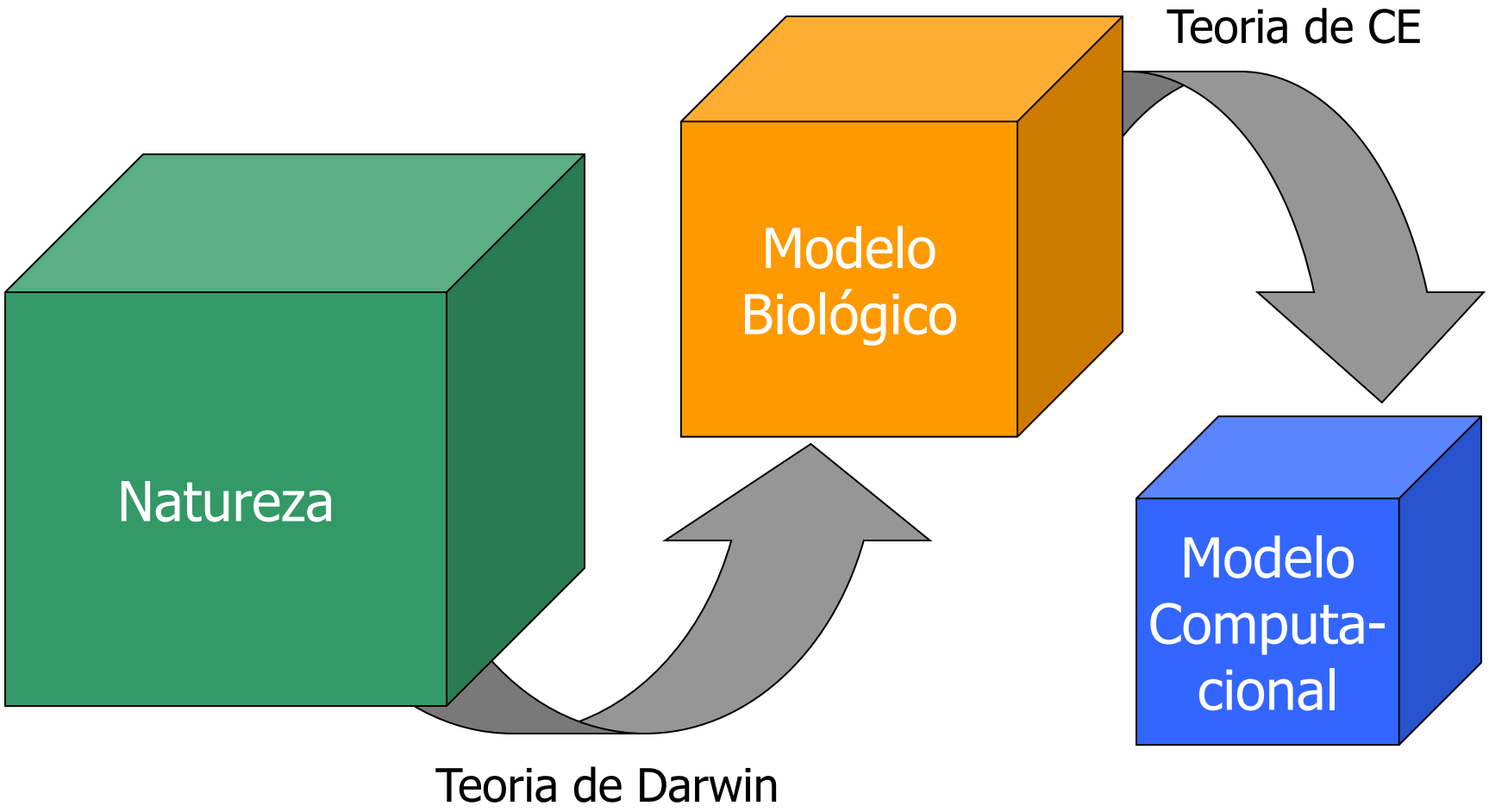
Otimização

- As técnicas de otimização, geralmente, apresentam:
  - **Espaço de busca**: onde estão todas as possíveis soluções do problema
  - **Função objetivo**: utilizada para avaliar as soluções produzidas, associando a cada uma delas uma “nota” ou “qualidade” da solução

Computação Evolucionária – CE

- Área da IA que engloba um conjunto de métodos computacionais inspirados na Teoria da Evolução das espécies:
  - Estratégias Evolucionárias
  - ...
  - **Algoritmos Genéticos**
- Aplicação da **Seleção Natural** na computação: *os indivíduos (possíveis soluções) mais adaptados ao meio (problema) têm maior chance de sobrevivência*

Simplificações



Elementos Básicos de AG

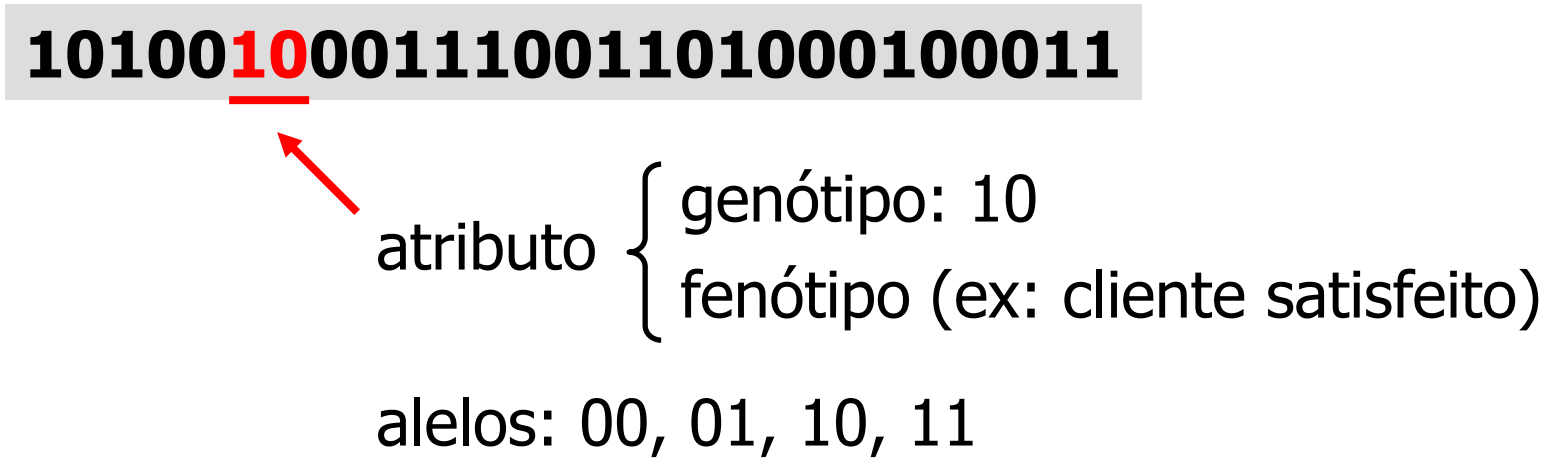
- Definição de um problema
  - Restrições ao espaço de busca
  - Ambiente – domínio do problema
- Indivíduo
  - Formas de representação
- População
- Função de Aptidão/Objetivo
  - Avaliação
  - Seleção
- Procriação
  - Herança genética

Aplicações

- Atividades
  - Otimização
  - Busca
- Áreas
  - Indústria (processos)
  - Planejamento (grade de horários, rota de veículos)
  - Mineração de dados (técnicas como redes neurais)
  - Processamento de imagem (médica, de satélite)
  - Processamento e reconhecimento de linguagem
  - Economia

Indivíduos

- Conjunto de **atributos** que formam uma **possível solução**
- Estrutura dos Indivíduos em AG:



Conceitos da Biologia
Equivalente em AG

- **Gene**
  Unidade básica do Ácido Desoxirribonucleico (ADN)
  1 bit
- **Locus**
  Posição de um gene no ADN
  Posição do bit na palavra

Conceitos da Biologia
Equivalente em AG

- **Alelo**
  Configuração possível do(s) gene(s)
  0, 1, 00, 01, 10, 11, 001, 010, ...

Conceitos da Biologia
Equivalente em AG

- **Genótipo**
  Conjunto de alelos que o indivíduo possui e que determinam atributos observáveis
  Seqüência de bits que determina um indivíduo
- **Fenótipo**
  Atributos observáveis em um indivíduo (cor, tamanho, forma...)
  Significado do genótipo

Conceitos da Biologia
Equivalente em AG

- **Genoma**
  Indivíduo, representado por uma sequência completa de ADN
  Indivíduo
- **Cromossomo**
  Uma unidade do genoma que contém uma subsequência de ADN
  Indivíduo

População

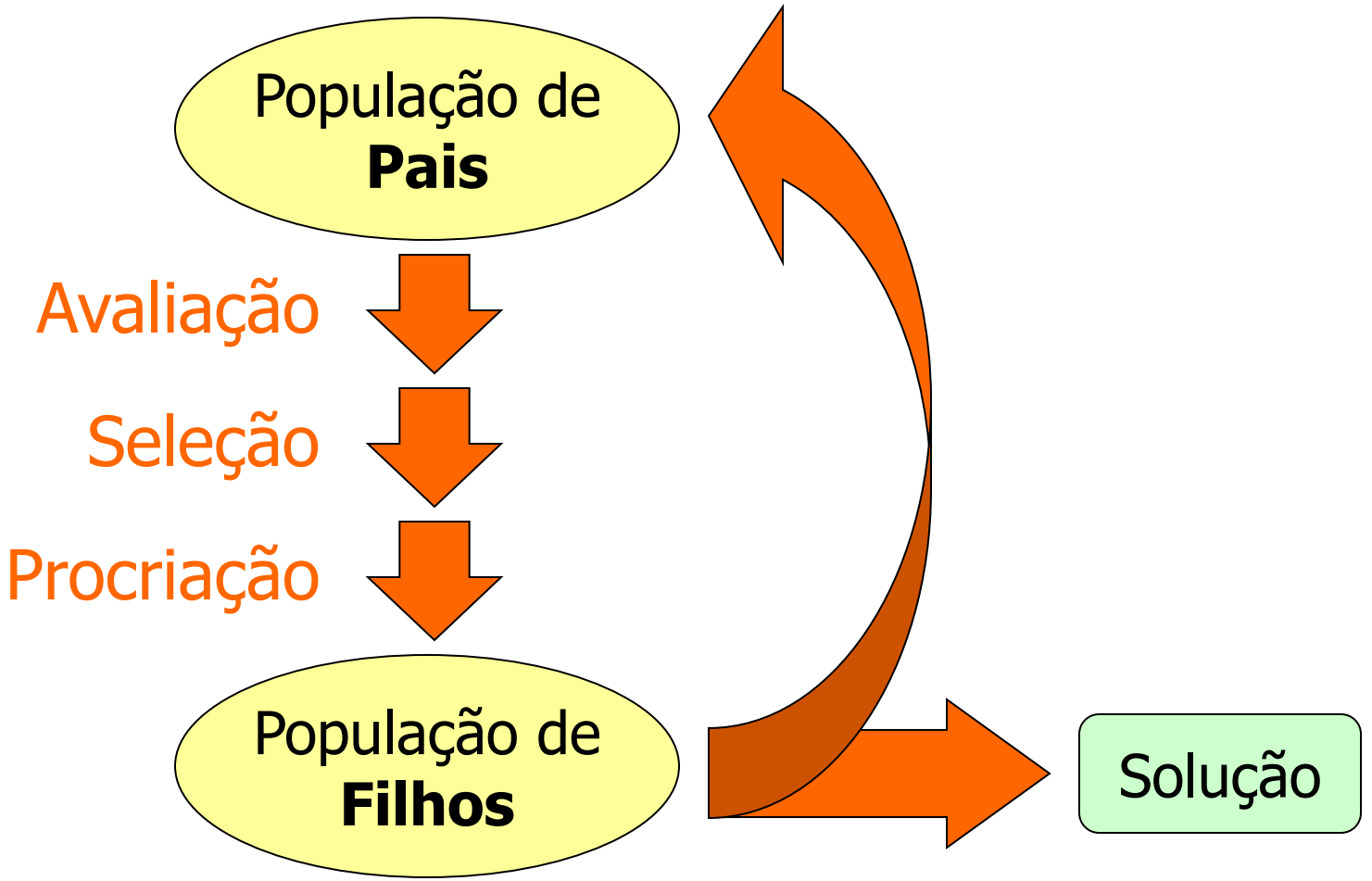
- Conjunto de indivíduos
- Número de pontos de exploração no espaço de busca
- **Tamanho** da população é importante
  - Pequeno demais » Perda da diversidade
  - Grande demais » Custo computacional
- **Representatividade**

Visão Geral do Algoritmo

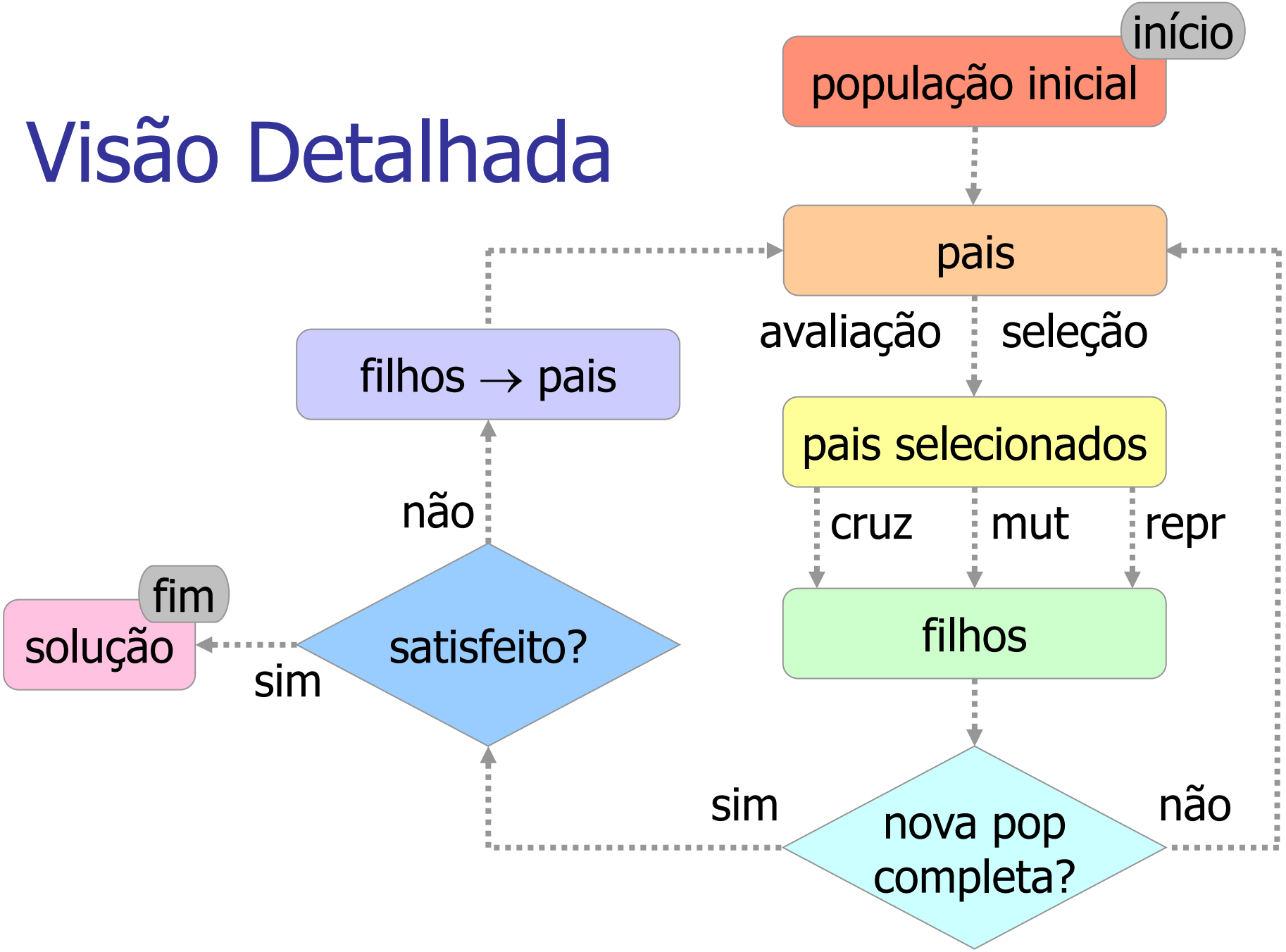
- Gerar uma população inicial **aleatoriamente**, em geral
- Enquanto não atingir o critério de término
  - Avaliar indivíduos
  - Selecionar indivíduos para serem pais
  - Produzir filhos
- Retornar uma solução

Nota: relação entre algoritmo e natureza

Visão Geral do Algoritmo



Visão Detalhada



Criação da População Inicial

- Em AG, indivíduos têm o mesmo tamanho
- Restrições?
- Muito importante: **Diversidade**

Fase de Avaliação

- Estabelece quais indivíduos são adequados ao problema
- Bom? Ruim? » **Adequado** ao problema
- Cálculo do valor de **aptidão** de todos os indivíduos da população

Função de Aptidão (*Fitness*)

- Mede o grau de **adaptação** do indivíduo ao meio
- Função heurística que **guia** a busca
- **Específica** para cada problema
- Muito complexa » Custo computacional
- Estabelece uma forma de se diferenciar os melhores dos piores, servindo como a **força mestre** do processo evolutivo

(Gritz, M., 1993, "The Impact of Training on the Frequency and Duration of Employment," *Journal of Econometrics* 57, 21-51)

Fase de Seleção

- Método de Seleção
  - Ordem (*Ranking*)
  - Roleta
  - Torneio

Seleção por Ordem

- Indivíduos são ordenados de acordo com a **aptidão**
- Os melhores são escolhidos
- Simples demais
- **Baixa diversidade**

Nota: exemplo da otimização de parâmetro, partindo de população inicial aleatória

Seleção por Roleta

- Calcula-se **R** (tamanho da roleta): soma da aptidão de todos os indivíduos
- Cada indivíduo ocupa uma porção da roleta proporcional à sua aptidão
- Sorteia-se um valor de 1 a R
- O indivíduo que estiver na posição referente a esse valor é selecionado

Seleção por Roleta

- Ex: população com 5 indivíduos, aptidão igual à quantidade de bits com valor 1

Indivíduo	Aptidão	Aptidão %
10011101	5	
11111111	8	
10000001	2	
11101011	6	
10011010	4	
Total	25	

Seleção por Roleta

- Ex: população com 5 indivíduos, aptidão igual à quantidade de bits com valor 1

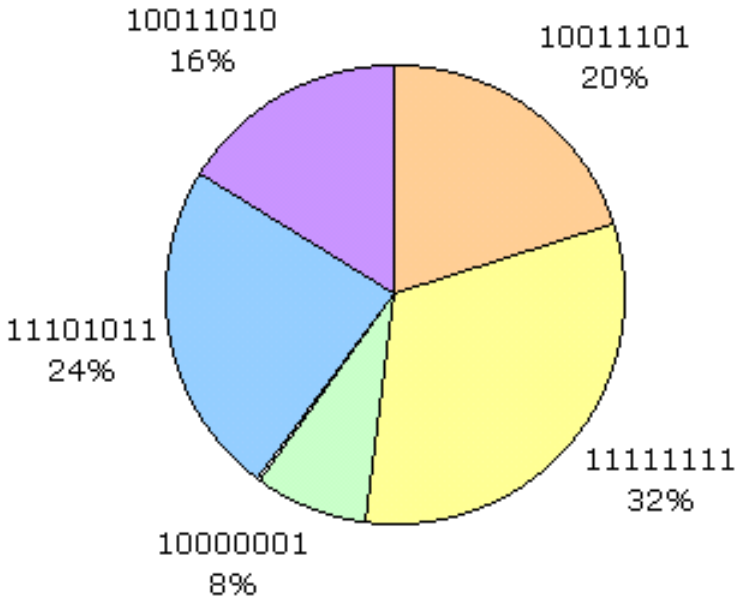
Indivíduo	Aptidão	Aptidão %
10011101	5	20
11111111	8	32
10000001	2	8
11101011	6	24
10011010	4	16
Total	25	100

A transformação dos valores de aptidão facilita a roleta

Seleção por Roleta

- Ex: população com 5 indivíduos, aptidão igual à quantidade de bits com valor 1

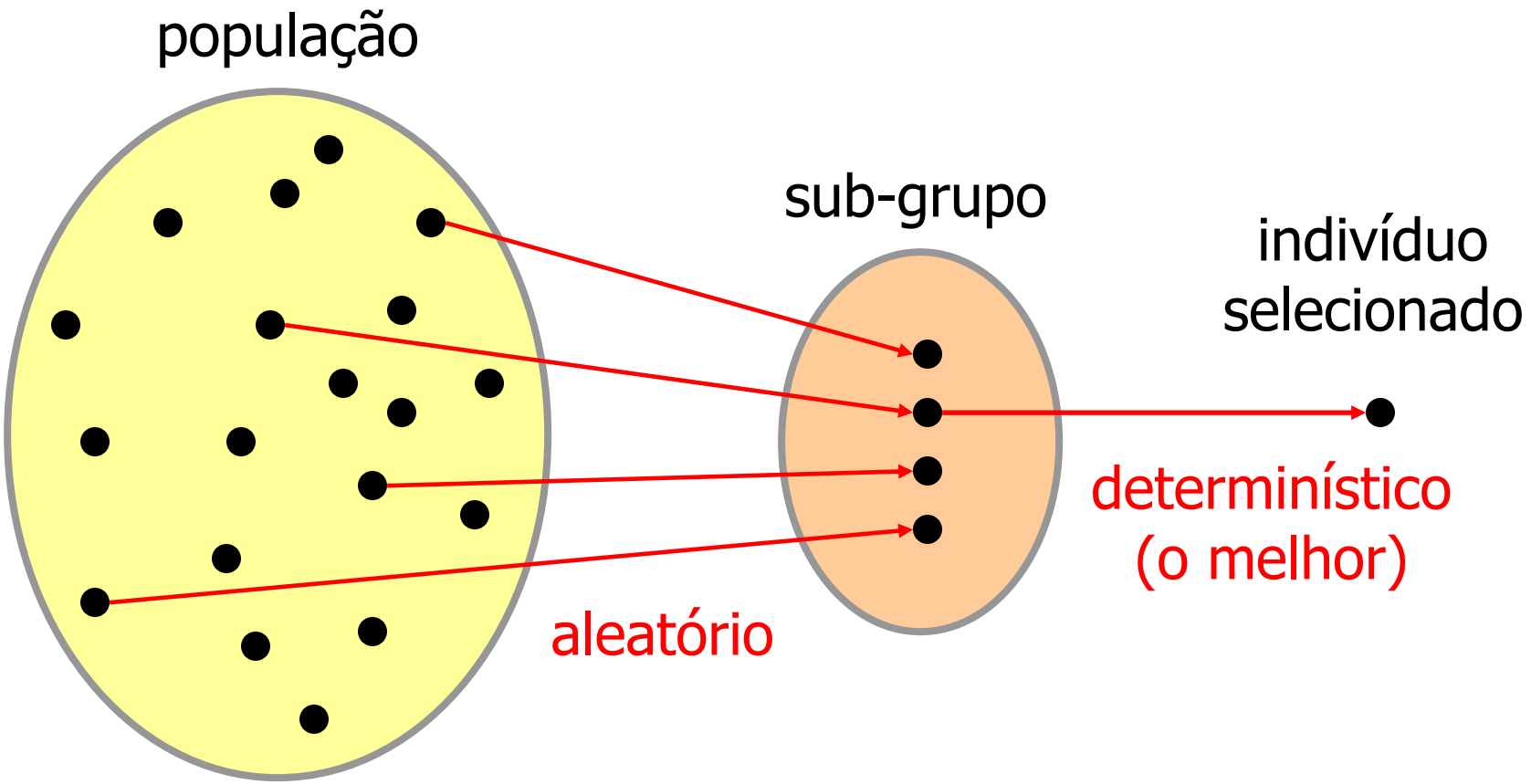
Indivíduo	Aptidão	Aptidão %
10011101	5	20
11111111	8	32
10000001	2	8
11101011	6	24
10011010	4	16
Total	25	100



Seleção por Roleta

- Escolha **probabilística**
  - Maior aptidão » Maior probabilidade de ser selecionado
- Competição sempre se dá entre **todos** os indivíduos da população » Custo
- Como seleção por ordem, a convergência pode ainda ser muito rápida
- Pouca **diversidade**

Seleção por Torneio



Seleção por Torneio

- **T** indivíduos selecionados **randomicamente** usando uma distribuição uniforme » **Melhor diversidade**
- O **melhor** dos T é selecionado
- Avaliação/competição apenas entre os indivíduos do sub-grupo » Menor custo
- Reflete melhor a seleção na natureza

Fase de Procriação

- Aplicação dos Operadores Genéticos
  - Cruzamento
  - Mutação
  - Reprodução

Cruzamento

- **Recombina** características de 2 pais, gerando 2 filhos
- Equivale à reprodução sexuada dos seres vivos

Cruzamento em AG

- Considerando
  - Pai 1: 10101010110101010111
  - Pai 2: 00001001010101110010
- Cruzamento em um ponto
  - Filho 1: 10101010110101110010
  - Filho 2: 00001001010101010111

Cruzamento em AG

- Considerando
  - Pai 1: 10101010110101010111
  - Pai 2: 00001001010101110010
- Cruzamento em um ponto
  - Filho 1: 10101010110101110010
  - Filho 2: 00001001010101010111
- Cruzamento multi-ponto
  - Filho 1: 10101001010101010010
  - Filho 2: 00001010110101110111

Mutação

- Substitui uma parte de 1 indivíduo por outra gerada aleatoriamente
- **Cria novas** características » Diversidade

Mutação em AG

- Considerando
  - Pai: 10101010110101010111
- Mutação
  - Filho: 10101010110111010111

Reprodução

- Cópia simples de um indivíduo para a próxima geração
- **Preserva** características

Estratégia Elitista

- Os **E** melhores indivíduos são **sempre** reproduzidos, *i.e.*, copiados para a próxima geração
- Aumenta E » Diminui diversidade

Critério de Término

- Nível de aceitação de uma solução
  - Número máximo de gerações
  - Margem de erro
  - Detecção de Estado-estável (*Steady-state*) – solução estabiliza e não muda mais

Parâmetros

- Controlam o algoritmo
- Impacto em
  - Probabilidade de Sucesso
  - Eficiência
    - Custo de Processamento
    - Custo de Memória
  - Qualidade da solução final
- Configurados de acordo com o problema

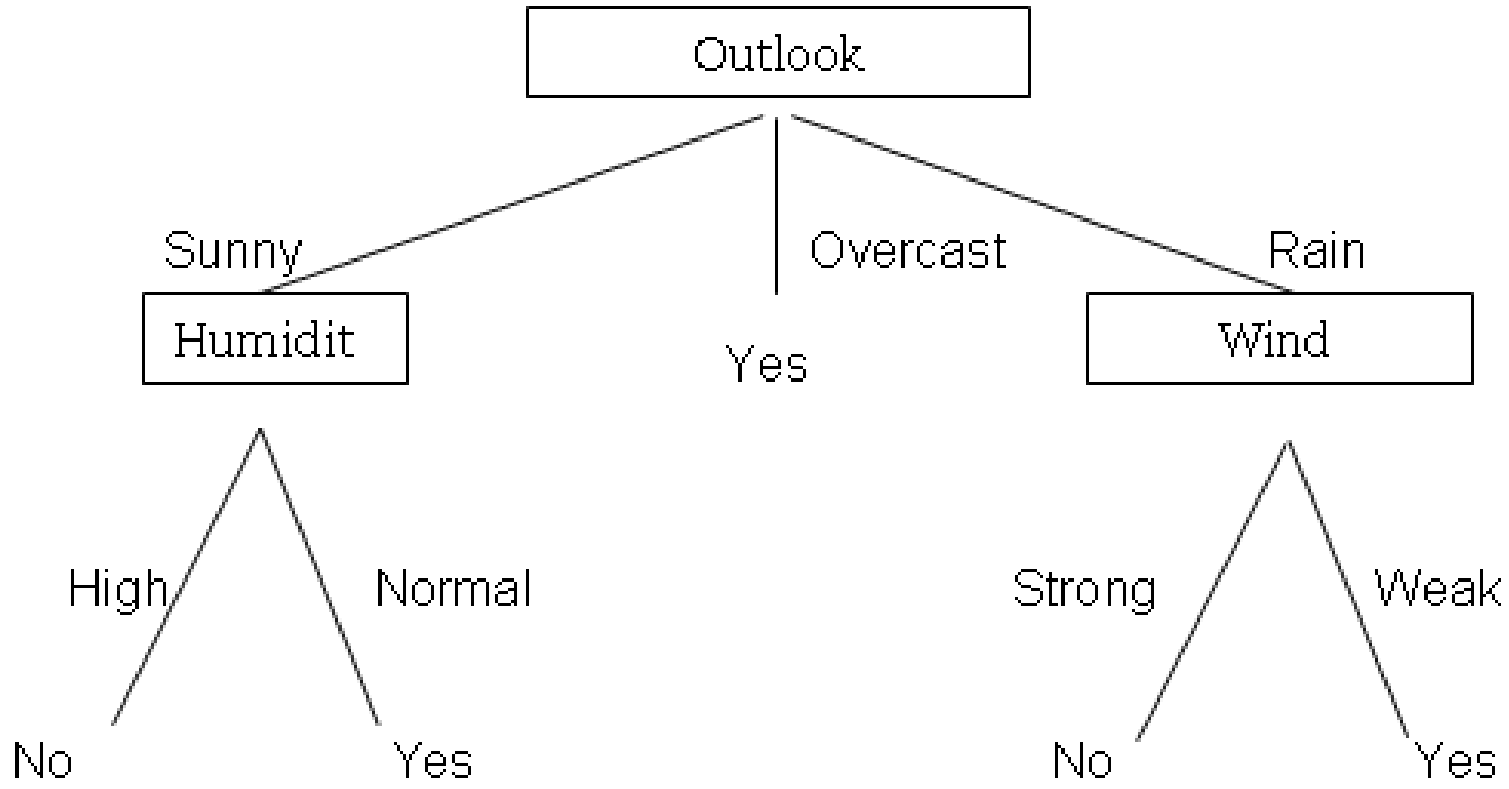
Parâmetros

- **Iniciação**
  - Método de iniciação
- **Repetições**
  - Tamanho da população
  - Número máximo de gerações
- **Avaliação**
  - Função de aptidão
  - Limiar (Margem de erro)
- **Seleção**
  - Método de seleção
  - Tamanho do torneio
  - Tamanho da elite
- **Procriação**
  - Taxa de cruzamento
  - Taxa de mutação
  - Taxa de reprodução

Exemplo de AG - 1

Exemplo de AG – Aprender regras

Aprender regras para o problema "Playtennis":



Exemplo de AG – Aprender regras

- **Codificação dos Indivíduos:**
  - Indivíduos neste problema: regras de decisão, compostas por cabeça e corpo
  - Árvore de decisão (conceito "Playtennis"): nem todos os atributos, inicialmente disponíveis, precisam ser usados para a concepção deste conceito

Exemplo de AG – Aprender regras

- **Codificação dos Indivíduos:**
  - Codificação mais completa: todos os atributos devem ser considerados
  - Assim, indivíduos podem ser codificados usando:
    - uma string de bits
    - um bit para cada valor possível para cada atributo

Exemplo de AG – Aprender regras

Outlook			Humidity		Wind		Temperature			Class	
Sunny	Overcast	Rain	High	Normal	Strong	Weak	Hot	Mild	Cool	Yes	No
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- Indivíduo: composto por 12 bits (10 bits corpo e 2 bits cabeça da regra):
  - *Outlook*: três bits (três valores possíveis)
  - *Humidity*: dois bits
  - *Wind*: dois bits
  - *Temperature*: três bits
  - Dois bits representam a cabeça da regra (*Class* "Play")
- O valor 1 representa "sim" para o valor do atributo correspondente e 0 caso contrário

Exemplo de AG – Aprender regras

- Tratamento de genótipos sem significado útil para atributos:
  - Se houver, por exemplo, uma seqüência 111 para os três bits do atributo *Temperature*, significa que esse atributo não importa naquele caso específico
  - Vide exemplo dessa sequência a seguir

Exemplo de AG – Aprender regras

IF Outlook=Sunny AND Humidity=Normal
THEN Playtennis=Yes

Outlook			Humidity		Wind		Temperature			Class	
Sunny	Overcast	Rain	High	Normal	Strong	Weak	Hot	Mild	Cool	Yes	No
1	0	0	0	1	1	1	1	1	1	1	0

Exemplo de AG – Aprender regras

IF Outlook=Sunny AND Humidity=Normal
THEN Playtennis=Yes

Outlook			Humidity		Wind		Temperature			Class	
Sunny	Overcast	Rain	High	Normal	Strong	Weak	Hot	Mild	Cool	Yes	No
1	0	0	0	1	1	1	1	1	1	1	0

IF Outlook=Rain AND Wind=Strong
THEN Playtennis=No

Outlook			Humidity		Wind		Temperature			Class	
Sunny	Overcast	Rain	High	Normal	Strong	Weak	Hot	Mild	Cool	Yes	No
0	0	1	1	1	1	0	1	1	1	0	1

Exemplo de AG - 2

Otimizar medida de importância de atributos (seleção de atributos)

Exemplo de AG - 3

Otimizar função (programação linear)

Exemplo de AG – Otimizar função

- **Problema de programação linear:**

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{sujeito a} & A\mathbf{x} \leq \mathbf{b} \\ \text{e} & \mathbf{x} \geq \mathbf{0} \end{array}$$

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Problema de programação linear:**

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{sujeito a} & A\mathbf{x} \leq \mathbf{b} \\ \text{e} & \mathbf{x} \geq \mathbf{0} \end{array}$$

  - c e b: vetores de coeficientes (já conhecidos)
  - c^T: vetor coluna transposto, *i.e.*, vetor linha
  - x: vetor de variáveis (a ser encontrado)
  - A: matriz de coeficientes (já conhecidos)

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Exemplo:**
  - Um fazendeiro tem uma área de L km² para plantar trigo, cevada ou uma combinação dessas culturas
  - A área plantada (em km²) com trigo e com cevada corresponde a x_1 e x_2, respectivamente

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Exemplo:**
  - Logo, há uma restrição: a soma das áreas plantadas de trigo (x_1) e cevada (x_2) deverá ser menor ou igual a L km²
  - Mais duas restrições surgem porque, eventualmente, o fazendeiro não plantará nada em parte da área que possui, *i.e.*, x_1 e x_2 podem ser maiores ou iguais a zero

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Exemplo:**
  - O fazendeiro tem uma quantidade limitada de fertilizante (F kg) e de inseticida (P kg) por km², definindo mais restrições
    - Cada km² do total de trigo (x_1) necessita de F_1 kg e P_1 kg de fertilizante e inseticida, respectivamente
    - Cada km² do total de cevada (x_2) precisa de F_2 kg e P_2 kg de fertilizante e inseticida

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Exemplo:**
  - O preço de venda por km² do total de trigo (x_1) e do total de cevada (x_2) é de S_1 e S_2, respectivamente
  - Nesse cenário, o objetivo do fazendeiro é maximizar o lucro (preço de venda) que combina trigo e cevada

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Exemplo:**

maximize $S_1x_1 + S_2x_2$ (maximize lucro - "função objetivo")
sujeito a $x_1 + x_2 \leq L$ (limite da área total)
$F_1x_1 + F_2x_2 \leq F$ (limite do fertilizante)
$P_1x_1 + P_2x_2 \leq P$ (limite do inseticida)
$x_1 \geq 0, x_2 \geq 0$ (não semear uma área negativa)

fonte: https://en.wikipedia.org/wiki/Linear_programming

Exemplo de AG – Otimizar função

- **Codificação dos Indivíduos:**
  - Indivíduos neste problema: valores para x_1 e x_2 (áreas plantadas)
  - Uma possível codificação:
    - Arredondar cada área para um número inteiro
    - Representar o valor inteiro como uma string de bits (valor decimal transformado para base binária)
    - Concatenar as duas strings

Exemplo de AG – Otimizar função

x1										x2									
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- Indivíduo: composto por 20 bits (10 bits por área, representando valor de 0 a 1023 km²):
  - Área de trigo (x_1): bits 0 a 9
  - Área de cevada (x_2): bits 10 a 19
- Para obter o valor aproximado da área, basta converter o valor binário para o valor decimal correspondente

Exemplo de AG – Otimizar função

$x_1 = 10$ km² e $x_2 = 50$ km²

x1										x2									
0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	1	0

$x_1 = 30$ km² e $x_2 = 65$ km²

x1										x2									
0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1

O Futuro?

fonte: www.geneticprogramming.org

Para saber mais...

- Demonstrações

  http://math.hws.edu/eck/js/genetic-algorithm/GA.html

  http://rednuht.org/genetic_walkers/

- Introdução a AG com exemplo

  Página de J. Holland
  http://www2.econ.iastate.edu/tesfatsi/holland.gaintro.htm

Para saber mais...

- Busca em Publicações
  - **GP Bibliography**
    http://liinwww.ira.uka.de/bibliography/Ai/genetic.programming.html
  - **Google Acadêmico**
    https://scholar.google.com
  - **Citeseer**
    http://citeseerx.ist.psu.edu

Para saber mais...

- Ferramentas AG
  - **Weka**

    http://www.cs.waikato.ac.nz/ml/weka
  - **Outras ferramentas**

  https://sci2s.ugr.es/keel/links.php
- Outros algoritmos evolucionários
  - **GP Notebook**
    http://www.geneticprogramming.com/
  - **Livro de Sean Luke**
    http://cs.gmu.edu/~sean/book/metaheuristics/