Análise e Revisão do Problema da Mochila 0-1

Alberto Magno Machado[1], **Josué Pereira Nogueira**[1], **Leonardo Henrique Saraiva de Avelar**[1]

[1]Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Projeto e Análise de Algoritmos – Prof. Walisson Ferreira de Carvalho

Abstract. *The 0-1 knapsack problem is one of the most classic and studied problems in combinatorial optimization. This paper aims to provide an introductory and review analysis of this problem, addressing its mathematical formulation, computational complexity, variants, and both exact and approximate solution techniques. The discussion is grounded in classical works and recent scientific literature.*

Resumo. *Este artigo apresenta uma análise introdutória e revisional do problema da mochila 0-1, um problema clássico da otimização combinatória e exemplo canônico de problema NP-completo. São abordados sua formulação matemática, a complexidade computacional, as variantes do problema e os principais métodos de solução (exatos e aproximados), com base na literatura especializada.*

1. Introdução

O problema da mochila 0-1 é um clássico problema de otimização combinatória em que se deve selecionar, a partir de um conjunto finito de itens, aqueles que serão carregados em uma "mochila" com capacidade limitada, de modo a maximizar o valor total transportado. Cada item i possui um peso w_i e um valor v_i, e não é permitido fracionar ou repetir itens – cada item é escolhido ou não (daí o "0-1").

Formalmente, o objetivo é maximizar o valor total $\sum_{i=1}^{n} v_i x_i$ sujeito à restrição de que o peso total $\sum_{i=1}^{n} w_i x_i$ não exceda a capacidade da mochila W, com $x_i \in \{0, 1\}$ indicando a inclusão do item.

Esse nome vem da metáfora de um viajante que deve otimizar a carga de uma mochila limitada, buscando levar o máximo de valor possível. O problema apresenta grande relevância teórica e prática, sendo estudado há mais de um século e aparecendo em diversas aplicações reais, como corte de materiais, seleção de portfólios e sistemas de criptografia.

A forma de decisão do problema ("existe uma seleção de itens com valor $\geq V$?") é NP-completa. O objetivo deste trabalho é apresentar uma análise introdutória e revisional do problema da mochila 0-1, incluindo sua formulação, variantes, complexidade e métodos de resolução.

A Seção 2 apresenta o referencial teórico com foco na formulação matemática, complexidade e principais abordagens de resolução.

2. Revisão Bibliográfica

O problema da mochila 0-1 pode ser formalizado como um problema de programação inteira binária. Dado um conjunto de n itens numerados de 1 a n, cada um com peso w_i e valor v_i, e dado um limite de capacidade W, busca-se uma atribuição binária $x_i \in \{0, 1\}$ (incluir ou não o item i) que maximize o valor total:

$$\begin{aligned}
\text{Objetivo:} \quad & \max \sum_{i=1}^{n} v_i x_i \\
\text{Restrição:} \quad & \sum_{i=1}^{n} w_i x_i \leq W \\
\text{Domínio:} \quad & x_i \in \{0, 1\}, \quad i = 1, \ldots, n
\end{aligned}$$

Essa formulação reflete exatamente o desafio de maximizar o valor dos itens carregados sem exceder a capacidade da mochila. Em termos práticos, escolhe-se um subconjunto de itens cujo peso total não ultrapasse W e cujo valor total seja o maior possível.

Quanto à complexidade computacional, o problema é intrinsecamente difícil. A versão de decisão ("existe subconjunto com valor pelo menos V sem exceder W?") é *NP-completa*, e a versão de otimização é *NP-difícil*. Mais especificamente, o problema da mochila 0-1 é *fraco NP-difícil* (weakly NP-hard), pois admite algoritmos pseudo-polinomiais, como o de programação dinâmica com tempo $O(nW)$ [Pisinger 2005].

Esse tempo de execução é considerado pseudo-polinomial porque depende linearmente de n e de W, mas exponencialmente do tamanho da entrada binária (já que W pode ter $\log W$ bits). Por exemplo, dobrar o valor de W dobra o tempo do algoritmo, mas o tamanho da entrada cresce apenas linearmente em $\log W$. Assim, $O(nW) = O(n2^{\log W})$ é exponencial no tamanho da entrada, o que mantém a consistência com a complexidade *NP-difícil*.

Apesar disso, diversas abordagens exatas foram desenvolvidas. A mais clássica é a própria programação dinâmica. Outras envolvem técnicas como *branch-and-bound*, onde limites obtidos por relaxação linear (como no problema da mochila fracionária) são usados para podar ramos de busca [Martello and Toth 1990]. Heurísticas que partem de soluções fracionárias e as ajustam para soluções inteiras também têm sido efetivas [Pisinger 2005].

Outro algoritmo exato relevante é o *meet-in-the-middle*, proposto por Horowitz e Sahni [Horowitz and Sahni 1974], que, embora tenha complexidade exponencial em n, pode ser mais eficiente que a programação dinâmica quando n é grande e W é moderado.

Como alternativa à ausência de algoritmos exatos de tempo polinomial, foram propostos esquemas de aproximação. Em particular, o problema admite esquemas *FPTAS* (Fully Polynomial-Time Approximation Schemes), que garantem aproximações com fator $1 + \varepsilon$ em tempo polinomial em n e $1/\varepsilon$ [Ibarra and Kim 1975, Jin 2019].

Esses algoritmos baseiam-se na discretização dos valores ou pesos, e são capazes de produzir soluções arbitrariamente próximas do ótimo em tempo viável. Além disso, heurísticas metaheurísticas como algoritmos genéticos, colônia de formigas e enxame de partículas também são aplicadas em versões de larga escala, embora sem garantias formais de aproximação.

O problema possui ainda diversas variantes que ampliam sua complexidade e aplicabilidade:

- **Mochila limitada** (bounded knapsack): permite múltiplas cópias limitadas de cada item ($x_i \in \{0, 1, \dots, c_i\}$);
- **Mochila ilimitada** (unbounded knapsack): número ilimitado de cópias de cada item ($x_i \in \mathbb{N}$);
- **Mochila multidimensional**: considera múltiplas restrições (como peso e volume);
- **Mochila múltipla ou de escolha múltipla**: os itens são agrupados em classes, sendo permitido escolher um item por classe;
- **Mochila quadrática e outras extensões**: envolvem funções objetivo não lineares;
- **Subset Sum**: caso especial onde $v_i = w_i$ para todos os itens, sendo exatamente *NP-completo*, como listado por Karp [Karp 1972].

Essas variantes mostram a flexibilidade do problema da mochila como modelo para cenários reais e seu papel central na teoria da complexidade e otimização combinatória.

Referências

Horowitz, E. and Sahni, S. (1974). Computing partitions with applications to the knapsack problem. *Journal of the ACM*, 21(2):277–292.

Ibarra, O. H. and Kim, C. E. (1975). Fast approximation algorithms for the knapsack and subset-sum problems. *Journal of the ACM*, 22(4):463–468.

Jin, C. (2019). An improved fptas for 0-1 knapsack. Disponível em: `https://arxiv.org/abs/1904.09562`.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York.

Martello, S. and Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience, Chichester.

Pisinger, D. (2005). Where are the hard knapsack problems? *Computers & Operations Research*, 32(10):2271–2284.