Universidade Federal de Minas Gerais
Escola de Engenharia
Departamento de Engenharia Elétrica

ELE083 – Computação Evolucionária

Trabalho Prático:

1. Projete um **Algoritmo Genético** que evolui uma população de *strings* até convergir para a seguinte string:

   METHINKS*IT*IS*LIKE*A*WEASEL

   - Assumindo que apenas o caracter de espaço (representado por *), letras maísculas e dígitos são permitidos, calcule a dimensão do espaço de busca para esse problema. Qual a probabilidade de uma *string* gerada aleatoriamente ser exatamente igual à *string* alvo?
   - Explique como o algoritmo genético consegue superar essa probabilidade incrivelmente pequena.
   - Calcule o número médio de gerações para convergência.
   - Faça experimentos com o parâmetro probabilidade de mutação. O que acontece com o desempenho do algoritmo?
   - Faça experimentos com o parâmetro probabilidade de cruzamento. O que acontece com o desempenho do algoritmo?
   - Apresente um relatório descrevendo sua implementação, testes e res[1]ultados obtidos.

[1] Extraído de Hamlet – W. Shakespeare

2. Projete e implemente um Algoritmo Genético Geracional (GGA) com **codificação binária** para solucionar o problema da Mochila 0/1 o qual pode ser descrito da seguinte forma: “Dados *N* itens, onde cada item possui um benefício (v_j) e um peso associado (w_j), o problema consiste em selecionar o subconjunto de itens que maximiza a soma dos benefícios sem ultrapassar a capacidade (cap) da Mochila”.

A modelagem matemática deste problema é dada a seguir

$$\max_{\vec{x}} \quad f(\vec{x}) = \sum_{j=1}^{N} v_j x_j$$

$$\text{sa.} \quad \sum_{j=1}^{N} w_j x_j \leq cap$$

$$x_1, x_2, \ldots, x_n \in \{0, 1\}$$

o vetor $\vec{x} = \{x_1, x_2, \ldots, x_n\}$ é o vetor de variáveis de decisão tal que

$$x_j = \begin{cases} 1 & \text{se o item for colocado na Mochila} \\ 0 & \text{caso contrario} \end{cases}$$

- **Função de Aptidão**: o critério essencial para determinar a qualidade de uma solução é o benefício total proporcionado pelos itens escolhidos ($\sum_{j=1}^{n} v_j x_j$). No entanto, dada a codificação binária adotada para representar cada solução candidata, é necessário incluir na função de aptidão um mecanismo que penalize as soluções inválidas, isto é, as soluções cujo peso total ($\sum_{j=1}^{n} w_j x_j$) ultrapassa a capacidade da Mochila. Se este mecanismo não for considerado, é trivial verificar que a melhor solução para o problema seria a que escolhe todos os itens para serem incluídos na Mochila. Assim, a avaliação da qualidade de uma dada solução candidata deverá ser feita com base na seguinte expressão

$$f'(\vec{x}) = \sum_{j=1}^{N} v_j x_j - Pen(\vec{x})$$

onde

$$Pen(\vec{x}) = \begin{cases} 0 & \text{se } \vec{x} \text{ é válida} \\ \rho \times \left(\sum_{j=1}^{N} w_j x_j - cap\right) & \text{caso contrario} \end{cases}$$

com $\rho = \max \frac{v_j}{w_j}, \quad \forall j = 1, \ldots, N.$

Sugestões a serem usadas na implementação:

1. Mecanismo de Seleção dos Pais:
   a. Operador de Seleção Proporcional ao Fitness + Roleta;
2. Operador de Cruzamento:

    a. Crossover com 1 Ponto de Corte com probabilidade > 0.6;
3. Operador de Mutação:
    a. Bit Flip com probabilidade entre 0.02 e 0.10;
4. Mecanismo de Seleção dos Sobreviventes:
    a. Geracional ($\mu = \lambda$). Logo, todos os pais são substituídos pelos descendentes.

Exemplo de Instância a ser usada para teste:

- Capacidade da Mochila: **35**
- Número de objetos: **8**
- Propriedades dos objetos:

	Obj_1	Obj_2	Obj_3	Obj_4	Obj_5	Obj_6	Obj_7	Obj_8
Peso	10	18	12	14	13	11	8	6
Valor	5	8	7	6	9	5	4	3