Exercício: Algoritmo Genético para Alocação de Gateways em Aeroportos

Objetivo:
Usar um **Algoritmo Genético (AG)** para otimizar a alocação de **gateways** (pontos de acesso) em um aeroporto, minimizando o tempo médio de deslocamento dos passageiros entre os terminais e os gateways, considerando restrições de capacidade e distância.

Problema:

Um aeroporto possui:

- **5 terminais** (T1 a T5) com demanda de passageiros:
  - T1=200, T2=150, T3=300, T4=250, T5=100 (passageiros/hora).
- **3 gateways disponíveis** (G1,G2,G3), cada um com capacidade máxima de **500 passageiros/hora**.
- **Matriz de distâncias** (em metros) entre terminais e gateways:

	G1	G2	G3
T1	100	300	500
T2	200	100	400
T3	300	200	100
T4	400	500	200
T5	500	400	300

Objetivo:
Alocar cada terminal a **um único gateway**, respeitando a capacidade, de forma a **minimizar a distância total ponderada**:

$$\text{Minimizar } Z = \sum_{i=1}^{5} \sum_{j=1}^{3} d_{ij} \cdot x_{ij} \cdot p_i$$

Onde:

- dij: distância entre terminal i e gateway j.
- xij=1 se Ti está alocado a Gj, senão 0.
- pi: demanda do terminal i.

Implementação do Algoritmo Genético:

1. Representação do Cromossomo (Solução):

- **Codificação binária:** Cada gene representa a alocação de um terminal a um gateway.
  - Exemplo: `[1, 0, 0 | 0, 1, 0 | 0, 0, 1 | 1, 0, 0 | 0, 1, 0]`
    - T1→G1, T2→G2, T3→G3, T4→G1, T5→G2.

2. População Inicial:

- Gerar aleatoriamente **6 cromossomos** válidos (respeitando a capacidade dos gateways).

3. Função de Fitness:

- Avaliar cada solução usando:

  Fitness=11+ZFitness=1+Z1

  (Quanto maior o fitness, melhor a solução).

4. Seleção (Método da Roleta):

- Selecionar pais com probabilidade proporcional ao fitness.

5. Crossover (Recombinação):

- **Ponto de corte aleatório:** Trocar partes dos cromossomos dos pais.
  - Exemplo:
    - Pai 1: `[1,0,0 | 0,1,0 | 0,0,1 | 1,0,0 | 0,1,0]`
    - Pai 2: `[0,1,0 | 1,0,0 | 0,1,0 | 0,0,1 | 1,0,0]`
    - Filho: `[1,0,0 | 0,1,0 | 0,1,0 | 0,0,1 | 0,1,0]`

6. Mutação:

- Trocar aleatoriamente um gene (ex.: T3 de G3 para G1).

7. Critério de Parada:

- **Nº máximo de gerações (ex.: 50)** ou **solução estável por 10 gerações**.

Exemplo de Solução Ótima Esperada:

- **Alocação:**
  - G_1: T_1, T_4 (Total = 450 ≤ 500)
  - G_2: T_2, T_5 (Total = 250 ≤ 500)
  - G_3: T_3 (Total = 300 ≤ 500)
- **Distância Total Ponderada:**

$$Z = (100 \cdot 200) + (100 \cdot 150) + (100 \cdot 300) + (200 \cdot 250) + (100 \cdot 100) = 140.000 \text{ metros}$$

Tarefas:

1. Implemente o AG em Python e mostre a evolução das soluções.
2. Verifique se a solução encontrada respeita as restrições de capacidade.
3. Modifique o problema para **5 gateways** e compare os resultados.

Dica: Use penalizações no fitness para soluções inválidas (ex.: gateway superlotado).