

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Departamento de Informática e Estatística - INE
Ciências da Computação
INE5430 - Inteligência Artificial

Trabalho Algoritmos Genéticos

Henrique Pereira Ramos (21203360)
Luiz Maurício do Valle Pereira (21104157)

**Florianópolis,
5 de maio de 2025**

1 Problema abordado

Neste trabalho foi abordado o problema da mochila, problema de otimização já conhecido na literatura.

2 Solução

Para a implementação da solução foi adotada a biblioteca *pygad* (<https://pygad.readthedocs.io/en/latest/>). Código desenvolvido e instruções para execução do *jupyter notebook* estão disponíveis no seguinte repositório: https://github.com/hpereira1/knapsack_GA.

2.1 Parâmetros adotados

Foram adotados os seguintes parâmetros para o algoritmo genético:

- 300 itens aleatórios distintos, cada qual com seus próprios pesos e valores
- Cada item distinto é considerado um gene
- Mochila com capacidade de 5000
- 1000 gerações
- 100 soluções por população
- 20 indivíduos da população são utilizados para *crossover*
- O melhor indivíduo da população será mantido para a próxima geração
- Seleção dos "pais" de tipo *sss*
- *Crossover* uniforme
- Mutações aleatórias
- Chance de mutação por gene de 10%

A função de *fitness* adotada é como segue:

```
1 def fitness_func(ga_instance, solution, solution_idx):
2     total_w = np.sum(solution * np.array(items_df['weight']))
3     total_val = np.sum(solution * np.array(items_df['value']))
4     if total_w > knapsack_capacity:
5         # return 0.0 # hard penalty
6         return total_val - 2*(max(0, total_w - knapsack_capacity)) # soft penalty
7     return float(total_val)
```

Figura 1: Função de *fitness*

3 Resultados

A melhor solução encontrada para o problema com os parâmetros pré estabelecidos foi:

- Peso final = 4965
- Valor final = 4253

Figuras 2, 3 e 4 demonstram visualmente os resultados obtidos. A Figura 2 demonstra a máxima *fitness* por cada geração. Figura 3 é um comparativo entre a melhor e a segunda melhor *fitness* de cada geração. A Figura 4 compara os desempenhos do algoritmo de geração a geração, destacando o máximo, a média e a mínima *fitness* encontrada.

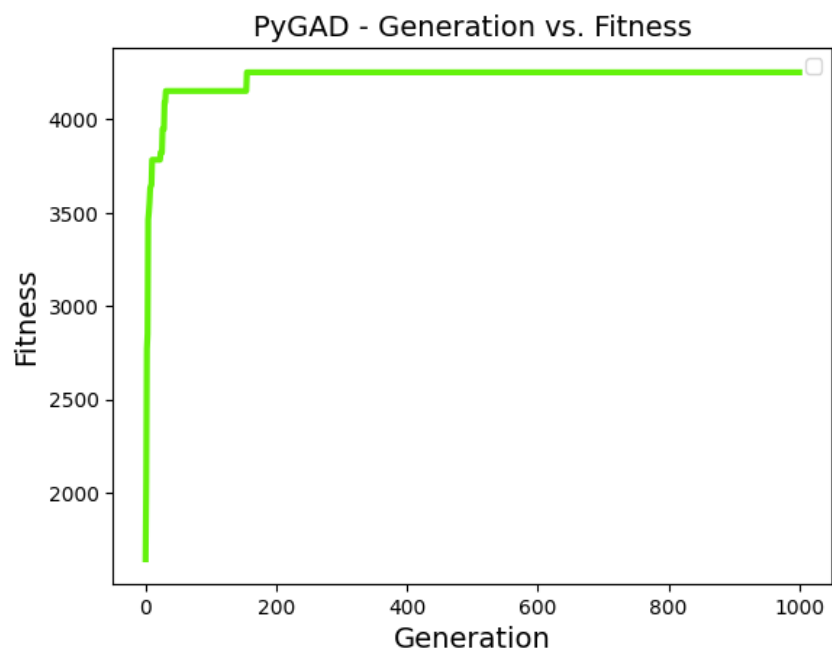


Figura 2: Gráfico de *fitness* por geração

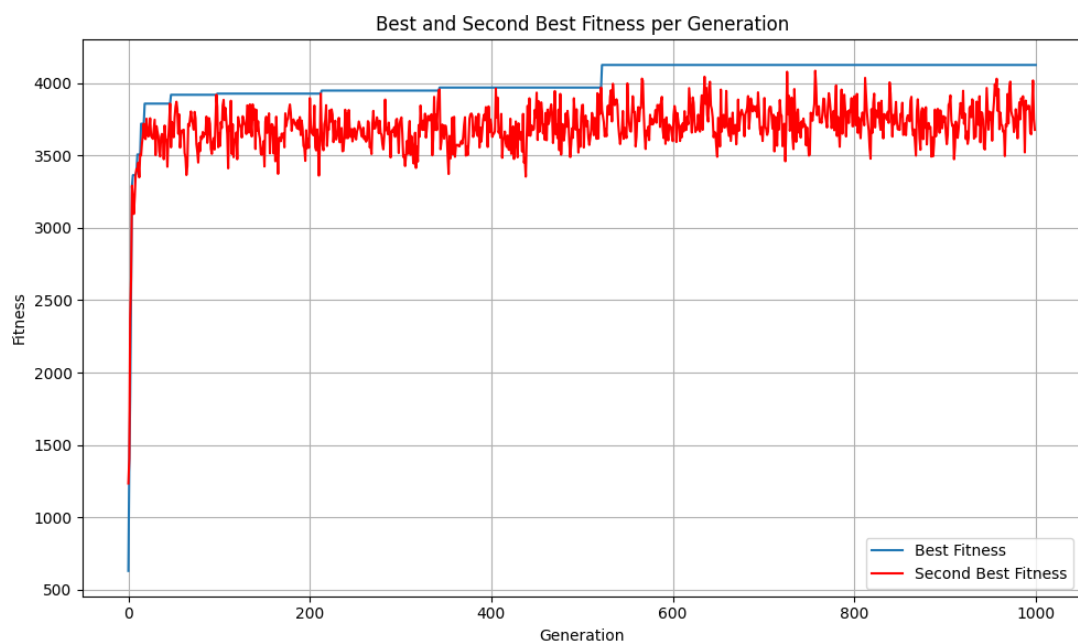


Figura 3: Comparativo entre a melhor e a segunda melhor *fitness* de cada geração



Figura 4: Comparativo entre máxima, média e mínima *fitness* encontrada