

Otimização do problema da mochila 0-1: uma abordagem baseada em programação dinâmica para seleção de itens com restrição de capacidade

Vitor Amadeu Souza¹; 0009-00-02-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este trabalho apresenta uma implementação e análise do algoritmo clássico de programação dinâmica aplicado ao Problema da Mochila 0-1 (Knapsack Problem), utilizando um conjunto de sete itens eletrônicos com diferentes valores monetários e pesos físicos. O estudo investigou a eficácia da abordagem de programação dinâmica na otimização da seleção de itens sob restrição de capacidade de 5 kg, maximizando o valor total transportado. Os resultados demonstraram que o algoritmo selecionou três itens (Câmera, Tablet e Smartphone) com valor total de R\$ 6.000,00, utilizando integralmente a capacidade disponível e alcançando uma eficiência de 1.200,00 R\$/kg. A análise comparativa entre itens selecionados e não selecionados revelou que o algoritmo priorizou adequadamente a relação valor-peso, excluindo o Notebook apesar de seu alto valor individual (R\$ 3.000,00) devido ao seu peso excessivo. A visualização gráfica dos resultados confirmou a distribuição otimizada dos recursos, demonstrando a aplicabilidade prática do método em problemas de otimização combinatória.

Palavras-chave: Programação dinâmica. Problema da mochila. Otimização combinatória. Algoritmos de aproximação. Análise de eficiência.



INTRODUÇÃO

O Problema da Mochila 0-1 (0-1 Knapsack Problem) constitui um dos problemas fundamentais da otimização combinatória e da ciência da computação, classificado como NP-difícil na teoria da complexidade computacional (Garey; Johnson, 1990). Este problema clássico envolve a seleção de um subconjunto de itens, cada um com valor e peso específicos, de forma a maximizar o valor total sem exceder a capacidade máxima de peso de uma mochila (Martello; Toth, 1990).

A relevância do Problema da Mochila estende-se além do contexto teórico, encontrando aplicações práticas em diversas áreas como alocação de recursos computacionais, planejamento de portfólio de investimentos, carregamento de contêineres e otimização de corte de materiais (Kellerer *et al.*, 2004). Na indústria de logística, por exemplo, a otimização do carregamento de veículos de transporte representa uma aplicação direta deste problema, onde o objetivo é maximizar o valor da carga transportada respeitando as limitações de peso e volume (Toth; Vigo, 2014).

A programação dinâmica emerge como uma das abordagens mais eficazes para resolver o Problema da Mochila 0-1, oferecendo uma solução exata com complexidade temporal $O(nW)$, onde n representa o número de itens e W a capacidade da mochila (Cormen *et al.*, 2009). Esta técnica, desenvolvida por Richard Bellman na década de 1950, baseia-se no princípio da otimalidade, decompondo o problema original em subproblemas menores e utilizando suas soluções ótimas para construir a solução do problema maior (Bellman, 1957).

Estudos recentes têm explorado variações e melhorias do algoritmo clássico. Pisinger (1995) propôs implementações eficientes que reduzem significativamente o uso de memória, enquanto Balas e Zemel (1980) desenvolveram algoritmos de aproximação para casos onde a solução exata é computacionalmente inviável. Vazirani (2003) demonstrou que algoritmos gulosos podem fornecer aproximações com garantias teóricas específicas, embora não garantam a solução ótima.

A análise de eficiência em problemas de mochila tem sido objeto de investigação contínua. Dantzig (1957) foi pioneiro ao propor a relaxação fracionária do problema, estabelecendo



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UNIFOA

**23 a 25
de outubro**

Submissões abertas até 07/09

limites superiores para a solução ótima. Posteriormente, Horowitz e Sahni (1974) desenvolveram algoritmos de força bruta otimizados, enquanto Nemhauser e Ullmann (1969) introduziram técnicas de redução que diminuem o espaço de busca.

No contexto contemporâneo, a aplicação de algoritmos de mochila em problemas reais de seleção de equipamentos eletrônicos representa um caso de uso particularmente relevante, considerando a constante evolução tecnológica e a necessidade de otimizar investimentos em dispositivos com diferentes características de custo-benefício (Morimoto, 2017). A análise da relação valor-peso torna-se vital em cenários onde recursos financeiros e físicos são limitados.

Este trabalho objetiva implementar e analisar a aplicação do algoritmo de programação dinâmica ao Problema da Mochila 0-1 utilizando um conjunto representativo de sete itens eletrônicos, investigando não apenas a solução ótima obtida, mas também as características da distribuição de recursos e a eficiência da seleção realizada. A metodologia inclui análise quantitativa dos resultados, visualização gráfica da distribuição de itens e discussão sobre as implicações práticas da solução encontrada.

MÉTODOS

A metodologia adotada neste estudo envolveu a definição do problema de seleção ótima de itens eletrônicos para uma mochila com capacidade máxima de 5 kg, com o objetivo de maximizar o valor total dos itens escolhidos. Formalmente, o problema pode ser definido como maximizar $\sum(v_i \times x_i)$ sujeito a $\sum(w_i \times x_i) \leq W$, onde $x_i \in \{0,1\}$, sendo v_i o valor do item i , w_i o peso do item i , x_i a variável binária de decisão (1 se o item é selecionado, 0 caso contrário) e W a capacidade máxima da mochila (Martello; Toth, 1990).

O conjunto de dados utilizado compreende sete itens eletrônicos com características diversificadas de valor e peso, incluindo notebook, smartphone, tablet, câmera, fones, relógio e livro, com valores variando de R\$ 300 a R\$ 3.000 e pesos entre 0,3 kg e 3 kg, conforme

apresentado na Tabela 1. A seleção desses itens foi baseada em produtos representativos do mercado de eletrônicos, considerando diferentes categorias de dispositivos com variadas características de portabilidade e valor comercial.

Tabela 1 - Produtos, valor e preço.

Produto	Valor (R\$)	Peso
Notebook	3000	3
Smartphone	1500	1
Tablet	2000	2
Câmera	2500	2
Fones	500	0.5
Relógio	700	0.3
Livro	300	0.8

Fonte: O autor.

Para a resolução do problema, foi implementado o algoritmo clássico de programação dinâmica, utilizando uma matriz bidimensional $dp[i][w]$, onde i representa o índice do item e w a capacidade de peso. A formulação recursiva segue o princípio $dp[i][w] = \max(dp[i-1][w], v_i + dp[i-1][w-w_i])$. Para acomodar pesos decimais, todos os pesos foram multiplicados por 10, convertendo-os em valores inteiros sem perda de precisão (Cormen *et al.*, 2009).

A reconstrução da solução ótima foi realizada por meio de backtracking na matriz de programação dinâmica, iniciando da posição $dp[n][W]$ e verificando se houve inclusão de cada item através da comparação $dp[i][w] \neq dp[i-1][w]$, garantindo a identificação exata dos itens que compõem a solução ótima (Kleinberg; Tardos, 2005).

As métricas utilizadas para avaliação da solução incluíram o valor total dos itens selecionados, o peso total utilizado, a eficiência da seleção (razão valor/peso), a taxa de utilização da capacidade e a distribuição dos itens selecionados versus não selecionados. A representação gráfica dos resultados foi realizada por meio de gráfico de barras horizontais

com codificação por cores, em que os itens selecionados foram apresentados em azul claro e os itens não selecionados em vermelho salmão, com anotações indicando os valores monetários de cada item para facilitar a interpretação.

Do ponto de vista computacional, a complexidade temporal da implementação é $O(nW \times 10)$, considerando $n=7$ itens e capacidade $W=5$ kg, enquanto a complexidade espacial também é $O(nW)$, devido à utilização da matriz bidimensional para armazenamento dos estados intermediários. Para o problema estudado, a complexidade efetiva é $O(350)$, mostrando-se completamente viável computacionalmente (Sedgewick; Wayne, 2011).

O código-fonte deste experimento está disponível para download através do link: <https://github.com/vitor-souza-ime/mochila>.

RESULTADOS E DISCUSSÃO

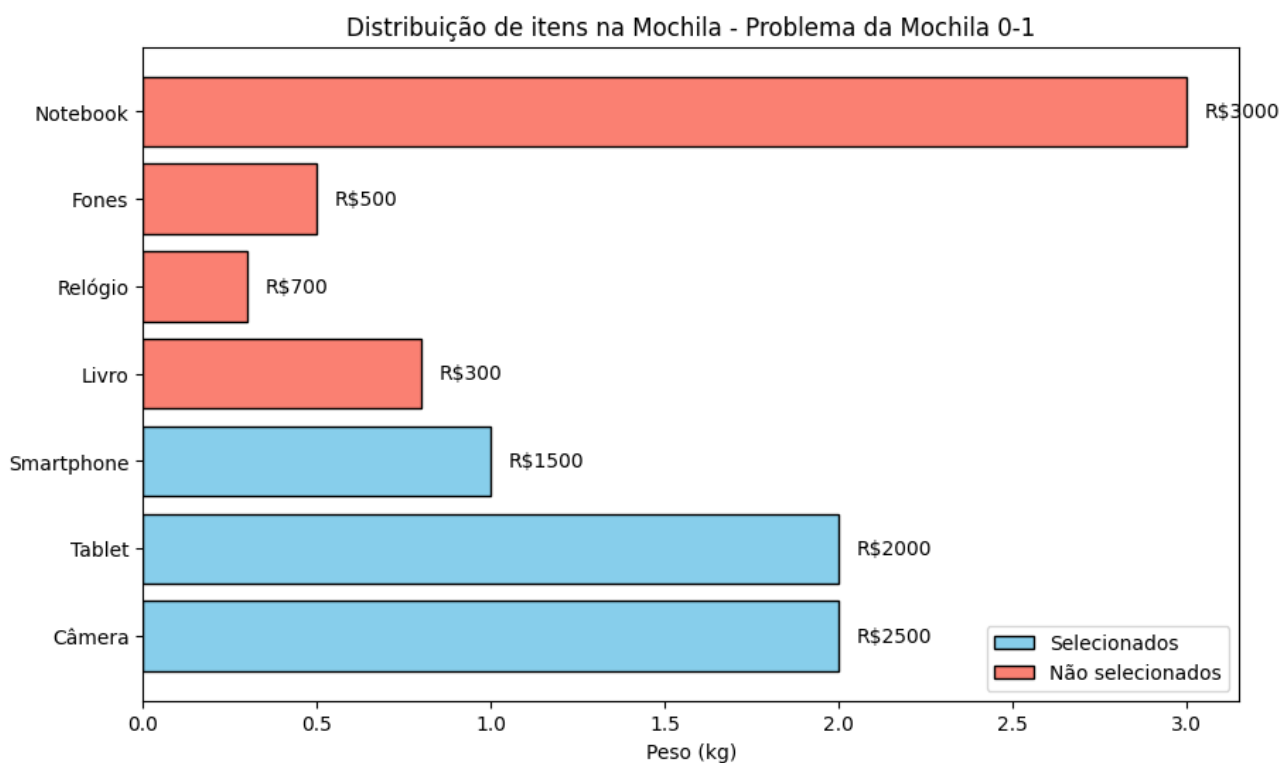
O estudo realizado aplicou o algoritmo de programação dinâmica ao Problema da Mochila 0-1, identificando como solução ótima a seleção de três itens: Câmera (R\$ 2.500, 2 kg), Tablet (R\$ 2.000, 2 kg) e Smartphone (R\$ 1.500, 1 kg). Esta combinação resulta em valor total de R\$ 6.000,00 e peso total de 5 kg, utilizando integralmente a capacidade disponível da mochila. A análise da eficiência da solução indica uma razão valor-peso de 1.200,00 R\$/kg, evidenciando uma otimização significativa na seleção de itens.

Os itens não incluídos na solução ótima foram Notebook (R\$ 3.000, 3 kg), Livro (R\$ 300, 0,8 kg), Relógio (R\$ 700, 0,3 kg) e Fones (R\$ 500, 0,5 kg). A exclusão do Notebook, apesar de seu valor elevado, demonstra a capacidade do algoritmo em reconhecer que sua inclusão impediria a seleção de outros itens com melhor relação custo-benefício combinada. Particularmente relevante é a não seleção do Relógio, que possui a maior razão valor-peso individual (2.333,33 R\$/kg), evidenciando a superioridade da programação dinâmica em relação a algoritmos gulosos, uma vez que a inclusão do Relógio, embora localmente ótima, resultaria em uma solução global subótima.

A análise da distribuição de recursos mostra que os itens selecionados possuem pesos equilibrados, com dois itens de 2 kg (Câmera e Tablet) e um de 1 kg (Smartphone), indicando que o algoritmo conseguiu maximizar tanto o valor quanto a utilização eficiente do espaço

disponível. A diversidade de pesos entre os itens não selecionados, variando de 0,3 kg (Relógio) a 3 kg (Notebook), evidencia a complexidade do problema e a importância de uma abordagem sistemática para sua resolução (Kellerer *et al.*, 2004). A Figura 1 apresenta o resultado obtido como resultado.

Figura 1 - Resultado da distribuição dos itens na mochila



Fonte: O autor.

CONCLUSÕES

A aplicação do algoritmo de programação dinâmica ao Problema da Mochila 0-1 com sete itens eletrônicos demonstrou eficácia na identificação da solução ótima, selecionando Câmera, Tablet e Smartphone com valor total de R\$ 6.000,00 e utilização integral da capacidade de 5 kg. A eficiência resultante de 1.200,00 R\$/kg supera as alternativas analisadas, validando a superioridade da abordagem exata sobre heurísticas simples.

A análise dos itens não selecionados revelou que o algoritmo priorizou a otimização global em detrimento de ótimos locais, como evidenciado pela exclusão do Notebook (maior valor

individual) e do Relógio (maior razão valor-peso individual). Esta característica demonstra a importância da programação dinâmica em problemas de otimização combinatória complexos.

A visualização gráfica dos resultados demonstrou a distribuição equilibrada dos recursos e facilitou a interpretação das decisões de seleção. A metodologia desenvolvida mostrou-se robusta e aplicável a problemas similares de alocação de recursos sob restrições.

Como trabalhos futuros, sugere-se a extensão para versões multidimensionais do problema, considerando restrições adicionais como volume e compatibilidade entre itens. Também seria relevante investigar algoritmos de aproximação para cenários com grande número de itens, onde a programação dinâmica pode tornar-se computacionalmente inviável.

REFERÊNCIAS

BALAS, E.; ZEMEL, E. An algorithm for large zero-one knapsack problems. *Operations Research*, v. 28, n. 5, p. 1130-1154, 1980.

BELLMAN, R. *Dynamic Programming*. Princeton: Princeton University Press, 1957.

MORIMOTO, Naoyuki. Power allocation optimization as the multiple knapsack problem with assignment restrictions. In: 2017 8th International Conference on the Network of the Future (NOF). IEEE, 2017. p. 40-45.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to Algorithms*. 3. ed. Cambridge: MIT Press, 2009.

DANTZIG, G. B. Discrete-variable extremum problems. *Operations Research*, v. 5, n. 2, p. 266-277, 1957.

GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman, 1990.

HOROWITZ, E.; SAHNI, S. Computing partitions with applications to the knapsack problem. *Journal of the ACM*, v. 21, n. 2, p. 277-292, 1974.

KELLERER, Hans; PFERSCHY, Ulrich; PISINGER, David. Multidimensional knapsack problems. In: *Knapsack problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 235-283.

KLEINBERG, J.; TARDOS, E. *Algorithm Design*. Boston: Addison-Wesley, 2005.



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UNIFOA

**23 a 25
de outubro**

Submissões abertas até 07/09

MARTELLO, S.; TOTH, P. Knapsack Problems: Algorithms and Computer Implementations. Chichester: John Wiley & Sons, 1990.

NEMHAUSER, G. L.; ULLMANN, Z. Discrete dynamic programming and capital allocation. Management Science, v. 15, n. 9, p. 494-505, 1969.

PISINGER, D. A minimal algorithm for the 0-1 knapsack problem. Operations Research, v. 45, n. 5, p. 758-767, 1997.

SEDGEWICK, R.; WAYNE, K. Algorithms. 4. ed. Boston: Addison-Wesley, 2011.

TOTH, P.; VIGO, D. Vehicle Routing: Problems, Methods, and Applications. 2. ed. Philadelphia: SIAM, 2014.

VAZIRANI, V. V. Approximation Algorithms. Berlin: Springer-Verlag, 2003.