Computação Evolutiva
Trabalho Prático

Objetivo: O trabalho consiste em implementar um algoritmo evolutivo para resolver um problema de otimização, elaborar um relatório sobre o trabalho e fazer uma apresentação para a turma.

Observações sobre a apresentação/entrega:

(a) O trabalho será feito individualmente ou em dupla (apenas para graduação).
(b) Cada grupo deve implementar um algoritmo evolutivo a ser aplicado ao problema escolhido.
(c) O trabalho consiste ainda de apresentação e parte escrita.
(d) Parte escrita: consiste de um relatório sobre o assunto, contendo Introdução, Desenvolvimento, Conclusão e Referências.
(e) A data de entrega da parte escrita e código será: 24/03/2025.
(g) As datas previstas de apresentação são: 24/03/2025 a 04/04/2025.
(h) As apresentações devem ter aproximadamente 15 minutos (+5 min para discussão).
(i) Todos alunos deverão estar aptos a apresentar nos dias 24/03/2025.

Pontos a serem considerados:
(a) Implementação do algoritmo evolutivo utilizado a biblioteca DEAP
(b) Análise de desempenho utilizando algum benchmark da literatura
(c) Apresentação de gráficos e/ou tabelas com resultados e estatística

Sugestões:
Alguns problemas e sites que contém instâncias de teste são listados no **Anexo 1**.

Anexo 1

1. Problema do Caixeiro Viajante (TSP - Traveling Salesman Problem)
Descrição: Dado um conjunto de cidades e as distâncias entre elas, encontrar o caminho mais curto que visita todas as cidades exatamente uma vez e retorna à cidade inicial.
Aplicação: Planejamento de rotas, logística, otimização de circuitos.
Benchmarks:
TSPLIB: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

2. Problema da Mochila (Knapsack Problem)
Descrição: Dado um conjunto de itens, cada um com um peso e valor, determinar quais itens devem ser incluídos em uma mochila de capacidade limitada para maximizar o valor total.
Aplicação: Alocação de recursos, finanças, logística.
Benchmarks:
OR-Library (Knapsack):
https://www.or.deis.unibo.it/research_pages/ORinstances/ORinstances.html
Pisinger Knapsack Benchmarks: http://hjemmesider.diku.dk/~pisinger/codes.html

3. Problema de Corte de Estoque (Cutting Stock Problem)
Descrição: O objetivo é minimizar o desperdício ao cortar itens menores a partir de materiais maiores disponíveis (chapas, rolos, barras).
Aplicação: Indústria de manufatura e planejamento de produção.
Benchmarks:
OR-Library (Cutting Stock):
https://www.or.deis.unibo.it/research_pages/ORinstances/ORinstances.html

4. Problema de Roteamento de Veículos (VRP - Vehicle Routing Problem)
Descrição: Determinar as rotas ideais para um conjunto de veículos que devem atender clientes com demandas específicas, minimizando custos como distância ou tempo.
Aplicação: Logística, transporte, entrega de produtos.
Benchmarks:
CVRPLIB: http://vrp.atd-lab.inf.puc-rio.br/index.php/en/
SINTEF VRP Benchmarks: https://www.sintef.no/projectweb/top/vrptw/

5. Problema do Corte de Grafos (Graph Partitioning Problem)
Descrição: Dividir um grafo em subconjuntos de nós, minimizando o número de arestas cortadas entre os subconjuntos.
Aplicação: Redes, processamento paralelo, balanceamento de carga.
Benchmarks:
10th DIMACS Challenge (Graph Partitioning):
https://sites.cc.gatech.edu/dimacs10/

6. Problema de Satisfatibilidade Booleana (SAT - Satisfiability Problem)
Descrição: Determinar se existe uma atribuição de valores verdadeiros e falsos às variáveis de uma fórmula booleana que torne a fórmula verdadeira.
Aplicação: Verificação de circuitos, inteligência artificial, planejamento.
Benchmarks:
SATLIB: http://www.cs.ubc.ca/~hoos/SATLIB/
DIMACS SAT Benchmarks: https://www.satcompetition.org/

7. Problema de Sequenciamento de Tarefas (Job Shop Scheduling Problem)
Descrição: Dado um conjunto de tarefas com restrições de precedência e máquinas específicas, encontrar uma sequência de execução que minimize o tempo total (makespan).
Aplicação: Planejamento de produção, otimização de processos industriais.
Benchmarks:
OR-Library (Job Shop):
https://www.or.deis.unibo.it/research_pages/ORinstances/ORinstances.html
Taillard Scheduling Benchmarks:
https://web.imt-atlantique.fr/x-auto/clahlou/mdl/Benchmarks.html

8. Problema de Empacotamento de Caixas em Containers (Bin Packing Problem)
Descrição: Dado um conjunto de itens de tamanhos variados, determinar a melhor forma de empacotá-los em contêineres de capacidade fixa, minimizando o número de contêineres usados.
Aplicação: Logística, alocação de recursos, computação em nuvem.
Benchmarks:
BPPLIB:
https://site.unibo.it/operations-research/en/research/bpplib-a-bin-packing-problem-library

9. Problema de Coloração de Grafos (Graph Coloring Problem)
Descrição: Atribuir cores aos vértices de um grafo de forma que vértices adjacentes não compartilhem a mesma cor, minimizando o número total de cores usadas.
Aplicação: Alocação de frequências em redes, planejamento escolar, otimização de registro.
Benchmarks:
DIMACS Graph Coloring Benchmarks: https://mat.tepper.cmu.edu/COLOR04/

10. Seleção de Conjuntos de Recursos (Feature Selection)
Descrição: Dado um conjunto de variáveis (features), o objetivo é selecionar um subconjunto que maximize o desempenho de um modelo de Machine Learning (ex.: acurácia, F1-score), minimizando a quantidade de features utilizadas.
Aplicação: Aprendizado supervisionado, análise de dados, big data.
Função de Aptidão: Maximizar o desempenho do modelo e minimizar o número de features selecionadas.
Benchmarks:
UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php
Kaggle Datasets: https://www.kaggle.com/datasets