	Universidade Federal de Goiás – Instituto de Informática Disciplina: INF 0092 - INTELIGÊNCIA COMPUTACIONAL Docente: Prof. Me. Bruno Brandão	
	Trabalho (Algoritmos Genéticos)	25/03/2025

Orientações Gerais:

1. O trabalho deve ser entregue na data combinada em sala. Atrasos serão aceitos mediante apresentação de justificativa plausível. Ademais, atrasos terão dedução de nota.
2. Não é permitido plagiar código ou relatório de outros grupos. Quaisquer suspeitas de plágio serão confrontadas com os grupos de forma a
3. A entrega deverá ser feita pelo SIGAA em um único arquivo comprimido por APENAS UM integrante do grupo.

1. Objetivos e Organização

- **Objetivo principal:** Resolver um dos problemas disponibilizados utilizando Algoritmos Genéticos
- **Formação de grupos:** Até 3 integrantes
- **Entregas:** Duas partes complementares
 - Código-fonte em Python
 - Relatório técnico detalhado

2. Requisitos do Código

2.1 Desenvolvimento

- Implementar o AG completamente sem bibliotecas especializadas para operadores genéticos
- É permitido usar bibliotecas auxiliares como numpy, pandas e matplotlib para cálculos e visualizações

2.2 Estrutura do Código

O código deve conter funções bem definidas para:

- Inicialização da população
- Seleção de indivíduos
- Operações de cruzamento
- Operações de mutação
- Critério de parada do algoritmo

2.3 Formatos de Entrega

- Arquivos .py com README.md explicativo
- Jupyter Notebook
- Google Colab

Independentemente do formato escolhido, o código deve estar legível e conter instruções claras de execução. No README ou nas células de texto.

3. Estrutura do Relatório

O relatório deve ser entregue em formato PDF, máximo de 8 páginas (formato coluna simples) ou 5 páginas (formato coluna dupla), contendo:

3.1 Identificação

- Nome e matrícula de todos os integrantes do grupo na primeira página

3.2 Seções Obrigatórias

3.2.1 Problema e Restrições

- Descrição do problema escolhido, seus objetivos e restrições.
- Explicação das restrições e como foram tratadas (hard ou soft constraints) descrevendo como seu tratamento reflete no AG, por exemplo: "a restrição de caminho máximo por dia foi tratada como soft constraint, no algoritmo, permitimos que soluções apresentassem um caminho maior que o máximo."

3.2.2 Representação da Solução

- Explicação de como um indivíduo é traduzido para uma solução
- Descrição do que cada gene representa
- Recomendação: incluir imagens ou diagramas para ilustrar a representação

3.2.3 Operador de Cruzamento

- Detalhamento do(s) método(s) de cruzamento utilizado(s)
- Se mais de um método foi testado: explicação breve de cada um e justificativa da escolha destes métodos.
- Se apenas um método foi usado: descrição detalhada com diagrama recomendado.

3.2.4 Operador de Mutação

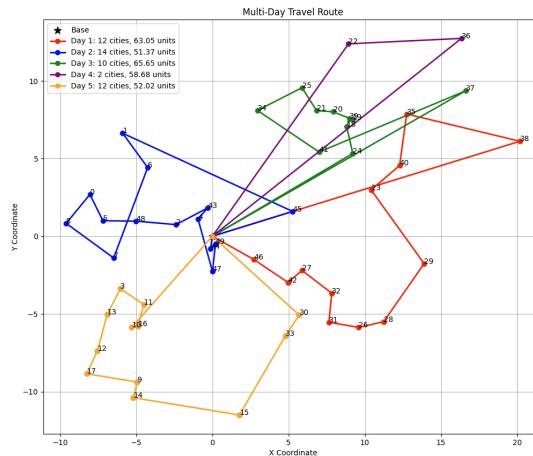
- Detalhamento do(s) método(s) de mutação utilizado(s)
- Mesmo formato da seção de cruzamento: breve se vários foram testados, detalhado se apenas um

3.2.5 Experimentos

- Uma tabela com os parâmetros do AG, como a taxa de mutação, tamanho da população, tamanho do torneio (se usado), e quaisquer outras configurações para reproduzir o trabalho no código fornecido.
- É possível adicionar uma breve discussão sobre estes valores da tabela se necessário.
- Gráficos comparativos de diferentes operadores e/ou configurações de parâmetros (se testados)
- Análise detalhada do melhor experimento, incluindo:
 - Fitness do melhor indivíduo em cada geração
 - Média de fitness da população por geração
 - Variância ou desvio padrão do fitness por geração
- Representação gráfica da melhor solução obtida
- Discussão sobre as características da solução que demonstram sua qualidade ou peculiaridades, por exemplo: "*os caminhos do TSP estão sem cruzamentos e mostram claramente uma preferência por cidades próximas para cada movimentação...*" ou "*as curvas de controle demonstram que as juntas X Y e Z se comportam melhores em baixa frequência, já as juntas A, B, e C apresentam frequência alta.*"

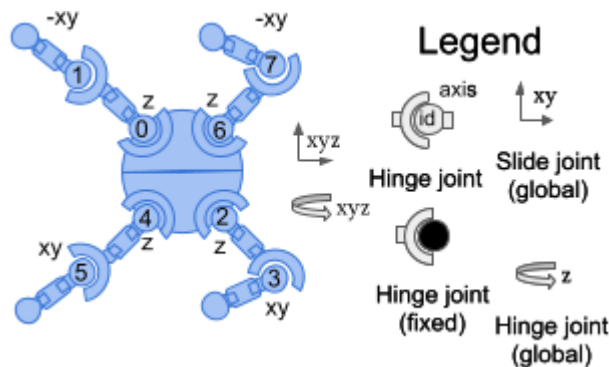
4. Problemas Disponíveis

4.1 Problema do Caixeiro Viajante



- **Objetivo:** Maximizar o número de pontos turísticos visitados em 5 dias
- **Restrição:** Não é possível viajar mais que 60km por dia
- **Recurso:** O código fornecido contém funções para visualizar cidades e caminhos

4.2 Controle do Agente Formiga



- **Objetivo:** Controlar um robô formiga para caminhar o mais rápido possível no eixo X utilizando curvas de controle senoidais para cada uma de suas juntas.
- **Parâmetros:**
 - 8 juntas para controlar
 - Frequência, amplitude $[0-1]$ e deslocamento de fase $[0-2\pi]$ para cada junta
- **Simulação:** Ambiente disponível em <https://gymnasium.farama.org/environments/mujoco/ant/>
- **Recurso:** No código fornecido para o trabalho é possível encontrar uma função especializada em avaliar um conjunto de frequências, amplitudes, e deslocamentos.

5. Dicas e Recursos

- É recomendável utilizar um template LaTeX para a escrita do relatório.
- Um ambiente virtual conda com as bibliotecas necessárias está disponível no arquivo `environment.yml`. (opcional)
- Dedique tempo suficiente para a análise dos resultados e a qualidade das visualizações.
- Ao longo da implementação, pergunte-se o porquê de cada decisão tomada.

Boa sorte com o trabalho! Lembre-se que tanto a implementação quanto a documentação são importantes para demonstrar seu entendimento dos algoritmos genéticos.