

Pós Graduação em Data Science

Machine Learning

Universidade do Oeste Paulista

Prof. Me. Caio Carneloz

8 de junho de 2024

Sumário

- 1 Objetivos da Matéria
- 2 Critério de Avaliação
- 3 Computação Bioinspirada
- 4 Abordagens
- 5 Algoritmos
 - Hill Climbing
 - Simulated Annealing
 - Algoritmos Evolutivos

Objetivos da Matéria

Objetivos

- Entender o dia a dia e as funções de profissionais de dados
- Compreender os conceitos fundamentais de Machine Learning
- Diferenciar entre os principais tipos de aprendizado
- Conhecer o fluxo de trabalho típico de um projeto de Machine Learning

Critério de Avaliação

Critério de Avaliação

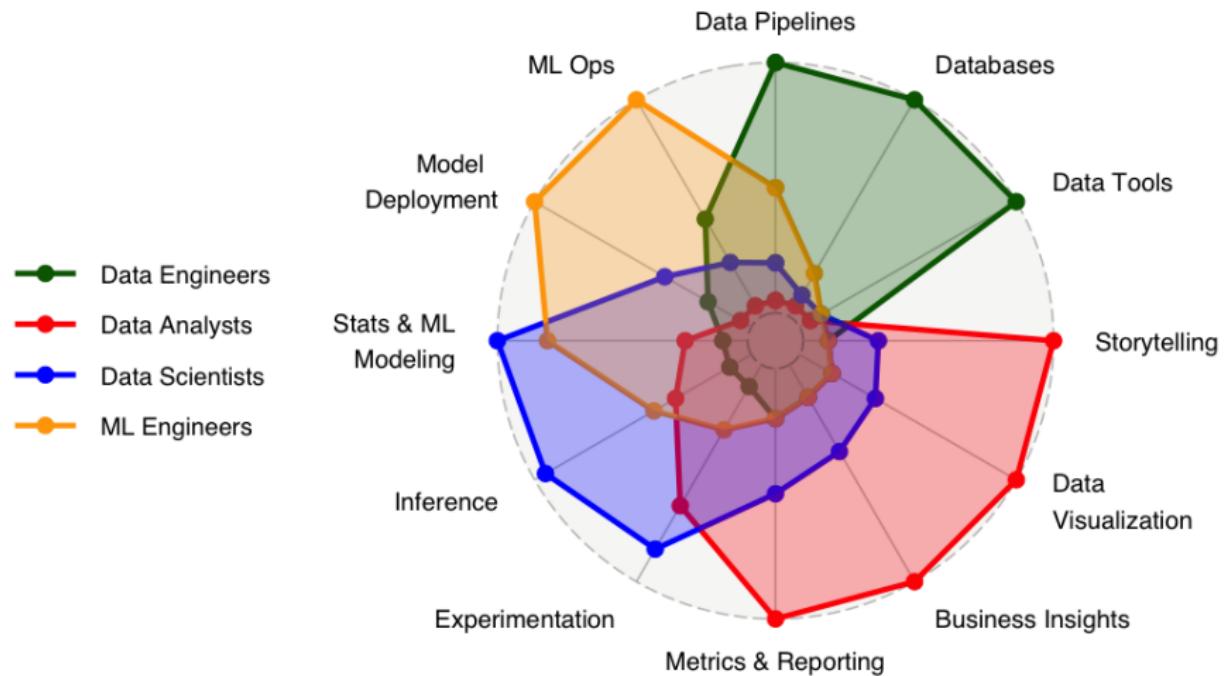
Critério	Peso
Participação Coletiva	30%
Trabalhos	20%
Seminário	50%

Principais Conceitos

Principais conceitos de Machine Learning:

- Otimização
- Features/Características
- Tipos de Aprendizado
 - Aprendizado Supervisionado
 - Aprendizado Não-Supervisionado
 - Aprendizado Semi-Supervisionado
 - Aprendizado por Reforço

Separando as Principais Funções da Área de Dados



Otimização Computação Bioinspirada

O que é?

Uso da natureza como inspiração no desenvolvimento de técnicas computacionais para solução de problemas complexos analisando:

- Comportamento de seres vivos
- Fenômenos naturais
- Fenômenos biológicos

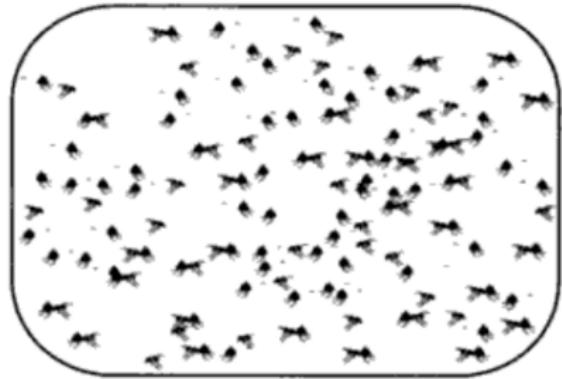
Conhecimentos Envolvidos

- Física
- Química
- Engenharia
- Biologia
- Ciência da Computação

Exemplo

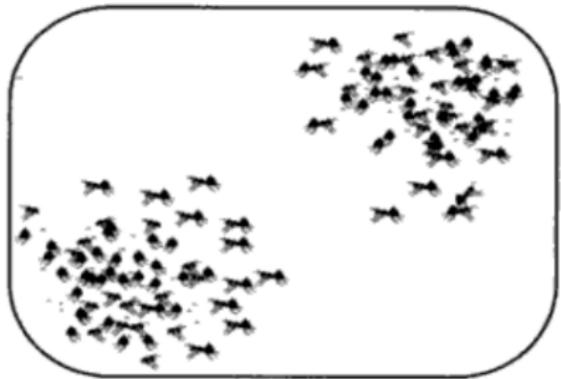
Agrupamento de corpos em colônias de formigas:

- Para limpar os ninhos, algumas espécies agrupam os corpos.
- Mecanismo básico:
 - Pegar um corpo.
 - Soltar um corpo.
- Se uma formiga encontra um corpo, ela o apanha.
- Durante o percurso, quanto mais corpos em uma determinada região, maior a probabilidade da formiga soltar o corpo que está carregando.



(a)

Distribuição inicial



(b)

Corpos agrupados

Questões

- Que tipo de problema poderia ser resolvido analisando o comportamento acima?
- Como desenvolver uma ferramenta a partir disto?

Abordagens

Quando usar abordagens de CIN?

Problema do caixeiro-viajante:

Tabela: Em um processador que calcula 100 rotas por segundo, seriam necessários cerca de 3.45×10^{18} anos para que todas as rotas fossem calculadas com 27 cidades.

Quando usar abordagens de CIN?

Em suma, é viável utilizar abordagens de CIN quando:

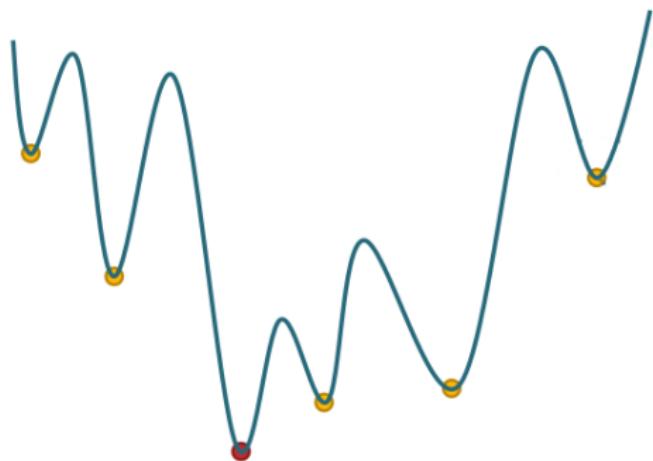
- Não há como modelar o problema corretamente
- O problema a ser solucionado é muito complexo
- Uma única solução do problema não é suficiente
- O problema não possui uma melhor solução definida

Como abordar os problemas?

Existem várias maneiras de abordar problemas complexos. Um dos mais comuns e utilizados são através de **TAREFAS DE BUSCA**.

Tarefas de busca

Neste contexto, um problema é uma coleção de informações das quais algum conhecimento será extraído ou inferido. Como por exemplo:
determinar os valores de x que minimizem uma função.



Tarefas de busca

Ou seja, resolver um problema a partir de uma tarefa de busca é tomar uma sequência de ações que:

- Levam ao desempenho desejado
- Aumentam o desempenho dos candidatos (indivíduos)

Resumindo:

- Um algoritmo de busca recebe um problema como entrada e retorna uma solução
- Um ou mais indivíduos serão usados como possíveis soluções

Definindo um problema de busca

Um problema de busca pode ser definido através dos seguintes passos:

- Especificar o objetivo
- Definir a função de avaliação
- Restringir o espaço de busca em função do objetivo
- Buscar a melhor solução através da função de avaliação
- Estipular um critério de parada

Definindo um problema de busca

No problema de maximizar a função

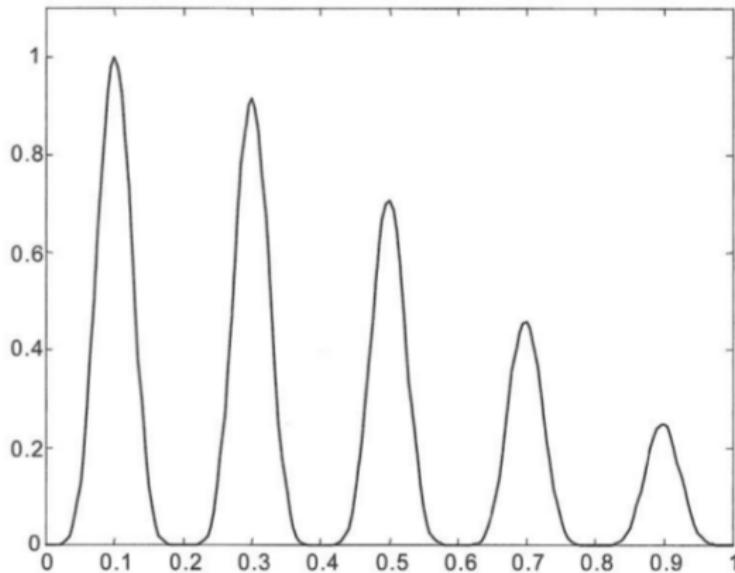
$$f(\mathbf{x}) = 2^{-2((\mathbf{x}-0.1)/0.9)^2} (\sin(5\pi \mathbf{x})^6)$$

Os passos seriam:

- Objetivo: $\max f(x)$
- Função de avaliação: $2^{-2((\mathbf{x}-0.1)/0.9)^2} (\sin(5\pi \mathbf{x})^6)$
- Espaço de busca: $x \in \mathbb{R} / x \geq 0 / x \leq 1$
- Buscar a melhor solução através da função de avaliação

Definindo um problema de busca

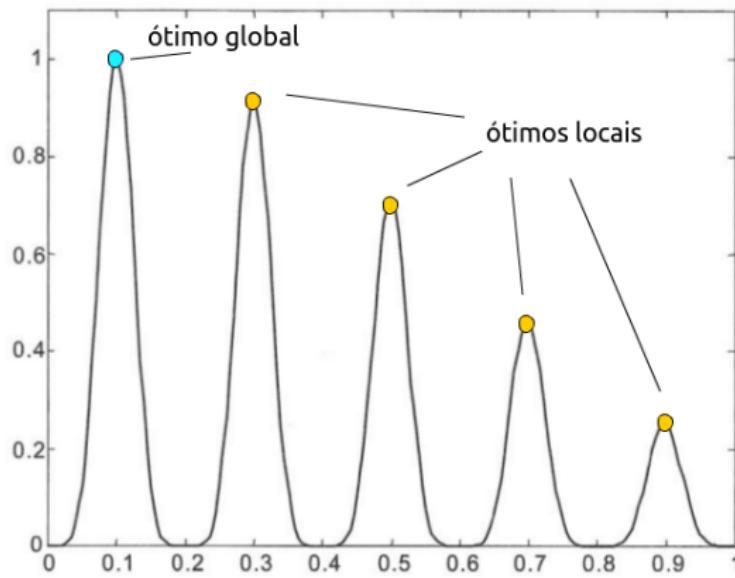
$$f(\mathbf{x}) = 2^{-2((\mathbf{x}-0.1)/0.9)^2} (\sin(5\pi\mathbf{x})^6)$$



Definindo um problema de busca

Como resultado, pode-se encontrar um:

- Ótimo local
- Ótimo global



Definindo um problema de busca

As maneiras de analisar as soluções podem ser definidas através de políticas de:

- **Exploração**

Que consiste em explorar o espaço de busca de forma local

- **Exploração**

Que consiste em explorar o espaço de busca de forma global

Algoritmos

Hill Climbing

Hill Climbing (Subida da Colina)

Inspirado na busca de um alpinista pelo ponto mais alto de uma montanha



Hill Climbing (Subida da Colina)

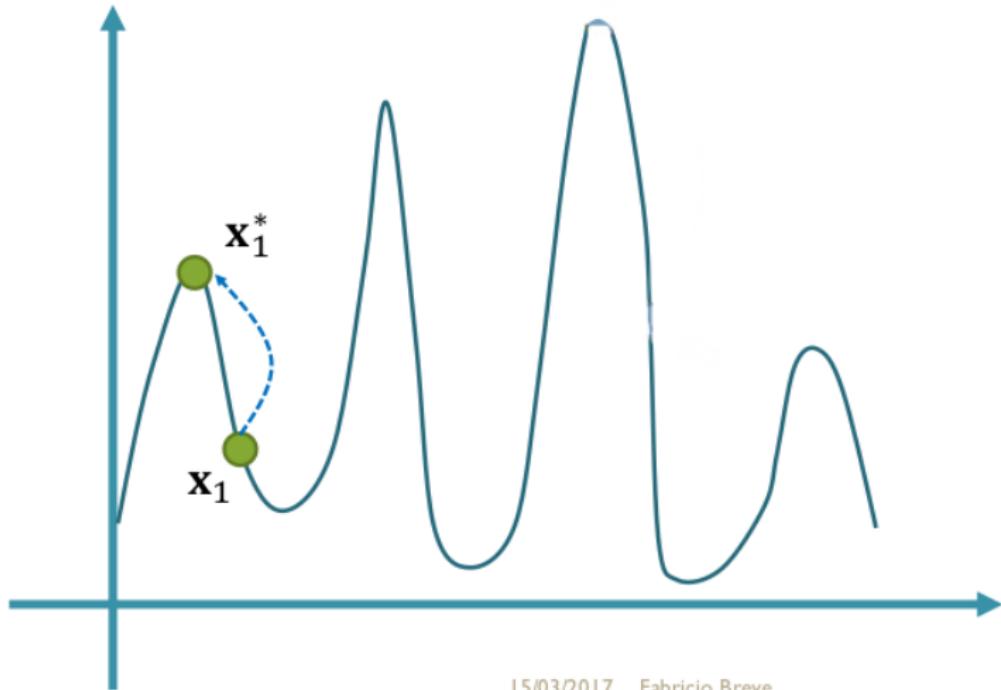
- Método de busca local que usa um aperfeiçoamento iterativo
- Aplicado a um único ponto no espaço de busca

Implementação Hill Climbing (Subida da Colina)

Funcionamento:

- Primeiramente, é definido o número máximo de iterações do algoritmo
- Em seguida, é gerado um ponto aleatório no espaço de busca
- A cada iteração, o alpinista recebe uma sugestão aleatória de movimento
- Só são aceitas sugestões que melhorem o estado atual

Implementação Hill Climbing (Subida da Colina)



15/03/2017 Fabricio Breve

Implementação Hill Climbing (Subida da Colina)

Implemente o Algoritmo Hill Climbing (sem usar o ChatGPT)

Hill Climbing (Subida da Colina)

Qual o problema deste algoritmo?

Hill Climbing (Subida da Colina)

Os problemas deste algoritmo são:

- Normalmente termina em soluções ótimas **locais**
- Não há informação sobre a distância para a ótima global
- O ótimo encontrado depende da configuração inicial
- Não é possível calcular um limite de tempo para o algoritmo

Hill Climbing (Subida da Colina)

Que tipo de melhoria poderia ser feita para que os problemas citados fossem minimizados?

Simulated Annealing

Simulated Annealing (Recozimento Simulado)

Inspirado no processo de recozimento de sistemas físicos



Simulated Annealing (Recozimento Simulado)

- Temperatura do material é elevada para que ocorra o derretimento e seus átomos possam se mover livremente
- Temperatura do sistema derretido é lentamente diminuída para que a cada nova temperatura os átomos possam se movimentar o suficiente para adotar uma orientação mais estável
- Se a temperatura for diminuída suficientemente devagar, os átomos irão repousar na orientação mais estável, produzindo um cristal

Implementação Simulated Annealing (Recozimento Simulado)

- Semelhante ao Hill Climbing
- São feitas perturbações se baseando na temperatura atual
- Temperaturas altas permitem uma maior exploração
- Temperaturas mais baixas abusam da explotação
- Resolve o maior problema do Hill Climbing

Implementação Simulated Annealing (Recozimento Simulado)

<https://i.makeagif.com/media/3-09-2014/XT0sjr.gif>

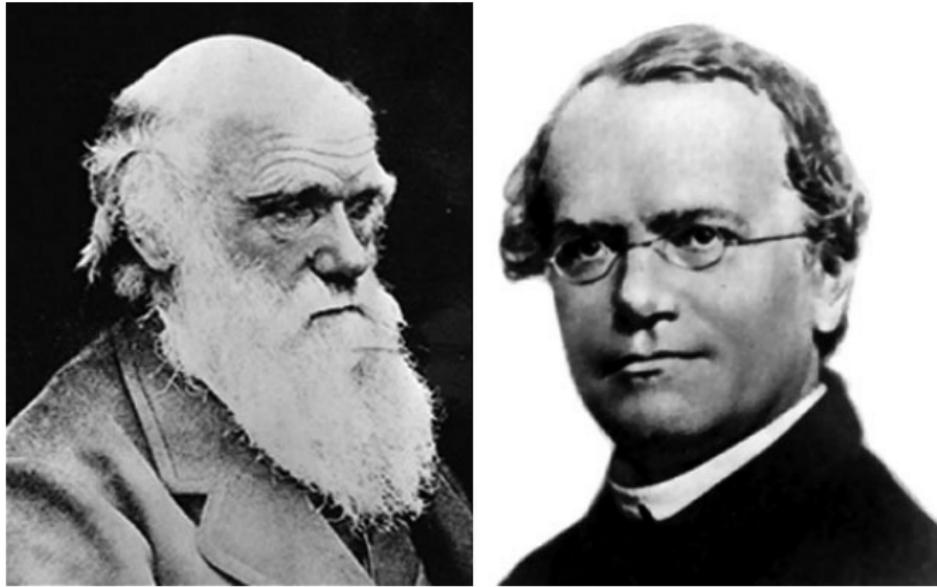
Implementação Hill Climbing (Subida da Colina)

Implemente o Algoritmo Simulated Annealing (sem usar o ChatGPT)

Algoritmos Genéticos

Algoritmos Genéticos

Inspirados na biologia evolutiva, envolvendo conceitos de hereditariedade, mutação, seleção natural e crossover.



Biologia Evolutiva

Sub-área da Biologia que tem como foco estudar a origem, descendência e evolução das espécies

Quem veio primeiro, a galinha ou o ovo? | Nerdologia



Biologia Evolutiva

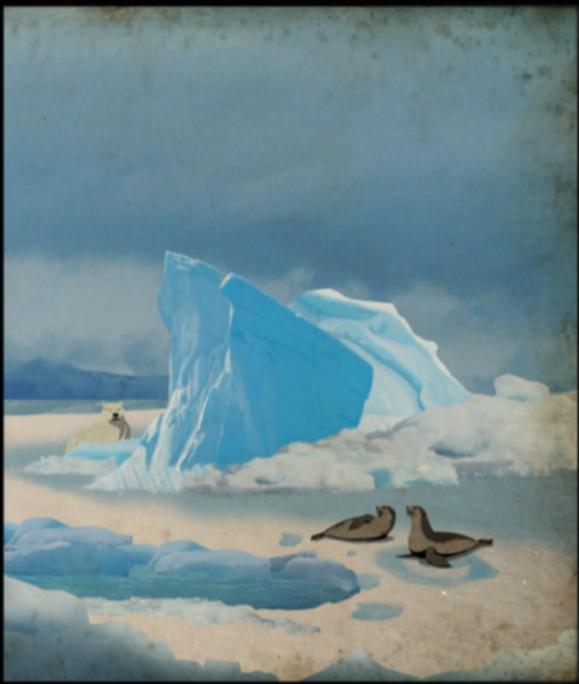
Exemplo clássico de evolução:



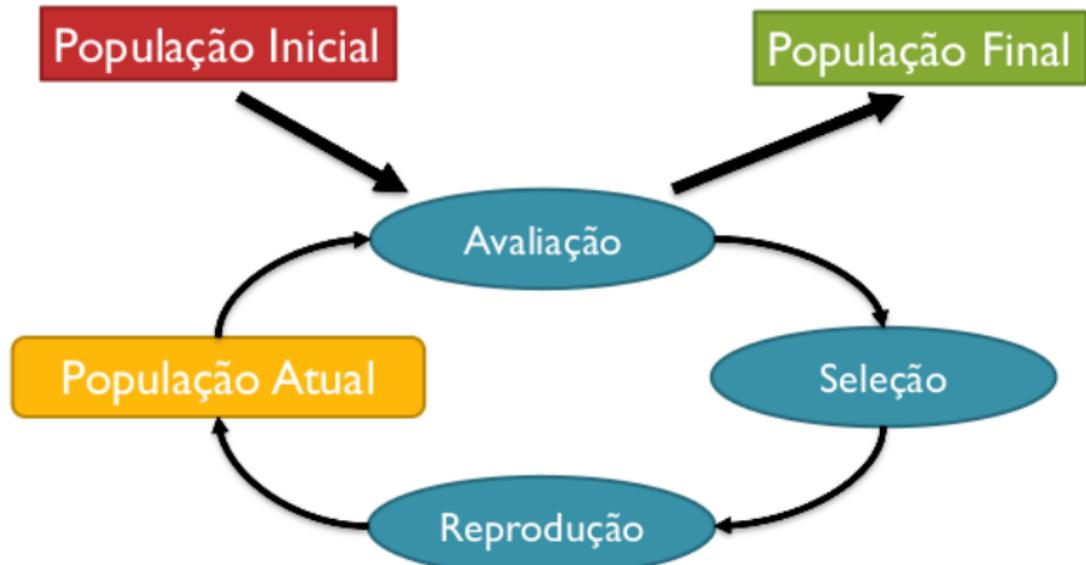
- Durante o dia, mariposas descansam no tronco das árvores
- Sua cor é camuflagem natural contra predadores
- Nas áreas mais industrializadas, predominam mariposas escuras
- Revolução industrial na Inglaterra causou a diminuição da população clara e aumentou a população escura

Biologia Evolutiva

Outro exemplo, da série cosmos, retratando a seleção natural e artificial:



Algoritmos Evolutivos



O algoritmo evolutivo padrão é:

- Genérico, iterativo e probabilístico
- Mantém uma população **P** de **N** indivíduos em cada iteração **t**
- Cada indivíduo é uma possível solução para o problema
- Indivíduos são avaliados de acordo com sua aptidão (obtida através da função de aptidão)

Algoritmos Evolutivos

A evolução ocorre a partir de:

- Cruzamento entre diferentes indivíduos (mais aptos)
- Variações genéticas que ocorrem por mutação (urso)

Algoritmos Evolutivos

Principais fatores fazem um algoritmo evolutivo:

- Indivíduo
- Avaliação
- Seleção
- Cruzamento
- Mutação
- Critério de parada

Indivíduo

- Possível solução para um dado problema
Também chamado de cromossomo ou string
- Codificado como vetor de características
- Cada indivíduo é associado a seu valor de aptidão
- Conjunto de indivíduos forma a população
- Tradicionalmente representados por vetores binários

Avaliação

- Habilidade do indivíduo para resolver o problema
- Ou, probabilidade do indivíduo de sobreviver para a próxima geração
- Função de avaliação/aptidão muda para cada tipo de aplicação

Seleção

- Forma de escolher os indivíduos mais aptos
- Dependendo da política, se baseia em porcentagens
- Indivíduos selecionados podem reproduzir e gerar descendentes

Seleção

Principais tipos de seleção:

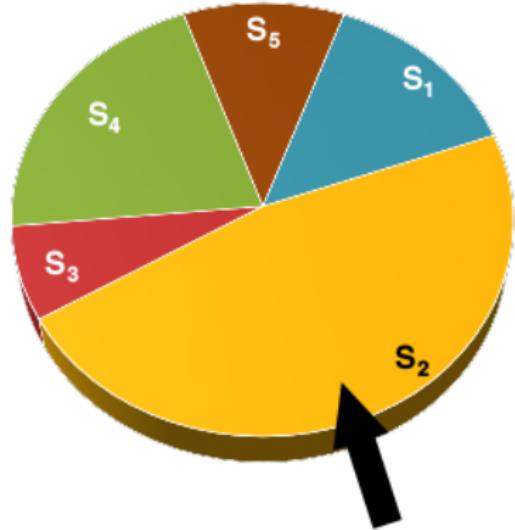
- Seleção por roleta
- Seleção por torneio
- Seleção por amostragem universal estocástica

Seleção por roleta

- Método mais simples e utilizado
- Escolhe indivíduos para a próxima geração por meio de um sorteio
- Cada indivíduo representa uma fatia da roleta

Seleção por roleta

Indivíduo (S_i)	Aptidão $f(S_i)$	Aptidão Relativa
S_1 10110	2,23	0,14
S_2 11000	7,27	0,47
S_3 11110	1,05	0,07
S_4 01001	3,35	0,21
S_5 00110	1,69	0,11



Seleção por torneio

- Coloca indivíduos em confronto
- A variável **n** define quantos indivíduos participam de cada disputa
- Favorece a pressão seletiva (grau de favorecimento dos mais aptos)

Seleção por torneio

Problema:

- Muita pressão seletiva atrapalha a convergência do algoritmo
- Falta diversidade, a população fica com poucas soluções

Seleção por torneio

Indivíduo (S _i)	Aptidão f(S _i)	Aptidão Relativa
S ₁ 10110	2,23	0,14
S ₂ 11000	7,27	0,47
S ₃ 11110	1,05	0,07
S ₄ 01001	3,35	0,21
S ₅ 00110	1,69	0,11

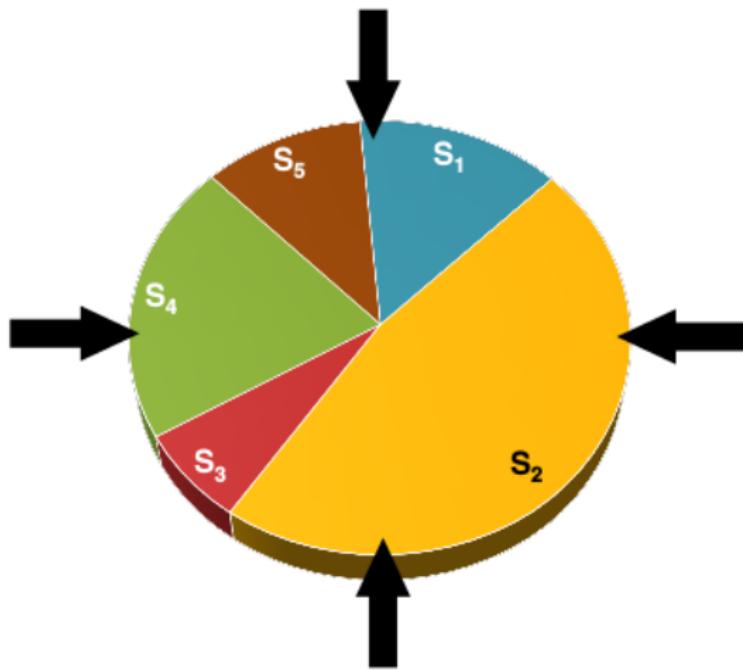
Com n=3

Candidatos			Vencedor
S ₁	S ₂	S ₅	S ₂
S ₂	S ₄	S ₅	S ₂
S ₅	S ₁	S ₃	S ₁
S ₄	S ₅	S ₃	S ₄
S ₃	S ₁	S ₅	S ₁

Seleção amostragem universal estocástica

- Também utiliza roleta, mas dá espaço para menos favorecidos
- Ao invés de uma, utiliza n agulhas para escolher indivíduos

Seleção amostragem universal estocástica



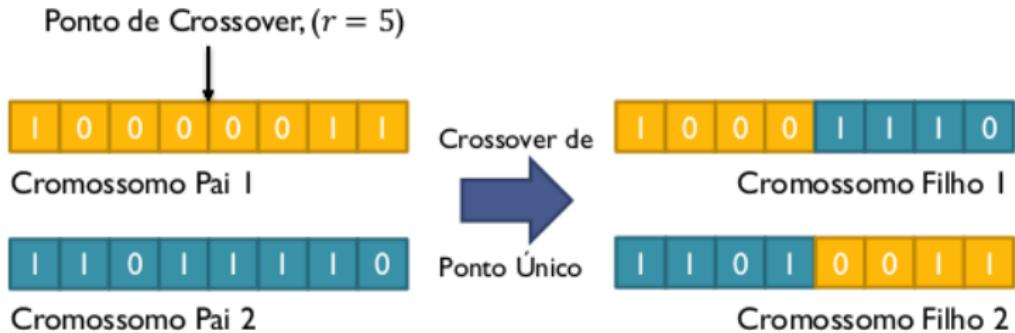
Operadores Genéticos

- Crossover
- Mutação
- Elitismo

Crossover

- Cruzamento entre indivíduos
- Filhos herdam partes das características dos pais
- Envolve os conceitos de hereditariedade

Crossover



Crossover

- Forma mais rápida de explorar o espaço de busca
- Nem todos indivíduos irão cruzar, isso depende de uma taxa
- A taxa de crossover deve SEMPRE ser maior que a de mutação

Mutação

- Permite a introdução e manutenção da DIVERSIDADE
- Aplicado a cada indivíduo após o crossover
- Normalmente é uma taxa pequena (por que?)
- Assegura a chance de explorar TODO espaço de busca

Mutação



Cromossomo Original

Mutação



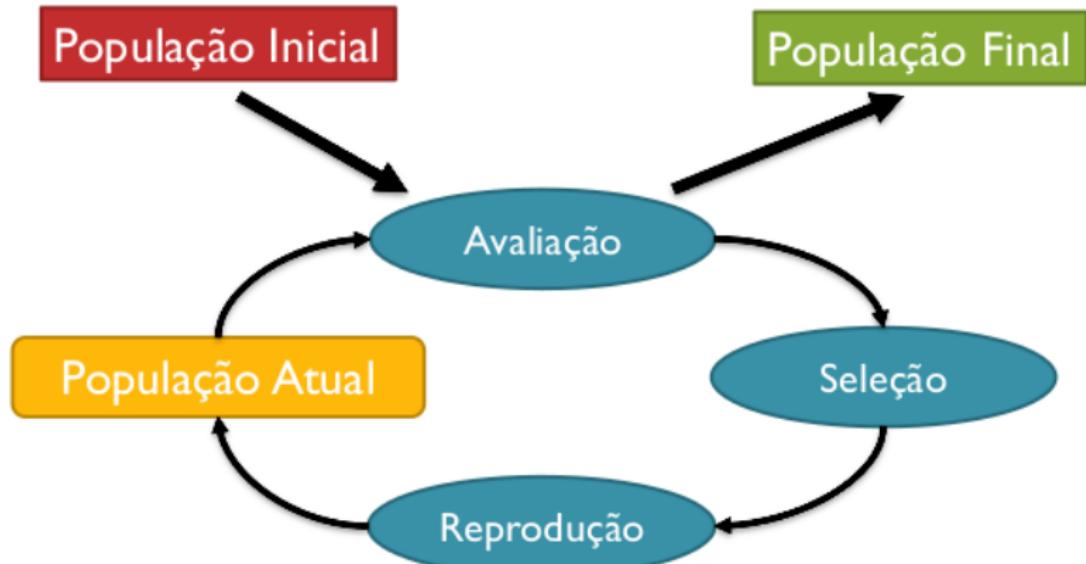
Cromossomo Mutado

- Seleção artificial dos algoritmos genéticos
- Passa indivíduos mais aptos para próxima geração
- Evita modificações no indivíduo por operadores genéticos

Critérios de parada

- Convergência
- Tempo de execução
- Número de gerações
- Valor mínimo/médio/máximo de aptidão

Funcionamento Algoritmo Genético



Exemplo de Funcionamento Algoritmo Genético

Exemplo de reconhecimento de padrões:

- Evoluir uma população para que a mesma reconheça o caractere 1



(a)

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(b)

Exemplo de Funcionamento Algoritmo Genético

População inicial

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

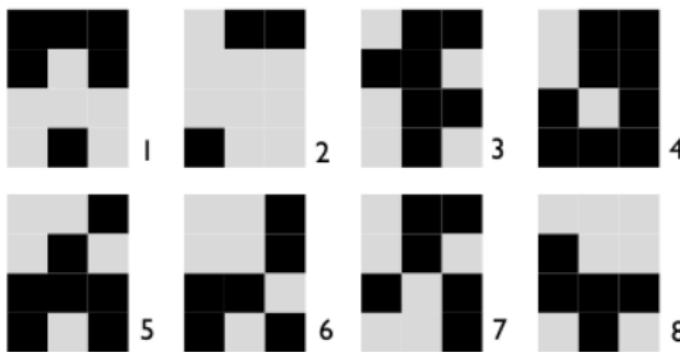


Figura: $h = [6, 7, 9, 5, 5, 4, 6, 7]$

Exemplo de Funcionamento Algoritmo Genético

Função de aptidão?

Exemplo - Função de aptidão

Distância de Hamming

- Contar o número de bits corretos na matriz
- $h = [6, 7, 9, 5, 5, 4, 6, 7]$
- Indivíduo ideal teria aptidão 12 (todos os bits corretos)

Exemplo - Seleção

Seleção por roleta:

- Somar a aptidão de todos os indivíduos
- Multiplicar a aptidão de cada indivíduo por 360 e dividir pela soma
- Gerar um número aleatório entre 0 e 360 e escolher o indivíduo sorteado

Exemplo - Seleção

- Assuma que os valores sorteados aleatoriamente foram:
 $s = [230, 46, 175, 325, 275, 300, 74, 108]$
- Portanto a população P será composta por
 $P = [x_6, x_1, x_5, x_8, x_6, x_7, x_2, x_4]$
- Observação:
 x_3 foi extinto
 x_6 foi selecionado duas vezes

Exemplo - Reprodução

Agora temos 4 pares de indivíduos:

- x_6 e x_1
- x_5 e x_8
- x_6 e x_7
- x_2 e x_4

Exemplo - Reprodução

Agora temos 4 pares de indivíduos:

- x6 e x1
- x5 e x8
- x6 e x7
- x2 e x4

Para cada par, geramos um número aleatório **r** entre 0 e 1

- Se o número gerado for maior que a taxa de crossover, ocorre o cruzamento
- Senão, os pais originais são repetidos na próxima geração

Exemplo - Reprodução

Seja $r = [0,5; 0,7; 0,3; 0,9]$ e a taxa de crossover 0,6:

- Foi selecionado o primeiro e terceiro par para crossover
- x_6 e x_1
- x_6 e x_7

Exemplo - Reprodução

Assumindo que foram sorteados $cp = 5$ e $cp = 9$ para cada um dos pares respectivamente:



Ponto de crossover, $cp = 5$

110110001010 Cromossomo pai 1 (x_6)
00001011101 Cromossomo pai 2 (x_1)

11011011101 Cromossomo filho 1
000010001010 Cromossomo filho 2



Ponto de crossover, $cp = 9$

110110001010 Cromossomo pai 1 (x_6)
100101010110 Cromossomo pai 2 (x_7)

110110001110 Cromossomo filho 1
000011010010 Cromossomo filho 2

Exemplo - Reprodução

Com isso, a nova população é:

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Exemplo - Mutação

Assumindo a probabilidade de mutação como 0.02, para cada posição da matriz é gerado um número aleatório e se tal número seja menor ou igual a 0.02, a mutação ocorre

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

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

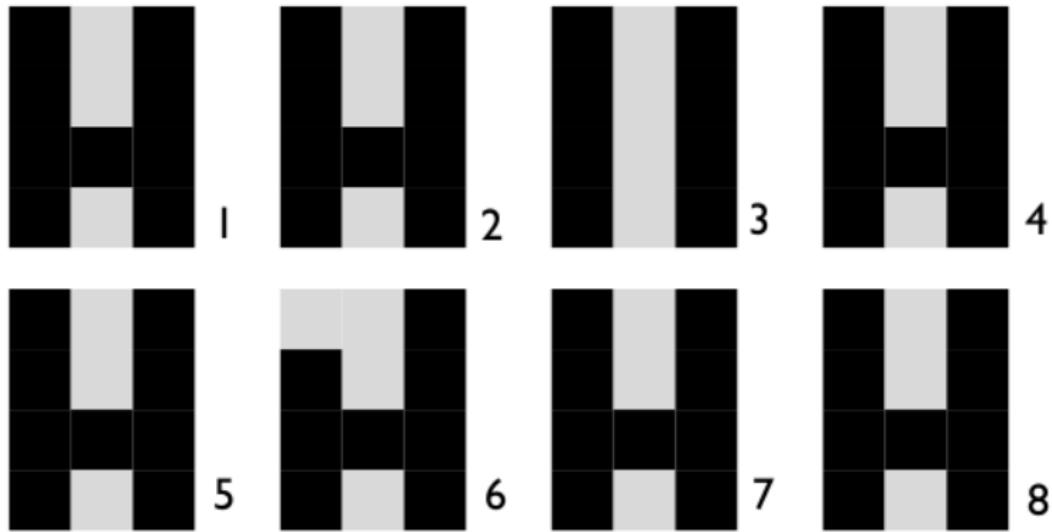
Exemplo - Evolução

Avaliando a aptidão atual da população, tem-se:

- 55 de aptidão total, sendo 6,875 de aptidão média
- A aptidão inicial era 47, sendo a média 5,875
- Da mesma forma, a distância de Hamming se aproxima mais de zero

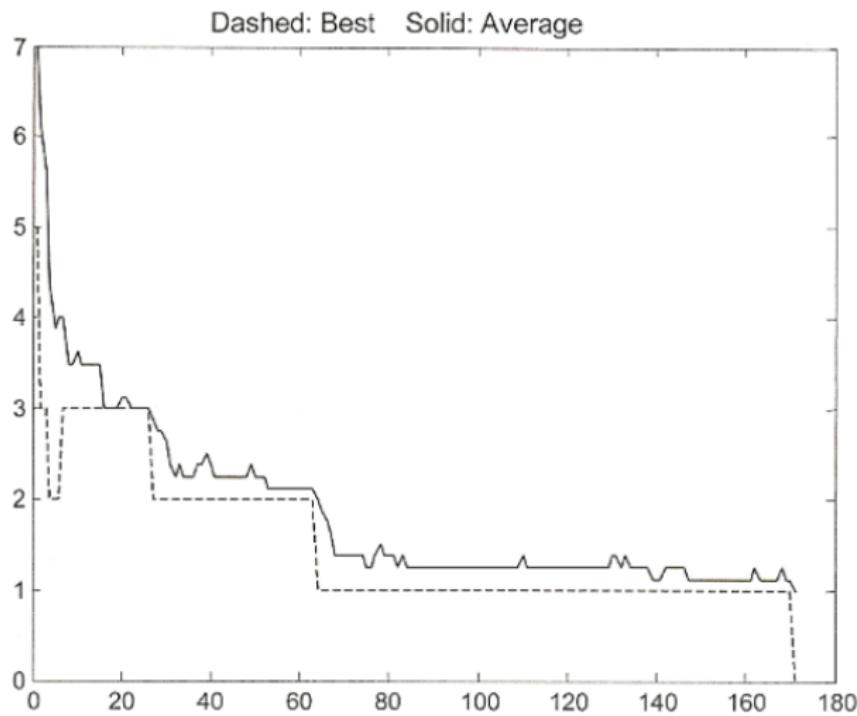
Exemplo - Evolução

Ao fim de 171 gerações, temos:



Exemplo - Evolução

Evolução da distância de hamming média e do melhor indivíduo:



Exemplo Algoritmo Genético

BoxCar 2D

[Home](#) | [Designer](#) | [Best Cars](#) | [Forum](#) | [News](#) | [FAQ](#) | [The Algorithm](#) | [Versions](#) | [Contact](#)

Computation Intelligence Car Evolution Using Box2D Physics (v3.2)

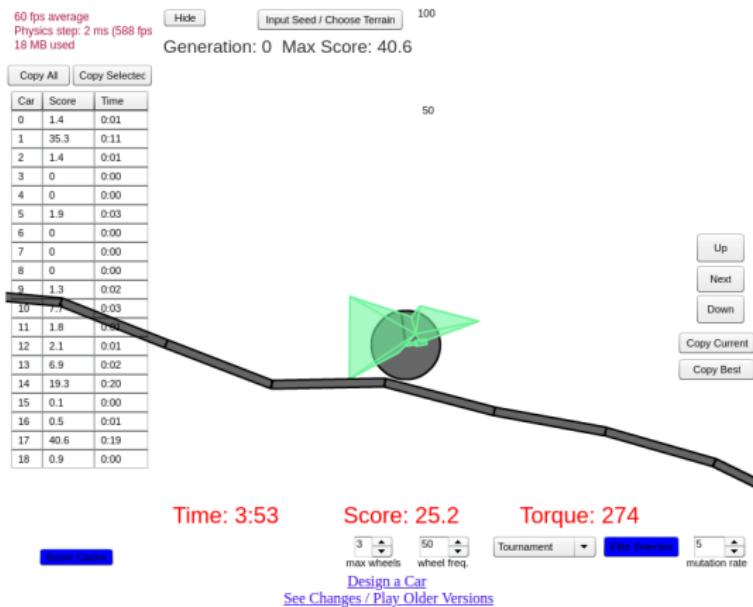


Figura: https://rednuht.org/genetic_cars_2/

E-mail: caio.carneloz@fc.unesp.br