

Atividade Tópicos Especiais em Inteligência artificial

Aluno: Elton Luiz Alves da Silva

Repositório: <https://github.com/eltonluiz178/Knapsack-Hill-Climbing>

1. A primeira questão solicitou a implementação do algoritmo knapsack operando em 20 dimensões. Para isso, foi utilizado o código disponibilizado pelo professor em seu repositório no Github. Os algoritmos foram configurados para apresentar o maior valor de fitness obtido na resolução do problema, assim como o total de iterações realizadas. O maior valor de fitness registrado foi 997.

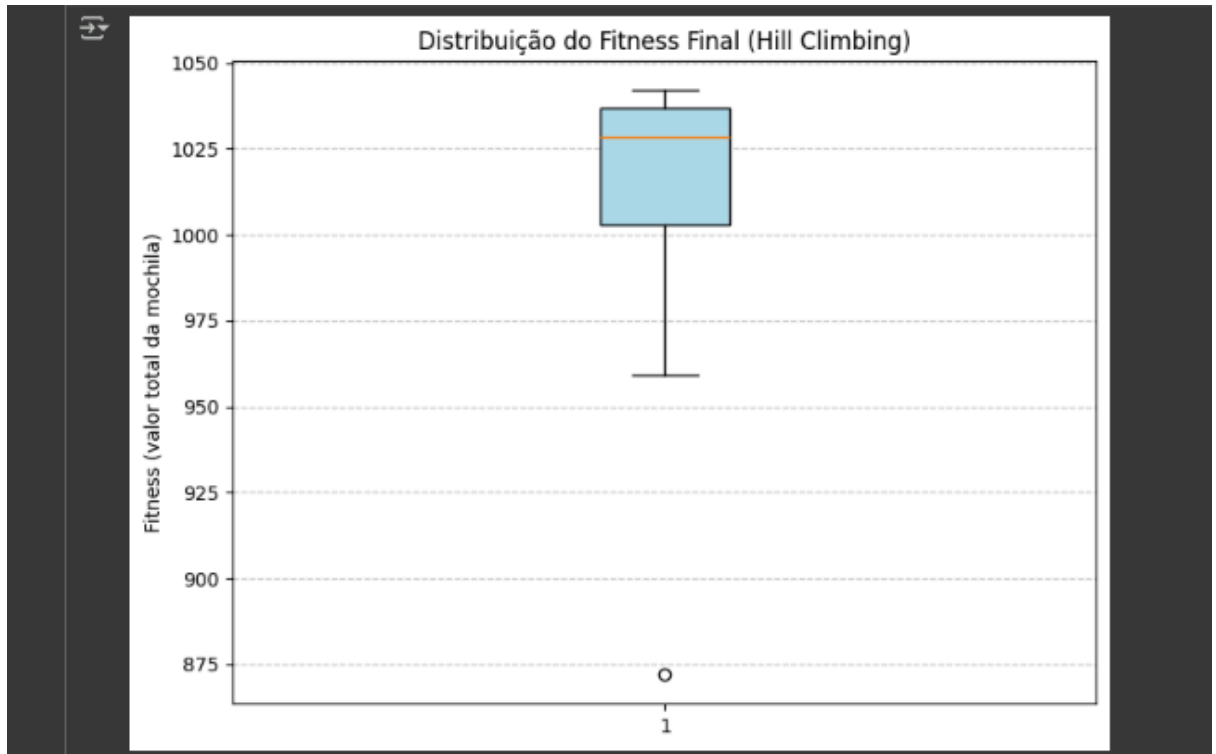
```
Iteração 0: Fitness = 203.0000
Iteração 1: Fitness = 299.0000
Iteração 2: Fitness = 391.0000
Iteração 3: Fitness = 475.0000
Iteração 4: Fitness = 567.0000
Iteração 5: Fitness = 650.0000
Iteração 6: Fitness = 733.0000
Iteração 7: Fitness = 803.0000
Iteração 8: Fitness = 861.0000
Iteração 9: Fitness = 917.0000
Iteração 10: Fitness = 949.0000
Iteração 11: Fitness = 997.0000
Convergiu na iteração 12
Melhorias realizadas: 11
Fitness final: 997.0000

=== RESULTADOS FINAIS ===
Solução inicial: [0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Melhor solução: [1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1]
Melhor valor total: 997
Peso total da melhor solução: 855

Histórico de fitness ao longo das iterações:
[203, 299, 391, 475, 567, 650, 733, 803, 861, 917, 949, 997]
```

2. No código fornecido pelo professor foi feita uma pequena adaptação: adicionou-se um laço FOR para executar 30 repetições e armazenar os valores de fitness em uma lista. Ao final, foi calculada a média e também o desvio padrão desses resultados. O gráfico evidenciou a estabilidade do método Hill Climbing em valores mais elevados, com a mediana próxima de 1020.

```
=== RESULTADOS FINAIS DO HILL CLIMBING ===  
Número de execuções: 30  
Média do fitness final: 1015.10  
Desvio padrão do fitness final: 33.69  
Melhor fitness obtido: 1042  
Pior fitness obtido: 872
```



3. No algoritmo de Hill Climbing foi possível ver a convergência na 15ª iteração. No Hill Climbing Stochastic, por sua vez, houve uma convergência mais rápida, apenas na 10ª iteração. As outras métricas reúnem os valores totais e o histórico. Na análise do boxplot, é possível perceber que o stochastic tem uma caixa mais comprida, o que o deixa menos consistente, mas explora mais possibilidades de vizinhos diferentes. O Hill Climbing tradicional é mais consistente, mas tem outlier.

```
Iteração 0: Fitness = 56.0000
Iteração 1: Fitness = 152.0000
Iteração 2: Fitness = 244.0000
Iteração 3: Fitness = 336.0000
Iteração 4: Fitness = 420.0000
Iteração 5: Fitness = 503.0000
Iteração 6: Fitness = 585.0000
Iteração 7: Fitness = 668.0000
Iteração 8: Fitness = 716.0000
Iteração 9: Fitness = 774.0000
Iteração 10: Fitness = 844.0000
Iteração 11: Fitness = 888.0000
Iteração 12: Fitness = 913.0000
Iteração 13: Fitness = 931.0000
Iteração 14: Fitness = 999.0000
Iteração 15: Fitness = 1042.0000
Convergiu na iteração 16
Melhorias realizadas: 15
Fitness final: 1042.0000

=== RESULTADOS FINAIS ===
Solução inicial: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
Melhor solução: [1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1]
Melhor valor total: 1042
Peso total da melhor solução: 878

Histórico de fitness ao longo das iterações:
[56, 152, 244, 336, 420, 503, 585, 668, 716, 774, 844, 888, 913, 931, 999, 1042]
```

```
↳ Iteração 0: Fitness = 272.0000
Iteração 1: Fitness = 368.0000
Iteração 2: Fitness = 426.0000
Iteração 3: Fitness = 518.0000
Iteração 4: Fitness = 610.0000
Iteração 5: Fitness = 693.0000
Iteração 6: Fitness = 775.0000
Iteração 7: Fitness = 843.0000
Iteração 8: Fitness = 899.0000
Iteração 9: Fitness = 943.0000
Iteração 10: Fitness = 975.0000
Convergiu na iteração 11
Melhorias realizadas: 10
Fitness final: 975.0000

=== RESULTADOS FINAIS ===
Solução inicial: [0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0]
Melhor solução: [1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1]
Melhor valor total: 975
Peso total da melhor solução: 851

Histórico de fitness ao longo das iterações:
[272, 368, 426, 518, 610, 693, 775, 843, 899, 943, 975]
```

```
plt.boxplot([resultados_tradicional, resultados_estocastico], labels=['Tradicional', 'Estocastico'], patch_artist=True)
```

