

Computação evolucionária

Inteligência artificial

Prof. Allan Rodrigo Leite

Computação evolucionária

- Princípios da evolução e sobrevivência na natureza
 - Competição
 - Indivíduos competem por recursos, como alimento, água ou abrigo
 - Membros da mesma espécie podem competir para atrair um par
 - Indivíduos fracos tem menos características necessárias para sobreviver
 - Adaptação
 - Indivíduos mais bem sucedidos terão, relativamente, mais descendentes
 - Indivíduos mal sucedidos geram poucos ou nenhum descendente
 - Genes dos indivíduos mais adaptados se espalham mais entre gerações

Computação evolucionária

- Princípios da evolução e sobrevivência na natureza (cont.)
 - Evolução
 - Combinação de boas características vindas de diferentes ancestrais podem produzir indivíduos mais aptos
 - Espécies evoluem para se tornarem cada vez mais adaptadas ao seu ambiente por meio da hereditariedade
 - Genética é a ciência que estuda mecanismos de transmissão de características de uma espécie de uma geração para outra

Computação evolucionária

- Fatos históricos
 - 1859 - Charles Darwin publica o livro *The origin of Species*
 - Mecanismos de transformação das espécies
 - 1865 - Gregor Mendel apresenta resultados de experimentos genéricos
 - Cruzamento genético de ervilhas
 - Considerado o pai da genética
 - 1960 - Início da computação evolucionária
 - Holland define princípios básicos de Algoritmos Genéticos
 - Rechemberg propõe as estratégias evolucionárias

Computação evolucionária

- Modelos computacionais baseados na teoria da evolução natural
 - Pesquisas iniciaram na década de 50 e continua ativamente até hoje
 - Evolução é um processo de otimização cujo objetivo consiste em
 - Melhorar a habilidade de um organismo ou sistema
 - Sobreviver em ambientes competitivos e que mudam dinamicamente
- Computação evolucionária reproduz processos evolucionários como
 - Seleção natural
 - Sobrevivência do mais apto
 - Reprodução

Computação evolucionária

- Podem ser agrupados em três categorias
 - Algoritmos genéticos
 - Estratégias evolutivas
 - Programação genética
- Algoritmos evolucionários
 - Busca estocástica por uma solução ótima para um dado problema
 - São compostos por
 - Codificação das soluções do problema como um cromossomo
 - Função para avaliar a aptidão (*fitness*) dos indivíduos
 - População inicial
 - Operadores de seleção e de reprodução

Algoritmos genéticos

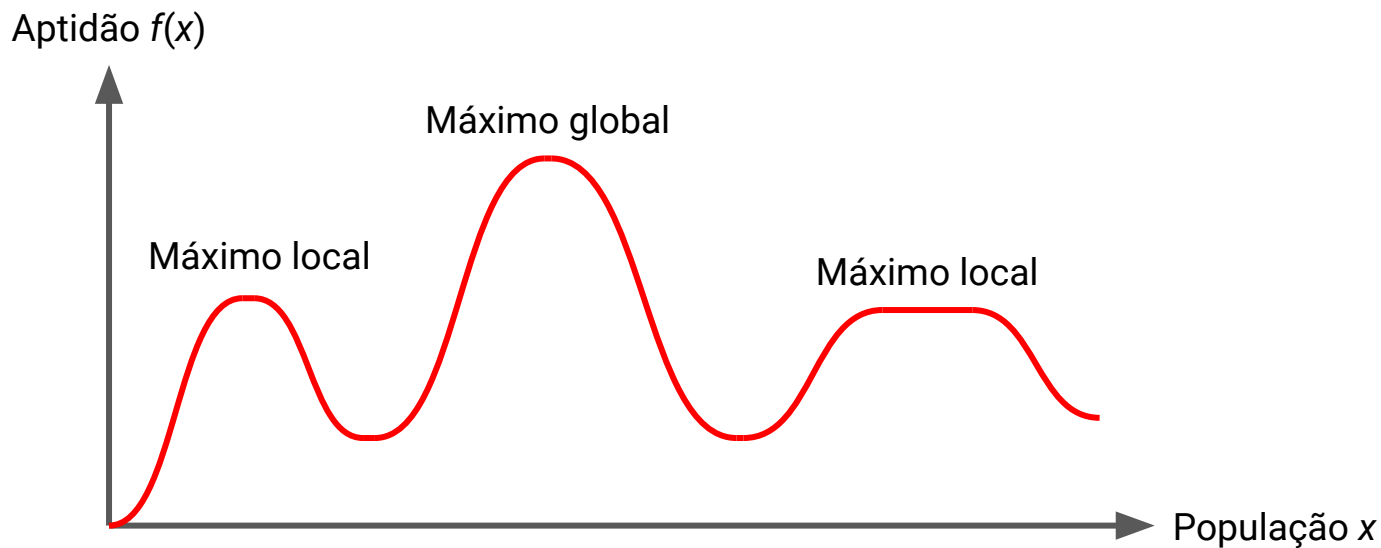
- Usam terminologia oriunda da teoria da evolução natural e genética
 - Desenvolvido por John Holland e popularizado por David Goldberg
- Objetivos de algoritmos genéticos
 - Formalizar matematicamente e explicar rigorosamente processos de adaptação em sistemas naturais
 - Desenvolver sistemas artificiais capazes de conservar tais mecanismos importantes dos sistemas naturais
 - Problemas devem ser adequadamente representados
 - Deve existir uma maneira de avaliar as soluções encontradas
 - Utilizam a metáfora de população para representar soluções candidatas

Algoritmos genéticos

- São úteis para resolver problemas complexos de busca e otimização
 - Gerações de populações evoluem de acordo com os princípios de seleção natural e sobrevivência dos mais aptos
 - A evolução corresponde a uma busca em um espaço de soluções potenciais para o problema
- Fornecem equilíbrio entre dois objetivos aparentemente conflitantes
 - Aproveitamento de soluções promissoras (*exploitation*)
 - Guiada por ótimos locais
 - Exploração do espaço de busca (*exploration*)
 - Busca por ótimos globais

Algoritmos genéticos

- Máximos globais e máximos locais



Algoritmos genéticos

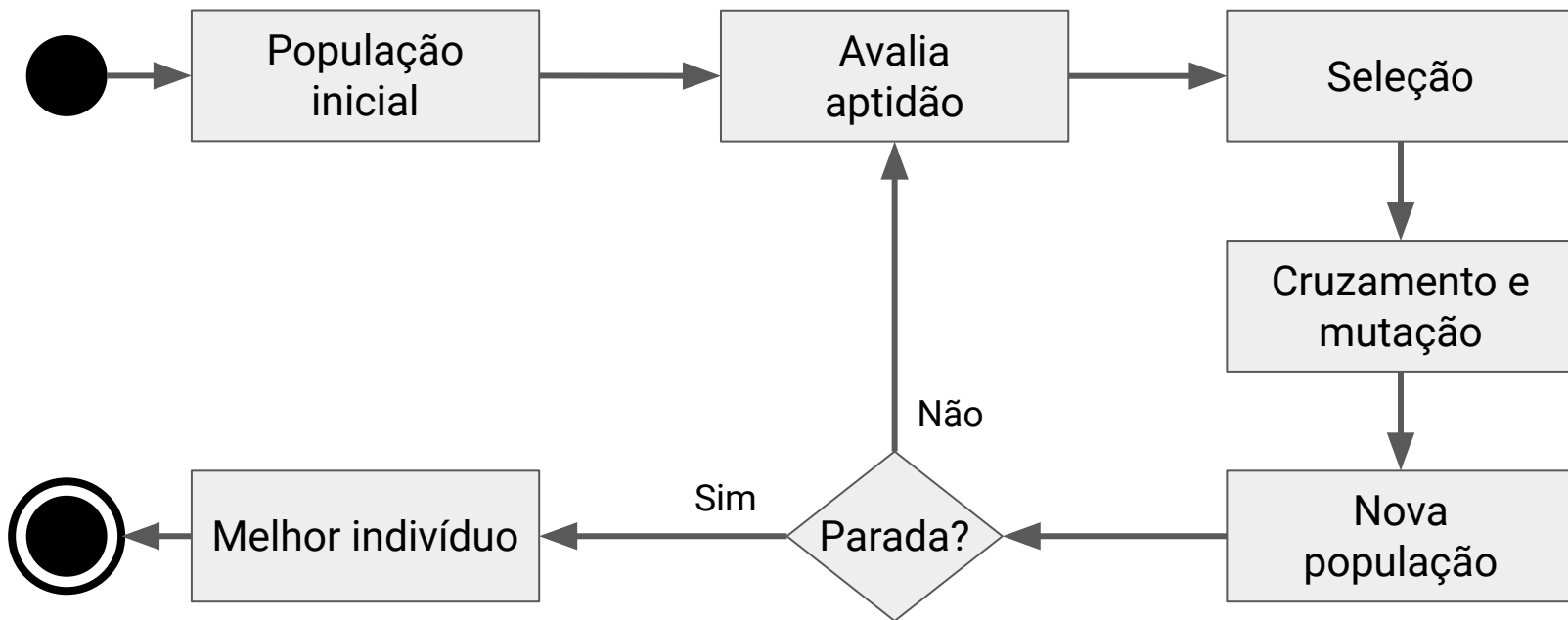
- O processo de otimização ocorre iterativamente em várias gerações
 - Mecanismos de seleção selecionam os indivíduos mais aptos
 - Operadores de reprodução geram novos indivíduos
 - Cada indivíduo representa uma possível solução para o problema
 - Cada indivíduo é associado um índice (*ranking*) de aptidão
 - Requer uma função de *fitness* para avaliar a qualidade da solução
 - Indivíduos mais aptos têm mais oportunidades de reprodução
 - Evolução ocorre a partir da produção de descendentes cada vez mais aptos

Algoritmos genéticos

- Aplicação de algoritmos genéticos são favoráveis para
 - Problemas de busca, combinatórios ou otimização
 - Problemas com um grande número de variáveis
 - Problemas com um amplo espaços de estados
- A concepção de um algoritmo genético envolve os conceitos
 - Representação do indivíduo
 - Definição da função de *fitness*
 - Processo de seleção
 - Processo de reprodução
 - Convergência da solução

Algoritmos genéticos

- Fluxo de um algoritmo genético



Algoritmos genéticos

- Representação do indivíduo
 - Tem por objetivo descrever as características do indivíduo
 - Também chamado de cromossomo ou *string*
 - Representado por meio de um vetor de parâmetros
 - Cada indivíduo possui um valor de aptidão
 - A aptidão mede a qualidade da solução que ele representa
 - Uma população é um conjunto de indivíduos
 - Uma população é produzida em uma dada geração
 - Uma geração é uma iteração completa que produz uma nova população

Algoritmos genéticos

- Representação do indivíduo (cont.)
 - Indivíduo é codificado por um conjunto de parâmetros chamado gene
 - Genes podem assumir valores representados por diferentes tipos dados
 - Por exemplo: valores binários, inteiros, reais, caracteres, etc.
 - Os valores para os genes são também conhecidos por alfabeto
 - Parâmetros são combinados para formar *strings* ou vetores
 - Esta combinação representa o cromossomo (indivíduo)

Algoritmos genéticos

- Definição da função de *fitness*
 - Representa a função objetivo do problema de busca ou otimização
 - Utilizada para calcular a aptidão de um indivíduo
 - Aptidão bruta
 - Saída gerada pela função objetivo para um indivíduo da população
 - Aptidão normalizada
 - Aptidão bruta normalizada usada como entrada para o método de seleção
 - Aptidão máxima
 - Melhor indivíduo da população corrente
 - Aptidão média
 - Aptidão média da população corrente

Algoritmos genéticos

- Processo de seleção
 - Objetiva a seleção dos indivíduos mais aptos da geração atual
 - Os selecionados serão utilizados para geração da próxima população
 - Um indivíduo tem probabilidade de seleção proporcional à sua aptidão
 - Método roleta
 - Considera cada indivíduo proporcionalmente pela aptidão
 - Os indivíduos mais aptos têm maiores chances de reprodução
 - Método torneio
 - Seleciona aleatoriamente pares de indivíduos da população
 - Escolhe o melhor indivíduo considerando uma probabilidade k
 - Do contrário, escolhe o pior indivíduo

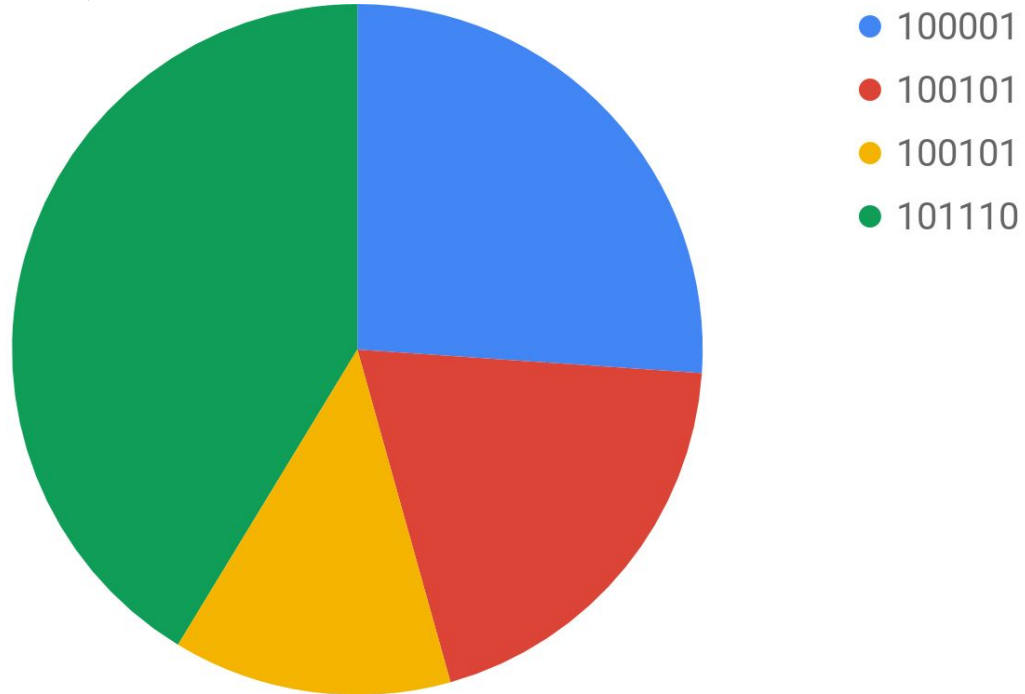
Algoritmos genéticos

- Método roleta

```
roleta( $P$ ,  $N$ ) //população e número de indivíduos a ser selecionado
 $S \leftarrow \emptyset$  //lista dos indivíduos selecionados
 $t \leftarrow \sum f(x) \mid \forall x \in P$  //soma de todas as aptidões na população
enquanto  $|S| \neq N$  faça //enquanto selecionados for menor que  $N$ 
     $r \leftarrow \text{random}(0, t)$  //escolhe aleatoriamente um número entre 0 e  $t$ 
     $s \leftarrow 0$  //acumulado das aptidões dos indivíduos percorridos
     $\forall x \in P$  faça //itera sobre indivíduos da população
         $s \leftarrow s + f(x)$  //acumula aptidão de indivíduos
        se  $s \geq r$  então
             $S \leftarrow S \cup \{x\}$  //seleciona indivíduo e encerra iteração
        fim se
    fim faça
fim enquanto
retorna  $S$ 
```

Algoritmos genéticos

- Método roleta (cont.)



Algoritmos genéticos

- Método torneio

```
torneio( $P$ ,  $N$ ,  $k$ ) //população, indivíduos selecionados e probabilidade  
   $S \leftarrow \emptyset$  //lista dos indivíduos selecionados  
  enquanto  $|S| \neq N$  faça //enquanto selecionados for menor que  $N$   
     $r \leftarrow \text{random}(0,1)$  //escolhe aleatoriamente um número entre 0 e 1  
     $\{x,y\} \leftarrow \text{random}(S)$  //escolhe dois indivíduos aleatoriamente  
    se  $r < k$  então  
       $S \leftarrow S \cup \{\text{argmax}(f(x), f(y))\}$  //seleciona o melhor indivíduo  
    senão  
       $S \leftarrow S \cup \{\text{argmin}(f(x), f(y))\}$  //seleciona o pior indivíduo  
    fim se  
  fim enquanto  
  retorna  $S$ 
```

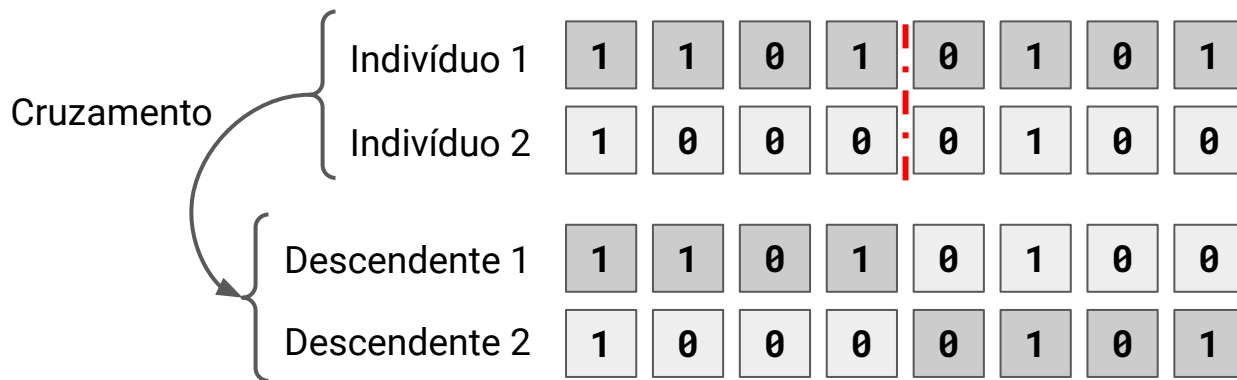
Algoritmos genéticos

- Processo de reprodução
 - O processo de reprodução ocorre por meio de operadores genéticos
 - São necessários para que a população se diversifique ao longo do tempo
 - Deve manter características de adaptação adquiridas em gerações anteriores
 - Operadores genético tem por objetivo
 - Transformar a população através de sucessivas gerações
 - Convergir a busca até alcançar a um resultado satisfatório
 - Os operadores genéticos são
 - Cruzamento (*crossover*)
 - Mutação

Algoritmos genéticos

- Cruzamento

- Produz novos indivíduos combinando pares de indivíduos selecionados
 - Os indivíduos são selecionados pelo processo de seleção
 - Procura manter parte das características dos descendentes
- É considerado o operador genético predominante ou primário
 - Deve ser aplicado com probabilidade definida pela taxa de *crossover* P_c
 - Deve ser maior que a taxa da operação de mutação



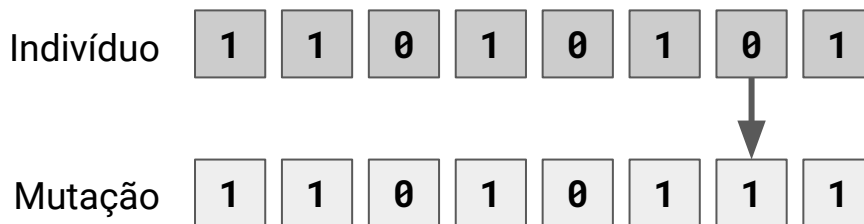
Algoritmos genéticos

- Cruzamento (cont.)
 - Estratégias comuns para *crossover*
 - Um ponto: um ponto de cruzamento é escolhido e, a partir deste ponto, as informações genéticas dos pais serão trocadas
 - Multi pontos: generalização da troca de material genético por meio de vários pontos, isto é, usa-se mais de um ponto de cruzamento
 - Uniforme: um parâmetro global determina a probabilidade de cada variável ser trocada entre os ancestrais

Algoritmos genéticos

- Mutação

- Modifica aleatoriamente alguma característica do indivíduo
 - Permite criar novos valores de características que não existiam anteriormente ou apareciam em pequena quantidade
- Este operador introduz e mantém a diversidade genética da população
 - Permite sair de uma zona de convergência da qualidade da solução
 - Causa uma ruptura da convergência em máximos locais



Algoritmos genéticos

- Mutação (cont.)
 - É considerado um operador genético secundário
 - Aplicado aos indivíduos com probabilidade dada pela taxa de mutação P_m
 - Geralmente se utiliza uma taxa de mutação baixa
 - Estratégias comuns para mutação
 - Inversão de bit: usado quando as propriedades binárias, escolhe-se uma propriedade e realiza a inversão do bit
 - Mudança de ordem ou permutação: usado quando as propriedades compartilham o mesmo domínio, dois valores são escolhidos e trocados

Algoritmos genéticos

- Estrutura básica do algoritmo

AlgoritmoGenetico()

$t \leftarrow 0$ //tempo inicial

$P \leftarrow \text{inicia}()$ //cria população inicial - normalmente aleatória

$f \leftarrow \text{avalia}(P)$ //avalia aptidão dos indivíduos

enquanto *<condição de parada não satisfeita>* **faça**

$t \leftarrow t + 1$ //incrementa tempo

$P \leftarrow \text{selecao}(P)$ //escolhe indivíduos mais aptos

$P \leftarrow \text{reproducao}(P)$ //realiza cruzamento e mutação

$f \leftarrow \text{avalia}(P)$ //avalia aptidão dos indivíduos

fim enquanto

retorna $\text{melhor}(P)$ //retorna melhor indivíduo

Algoritmos genéticos

- Ferramentas e bibliotecas de algoritmos genéticos
 - Jenetics (*Java Genetic Algorithm Library*)
 - Biblioteca desenvolvida em Java
 - Suporta otimização multi objetivos
 - Projeto *open-source*
 - GeneticAlgorithm
 - Biblioteca desenvolvida em Python
 - Suporta problemas combinatoriais e de otimização
 - Projeto *open-source*
 - GA
 - Biblioteca desenvolvida em R
 - Suporta diferentes formas de representação dos cromossomos
 - Projeto *open-source*

Exercícios

1. Implemente um algoritmo genético para resolver o problema da mochila. O problema consiste em encontrar uma combinação de objetos que maximize a recompensa e não exceda o peso que a mochila consegue carregar.

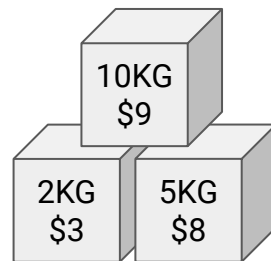
Dado um conjunto de objetos N e uma mochila de capacidade b , temos:

- Conjunto de objetos $[o_j, \dots, o_N]$
- Peso dos objetos $W(o_j) = w_j$
- Recompensa dos objetos $C(o_j) = c_j$
- Uma solução s é um vetor binário
 - Se o objeto o_j está na mochila, então $s_j = 1$, caso contrário $s_j = 0$
- Função objetivo

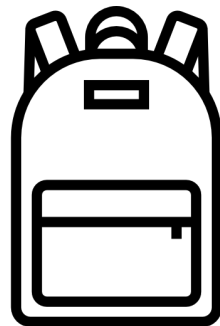
$$\text{Maximizar } z = \sum_{j=1}^n c_j s_j$$

$$\text{Sujeita a } \sum_{j=1}^n w_j s_j \leq b$$

$$s_j \in \{0, 1\}$$



?



Computação evolucionária

Inteligência artificial

Prof. Allan Rodrigo Leite