

O algoritmo Nemhauser-Ullmann para o problema de mochila multidimensional

Marcos Daniel Baroni

Vitória, 15 de janeiro de 2015

Sumário

- Introdução
- O problema da Mochila (Multidimensional)
- O Algoritmo de Nemhauser-Ullmann para o MKP
 - ▶ O Conceito de Soluções Dominantes
 - ▶ O Algoritmo para o KP
 - ▶ Adaptação do Conceito para o MKP
- Testes Computacionais

O Problema da Mochila

Knapsack Problem

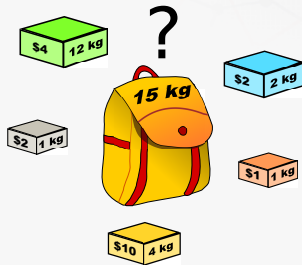
Utilizar a mochila para carregar o maior valor possível, respeitando a sua capacidade.

Modelagem matemática

$$\text{maximizar } z = \sum_{j=1}^n p_j x_j$$

$$\text{sujeito a } \sum_{j=1}^n w_j x_j \leq b$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n.$$



O Problema da Mochila Multidimensional

A *mochila* agora possui mais de uma capacidade (dimensão).

Modelagem matemática

$$\text{maximizar } z = \sum_{j=1}^n p_j x_j$$

$$\text{sujeito a } \sum_{j=1}^n w_{ij} x_j \leq b_i, \quad i = 1, \dots, m,$$

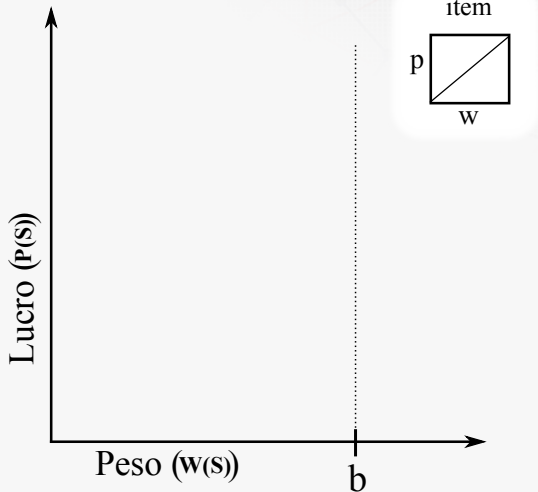
$$x_j \in \{0, 1\}, \quad j = 1, \dots, n.$$

Soluções dominantes para o KP

Solução dominante

Um subconjunto $S \subseteq [n]$ com peso $w(S) = \sum_{i \in S} w_i$ e lucro $p(S) = \sum_{i \in S} p_i$ **domina** sobre um outro subconjunto $T \subseteq [n]$ se $w(S) \leq w(T)$ e $p(S) \geq p(T)$.

Soluções dominantes para o KP



O Algoritmo de Nemhauser-Ullmann

Algorithm 1: O Algoritmo de Nemhauser-Ullmann para o KP

Input : Instancia do KP

Output: Lista $S(n)$ de todos as soluções dominantes

$S(0) \leftarrow \emptyset;$

para $i \leftarrow 1$ **to** n **faça:**

$S'(i) \leftarrow S(i-1) \cup \{s \cup \{i\} \mid s \in S(i-1)\};$

$S(i) \leftarrow \{s \in S'(i) \mid \text{dominates}(s, S'(i))\};$

fim

O Algoritmo de Nemhauser-Ullmann

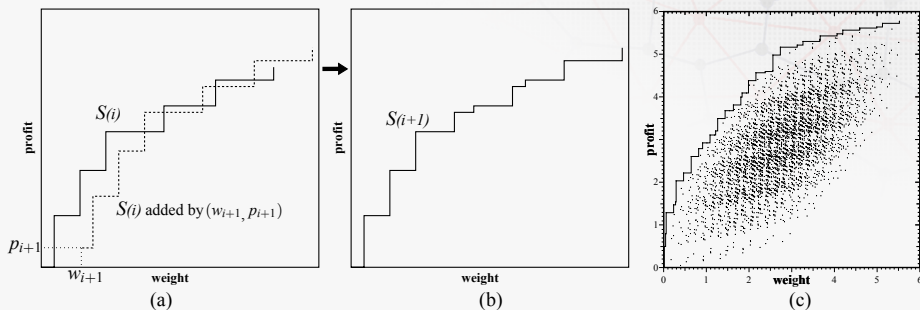


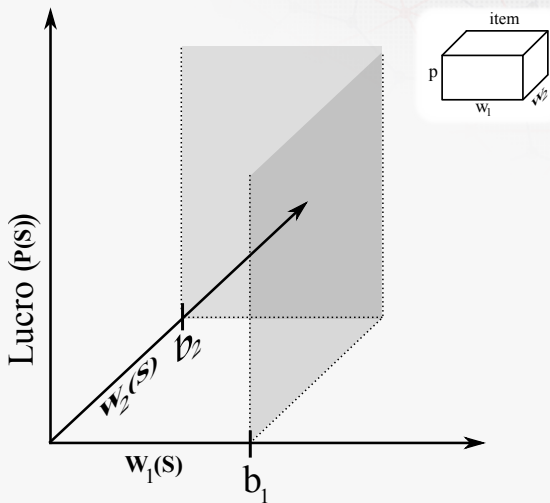
Figura: Representação gráfica das soluções dominantes (a) num passo intermediário $S(i)$, (b) em uma próxima solução $S(i+1)$ e (c) numa solução ótima (final).

O Algoritmo de Nemhauser-Ullmann para o MKP

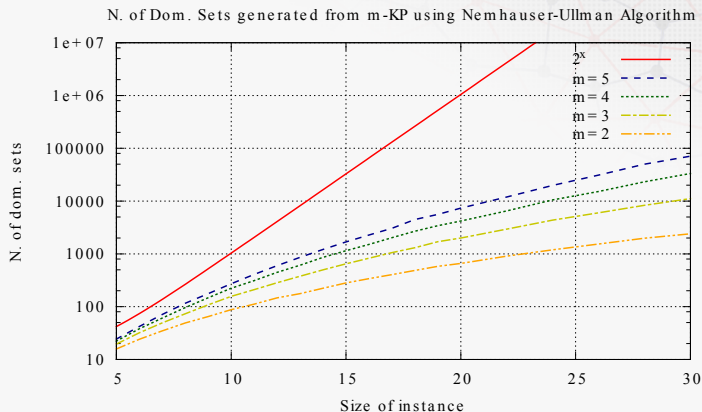
Solução dominante (para o MKP)

Um subconjunto $S \subseteq [n]$ com pesos $w_j(S) = \sum_{i \in S} w_{ji}, j = 1, \dots, m$ e lucro $p(S) = \sum_{i \in S} p_i$ **domina** sobre um outro subconjunto $T \subseteq [n]$ se $w_j(S) \leq w_j(T), \forall j = 1, \dots, m$ e $p(S) \geq p(T)$.

Soluções dominantes para o MKP

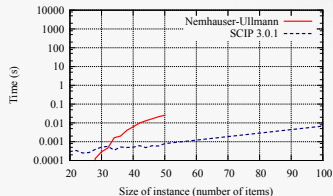


Testes computacionais

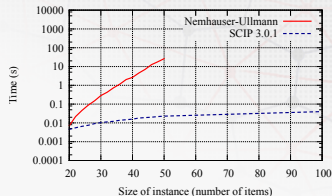


The "y" axis is on logarithmic scale. The number of generated *dominating sets* increases with the number of dimension but seems to maintain polynomial behavior.

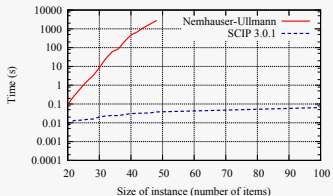
Testes computacionais



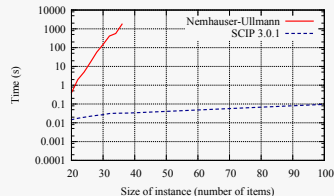
(a) $m = 1$



(b) $m = 2$



(c) $m = 3$



(d) $m = 4$

Figura: Tempo médio de execução para resolver instancias do MKP utilizando o algoritmo de Nemhauser-Ullmann e o SCIP solver para várias dimensões.

Referências Bibliográficas



R. Beier and B. Vöcking.

Random knapsack in expected polynomial time.

In Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, pages 232–241. ACM, 2003.



R. Beier and B. Vöcking.

An experimental study of random knapsack problems.

Algorithmica, 45(1):121–136, 2006.



G. L. Nemhauser and Z. Ullmann.

Discrete dynamic programming and capital allocation.

Management Science, 15(9):494–505, 1969.



H. M. Weingartner and D. N. Ness.

Methods for the solution of the multidimensional 0/1 knapsack problem.

Operations Research, 15(1):83–103, 1967.