

Algoritmo Genético - Problema do Caixeiro Viajante (TSP)

1. Descrição

O projeto implementa um **Algoritmo Genético (AG)**, uma meta-heurística bioinspirada, para encontrar uma solução otimizada para o **Problema do Caixeiro Viajante (TSP)**. O objetivo é determinar uma rota de baixa distância que visite um conjunto de 20 cidades aleatórias exatamente uma vez e retorne à origem.

2. Modelagem

- **Cromossomo (Indivíduo):** Uma solução candidata, representada por uma **rota** (uma permutação das 20 cidades).
 - **População:** Um conjunto de 100 rotas (indivíduos) que evoluem a cada geração.
 - **Função de Fitness:** Mede a "aptidão" de uma rota. Como o AG *maximiza* o fitness e queremos *minimizar* a distância, a função é o inverso da distância total ($\text{Fitness} = 1 / \text{DistânciaTotal}$).
-

3. Algoritmo

O AG simula o processo de seleção natural ao longo de 100 gerações, aplicando os seguintes operadores genéticos:

- **Seleção (Método de Torneio):** Indivíduos aleatórios da população "competem", e o de maior fitness é selecionado para ser "pai" da próxima geração.
 - **Crossover (Recombinação Ordenada - OX1):** Combina o material genético (um segmento da rota) de dois pais para criar um "filho", garantindo que o filho seja uma permutação válida.
 - **Mutação (Troca - Swap):** Troca a posição de duas cidades aleatórias em uma rota. Isso introduz diversidade e evita que o algoritmo estagne prematuramente.
 - **Elitismo:** O melhor indivíduo da geração atual é copiado diretamente para a próxima, garantindo que as melhores soluções encontradas nunca sejam perdidas.
-

4. Resultados

A execução do algoritmo para 20 cidades com uma população de 100 indivíduos por 100 gerações produziu os seguintes resultados:

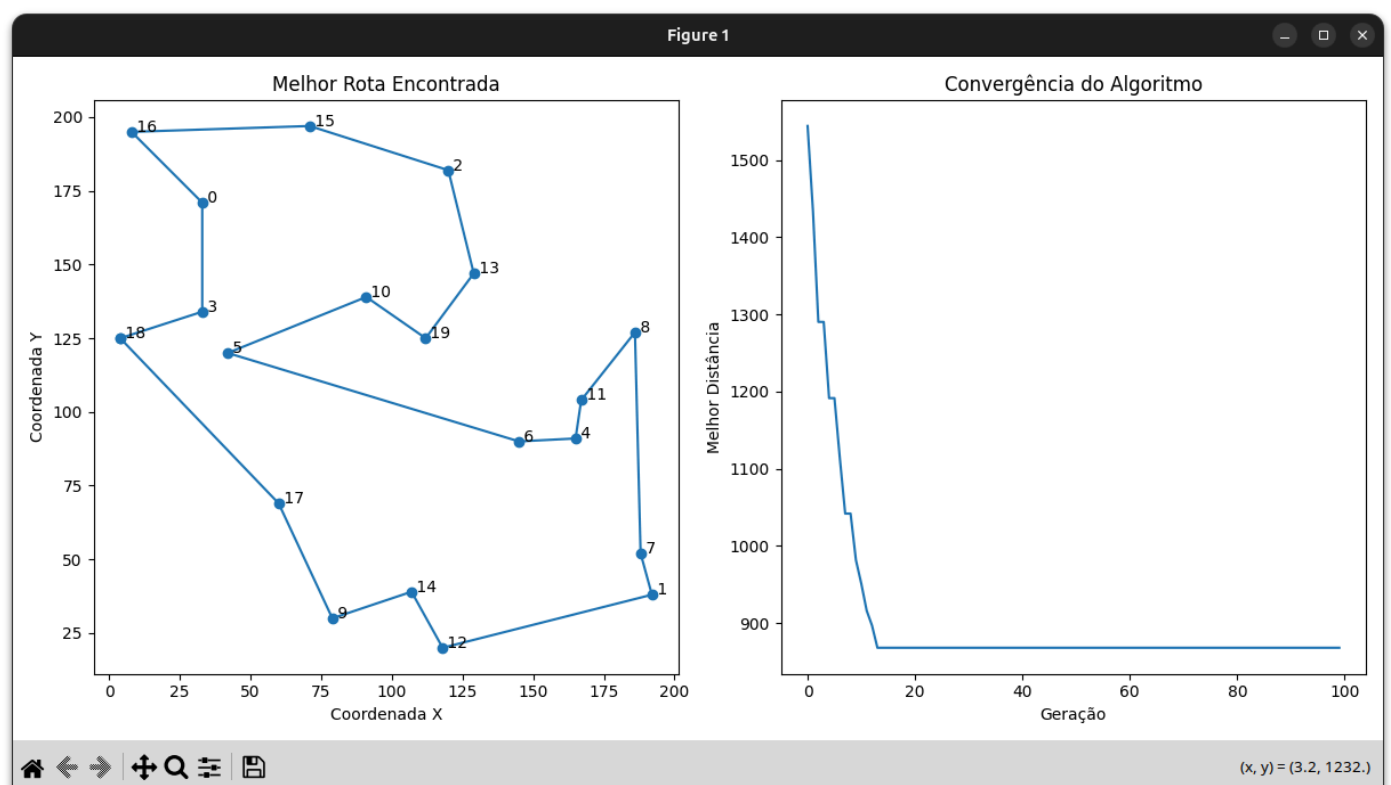
- **Melhor Distância Encontrada: 868.01**
- **Convergência:** Conforme o gráfico "Convergência do Algoritmo" (extraído da execução), o AG melhorou a solução após a geração 10, reduzindo a distância da rota de um valor inicial de 981.65 para 868.01.
- **Estagnação:** Após a geração 10, o algoritmo estabilizou, indicando que a população convergiu para uma solução ótima local (ou a solução global).
- **Rota Final:** O gráfico "Melhor Rota Encontrada" (extraído da execução) exhibe visualmente o caminho otimizado, que se mostra coeso e eficiente.

```
Executando o Algoritmo Genético por 100 gerações...
```

```
Geração 10 | Melhor Distância: 981.65
Geração 20 | Melhor Distância: 868.01
Geração 30 | Melhor Distância: 868.01
Geração 40 | Melhor Distância: 868.01
Geração 50 | Melhor Distância: 868.01
Geração 60 | Melhor Distância: 868.01
Geração 70 | Melhor Distância: 868.01
Geração 80 | Melhor Distância: 868.01
Geração 90 | Melhor Distância: 868.01
Geração 100 | Melhor Distância: 868.01
```

```
=====
RESULTADO FINAL DO ALGORITMO GENÉTICO
=====
```

```
Melhor distância encontrada: 868.01
```



5. Conclusão

O **Algoritmo Genético** provou ser uma meta-heurística eficaz para encontrar uma solução de alta qualidade para o Problema do Caixeiro Viajante, que é NP-Difícil. Diferente do A*, o AG não garante a solução *perfeita*, mas é capaz de explorar um espaço de busca massivo e convergir rapidamente para uma solução "boa o suficiente" (ótima local). O projeto ilustra o poder da computação bioinspirada para resolver problemas de otimização complexos.