

**Disciplina:** Computação Bioinspirada

**Professor:** Prof. Paulo Henrique Ribeiro Gabriel

**Nome:** Felipe Harrison Silva Cantarino **Matrícula:** 12011BSI226

## Projeto 1: Implementação de um algoritmo genético para o problema da mochila

### Composição do Algoritmo

Para o projeto foi criada um algoritmo genético com as seguintes características:

- **Método Seleção:** Seleção por torneio para composição da população intermediária, utilizando elitismo para sempre adicionar o melhor cromossomo.
- **Recombinação:** Crossover de 1-ponto
- **Mutação:** Bit Flip, com probabilidade de 20%
- **Substituição:** Substituição dos n piores cromossomos da geração

### Entrada de dados

Como entrada de dados para composição da mochila e do problema, foram escolhidos os 4 primeiros arquivos de entrada dentre os disponíveis no repositório <<https://github.com/neemiasbsilva/knapsack-problem-using-dp-grasp-tabu/tree/master/entradas>>, resultando nos arquivos: "input1.in", "input2.in", "input3.in", "input4.in".

### Algoritmo para comparação

Como algoritmo escolhido para comparar com o algoritmo genético criado, foi escolhido o algoritmo de Programação dinâmica, disponibilizado no OneDrive. Neste algoritmo foram feitas apenas alterações na parte de leitura de inputs.

### Valores ideais de cada entrada

Com base nas entradas coletadas, foi utilizado o algoritmo de programação dinâmica para definir o valor ótimo para cada uma das entradas. Por meio dessa execução chegou-se aos resultados apresentados na Tabela 1.

	Valor Ideal
input 1	31621
input 2	67829
input 3	143449
input 4	28840

Tabela 1 – Tabela de valor ideal para cada uma das mochilas

Para obter estes valores, o algoritmo foi executado 5 vezes sobre todas as entradas e em média cada execução gastou 0,38 segundos para fornecer os resultados.

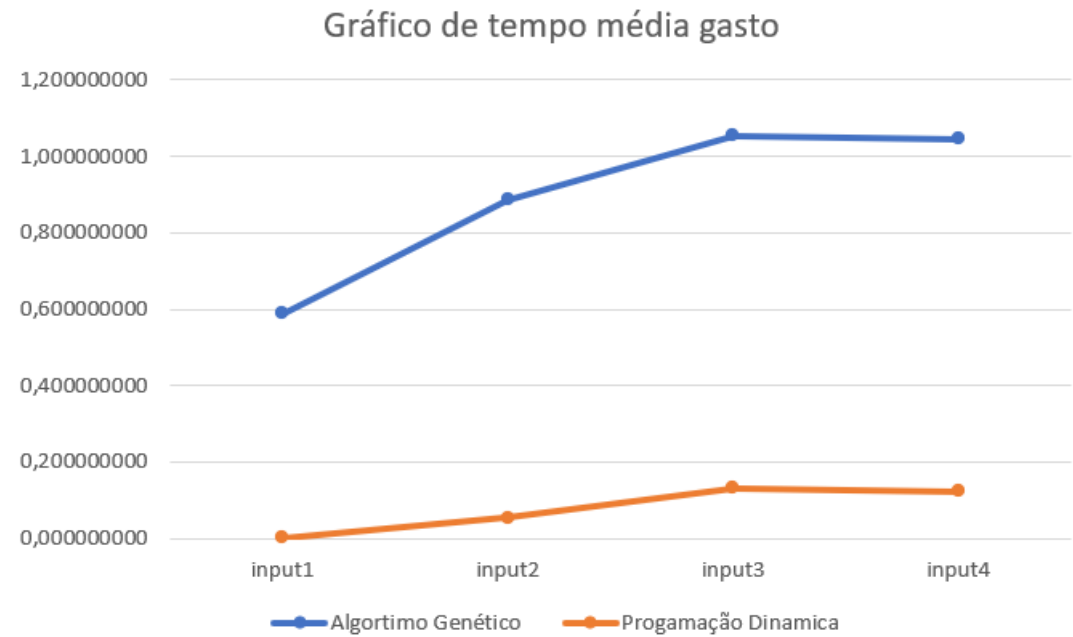
## Resultados obtidos

Para execução do algoritmo genético criado foi utilizado os seguintes parâmetros:

- Número de cromossomos no grupo: 20
- Número de interações do algoritmo: 50000
- Taxa de mutação: 0.15 (15%)
- Taxa de substituição dos indivíduos: 2 indivíduos (piores)

Para realizar a análise, para cada entrada ambos os algoritmos foram executados 30 vezes. Em seguida, foi calculado a média de tempo gasto e a frequência das respostas obtidas para comparar os algoritmos.

## Tempos de execução



Analisando o gráfico criado é possível notar que o algoritmo genético criado se mostrou inferior ao algoritmo dinâmico, no tempo de execução.

## Frequencia de respostas

Algoritmo Genetico		Dinamico	
31621	30	31621	30

Tabela 2 – Tabela de frequencia das respostas para o input1

Algoritmo Genetico		Dinamico	
67829	30	67829	30

Tabela 3 – Tabela de frequencia das respostas para o input2

Algoritmo Genetico		Dinamico	
131064	1	143449	30
132910	1		
133546	1		
136458	1		
136935	3		
138507	1		
138750	2		
143449	20		

Tabela 4 – Tabela de frequencia das respostas para o input3

Algortimo Genetico		Dinamico	
26466	1	28840	30
26887	1		
26948	1		
27137	2		
27140	1		
27188	1		
27641	1		
27657	2		
27680	2		
27709	5		
27762	1		
27830	1		
28293	1		
28395	1		
28445	1		
28446	2		
28463	2		
28615	1		
28616	2		
28670	1		

Tabela 5 – Tabela de frequencia das respostas para o input4

Numa visão geral, o algoritmo genético criado apresenta dificuldades para mochilas mais complexas, no caso para o "input3" e "input4". Analisando a Tabela 4, o algoritmo chegou na resposta ideal apenas 20 vezes, entretanto observando a Tabela 5, é possível notar que em todas as 30 execuções o algoritmo não chegou na resposta ideal.

## Resultado Final

Em síntese, com base no algoritmo criado e tabelas de resultados analisadas, é possível notar que os resultados obtidos foram inferiores ao esperado. Nesse sentido buscando melhorar o algoritmo, seria necessária uma análise no código gerado, afim de otimizar algumas partes, por exemplo, *loops for* desnecessários ou com complexidade alta. Além disso, alterar os parametros do algoritmo, como taxa de mutação e tamanho do grupo, podem melhor o resultado do algoritmo.