

Algoritmos Genéticos (GAs)

Tópicos de Inteligência Artificial

Prof. Dr. [Dieval Guizelini](#)

Fevereiro/2023

Algoritmos genéticos

- Essencialmente, é um método de busca.
- Na área da IA, os algoritmos genéticos (AGs) criaram uma subárea, conhecida como algoritmos evolutivo.
- É uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca,
- Foi proposta por John Henry Holland.



Computação evolucionária

“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.” (DARWIN, 1859)

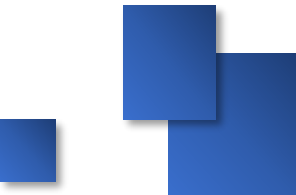
Também é considerado um método de busca paralela.

- Área da Inteligência Artificial, que engloba um conjunto de métodos computacionais;
- Inspirados na Teoria da Evolução das Espécies de Charles Darwin (DARWIN, 1859) para a solução de problemas:
 - Na natureza sobrevivem os indivíduos que possuem maior capacidade de se adaptarem ao meio ambiente;
 - Suas características genéticas são repassadas para as gerações seguintes e melhoradas;
 - A nova geração será composta por indivíduos com material genético melhor do que os da população anterior.

A computação evolucionária

Atualmente a computação evolucionária se divide:

- Programação evolucionária
 - Previsão do comportamento de máquinas de estado finitas.
- Estratégias evolucionárias
 - ênfase na auto-adaptação. O papel da recombinação é aceito, mas como operador secundário.
- Algoritmos genéticos
 - Indivíduos contém um genótipo formado por cromossomos
- Programação genética
 - Evolução de programas

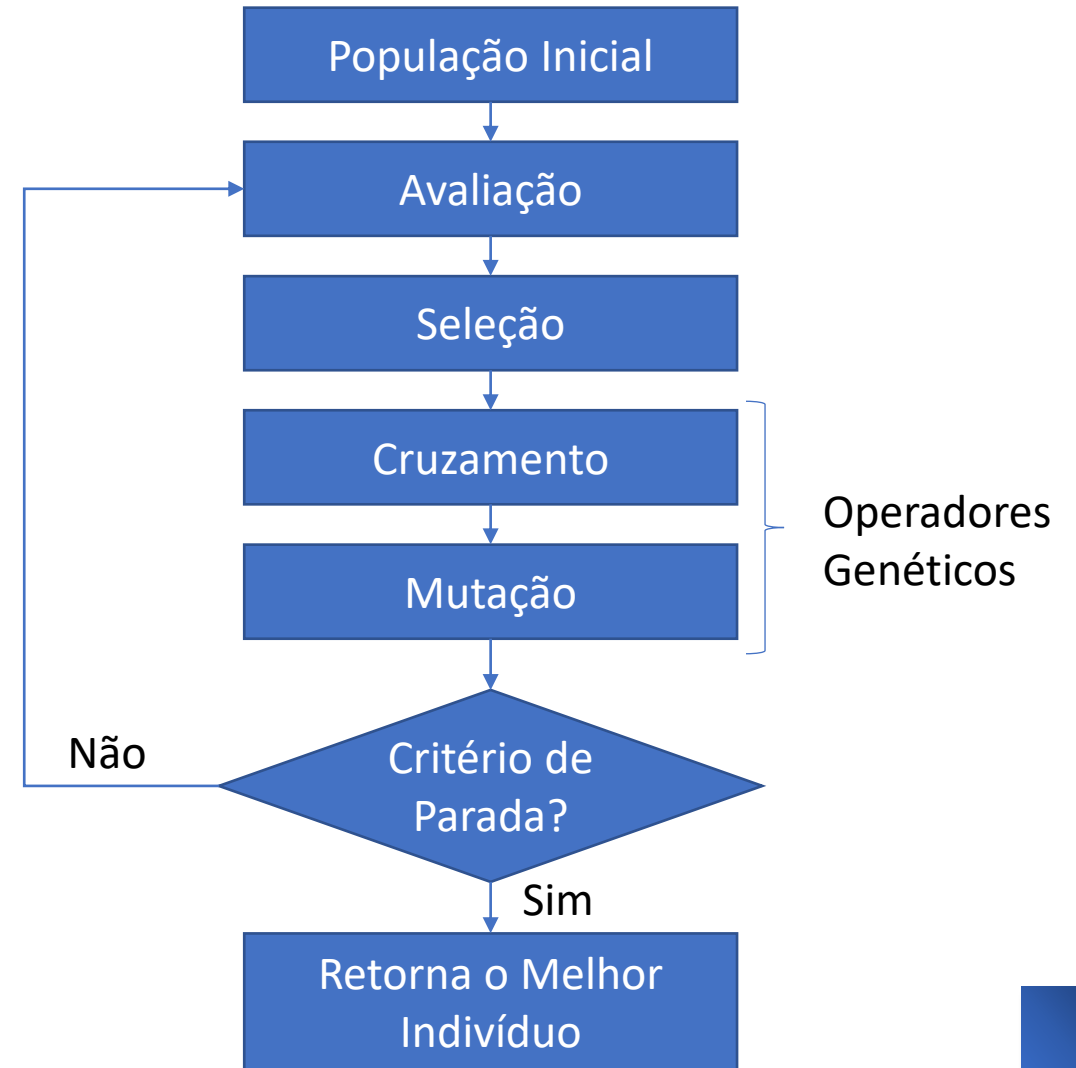


Terminologia

Termo	conceito
Indivíduo, Filho, pais	São propostas de soluções, um filho é uma solução derivada de soluções pais.
População	Um conjunto de soluções candidatas
Fitness	Qualidade
Genotipo ou genoma	Estrutura de dados do individuo
Cromossomo	Uma solução
Gene	Uma parte da solução
Alelo, fenótipo	um valor do gene
Geração	iteração
Função de avaliação	Função que avalia a solução e normalmente devolve a fitness (valor da solução).

Algoritmo

1. Gerar População Inicial
2. Descartar uma parte dos Indivíduos menos aptos
3. Aplicar operadores de reprodução
4. Aplicar operadores de mutação
5. Se o critério de parada foi satisfeito, encerrar. Senão, voltar ao passo 2.



Modelagem

- Indivíduos X Estados
- Cada indivíduo possui um *código genético*
- Esse código é chamado cromossomo
- Tradicionalmente, um cromossomo é um vetor de bits
- Vetor de bits nem sempre é o ideal



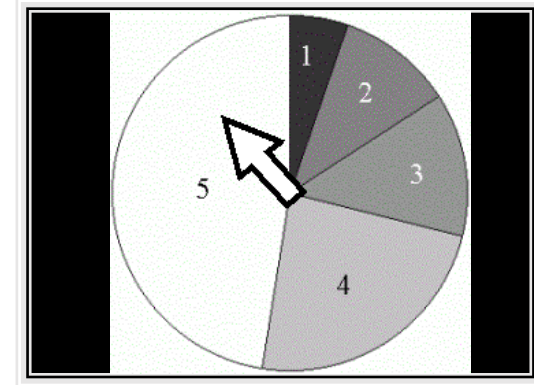
Operadores

As operações foram inspiradas nos processos biológicos, são elas:

- Seleção natural
- Manipulação genética por mutação
- Manipulação genética por reprodução

Principais métodos de seleção natural

- Roleta



- Torneio

- Utiliza sucessivas *disputas* para realizar a seleção
- Para selecionar k indivíduos, realiza k disputas, cada disputa envolvendo n indivíduos escolhidos ao acaso
- O indivíduo de maior aptidão na disputa é selecionado
- É muito comum utilizar $n = 3$

- Amostragem Universal estocástica

- *SUS – Stochastic Universal Sampling*
- Semelhante à Roleta, mas para selecionar k indivíduos utiliza k agulhas igualmente espaçadas, girando-as em conjunto uma só vez
- Apresenta resultados menos variantes que a Roleta

Algoritmo Roleta

Início

T = soma dos valores de aptidão de todos os indivíduos da população

Repita N vezes para selecionar n indivíduos

r = valor aleatório entre 0 e T

Percorra sequencialmente os indivíduos da população,
acumulando em S o valor de aptidão dos indivíduos já percorridos

Se $S \geq r$ então

 Selecione o indivíduo corrente

Fim se

Fim Repita

Fim



Algoritmo torneio

Inicio

$k = 0.75$

Repita N vezes

Escolha 2 indivíduos da população aleatoriamente

$r =$ valor aleatório entre 0 e 1

Se $r < k$

O melhor indivíduo é escolhido

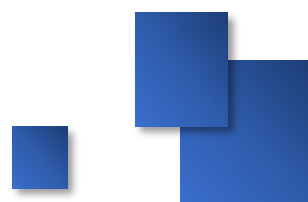
Senão

O pior indivíduo é escolhido

Fim se

Fim Repita

Fim



Operador de mutação

- Operador randômico de manipulação
- Introduz e mantém a variedade genética da população
- Garante a possibilidade de se alcançar qualquer ponto do espaço de busca
- Contorna mínimos locais
- É um operador genético secundário
- Se seu uso for exagerado, reduz a evolução a uma busca totalmente aleatória
- Logo um indivíduo sofre mutações com probabilidade baixa (normalmente entre 0,001 e 0,1)

Operador de cruzamento

- Também chamado de *reprodução* ou *crossover*
- Combina as informações genéticas de dois indivíduos (*pais*) para gerar novos indivíduos (*filhos*)
- Versões mais comuns criam sempre dois filhos para cada operação
- Operador genético principal
- Responsável por gerar novos indivíduos *diferentes* (sejam melhores ou piores) a partir de indivíduos já promissores
- Aplicado a cada par de indivíduos com alta probabilidade (normalmente entre 0,6 e 0,99)

Exemplo do cruzamento

0	1	0	1	0	1	1	0
1	0	1	0	1	1	0	0

0	1	0	1	0	1	0	0
1	0	1	0	1	1	1	0

Abordagens para o Cruzamento

- Cruzamento Um-Ponto
- Cruzamento Multi-Pontos
- Cruzamento Uniforme

Parâmetros genéticos

- Tamanho da população
- Taxa de cruzamento
- Taxa de mutação
- Intervalo de geração
- Critério de parada

Aplicações

- Ajuste de curva
- Alocação de tarefas
- Configuração de sistemas complexos
- Seleção de Rotas (caixeiro viajante)
- Problemas de Otimização e de Aprendizagem de Máquina
- Problemas cuja solução seja um estado final e não um caminho

Exemplo para uma modelagem não binária

Problema do timetables

TADS N1 4º Período		2ª	3ª	4ª	5ª	6ª	Sáb
	19h00	DS240	DS142	DS240 Rafaela A06	DS140 Razer A13	DS340 A (remoto) João E	xxxx
	20h00	DS240	DS142	DS240 Rafaela A06	DS140 Razer A13	DS340 A (remoto) João E	xxxx
	21h00	DS143	DS142	DS340 A João E A13	DS140 Razer A13	DS640 (remoto) Marly	xxxx
	22h00	DS143	DS142	DS340 A João E A13	DS140 Razer A13	DS640 (remoto) Marly	xxxx

Um cromossomo pode ser:

DS240	DS240	DS143	DS143	DS142	DS142	DS142	DS142	...
-------	-------	-------	-------	-------	-------	-------	-------	-----

Exercício

- Minimize a função:

$$f(x) = x^2 - 3x + 4$$

- ◆ Assumir que $x \in [-10, +10]$
- ◆ Codificar x como vetor binário
- ◆ Criar uma população inicial com 4 indivíduos
- ◆ Aplicar Mutação com taxa de 1%
- ◆ Aplicar Crossover com taxa de 60%
- ◆ Usar seleção por torneio.
- ◆ Usar 5 e 25 gerações.

Exercício 2

Considere uma equação no formato $ax^2+bx+c=0$

Considere os pontos conhecidos:

X	Eq
1	11
2	21
3	35
4	53
5	75

Desenvolva um algoritmo para encontrar os valores de a,b e c.
Considere a faixa de valores de 0 a 10.

Trabalho

Problema do Caixeiro Viajante

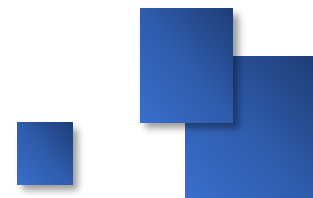
A Solução poderá ser apresentada individualmente:

Python (preferencialmente), ou em R, ou em Matlab, ou em C ou em Java.

Considere o seguinte problema de otimização (a escolha do número de 100 cidades foi feita simplesmente para tornar o problema intratável. A solução ótima para este problema não é conhecida):

Suponha que um caixeiro deva partir de sua cidade, visitar clientes em outras 99 cidades diferentes, e então retornar à sua cidade. Dadas as coordenadas das 100 cidades, descubra o percurso de menor distância que passe uma única vez por todas as cidades e retorne à cidade de origem.

Para tornar a coisa mais interessante, sorte as coordenadas das cidades (representada por x e y , em um espaço de 100 por 100 pixels. Para o desenvolvimento, recomendo definir a seed com um valor constante, para que as execuções comecem com um mesmo conjunto de dados.



Trabalho

Problema do Caixeiro Viajante

Codificação

- representação inteira: cada cromossomo conterá todos os números de 1 a 100 (cada número associado a uma cidade, e a ordem de aparecimento dos números no cromossomo vai indicar o percurso, sendo necessário fechar o percurso da última para a primeira cidade.
- Detalhe: como trata-se de um percurso fechado, a origem do percurso pode ser qualquer uma das cidades, ao menos para efeito da implementação computacional.
- número de possíveis percursos (soluções candidatas): $99! \cong 9,33 \times 10^{155}$
- função de adequação (fitness): o inverso da distância associada a cada percurso.
- solução ótima: desconhecida, em razão da impossibilidade de testar todas as soluções candidatas (único meio existente para se garantir a obtenção da solução ótima);

Trabalho

Problema do Caixeiro Viajante

Codificação

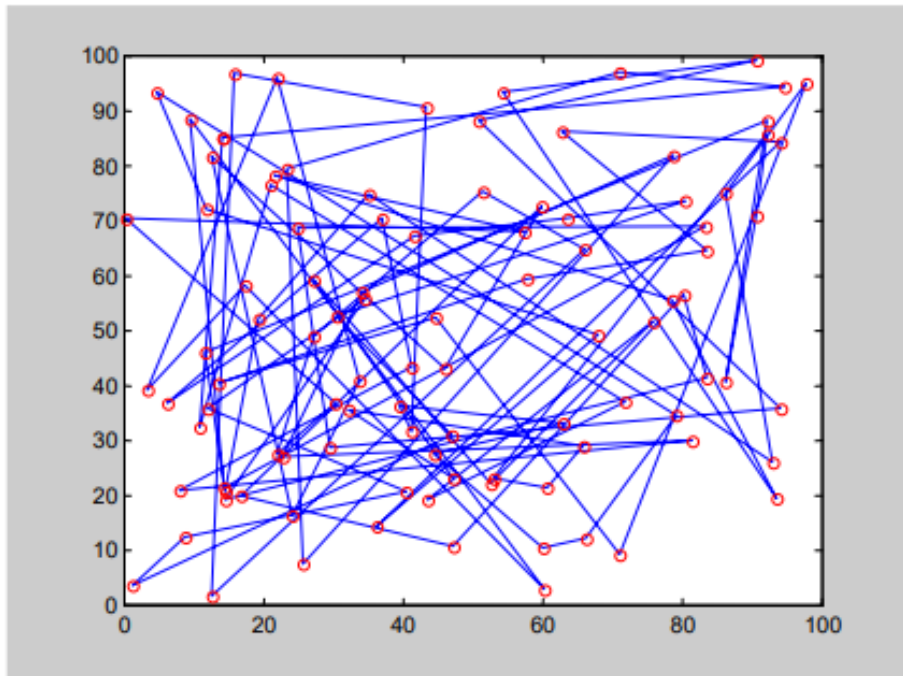
- valores arbitrados pelo usuário (ou por vocês):
 - tipo de mutação: sorteio de duas cidades para troca de posição
 - taxa de mutação: 1%
 - tipo de crossover: OX (uma espécie de crossover de um ponto, caracterizado pela junção de uma parte de um cromossomo com a parte de um outro, mas com a substituição das cidades repetidas pelas ausentes, na sequência)
 - taxa de crossover: 60%
 - tipo de seleção: rank ou torneio (50% dos melhores)

Trabalho

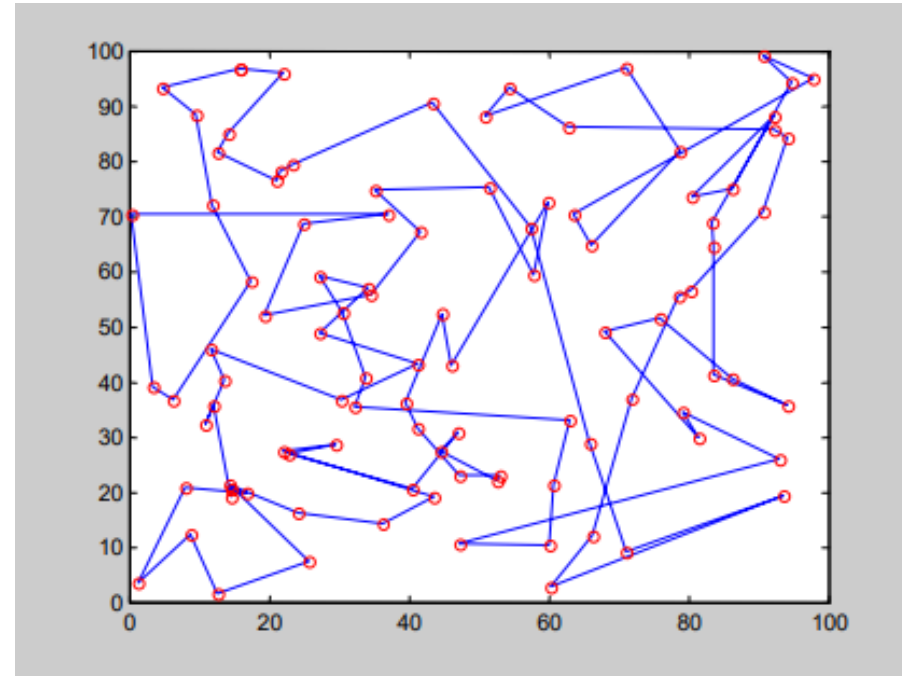
Problema do Caixeiro Viajante

Resultados esperados (naturalmente variável)

Melhor indivíduo na população inicial



Melhor indivíduo após 500 gerações

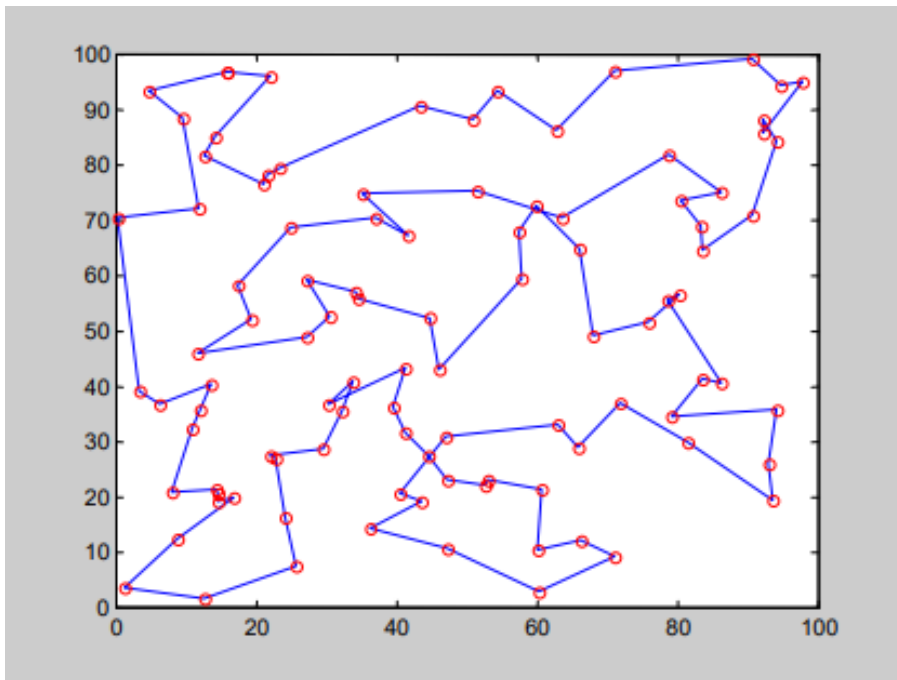


Trabalho

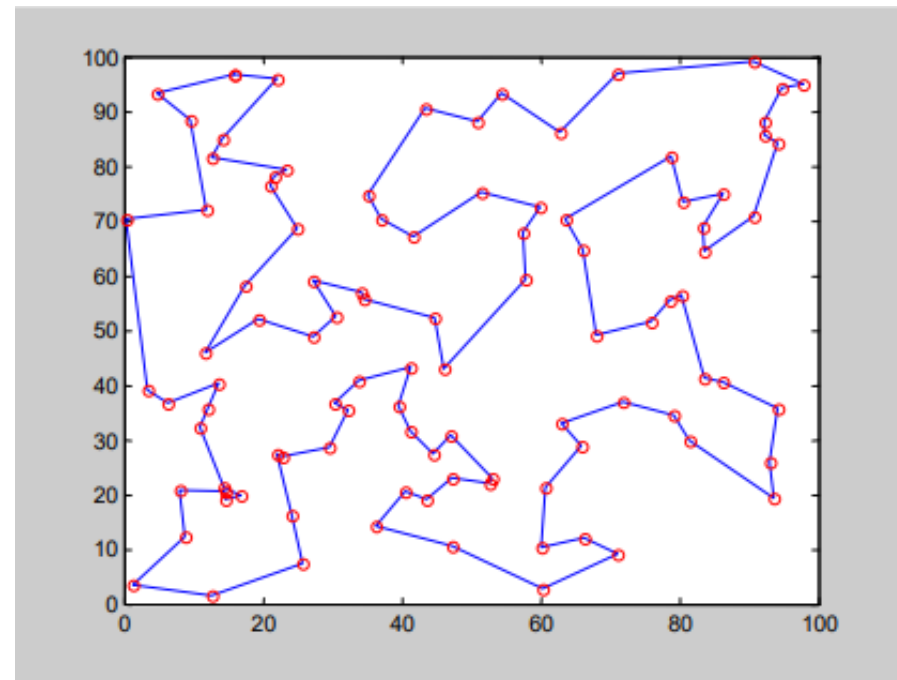
Problema do Caixeiro Viajante

Resultados esperados (naturalmente variável)

Melhor indivíduo após 2000 gerações



Melhor indivíduo após 4000 gerações



número de indivíduos testados: 400.000 (dentro os possíveis $9,33 \times 10^{155}$ candidatos)

Tempo da simulação de aproximadamente de 3 minutos.

Referências

- ZADEH, L. A. Fuzzy sets. 1965
- ABAR, Celina Aparecida Almeida Pereira. Lógica Fuzzy. PUC/SP. 2004.
- ZUBEN, Fernando J. V. Computação Evolutiva: Uma Abordagem Pragmática, 2011