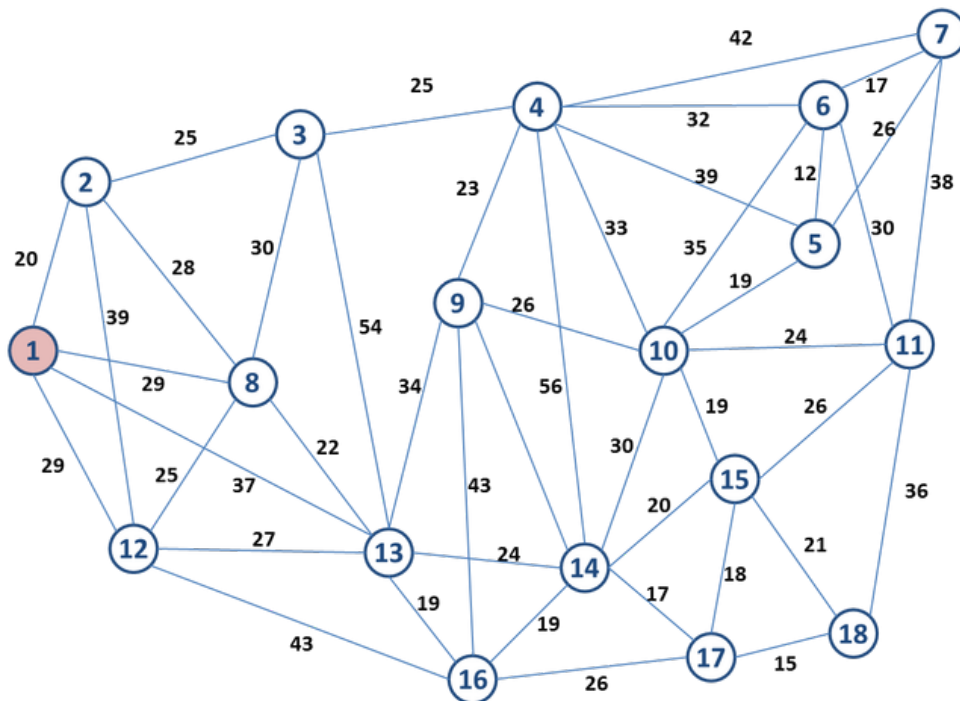


## Projeto 01 — Pós-Graduação

### 1 – Problema do Caixeiro Viajante com Metaheurísticas

Considere o grafo fixo abaixo, onde os nós representam cidades que devem ser visitadas por um caixeiro viajante. O objetivo é encontrar a menor rota possível que visite todas as cidades uma única vez e retorne ao ponto de partida (nó 1).



Implemente em Python os seguintes algoritmos de otimização:

- Algoritmo Genético (GA)
- Colônia de Formigas (ACO)
- Hill Climbing com permutação de cidades

Seu programa deve:

- Representar a instância do TSP com estrutura adequada (ex: matriz de adjacência);
- Comparar os algoritmos em termos de:

- Qualidade da solução (menor distância total);
- Tempo de execução;
- Convergência (melhoria ao longo das iterações);
- Visualizar os caminhos encontrados por cada algoritmo;
- Apresentar gráfico de evolução da solução (fitness, distância ou custo).

## 2 – Otimização Contínua

Considere a seguinte função Schwefel definida para  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) \in [-500, 500]^5$ :

$$f(\mathbf{x}) = 418.9829 \cdot 5 - \sum_{i=1}^5 x_i \cdot \sin(\sqrt{|x_i|})$$

Essa função apresenta múltiplos mínimos locais e um mínimo global conhecido, sendo amplamente utilizada para testar algoritmos de otimização em espaços contínuos desafiadores.

**Implemente em Python os seguintes algoritmos de otimização:**

- Algoritmo Genético (GA)
- Colônia de Formigas (ACO — versão combinatória adaptada para contínuo via discretização ou mapeamento)
- Hill Climbing com reinício aleatório

**Seu programa deve:**

- Avaliar e minimizar a função Schwefel no intervalo dado;
- Comparar os algoritmos em termos de:
  - Precisão da solução (distância ao mínimo global conhecido);
  - Número de iterações e tempo de execução;
  - Robustez (consistência em diferentes execuções).
- Gerar gráficos de convergência e registrar, se possível, a trajetória das soluções.

**Observação:** A função apresenta simetria e um número elevado de mínimos locais, exigindo algoritmos capazes de escapar de armadilhas no espaço de busca.

**Entrega:**

- Código Python bem organizado e comentado;
- Relatório contendo explicações, imagens geradas e comparações;
- Data de entrega e apresentação conforme cronograma da disciplina.