
Aula 03:

Computação Evolutiva e Conexão – Algoritmos Genéticos

Prof. Hugo Puertas de Araújo
hugo.puertas@ufabc.edu.br
Sala: 509.2 (5º andar / Torre 2)

■ Agenda

- Algoritmos genéticos
- Operadores genéticos
 - ❖ ex.: Maximização de função
 - ❖ ex.: Caixeiro viajante
- Efeito da mutação nas populações
- Bibliotecas de GA em Python

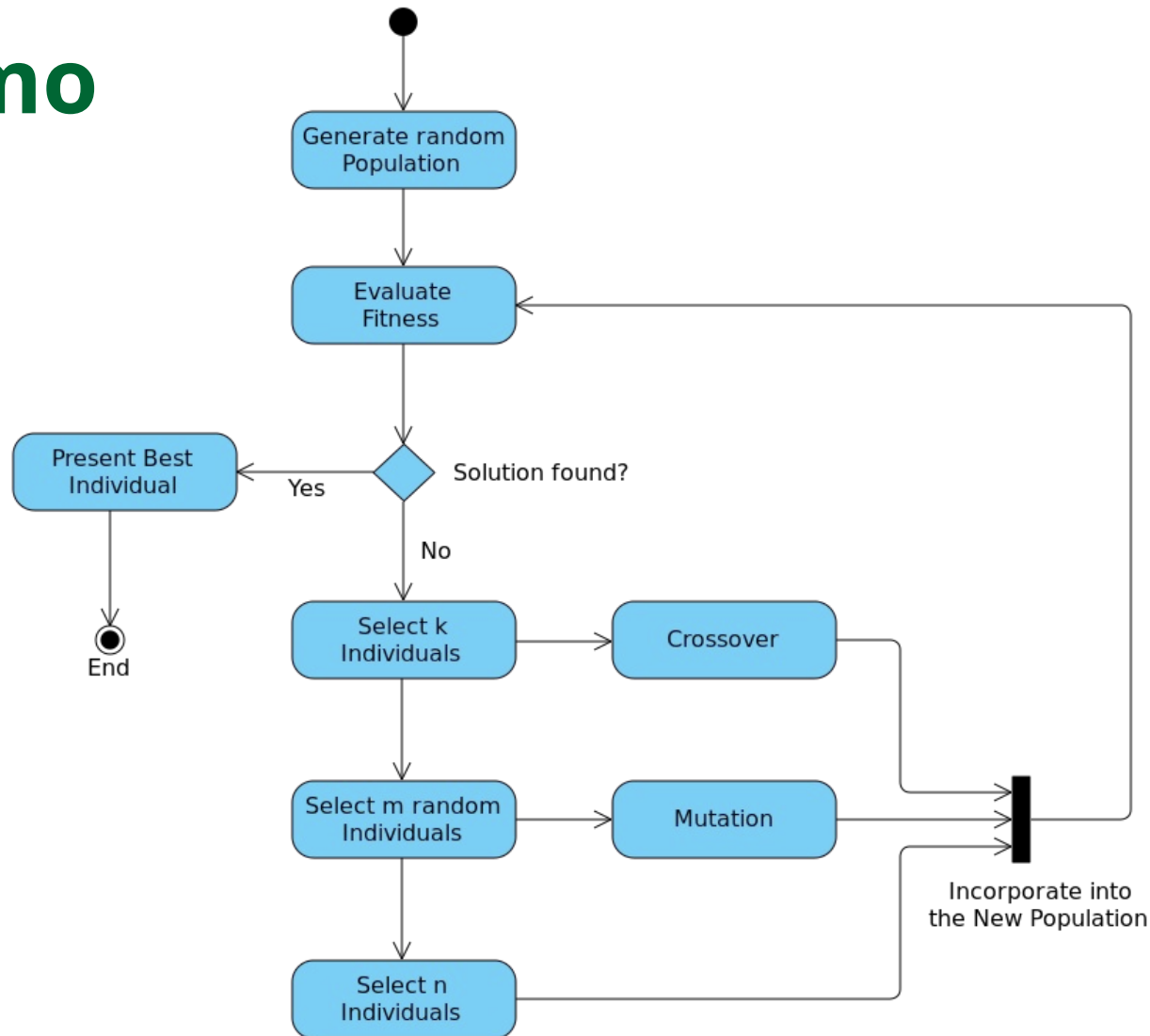
Computação Evolutiva e Conexionista



■ Meta-heurísticas evolucionárias

- **Algoritmos Genéticos**: proposto por Holland com o objetivo de estudar os fenômenos de adaptação
- **Estratégias Evolutivas**: introduzido por Rechenberg com o intuito de otimizar parâmetros de uma função não-linear
- **Programação Evolutiva**: representação de uma solução como uma máquina de estados finitos, proposta por Fogel et al.
- **Programação Genética**: Proposta por Koza com o intuito de evoluir programas de computador

O algoritmo genético



■ Características necessárias p/ uma população

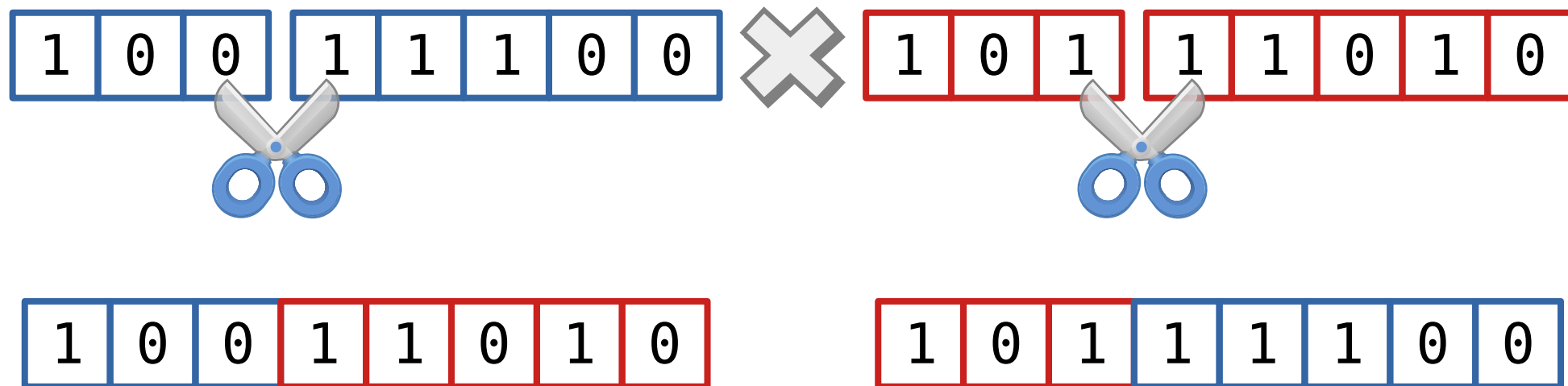
- Autorreplicação (Self-replication) – Reprodução / Recombinação
- Plano interno (Blueprint) – Genótipo vs Fenótipo
- Herança (Inherit blueprint)
- Mutação (Mutation)
- Seleção (Selection) – Baseada em Fitness

Operadores genéticos

- **Recombinação:** seleciona dois indivíduos parentais e os combina para gerar filhos com características mistas.
Baseado no fitness de cada pai
- **Mutação:** altera aleatoriamente uma característica de um indivíduo selecionado aleatoriamente
- **Seleção:** escolhe um indivíduo de uma geração para compor a próxima geração (diretamente ou via recombinação) com base em seu fitness

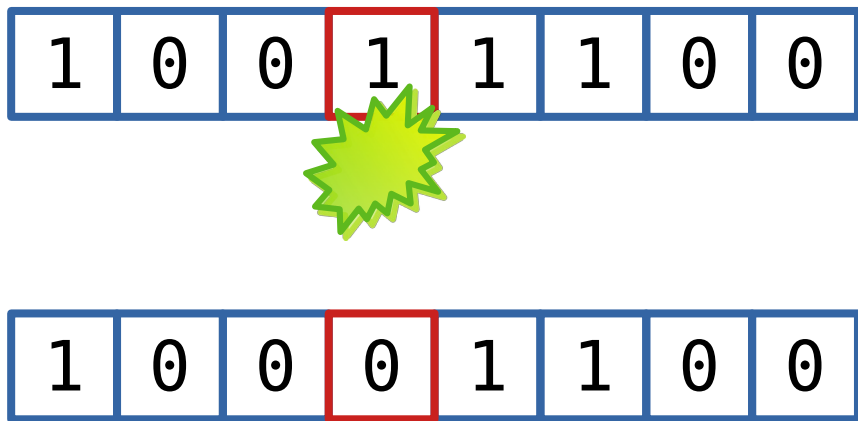
Operadores genéticos

Recombinação



Operadores genéticos

Mutação



Operadores genéticos

■ Seleção



1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

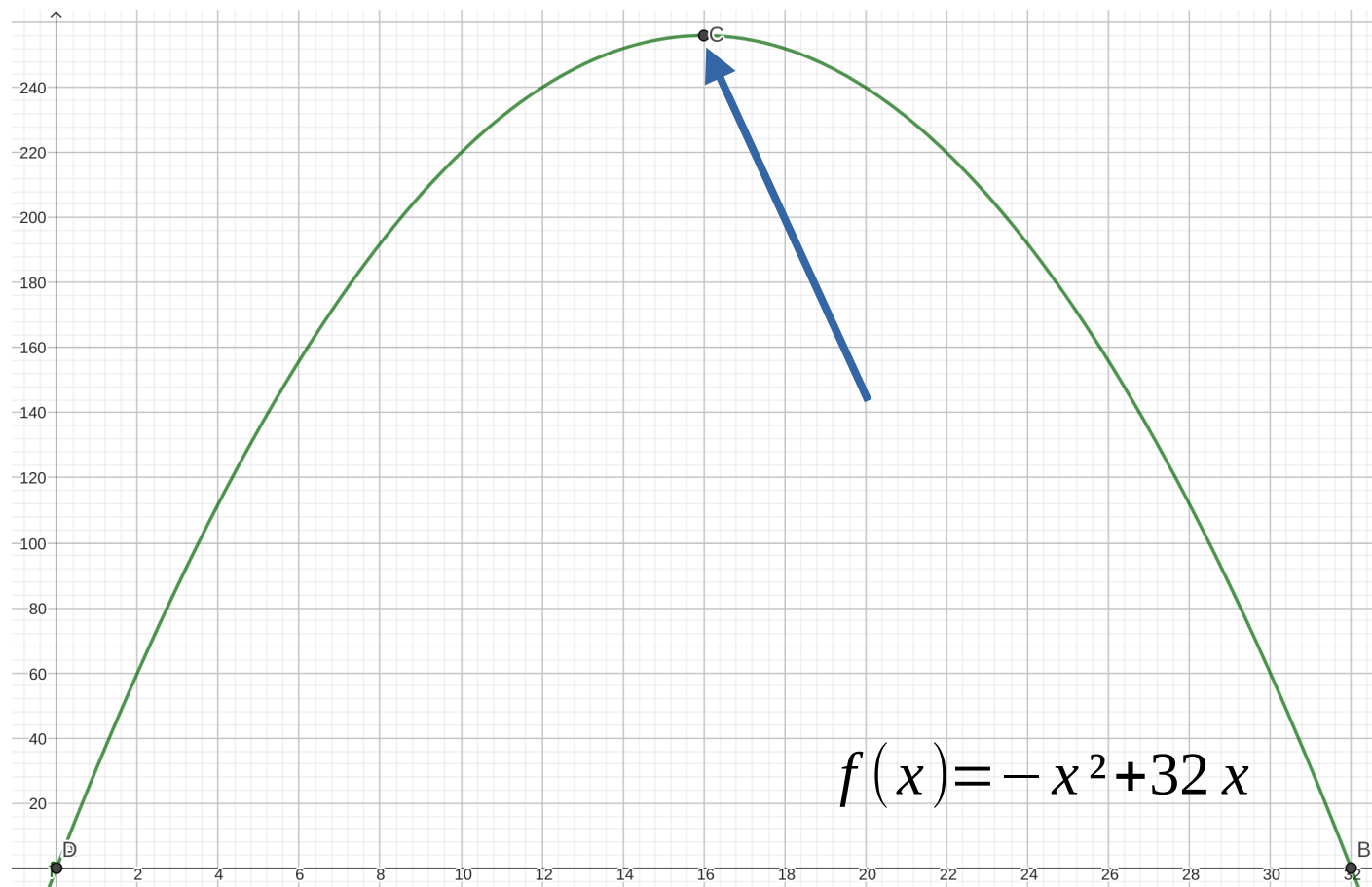


1	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---



GA: Localizar máximo da função

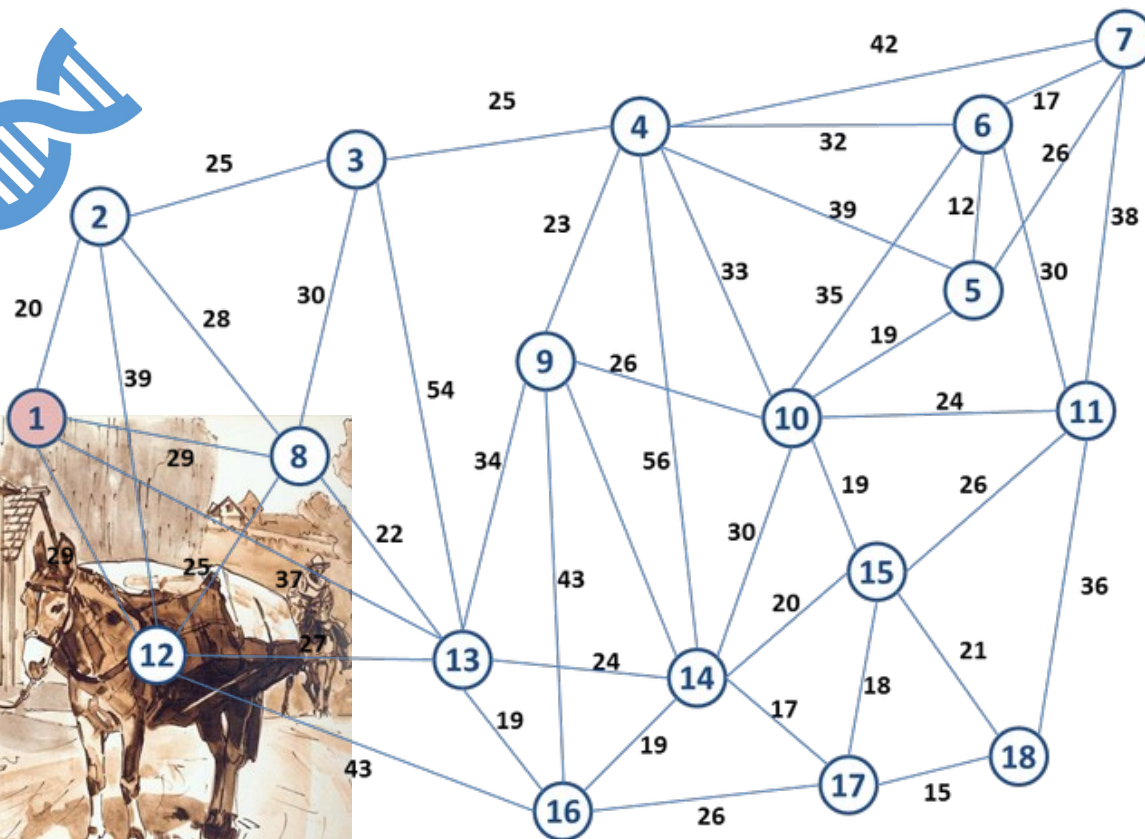
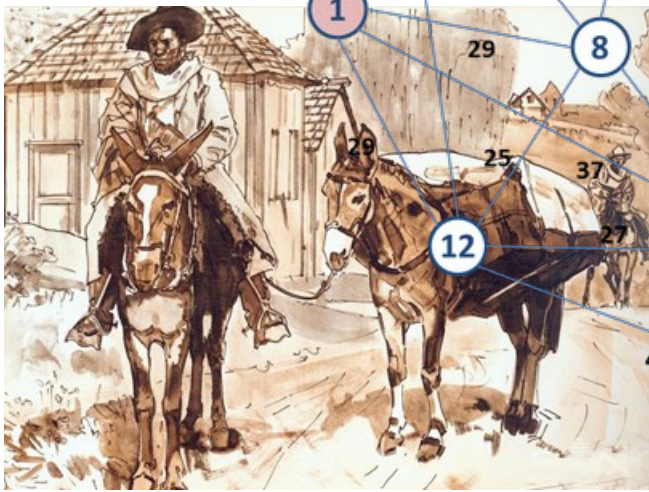
Usando GA, determinar o ponto de máximo da função ao lado, definida para x inteiro.



$$f(x) = -x^2 + 32x$$

https://colab.research.google.com/drive/1_GqNkGffp_qbWglS6lD4VMmObq_l1vuZ?usp=sharing

Problema prático: Caixeiro viajante



#	$[(n-1)!]/2$
3	1
4	3
5	12
6	60
7	360
8	2520
9	20160
10	181440
11	1814400
12	19958400
13	239500800
14	3113510400
15	43589145600
16	653837184000
17	10461394944000
18	177843714048000
19	3201186852864000
20	6,08225502044E+16
25	3,10224200867E+23
30	4,42088099687E+30

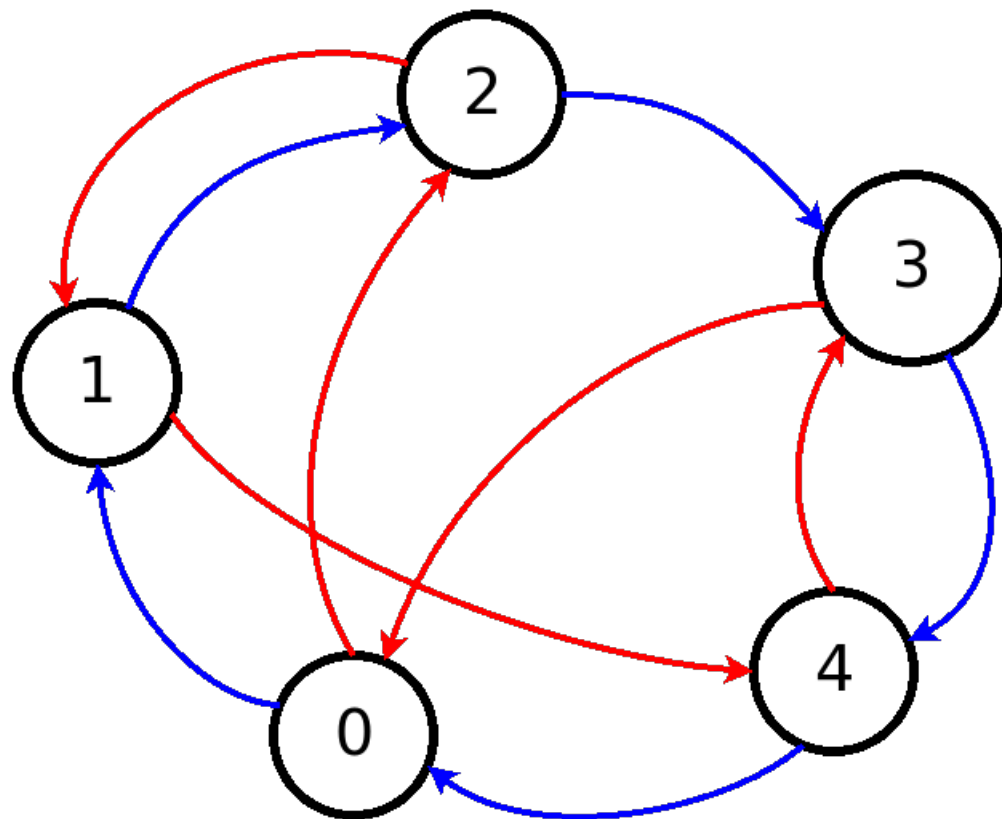
■ Conceitos dentro do contexto: TSP

- Gene: uma cidade (representada como coordenadas (x, y))
- Indivíduo (aka “cromossomo”): uma rota satisfazendo as condições
- População: uma coleção de possíveis rotas (i.e., coleção de indivíduos)
- Pais: duas rotas a serem combinadas para criar uma nova rota (filho)
- Mating pool: uma coleção de pais que são usados para criar a próxima população (nova geração de rotas)
- Fitness: uma função que avalia o quão boa cada rota é (o quão curta é a distância)
- Mutação: uma forma de introduzir variação na população ao trocar aleatoriamente duas cidades em uma rota
- Elitismo: uma forma de carregar os melhores indivíduos para a próxima geração

■ Representação cromossossomial

[0, 1, 2, 3, 4]

[0, 2, 1, 4, 3]



■ Seleção de indivíduos (cromossomos)

■ Sorteio por roleta

- ❖ O fitness de cada indivíduo determina sua probabilidade de seleção

■ Sorteio por torneio

- ❖ Sorteiam-se alguns indivíduos da população e daí escolhe-se o de maior fitness

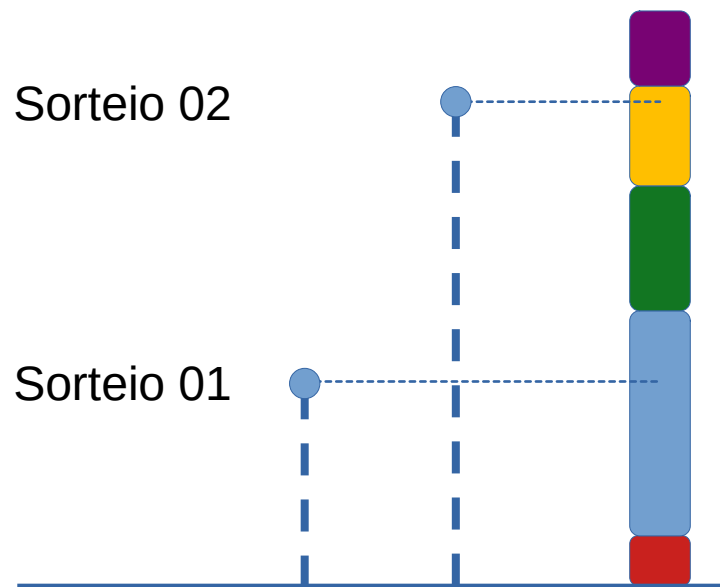
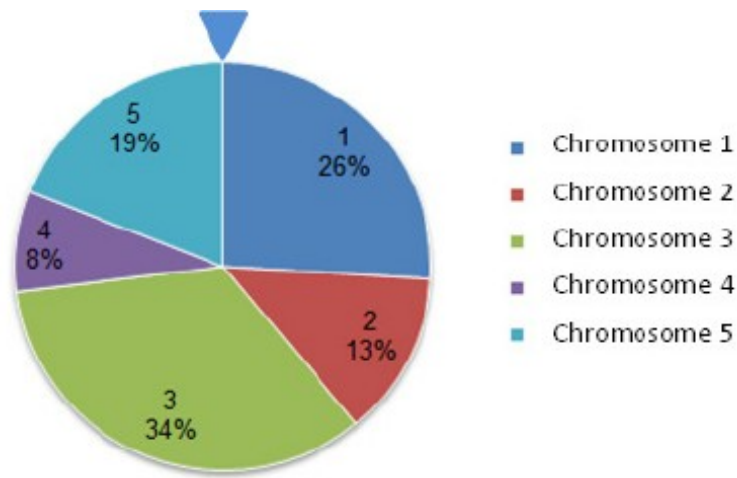
■ Elitismo

- ❖ Selecionam-se os indivíduos de maior fitness para integrarem a nova população

■ Seleção de indivíduos (cromossomos)

■ Sorteio por roleta

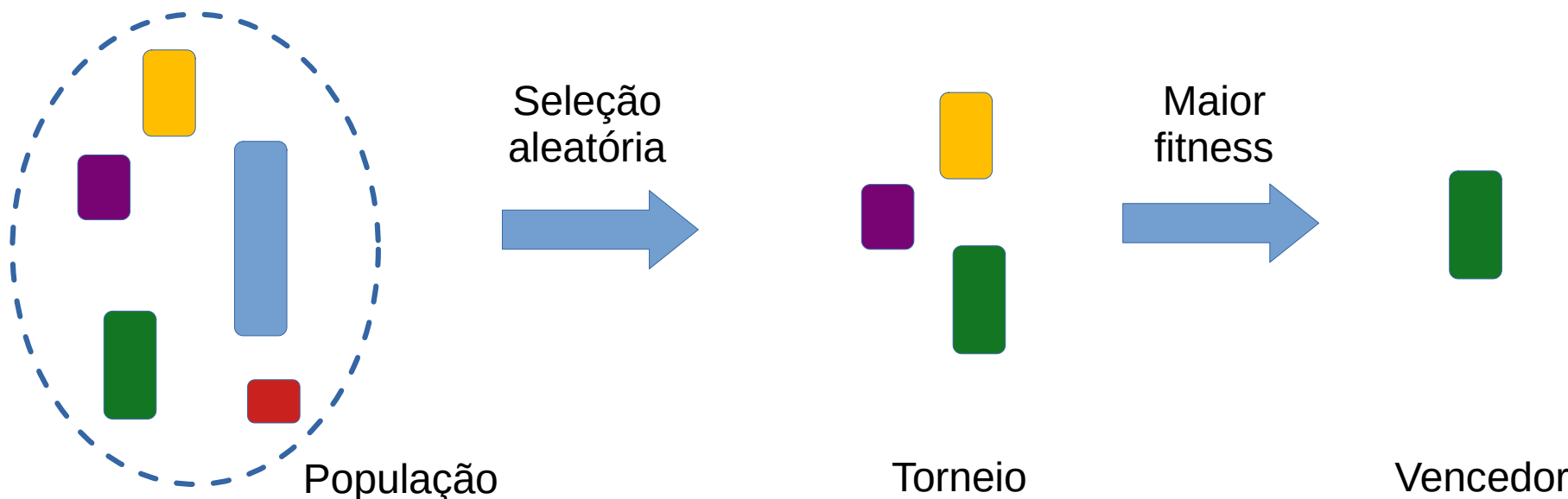
- ❖ O fitness de cada indivíduo determina sua probabilidade de seleção



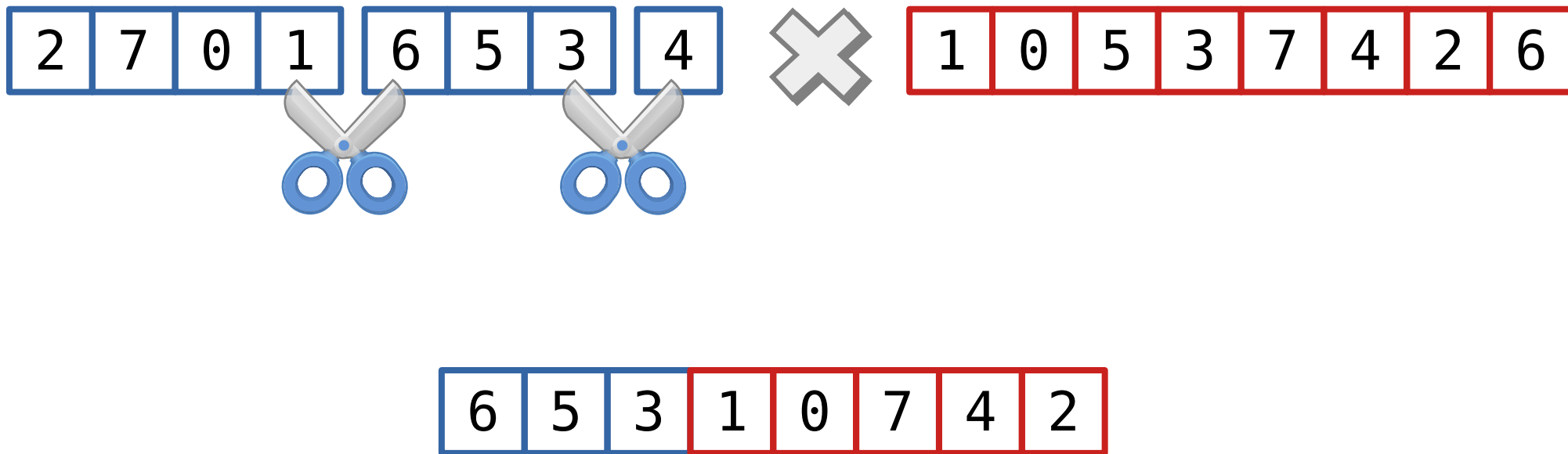
■ Seleção de indivíduos (cromossomos)

■ Sorteio por torneio

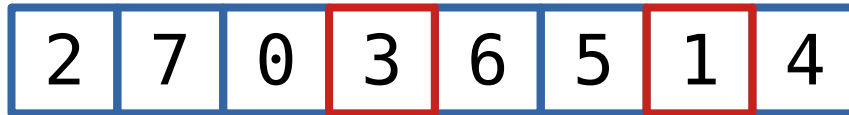
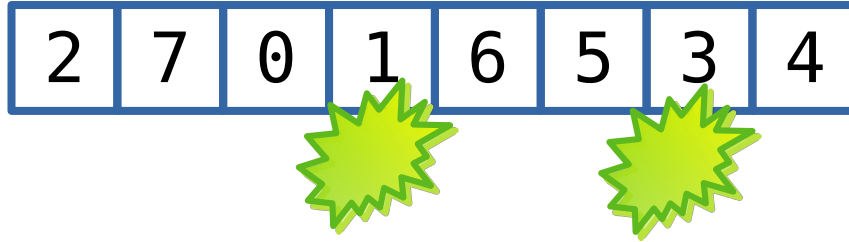
- ❖ Sorteiam-se alguns indivíduos da população e daí escolhe-se o de maior fitness



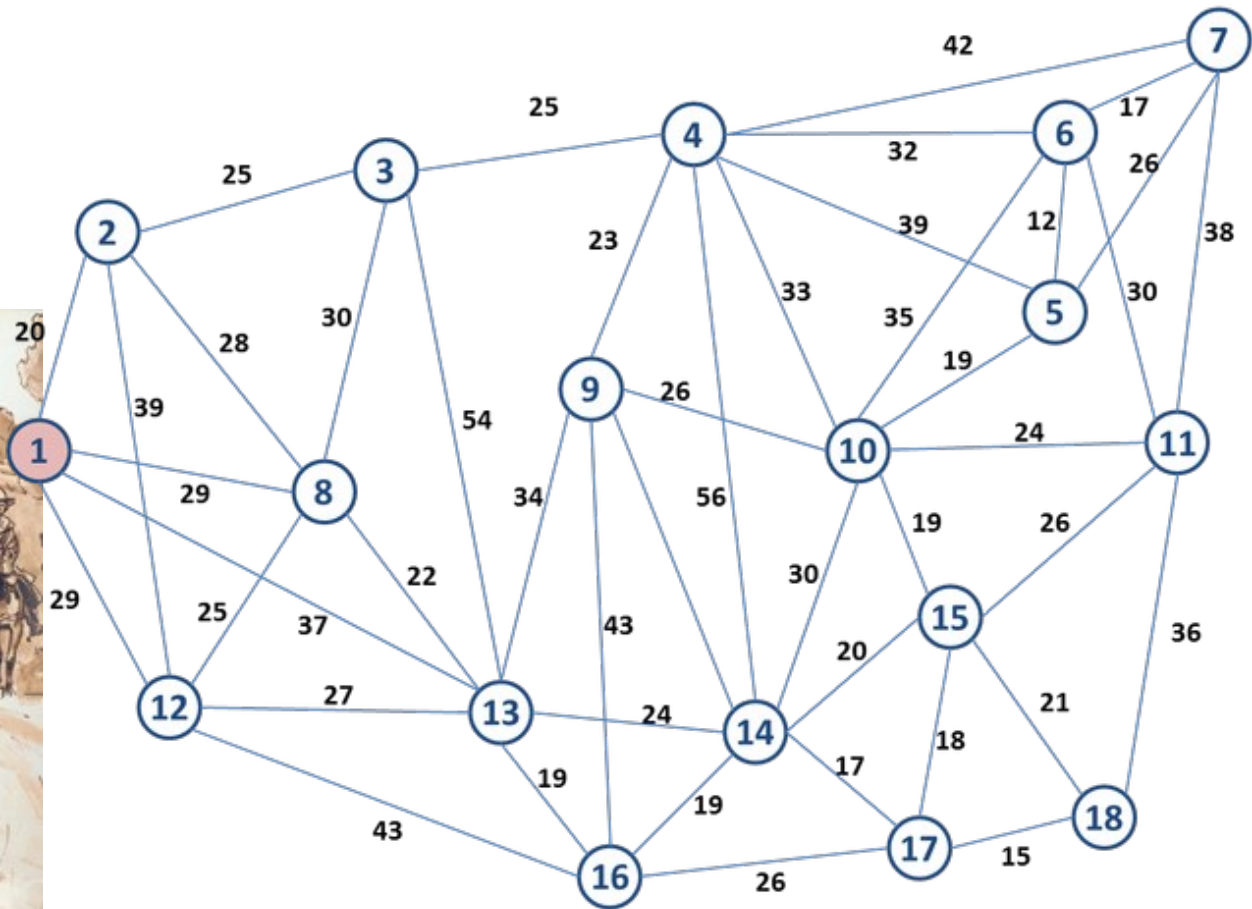
Crossover



Mutação



TSP & GA

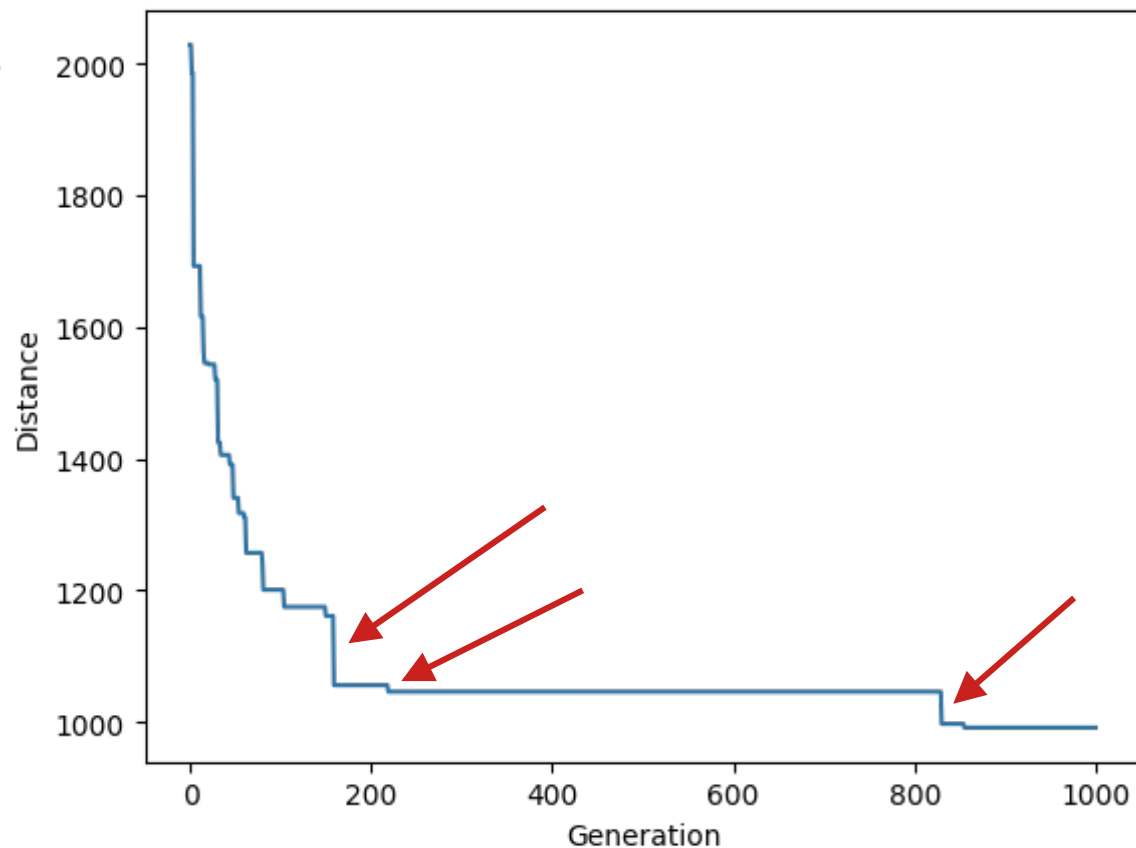


<https://colab.research.google.com/drive/1aN7KDJaqKGX1yGKhNyHvILbltenXIDs8?usp=sharing>

■ Introdução de inovações

■ Recombinação e Mutação

X-MEN



Bibliotecas de GA em Python



DISTRIBUTED
EVOLUTIONARY
ALGORITHMS IN
PYTHON

