

Algoritmos genéticos

• Inteligência Artificial •

INTRODUÇÃO

Este relatório descreve a implementação de algoritmos genéticos, proposto pela disciplina INE5430 (Inteligência Artificial). O tema a ser modelado corresponde ao Subset Sum Problem, ou problema da soma de subconjuntos. Foi utilizada a biblioteca PyGad do Python como ferramenta de suporte ao desenvolvimento da solução.

O PROBLEMA

O problema da Soma de Subconjuntos pode ser definido da seguinte maneira: dado um conjunto de inteiros I e um inteiro s , analisa-se a existência de algum subconjunto de I cuja soma resulte em s . O Subset Problem pode ser visto como um caso especial do Problema da Mochila, sendo assim NP-Completo.

SOLUÇÃO

Utilizamos a biblioteca PyGad para computar a solução.

Também foram adicionados casos de teste, em formato de lista de tuplas de exemplos.

- **Formato e codificação dos cromossomos:**

Os cromossomos foram apresentados como listas de binários, como é possível visualizar pela saída do algoritmo e o arquivo solution.txt gerado.

```
[1. 1. 0. 1. 0. 0. 0.]
```

- **Tamanho da população:**

O tamanho da população será sempre igual ao número de elementos no conjunto de entrada. Assim, varia conforme o teste que está sendo executado pelo usuário.

- **Função de fitness:**

A função de fitness, que avalia a qualidade do subconjunto gerado por um cromossomo, é definida como:

$$\text{fitness} = \frac{1}{|\text{diferença}| + 1}$$

E implementada como a `fitness_func`, que multiplica os valores da lista do teste atualmente sendo executado pelos pesos fornecidos na solução candidata e soma os resultados. A diferença absoluta entre essa soma e o valor alvo é usada para calcular o fitness, atribuindo maior pontuação às soluções que produzem uma soma mais próxima do alvo :

```
def fitness_func(ga_instance, solution, solution_idx):
    output_sum = numpy.sum(numpy.array(inputs) * solution)
    diff = abs(output - output_sum)
    fitness = 1.0 / (diff + 1)

    return fitness
```

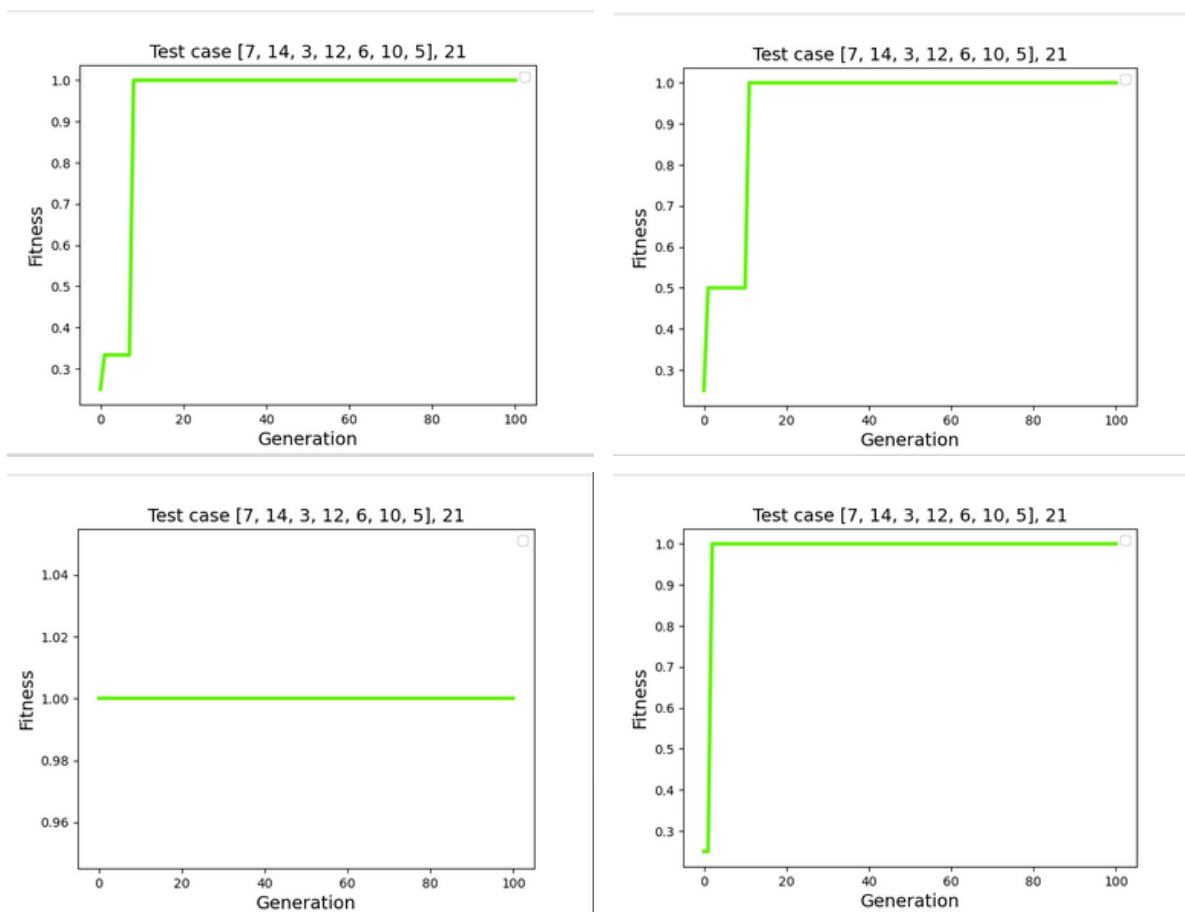
SOLUÇÃO

- **Plots de execução:**

Analisaremos, por meio de gráficos, dois casos de teste escolhidos aleatoriamente do conjunto de casos. Os gráficos trazem a relação entre porcentagem de fitness da solução encontrada em relação às gerações percorridas:

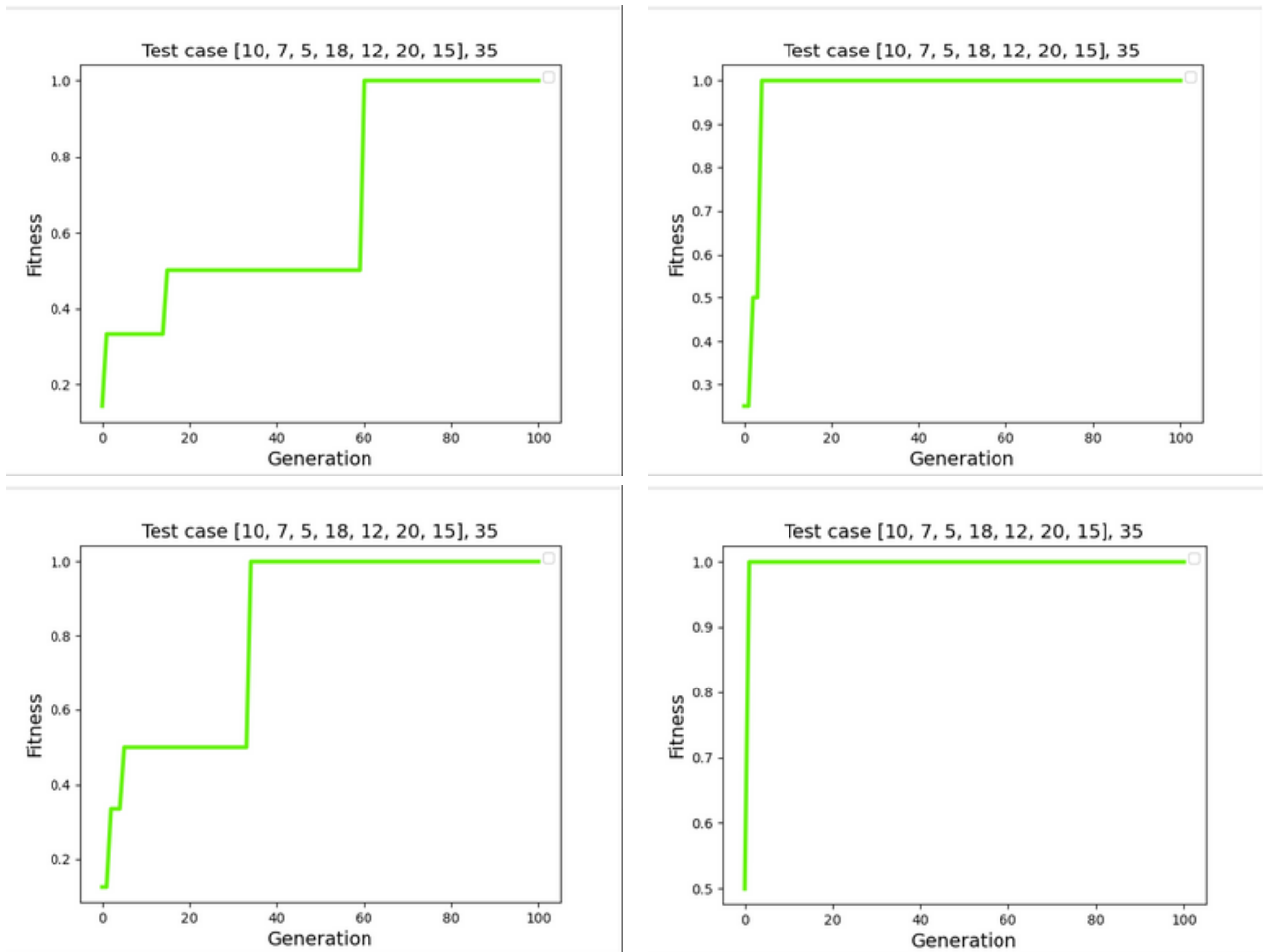
1. Input: [7, 14, 3, 12, 6, 10, 5], saída esperada: 21

Mediante gráfico a seguir, pode-se ver a rápida convergência do algoritmo, mesmo após poucas gerações:



Figuras 1 a 4: convergência do algoritmo para diferentes execuções do mesmo caso de teste

2. Input: [10, 7, 5, 18, 12, 20, 15], saída esperada: 35



Figuras 5 a 8: convergência do algoritmo para diferentes execuções do mesmo caso de teste

- **Critério para seleção, cruzamento e mutação:**

Foram utilizados os métodos padrão fornecidos pela biblioteca para mutação e tipos de crossover. A mutação ocorre em 20% dos genes, com uma probabilidade global de mutação de 10% por geração.

- **Critério de parada:**

Foi considerado um limite de 100 gerações como critério de parada.

DIFICULDADES E CONSIDERAÇÕES

O uso de algoritmos genéticos é uma abordagem interessante para problemas de otimização complexos como o Subset Sum, que não possuem soluções eficientes e determinísticas em tempo polinomial. No entanto, o custo computacional pode ser significativo, principalmente quando lidamos com conjuntos grandes ou muitas gerações.

CONCLUSÃO

Em resumo, o código desenvolvido provou-se eficiente para a resolução do problema proposto na disciplina.



ALUNOS

- Clara Rosa Oliveira Gonçalves - matrícula: 22103511
- Gian Carlo Figueiredo Ferrari - matrícula: 22100619
- Maykon Marcos Júnior - matrícula: 22102199