

**Supercomputação**  
**Entrega - Atividade 8**  
**Arthur Tamm**

## **Introdução**

Nesta aula, implementamos dois novos algoritmos heurísticos para resolver o problema da mochila: a Substituição de Objeto e o Hill Climbing. O objetivo foi melhorar as heurísticas abordadas na última aula, baseadas em embaralhamento e seleção probabilística, comparando suas performances em termos de tempo de execução e qualidade das soluções. Para cada método, rodamos os algoritmos 10 vezes em quatro arquivos de entrada, retornando a melhor solução obtida e avaliando os ganhos em relação às abordagens anteriores.

## **Substituição de Objeto**

### **Descrição do Método**

A Substituição de Objeto consiste em gerar uma solução aleatória e, em seguida, verificar se é possível aumentar o valor da mochila trocando itens selecionados por outros que não foram incluídos, respeitando a restrição de capacidade. Se a troca melhora a solução, o processo é repetido até que não haja mais ganhos possíveis.

### **Resultados**

<b>Entrada</b>	<b>Melhor Valor</b>	<b>Tempo Médio de Execução (s)</b>
input1.txt	509	0.005994
input2.txt	620	0.005297
input3.txt	3320	0.005543
input4.txt	639	0.005775

### **Análise**

A Substituição de Objeto apresentou uma melhoria significativa na qualidade das soluções comparada com as heurísticas probabilística e de embaralhamento, especialmente para input1.txt e input2.txt, onde os valores máximos atingidos foram superiores aos das abordagens anteriores. No caso do input3.txt, o resultado ótimo (3320) foi facilmente alcançado, já que todos os itens podem ser incluídos na mochila, simplificando o problema. O método manteve um tempo de execução baixo, próximo ao das heurísticas mais simples, tornando-o uma abordagem promissora para problemas onde a capacidade da mochila é mais restritiva.

# Hill Climbing

## Descrição do Método

O Hill Climbing é uma técnica de otimização local que tenta melhorar uma solução inicial, gerando vizinhos (soluções semelhantes) a partir de modificações de bits na representação binária dos itens incluídos na mochila. O algoritmo seleciona o melhor vizinho e repete o processo até que não haja mais melhorias.

## Resultados

Entrada	Melhor Valor	Tempo Médio de Execução (s)
input1.txt	379	0.005398
input2.txt	412	0.005771
input3.txt	3320	0.005944
input4.txt	546	0.005824

## Análise

O Hill Climbing apresentou resultados satisfatórios para o input3.txt, atingindo consistentemente o valor ótimo de 3320. Isso se deve à característica específica desse arquivo, que permite incluir todos os itens na mochila. Para os demais arquivos, especialmente os que possuem restrições mais severas de capacidade (como input1.txt e input2.txt), o método se mostrou menos eficaz, gerando soluções de menor qualidade em comparação com a Substituição de Objeto.

---

## Comparação das Heurísticas

Algoritmo	Melhor Valor (input1)	Melhor Valor (input2)	Melhor Valor (input3)	Melhor Valor (input4)
Embaralhamento	304	350	3320	566
Heurística Probabilística	351	381	1790	488
Substituição de Objeto	509	620	3320	639
Hill Climbing	379	412	3320	546

## **Avaliação dos Ganhos**

A técnica de Substituição de Objeto demonstrou-se a mais eficiente, tanto em termos de valor máximo alcançado quanto em tempo de execução. Ela superou consistentemente as heurísticas de embaralhamento e probabilística, mostrando uma capacidade superior de exploração do espaço de soluções. Já o Hill Climbing, apesar de alcançar o valor ótimo para input3.txt, mostrou-se menos eficaz nas outras entradas, especialmente em cenários com mais restrições.

---

## **Conclusão**

Os algoritmos desenvolvidos nesta aula, Substituição de Objeto e Hill Climbing, trouxeram melhorias em relação às heurísticas aleatórias implementadas anteriormente. A Substituição de Objeto destacou-se como a melhor estratégia para a maioria das entradas, combinando rapidez e qualidade das soluções. O Hill Climbing, por sua vez, demonstrou uma boa performance em problemas com menos restrições, mas ficou aquém em casos mais complexos. As análises indicam que a escolha do algoritmo deve considerar a estrutura do problema e o balanceamento entre qualidade e tempo de execução.