



**UNIVERSIDADE ESTADUAL DO MARANHÃO-UEMA**  
**CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**  
**DISCIPLINA: TÓPICOS ESPECIAIS EM PROGRAMAÇÃO**

**Bruno Rocha Gomes**

**ALGORITMOS GENÉTICOS: O Problema da Mochila Binária**

**SÃO LUÍS - MA**

**2017**

**Bruno Rocha Gomes**

**ALGORITMOS GENÉTICOS: O Problema da Mochila Binária**

Relatório apresentado a Universidade Estadual do Maranhão como requisito para obtenção de nota na disciplina Tópicos Especiais em Programação ministrada pelo professor Thiago Lemos.

**SÃO LUÍS - MA**

**2017**

## RESUMO

Este relatório abordará os resultados obtidos por um programa que utiliza um algoritmo típico da inteligência artificial, o algoritmo genético. A partir disto, o programa realizará uma busca a fim de achar soluções aproximadas para o Problema da Mochila Binária, um problema de otimização e busca muito comum na área de IA. O algoritmo se responsabilizará por encontrar uma combinação de valores (0 ou 1) que apresente o *fitness* (qualidade do indivíduo) mais próximo do *fitness* considerado como ótimo na base dados do programa. O programa foi desenvolvido na linguagem de programação python na IDE Spyder, com base nos assuntos aprendidos na disciplina Tópicos Especiais em Programação.

**Palavras-Chave:** Inteligência Artificial; Algoritmos Genéticos; Python.

## SUMÁRIO

<b>1. Introdução .....</b>	<b>5</b>
<b>2. Objetivos.....</b>	<b>5</b>
<b>3. Resultados e Discussões.....</b>	<b>5</b>
<b>4. Conclusão .....</b>	<b>11</b>
<b>5. Referências .....</b>	<b>12</b>

## 1. INTRODUÇÃO

O Problema da Mochila Binária é um problema muito comum na área de Inteligência Artificial e é denotado da seguinte forma:

“Dados os números  $P_1, P_2, \dots, P_n, V_1, V_2, \dots, V_n$  e  $C$ , encontrar um subconjunto de  $X$  de  $\{1, 2, \dots, n\}$  que maximize  $V(X)$  sob a restrição  $P(X) \leq C$ .”

Dessa forma, pode-se dizer que  $1, 2, \dots, n$  são os itens a serem colocados na mochila, que  $P_i$  é o peso de cada um desses itens e que  $V_i$  é o valor de cada item  $i$ . O problema deve respeitar uma restrição, que diz que  $C$  é a capacidade máxima da mochila. Assim, uma mochila viável é qualquer subconjunto  $X$  de  $\{1, 2, \dots, n\}$  tal que  $P(X) \leq C$ .

O valor de uma mochila  $X$  é o número  $V(X)$ . O problema será então encontrar uma combinação de valores 0 ou 1, onde 0 indica que o item está fora da mochila e 1 indica que o item está dentro. Essa combinação deve apresentar um resultado aproximado quanto à soma dos valores dos itens quando comparados ao valor considerado ótimo, presente nas bases de dados do programa. A base de dados informará tanto estes valores, quanto o número de itens, o valor individual de cada um, seus pesos e a soma dos valores de todos os itens. O programa trabalhará com 10 mochilas.

Para a resolução deste problema, será utilizado o algoritmo genético, uma técnica de busca comum na ciência da computação muito utilizada para achar soluções aproximadas em problemas de otimização e busca. Este algoritmo é baseado na evolução dos seres biológicos e simula que uma população de representações abstratas de solução é escolhida em busca de soluções melhores. A evolução costuma se iniciar a partir de um conjunto de soluções aleatórias e é realizada através das gerações, onde a cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para continuar na próxima geração, e recombinados ou mutados a fim de formar uma nova população. Dessa forma, nova população é utilizada como entrada para a próxima iteração do algoritmo.

Para o desenvolvimento deste programa, foi utilizada a versão 3.6 do python e a IDE Spyder.

## 2. OBJETIVOS

Dada uma base de dados com 10 mochilas diferentes, deve-se utilizar o algoritmo genético para encontrar uma combinação de 0 ou 1 que apresente o *fitness* mais próximo ao *fitness* presente na outra base de dados. O *fitness* corresponderá ao valor de cada item da mochila somado um ao outro. Para isso, as restrições quanto à capacidade máxima de peso suportada pela mochila, devem ser respeitadas, tornando a mochila viável.

## 3. RESULTADOS E DISCUSSÕES

Uma vez que os arquivos contendo a base de dados do programa foram lidos, foi possível trabalhar diretamente com os mesmos no algoritmo genético. Para o programa foi utilizado o total 100 populações, uma vez que o critério de parada do programa é após ocorrerem 100 gerações, onde cada população contém dez indivíduos e cada indivíduo é formado por uma combinação de  $n$  itens, que será indicado na base de dados. Além disso, o programa utiliza o valor de 20% como taxa de mutação a quantidade de 3 indivíduos utilizados para se reproduzir.

Sobre a execução do programa, o mesmo foi capaz de encontrar um *fitness* bem próximo e às vezes igual ao *fitness* ótimo indicado na base de dados. Além disso, o programa aponta o indivíduo detentor daquele *fitness*, exibindo um vetor de 0 e 1. O programa também aponta o indivíduo de menor *fitness*.

Foi utilizado um menu que solicita ao usuário qual a mochila ele quer analisar dentre dez que

foram disponíveis na base de dados.

- **Mochila 1**

```
População inicial:
[[0, 0, 0, 1, 1, 0, 0, 0, 0, 0], [0, 1, 1, 0, 1, 0, 0, 0, 0, 1], [0, 1, 1, 0, 1,
0, 0, 1, 0, 0], [1, 0, 0, 0, 1, 0, 0, 0, 1, 0], [0, 1, 0, 1, 0, 0, 1, 1, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 1], [0, 0, 1, 1, 0, 0, 1, 0, 1, 0], [0, 0, 0, 1, 0,
0, 1, 1, 0, 0], [1, 0, 0, 1, 0, 0, 0, 0, 0, 0], [0, 1, 1, 0, 1, 0, 1, 1, 0, 0]]

População final:
[[0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 1, 1, 0,
0, 0, 1, 1, 1], [0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 1, 1, 0, 0, 0, 1, 1, 1],
[0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 1, 1, 0,
0, 0, 1, 1, 1], [0, 1, 1, 1, 0, 0, 0, 1, 1, 1], [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]]

Numero de itens:
10

Capacidade Maxima da mochila:
269

O melhor individuo foi:
[0, 1, 1, 1, 0, 0, 0, 1, 1, 1]
Fitness:
295.0

O pior individuo foi:
[0, 1, 0, 1, 0, 0, 1, 1, 1, 0]
Fitness
169.0

Pressione qualquer tecla para continuar. . . _
```

- **Mochila 2**

```
===== Resolvendo a Mochila 2 =====

População inicial:
[[0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0], [1, 0, 0, 1, 1, 1,
0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0], [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
0, 1, 1, 1, 1, 0], [0, 1, 1, 1, 0, 0, 0], [0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1, 1, 1, 1, 0], [0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1],
[0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0], [0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 0, 0, 0, 1, 0, 1, 0], [1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0,
0, 0, 0, 0, 1], [1, 1, 0, 0, 1, 1], [1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
1, 0, 0, 0, 0, 1], [1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0]]

População final:
[[1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 0,
1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1], [1, 0, 1, 1, 1, 1, 1, 1, 1], [1,
1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]

Numero de itens:
20

Capacidade Maxima da mochila:
878

O melhor individuo foi:
[1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
Fitness:
995.0

O pior individuo foi:
[1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1]
Fitness
966.0

Pressione qualquer tecla para continuar. . . _
```

- Mochila 3

```

===== Resolvendo a Mochila 3 =====

População inicial:
[[0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1], [0, 1, 0, 1], [0, 1, 1, 0], [0, 1, 0,
 1], [1, 1, 1, 0], [0, 0, 1, 0], [0, 0, 1, 1], [1, 0, 1, 0]]

População final:
[[1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0,
 1], [1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0, 1], [1, 1, 0, 1]]

Numero de itens:
4

Capacidade Maxima da mochila:
20

O melhor individuo foi:
[1, 1, 0, 1]
Fitness:
35.0

O pior individuo foi:
[1, 1, 0, 1]
Fitness
35.0

Pressione qualquer tecla para continuar. . . _

```

- Mochila 4

```

===== Resolvendo a Mochila 4 =====

População inicial:
[[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 1, 0], [1, 0, 1, 0], [0, 1, 0, 1], [1, 0, 1,
 0], [0, 1, 1, 0], [1, 0, 0, 0], [0, 1, 0, 0], [0, 1, 0, 0]]

População final:
[[0, 1, 0, 1], [0, 1, 0, 1], [0, 1, 0, 1], [0, 1, 0, 1], [0, 1, 0, 1], [0, 1, 0,
 1], [0, 1, 0, 1], [0, 1, 0, 1], [0, 1, 0, 1], [0, 0, 0, 1]]

Numero de itens:
4

Capacidade Maxima da mochila:
11

O melhor individuo foi:
[0, 1, 0, 1]
Fitness:
23.0

O pior individuo foi:
[0, 0, 0, 1]
Fitness
13.0

Pressione qualquer tecla para continuar. . . _

```

- Mochila 5

```

===== Resolvendo a Mochila 5 =====
População inicial:
[[0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0], [0, 0, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1], [0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1], [0, 0, 1, 0, 0, 1, 0, 1, 0, 1], [0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0], [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1], [0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0], [1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1], [1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0], [1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1]]

População final:
[[0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1], [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1]]

Numero de itens:
15

Capacidade Maxima da mochila:
375

O melhor individuo foi:
[0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1]
Fitness:
477.0

O pior individuo foi:
[0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0]
Fitness
378.0

Pressione qualquer tecla para continuar. . .

```

- Mochila 6

```

===== Resolvendo a Mochila 6 =====
População inicial:
[[1, 1, 0, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 1, 0, 1, 0, 0, 0], [0, 1, 0, 1, 0, 1, 0, 1, 1, 0], [1, 0, 1, 1, 0], [1, 0, 0, 0, 1, 1, 0, 0, 0, 1], [0, 0, 1, 0, 1, 0, 1, 0, 1, 0], [0, 0, 0, 1, 1, 1, 0, 0, 0], [0, 0, 1, 1, 0, 1, 1, 0, 1, 1], [0, 1, 1, 0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 1, 0, 0, 1, 0], [0, 0, 0, 0, 1, 0, 0, 0, 1, 1]]

População final:
[[0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [1, 1, 0, 1, 0], [0, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 0], [0, 0, 1, 1, 0, 1, 1, 0, 1, 0], [0, 0, 0, 1, 1, 1, 1, 0, 0, 0]]

Numero de itens:
10

Capacidade Maxima da mochila:
60

O melhor individuo foi:
[0, 1, 0, 1, 0, 1, 1, 0, 1, 0]
Fitness:
49.0

O pior individuo foi:
[0, 0, 0, 1, 1, 1, 1, 0, 0, 0]
Fitness
45.0

Pressione qualquer tecla para continuar. . .

```



- **Mochila 7**

```

===== Resolvendo a Mochila ? =====
População inicial:
[[1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 1, 1], [1, 0, 0, 1, 0, 0, 0], [0, 0, 1,
1, 1, 0, 0], [1, 0, 0, 1, 0, 0, 0], [0, 0, 1, 0, 1, 1, 0], [0, 0, 0, 0, 0, 1, 0],
, [0, 0, 1, 0, 1, 0, 1], [1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 0, 0, 1, 0]]
População final:
[[1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0], [1, 0, 0,
1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0],
, [1, 0, 0, 1, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]]
Numero de itens:
?
Capacidade Maxima da mochila:
50
O melhor individuo foi:
[1, 0, 0, 1, 0, 0, 0]
Fitness:
107.0
O pior individuo foi:
[0, 0, 0, 1, 0, 0, 0]
Fitness
37.0
Pressione qualquer tecla para continuar. . . _

```

- **Mochila 8**

```

===== Resolvendo a Mochila 8 =====
População inicial:
[[1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1], [1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0], [1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0], [0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0], [1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1], [1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0], [1, 0, 0, 1, 1, 0, 0, 0, 1, 0], [1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1], [1, 1, 0, 1, 0, 0, 1, 1], [0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0], [1, 1, 0, 0, 0], [0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1], [0, 1], [0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0], [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1]]
População final:
[[1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 0, 0, 0, 0], [1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0], [0], [1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0], [1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
Numero de itens:
23
Capacidade Maxima da mochila:
10000
O melhor individuo foi:
[1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]
Fitness:
9765.0
O pior individuo foi:
[1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]
Fitness:
8288.0

```

- Mochila 9

```

===== Resolvendo a Mochila 9 =====

População inicial:
[[1, 1, 0, 0, 0], [0, 1, 0, 0, 1], [1, 0, 1, 1, 1], [1, 1, 0, 0, 1], [0, 0, 0, 1, 0], [0, 1, 0, 1, 0], [0, 1, 0, 1, 0], [0, 0, 0, 0, 1], [1, 0, 1, 0, 0], [0, 1, 0, 1, 0]]

População final:
[[1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0], [1, 1, 1, 1, 0]]

Numero de itens:
5

Capacidade Maxima da mochila:
30

O melhor individuo foi:
[1, 1, 1, 1, 0]
Fitness:
130.0

O pior individuo foi:
[0, 1, 1, 1, 0]
Fitness
97.0

Pressione qualquer tecla para continuar. . .

```

- Mochila 10

```

===== Resolvendo a Mochila 10 =====

População inicial:
[[0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0], [0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 0, 0, 0, 1, 1], [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1], [0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0], [0, 0, 0, 0, 1, 0, 1, 0, 1, 0], [0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0], [1, 0, 1, 1, 1], [0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1], [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1]]

População final:
[[1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1], [1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]

Numero de itens:
20

Capacidade Maxima da mochila:
879

O melhor individuo foi:
[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1]
Fitness:
1025.0

O pior individuo foi:
[1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1]
Fitness
985.0

Pressione qualquer tecla para continuar. . . _

```

#### **4. CONCLUSÃO**

Utilizando um algoritmo genético, foi possível ter um contato a mais com o ramo da inteligência artificial e aprendizado de máquina. Foi visto na prática, como o algoritmo utilizou uma base de dados pré-estabelecida como referência para chegar a uma solução aproximada e às vezes exatamente igual do problema, no que se refere ao *fitness* obtido, juntamente com o seu indivíduo correspondente. Dessa forma, pode-se afirmar que os resultados obtidos através do programa foram considerados satisfatórios

## REFERÊNCIAS

- 1- Algoritmos Genéticos. Disponível em: ([https://pt.wikipedia.org/wiki/Algoritmo\\_gen%C3%A9tico](https://pt.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico))
- 2- Problema da Mochila Binária. Disponível em: ([https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/mochila-bool.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/mochila-bool.html)).