# **Documentação do Código**

## **Introdução**

Este é um programa em Python que implementa um algoritmo genético para resolver o Problema da Mochila. O Problema da Mochila é um problema de otimização combinatória em que o objetivo é selecionar um conjunto de itens, cada um com um peso e um valor, de modo a maximizar o valor total, sem exceder um peso máximo predefinido.

## **Módulo tkinter**

### **Classe ProblemaMochilaInterface**

### **Método \_\_init\_\_(self, master)**

* **Descrição**: Construtor da classe, responsável por inicializar a interface gráfica.
* **Parâmetros**:
  + master: O mestre (root) da interface gráfica.
* **Retorno**: Nenhum.

### **Método inicializar\_interface(self)**

* **Descrição**: Configura a interface gráfica com rótulos, campos de entrada e botão de execução.
* **Parâmetros**: Nenhum.
* **Retorno**: Nenhum.

### **Método executar\_algoritmo\_genetico(self)**

* **Descrição**: Método chamado quando o botão "Executar Algoritmo Genético" é pressionado. Ele recupera os parâmetros inseridos pelo usuário, executa o algoritmo genético e exibe os resultados na interface gráfica.
* **Parâmetros**: Nenhum.
* **Retorno**: Nenhum.

## **Módulo de Algoritmo Genético**

### **Função individual(n\_de\_itens)**

### **Descrição**

Cria um indivíduo (solução) da população com n\_de\_itens genes, onde cada gene representa a presença (1) ou ausência (0) de um item na mochila.

### **Parâmetros**

* n\_de\_itens: Número de itens disponíveis.

### **Retorno**

* Lista representando o indivíduo.

### **Função population(n\_de\_individuos, n\_de\_itens)**

### **Descrição**

Cria uma população de indivíduos.

### **Parâmetros**

* n\_de\_individuos: Número de indivíduos na população.
* n\_de\_itens: Número de itens disponíveis.

### **Retorno**

* Lista de listas representando a população.

### **Função fitness(individuo, peso\_maximo, pesos\_e\_valores)**

### **Descrição**

Calcula a aptidão (fitness) de um indivíduo com base no peso total e no valor total dos itens na mochila.

### **Parâmetros**

* individuo: Lista representando o indivíduo.
* peso\_maximo: Peso máximo permitido na mochila.
* pesos\_e\_valores: Lista de tuplas representando o peso e o valor de cada item.

### **Retorno**

* Valor da aptidão do indivíduo.

### **Função media\_fitness(populacao, peso\_maximo, pesos\_e\_valores)**

### **Descrição**

Calcula a média da aptidão de uma população, excluindo indivíduos com aptidão zero.

### **Parâmetros**

* populacao: Lista de listas representando a população.
* peso\_maximo: Peso máximo permitido na mochila.
* pesos\_e\_valores: Lista de tuplas representando o peso e o valor de cada item.

### **Retorno**

* Média da aptidão da população.

### **Função selecao\_roleta(pais)**

### **Descrição**

Realiza a seleção de pais usando o método da roleta.

### **Parâmetros**

* pais: Lista de listas contendo pares (aptidão, indivíduo).

### **Retorno**

* Indivíduo selecionado como pai/mãe.

### **Função crossover(pai1, pai2)**

### **Descrição**

Realiza o crossover de ponto único entre dois pais.

### **Parâmetros**

* pai1: Lista representando o primeiro pai.
* pai2: Lista representando o segundo pai.

### **Retorno**

* Duas listas representando os filhos resultantes do crossover.

### **Função mutacao(individuo, taxa\_mutacao)**

### **Descrição**

Realiza a mutação por inversão de bit em um indivíduo.

### **Parâmetros**

* individuo: Lista representando o indivíduo.
* taxa\_mutacao: Taxa de mutação.

### **Retorno**

* Nenhum.

### **Função evolve(populacao, peso\_maximo, pesos\_e\_valores, n\_de\_cromossomos, taxa\_crossover, taxa\_mutacao)**

### **Descrição**

Executa uma geração do algoritmo genético, realizando a seleção, crossover e mutação.

### **Parâmetros**

* populacao: Lista de listas representando a população.
* peso\_maximo: Peso máximo permitido na mochila.
* pesos\_e\_valores: Lista de tuplas representando o peso e o valor de cada item.
* n\_de\_cromossomos: Número de cromossomos (indivíduos) a serem gerados.
* taxa\_crossover: Taxa de crossover.
* taxa\_mutacao: Taxa de mutação.

### **Retorno**

* Lista de listas representando a nova população gerada.

## **Programa Principal**

O programa principal cria uma instância da classe ProblemaMochilaInterface, que inicia a interface gráfica usando a biblioteca Tkinter. O usuário fornece parâmetros para o algoritmo genético por meio da interface, e os resultados são exibidos na própria interface após a execução do algoritmo.