



**Universidade Federal de Sergipe**

# **Problema do Caixeiro Viajante (PCV)**

30 de setembro de 2025

# Equipe

- Reinan Gabriel Dos Santos Souza
- João Rabelo de Araujo Filho
- Paulo Ezequiel Dias Dos Santos

# Sumário

1. Introdução
2. Definição do problema
3. A solução
4. Como Algoritmo Genético resolve o problema
5. Código/Experimento
6. Resultados
7. Referências

# Introdução

O **Problema do Caixeiro Viajante (TSP)** é um dos problemas clássicos em **teoria dos grafos e otimização combinatória**. Ele pode ser formulado da seguinte forma: dado um conjunto de cidades e as distâncias entre cada par delas, encontrar o **menor caminho possível** que permita ao caixeiro visitar todas as cidades **uma única vez** e retornar ao ponto de partida.

Matematicamente, trata-se de encontrar um **ciclo Hamiltoniano de custo mínimo** em um grafo ponderado.

Esse problema é notoriamente difícil (NP-difícil), pois o número de possíveis rotas cresce **fatorialmente** com a quantidade de cidades. Por isso, não há um algoritmo eficiente conhecido que resolva o TSP de forma exata para instâncias muito grandes.

# Definição do problema

O PCV é um problema de otimização combinatória.

- **Objetivo:** Encontrar o caminho mais curto que visita cada vértice (cidade) de um grafo exatamente uma vez e retorna ao vértice de origem.
- **Requisitos:** Dado um conjunto de  $n$  vértices e as distâncias (pesos) entre eles.
- **Solução:** Em um grafo completo com pesos nas arestas, queremos o ciclo de menor custo que passa uma única vez por cada vértice e retorna ao início.

# Definição do problema

Uma forma simples de entender a dificuldade do PCV é contar as rotas possíveis.

- O caixeiro pode começar em qualquer uma das  $n$  cidades.
- Dali ele pode partir para qualquer uma das outras  $n - 1$  cidades.
- Da segunda, ele pode partir para qualquer uma das  $n - 2$  cidades restantes.
- E assim por diante, até chegar na última cidade.

Isto nos dá um número de opções igual a:

$$n \cdot (n - 1) \cdot (n - 2) \cdots 2 \cdot 1 = n!$$

Isso gera um número fatorial que cresce muito rapidamente.

# Definição do problema

- Por exemplo, se tivermos 100 cidades temos  $10^{158}$  opções.
- Se pudermos testar um bilhão de soluções por segundo.
- Demoraríamos um tempo igual a  $10^{140}$  anos para encontrar a melhor solução.

Isso é maior que a idade do universo, estimada em  $10^{10}$ .

# Definição do problema

O problema do caixeiro viajante, apesar de seu nome, é análogo a muitos desafios da vida real.

- **Logística:** Otimização de rotas para entregas de mercadorias, serviços de transporte ou até mesmo coleta de lixo.
- **Roteamento em redes:** Otimização de pacotes de dados em redes de computadores, buscando o caminho mais eficiente entre os roteadores.
- **Inspeção com drones:** Planejar a rota que visita todos os pontos de inspeção com o menor percurso total antes de retornar à base.

# Solução

O problema do Caixeiro Viajante possui diversas abordagens de resolução que podem ser divididas em abordagens **Exatas, Aproximativas e Heurísticas/Meta-heurísticas**.

**Algoritmo Genético** - Inspirado na evolução natural. Não garante uma solução ótima, mas encontra rotas eficientes em pouco tempo.

- Solução Escalável
- Qualidade na solução
- Flexibilidade
- Equilibrado

# Como o Algoritmo Genético resolve o problema

1. **Representação** - Cada rota é um cromossomo.
2. **População inicial** - Gera várias rotas aleatórias.
3. **Fitness** - Quanto menor a distância total, maior a pontuação.
4. **Seleção** - Escolhe as melhores rotas.
5. **Crossover** - Combina rotas para gerar novas soluções.
6. **Mutação** - São inseridas pequenas mudanças para evitar estagnação.
7. **Nova Geração** - Processo se repete, promovendo evolução através das gerações até que uma solução aceitável for detectada.

# Apresentação do código fonte

O código-fonte deste projeto está público no GitHub e pode ser acessado diretamente pelo navegador em:

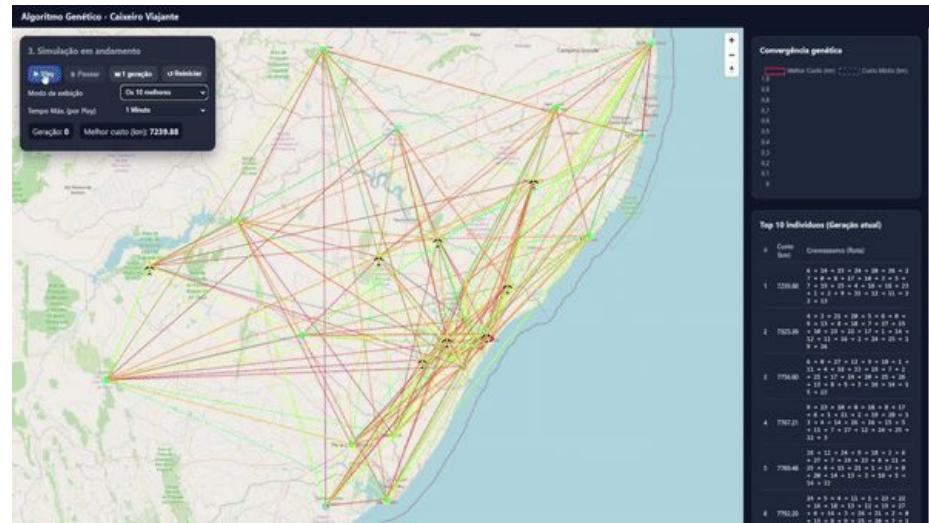
- [Github.com/ReinanHS/paa-ufs-seminario-01](https://github.com/ReinanHS/paa-ufs-seminario-01)

Faremos uma breve pausa na apresentação para abrir o repositório, executar o código e demonstrar o fluxo (Python) e as visualizações geradas, explicando passo a passo os requisitos, a instalação das dependências e o comando de execução necessário para reproduzir os resultados.

# Resultados

Os resultados demonstram a eficácia do Algoritmo Genético na resolução do problema do caixeiro viajante.

Além disso, a visualização web do projeto proporciona uma melhor compreensão do funcionamento do algoritmo e de sua evolução ao longo do tempo para identificar o caminho ótimo, sem a necessidade de percorrer todas as possibilidades.



# Referências

- LINDEN, Ricardo. Algoritmos genéticos. 3. ed. Rio de Janeiro: Ciência Moderna, 2012. ISBN 978-85-399-0195-1.
- POMPERMAYER, Mateus Platinetty; RUFINO, Samuel Mayer; VALDO, Clayton Augusto. Uso de algoritmo genético para análise de trajetos e percursos. Revista de Ubiquidade, v. 5, n. 2, 2022. Publicado em: 1 mar. 2023. Disponível em: <https://revistas.anchieta.br/index.php/RevistaUbiquidade/article/view/2031>. Acesso em: 27 set. 2025.