

Supercomputação
Entrega - Atividade 9
Arthur Tamm

Introdução

Neste relatório, analisaremos duas heurísticas clássicas para resolver o problema da mochila binária: "Mais Caro Primeiro" e "Mais Leve Primeiro". Implementamos e comparamos as duas abordagens em termos de complexidade computacional e eficácia, e avaliamos suas performances em diferentes entradas, demonstrando situações onde cada uma é mais eficaz.

Mais Caro Primeiro

A heurística "Mais Caro Primeiro" se baseia em priorizar os itens de maior valor para maximizar o valor total da mochila. A lógica é simples: começamos pelos itens mais caros e continuamos adicionando até que a capacidade da mochila seja atingida. Caso um item não caiba, passamos para o próximo item mais barato.

Complexidade Computacional A complexidade do algoritmo é dominada pela operação de ordenação dos itens com base no valor, que é $O(n \log n)$. Após a ordenação, percorremos os itens, o que requer $O(n)$ operações. Assim, a complexidade final é $O(n \log n)$.

É a Melhor Implementação Possível? Apesar de ser uma abordagem simples e eficiente, ela não garante a solução ótima, pois é uma heurística gananciosa que não explora combinações que poderiam resultar em valores maiores. Pode ser eficiente em muitos casos, mas não é a melhor implementação possível, uma vez que algoritmos como programação dinâmica ou busca exaustiva conseguem garantir o resultado ótimo.

Mais Leve Primeiro

Na heurística "Mais Leve Primeiro", o foco é maximizar o número de itens na mochila. Ordenamos os itens por peso crescente e adicionamos o maior número possível de itens, na esperança de que, somados, tragam um bom valor total.

Complexidade Computacional Assim como na heurística "Mais Caro Primeiro", a parte dominante da complexidade é a ordenação, que é $O(n \log n)$. Depois de ordenar, percorremos os itens e verificamos quais cabem na mochila, o que toma $O(n)$. Portanto, a complexidade final é $O(n \log n)$.

Analisando as Heurísticas

Situações em que Cada Heurística é Melhor:

- **Mais Caro Primeiro** é melhor quando existem poucos itens de valor muito alto que podem ser incluídos na mochila sem ultrapassar sua capacidade. Isso garante que o valor total seja maximizado, mesmo com menos itens.
- **Mais Leve Primeiro** é melhor quando há muitos itens de valor moderado, mas de peso muito baixo. Nesses casos, a mochila consegue conter mais itens, o que resulta em um valor total maior.

Entrada em que a Heurística do "Mais Caro Primeiro" funciona melhor:

Itens disponíveis	Capacidade da mochila
5	10

Item	Peso	Valor
1	5	6
2	4	5
3	3	3
4	2	2
5	1	1

Como nesse caso os itens mais caros têm valores altos, a heurística "Mais Caro Primeiro" é superior.

Mais Caro Primeiro:

Max value: 12

Final weight: 10

Items selected: 1 2 5

Mais Leve Primeiro:

Max value: 11

Final weight: 10

Items selected: 5 4 3 2

Entrada em que a Heurística do "Mais Leve Primeiro" funciona melhor:

Itens disponíveis	Capacidade da mochila
5	10

Item	Peso	Valor
1	5	5
2	4	4
3	3	4
4	2	3
5	1	2

Como nesse caso os itens mais leves têm valores moderados, a heurística "Mais Leve Primeiro" é superior.

Mais Caro Primeiro:

Max value: 11

Final weight: 10

Items selected: 1 2 5

Mais Leve Primeiro:

Max value: 13

Final weight: 10

Items selected: 5 4 3 2

Nova Heurística: Melhor Ratio Valor/Peso

Após a análise das heurísticas "Mais Caro Primeiro" e "Mais Leve Primeiro", decidi testar uma nova abordagem que combina os pontos fortes de ambas. Esta heurística se baseia em selecionar os itens com o **melhor ratio valor/peso**, ou seja, os itens que oferecem o maior valor por unidade de peso. A ideia é maximizar a eficiência da mochila, garantindo que cada unidade de capacidade seja usada de forma otimizada.

Descrição do Método

A heurística começa calculando o ratio valor/peso para cada item. Em seguida, os itens são ordenados de acordo com esse ratio em ordem decrescente. A mochila é preenchida da mesma forma que nas abordagens anteriores: percorremos os itens e os adicionamos se ainda houver capacidade.

Complexidade Computacional

A complexidade computacional desta heurística é semelhante às abordagens anteriores, com $O(n \log n)$ devido à necessidade de ordenar os itens de acordo com o ratio valor/peso.

Resultados

Esta heurística se mostrou muito eficaz, alcançando o melhor valor em ambas as entradas testadas, tanto quando há itens muito valiosos quanto quando há muitos itens leves com valores moderados. Ao balancear o valor e o peso de cada item, a heurística conseguiu combinar o melhor de ambos os mundos.

Entrada 1

Itens disponíveis	Capacidade da mochila
5	10

Item	Peso	Valor
1	5	6
2	4	5
3	3	3
4	2	2
5	1	1

Resultados

Max value: 12

Final weight: 10

Items selected: 2 1 5

Entrada 2

Itens disponíveis	Capacidade da mochila
5	10

Item	Peso	Valor
1	5	5
2	4	4
3	3	4
4	2	3
5	1	2

Resultados

Max value: 13

Final weight: 10

Items selected: 5 4 3 2

Além disso, essa heurística alcançou a melhor solução possível para as 4 entradas testadas nas aulas passadas.

Conclusão

Neste relatório, analisamos e implementamos três heurísticas para o problema da mochila binária: "Mais Caro Primeiro", "Mais Leve Primeiro", e uma nova heurística baseada no **ratio valor/peso**. As duas primeiras abordagens apresentaram bons resultados em cenários específicos, sendo a "Mais Caro Primeiro" mais eficaz em situações com poucos itens de alto valor, enquanto a "Mais Leve Primeiro" se destacou quando havia muitos itens leves e moderadamente valiosos. No entanto, a heurística de **ratio valor/peso** se mostrou a mais eficiente e versátil, combinando o melhor das abordagens anteriores e alcançando a solução ótima nas entradas testadas. Com uma complexidade computacional similar às outras heurísticas ($O(n \log n)$), essa nova estratégia oferece uma solução robusta para maximizar o valor da mochila em uma variedade de cenários.

