

HEURÍSTICA DE REDUÇÃO DE PROBLEMAS DA MOCHILA ILIMITADA

*Autor 1 – Eduardo Henrique de Santana
Universidade Estadual de Londrina
eduardo.desantana0@uel.br*

*Co Autor 1 – Professor Dr. Robinson Samuel Vieira Hoto
Universidade Estadual de Londrina
hoto@uel.br*

*Co Autor 2 – Professor Dr. Fábio Sakuray
Universidade Estadual de Londrina
sakuray@uel.br*

Resumo:

Neste trabalho, discutiremos um método heurístico de resolução de Problemas da Mochila Ilimitada, uma importante classe de problemas de otimização. Inicialmente, definiremos formalmente o Problema da Mochila Ilimitada, além de alguns algoritmos tradicionais de resolução exata. Em seguida, introduziremos uma abordagem heurística, detalhando sua motivação e funcionamento. Finalmente, verificaremos a eficácia da heurística por meio de simulações computacionais, analisando seu desempenho, em termos de tempo de execução e proximidade ao valor ótimo, em comparação com soluções obtidas por resolução completa.

Palavras-chave: Problema da Mochila Ilimitada; Métodos Heurísticos; Simulações Computacionais.

1. Introdução

Através de diversas áreas das ciências e indústria, o processo de otimização de variados sistemas se caracteriza com excedente importância. Em específico, Problemas da Mochila configuram uma classe de problemas de otimização comumente utilizados para modelar diversas situações, tais como em criptografia, alocação de recursos, indústria de corte, entre outros. Por exemplo, conforme Gilmore e Gomory (1961), Problemas da Mochila aparecem como subproblemas na resolução de *Problemas de Corte de Estoque Unidimensionais*.

Deste modo, o desenvolvimento de técnicas heurísticas capazes de aproximar soluções ótimas dos problemas originais, reduzindo o tempo computacional necessário

para resolução, se tornam altamente relevantes. Neste trabalho, introduzimos e avaliamos uma heurística que atua reduzindo o tamanho do problema original, analisando sua eficácia por meio de experimentos computacionais.

2. Problema da Mochila Ilimitada

Um *Problema da Mochila Ilimitada* é um problema de otimização linear, de natureza de alocação de recursos. Dado um conjunto finito de variáveis independentes que assume valores inteiros não-negativos, com constantes de peso e utilidade associadas e um valor capacidade, o problema busca valores das variáveis que maximizam ou minimizam uma combinação linear de utilidades de modo que a combinação de pesos não supere a capacidade estabelecida.

Reescrevamos o problema em terminologia matemática. Considere os vetores $\mathbf{u}, \mathbf{p} \in \mathbb{N}^n$ e o escalar $C \in \mathbb{N}$. Sejam $\mathbf{x} \in \mathbb{N}^n$ um vetor variável e $z : \mathbb{N}^n \rightarrow \mathbb{R}$ uma função dada por $z(\mathbf{x}) = \mathbf{u} \cdot \mathbf{x}$, onde o operador $\cdot$ descreve o produto interno canônico de $\mathbb{R}^n$. Sem perda de generalidade, um *Problema da Mochila Ilimitada* busca maximizar a função z , de modo que a variável $\mathbf{x}$ satisfaça a expressão $\mathbf{p} \cdot \mathbf{x} \leq C$, na qual neste caso é chamada de *solução viável*. Denotemos por $\hat{z}$ e $\hat{\mathbf{x}}$ o valor ótimo e o vetor *solução ótima* do problema, respectivamente. Eis abaixo uma notação para o problema.

$$(PMI) \max z = \mathbf{u} \cdot \mathbf{x} : \mathbf{p} \cdot \mathbf{x} \leq C, \mathbf{x} \in \mathbb{N}^n$$

Nas próximas seções, serão introduzidos algoritmos clássicos da literatura para resolução de Problemas da Mochila Ilimitada.

3. Algoritmo de Chvátal

Nesta seção, introduziremos o algoritmo de Chvátal (1983) para resolução de Problemas da Mochila Ilimitada. Tal algoritmo se utiliza de um valor majorante para otimizar uma busca por enumeração de soluções viáveis.

Para a resolução de um Problema da Mochila, uma técnica natural é a da enumeração de todas as soluções viáveis possível, da qual se é determinado a solução ótima. Uma maneira comum de compreender tal processo é através de um enumeração por diagrama de árvores. Primeiramente, geramos uma solução inicial pela relação recursiva:

$$x_k = \left\lfloor \frac{C - \sum_{i=1}^{k-1} u_i x_i}{p_k} \right\rfloor$$

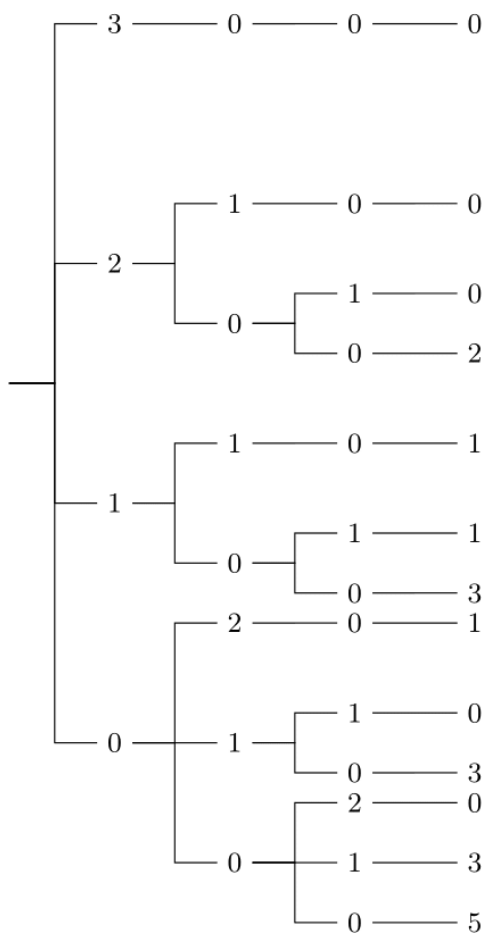
Em seguida, dada uma solução viável, outras podem ser obtidas alterando os valores das variáveis. Para isso, tome x_k como um valor pivô, diminui por um seu valor, e recalcule valores para $x_i, i > k$. Repita esse processo até $x_k = 0$, de onde tornaremos x_{k-1} nova variável pivô. Para iniciar o processo, considere x_{n-1} como pivô.

Como exemplo do processo, considere o seguinte Problema da Mochila.

$$(PMI) \max z = (4, 5, 5, 2) \cdot \mathbf{x} : (33, 49, 51, 22) \cdot \mathbf{x} \leq 120, \mathbf{x} \in \mathbb{N}^n$$

Utilizando a metodologia citada acima, seu diagrama correspondente é:

Figura 1 – Enumeração das soluções viáveis do Problema da Mochila Ilimitada



Fonte: O próprio autor

A partir de uma análise mais aprofundada, observa-se que a quantidade de soluções viáveis cresce significativamente à medida que variamos os valores de $\mathbf{u}$, $\mathbf{p}$ e n . Para evitar o gasto excessivo de tempo com o cálculo de soluções viáveis que não possuem chance de serem ótimas, Chvátal (1983) introduziu um valor majorante para restringir essa enumeração.

Sem perda de generalidade, assuma que as componentes de $\mathbf{x}$ estejam tais que as razões $\varepsilon_i = \frac{u_i}{p_i}$, denominadas *eficiência*, estejam ordenadas de maneira não-crescente. Sejam $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ a melhor solução viável obtida até o momento e $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n)$ a solução viável atual. Considere valor ótimo $\hat{z}$ associado à $\hat{\mathbf{x}}$. Buscamos uma maneira de verificar se a próxima solução gerada $\mathbf{x}$ tem capacidade de superar $\hat{\mathbf{x}}$, isto é, $z \geq \hat{z}$. Note que $\mathbf{x}$ é da forma:

$$\mathbf{x} = (\bar{x}_1, \dots, \bar{x}_{k-1}, \bar{x}_k - 1, x_{k+1}, \dots, x_n)$$

Para isso, buscamos majorante M tal que $z \leq M$. Caso seja verificado $M < \hat{z}$, segue necessariamente que $z < \hat{z}$, isto é, $\mathbf{x}$ não possui capacidade de superar a solução ótima atual $\hat{\mathbf{x}}$. Utilizando a ordenação não-crescente das eficiências, define-se o majorante:

$$M = \sum_{i=1}^{k-1} u_i \bar{x}_i + u_k (\bar{x}_k - 1) + \frac{u_{k+1}}{p_{k+1}} \left(W - \sum_{i=1}^{k-1} p_i \bar{x}_i - p_k (\bar{x}_k - 1) \right)$$

4. Algoritmo de Martello-Toth

Nesta seção, introduziremos parte do algoritmo de Martello e Toth (1990) para resolução de Problemas da Mochila Ilimitada, denominado pelo autor por MTU1. Semelhantemente ao algoritmo da Chvátal, ele se utiliza de um majorante para reduzir o número de soluções viáveis calculadas, porém com uma técnica de enumeração distinta. Pelas similaridades entre os dois, focaremos na determinação do valor majorante.

Primeiramente, suponha uma ordenação não-decrescente de utilidades e considere uma *relaxação contínua* do Problema da Mochila, isto é, suponha que $\mathbf{x}$ possa assumir valores reais não negativos. Neste caso, a solução ótima da relaxação é da forma $\hat{x}_1 = \frac{C}{p_1}$ e $\hat{x}_i = 0$, para todo $i > 1$. Tendo isto em mente, podemos determinar uma família de cotas superiores para os valores ótimos da mochila.

Denotemos por $a = \text{Mod}(b, m)$ o valor $0 \leq a < m$ tal que a equivalência

modular $a \equiv b \pmod{m}$ é válida. Para a execução do algoritmo MTU1, Martello e Toth (1990) utilizam-se do majorante $U_3 = \max\{U^0, \bar{U}^1\}$, conforme as especificações abaixo.

$$U_0 = \left\lfloor \frac{C}{p_1} \right\rfloor u_1 + \left\lfloor \frac{\bar{C}}{p_2} \right\rfloor u_2 + \left\lfloor C' \frac{u_3}{p_3} \right\rfloor$$

$$\bar{U}_1 = \left\lfloor \frac{C}{p_1} \right\rfloor u_1 + \left\lfloor \frac{\bar{C}}{p_2} \right\rfloor u_2 + \left\lfloor \left(C' + \left\lfloor \frac{p_2 - C'}{p_1} \right\rfloor p_1 \right) \frac{u_2}{p_2} - \left\lfloor \frac{p_2 - C'}{p_1} \right\rfloor u_1 \right\rfloor$$

Onde $\bar{C} = \text{Mod}(C, p_1)$ e $C' = \text{Mod}(\bar{C}, p_2)$. Tanto o algoritmo MTU1 quanto seu sucessor, o MTU2, ainda são considerados algoritmos de referência, comumente utilizados em artigos que comparam algoritmos resolutores de Problemas da Mochila Ilimitada. Esse fato se deve à inovação introduzida no uso de regras de dominância, estratégias eficientes de poda, e técnicas de *branch-and-bound*. Com essas introduções, possui desempenho superior às abordagens anteriores baseadas unicamente em programação dinâmica.

5. Heurística de Redução de Mochila

Durante a seção do algoritmo de Martello-Toth, foi-se discutido sobre a solução ótima de uma relaxação contínua de um Problema da Mochila Ilimitada. Com isso, obtemos uma intuição referente a composição ótima de um Problema da Mochila: a menos de correções para preservar a condição necessária do problema, a solução ótima é preferencialmente composta por variáveis que maximizam a eficiência, ou adicionalmente, que maximizam utilidade ou minimizam peso. Baseado nisso, desenvolveremos o método heurístico.

A heurística utilizada neste trabalho é baseada na redução do número de variáveis envolvidas na resolução do problema, levando em consideração o envolvimento de cada uma na solução ótima final. Seja $\eta \in \mathbb{N}$, $\eta < n$, um inteiro positivo. Primeiramente, realize um trio de ordenação, de natureza não-crescente, das variáveis x_i conforme peso, utilidade e eficiência. Em seguida:

- Selecione as η primeiras variáveis da ordenação da *eficiência* ε .
- Selecione as η primeiras variáveis da ordenação de *utilidade* u . Caso haja repetição com a seleção acima, continue o processo até selecionar η itens ou percorrer todas as variáveis.
- Selecione as η últimas variáveis da ordenação de *peso* p . Caso haja repetição com

as seleções acima, continue o processo até selecionar η itens ou percorrer todas as variáveis.

Finalmente, resolveremos uma redução do Problema da Mochila proposto, considerando somente as variáveis selecionadas nesse processo. A depender do valor η escolhido, obteremos uma solução aproximada da solução ótima completa com graus variados de precisão. Porém, de maneira geral, o tempo de computação necessário para resolver o problema será drasticamente reduzido.

6. Resultados Parciais

Para avaliar a eficiência da heurística, foram realizados experimentos computacionais que consistem em testes comparativos entre mochilas completas e suas reduções pela heurística, utilizando diferentes valores para o parâmetro η . Os testes foram implementados em linguagem C, visando desempenho e controle sobre tempo de execução e uso de memória.

A metodologia de teste consiste na criação de categorias de mochilas, cada uma composta por 1000 instâncias, definidas com base no tamanho das mochilas e escala de pesos e utilidades. O processo de geração das instâncias é baseado na formulação de mochilas fracamente correlacionadas, descritas no texto base de Martello e Toth (1990).

Para cada instância de uma categoria, os pesos e as utilidades são escolhidas de forma uniformemente aleatória, enquanto o tamanho da mochila e a capacidade máxima são determinados a partir dos valores acima. Segue abaixo um descritivo do procedimento de geração.

Quadro 1 – Configurações da geração de mochilas

Tamanho da Mochila	Peso	Utilidade	Capacidade
$n = p_{max} - p_{min} + 1$	$p_i \in [p_{min}, p_{max}]$	$u_i \in [p_i - r, p_i + r]$	$C = 0,5 \cdot \sum_{i=1}^n p_i$

Fonte: O próprio autor

Onde p_{min} e p_{max} representam os valores mínimo e máximo que o peso pode assumir, e $r > 0$ é uma constante de variância das utilidades em função do peso. Durante nossa implementação, adotamos $r = 10$.

Posteriormente, cada instância é resolvida utilizando o algoritmo MTU1, tanto de forma completa quanto com o auxílio da heurística, considerando os valores de $\eta = 3, 4, 5$. Para cada categoria e abordagem, são calculadas a média aritmética dos tempos de execução e das soluções ótimas. O tempo de execução da abordagem completa inclui

a ordenação por eficiência e a execução do algoritmo. Na heurística, adiciona-se o tempo de redução da instância. Valores *outliers* de tempo foram removidos de acordo com a amplitude interquartil das resoluções completas de cada categoria.

Para simplificar a apresentação dos resultados, denotaremos por $\mathcal{C}$ e $\mathcal{H}_\eta$ a resolução completa e a redução heurística com valor η . Além disso, será incluído o *erro percentual* da aproximação de cada heurística, em substituição aos valores médios das soluções obtidas. O erro será truncado até a quarta casa decimal.

Tabela 1 – Resultados comparativos da heurística proposta

Categoria		Tempo de Execução (μs)				Erro Percentual (%)		
p_{min}	p_{max}	$\mathcal{C}$	$\mathcal{H}_3$	$\mathcal{H}_4$	$\mathcal{H}_5$	$\mathcal{H}_3$	$\mathcal{H}_4$	$\mathcal{H}_5$
11	60	14	8	9	9	0,0334	0,0162	0,0141
11	110	37	15	16	16	0,0259	0,0176	0,0052
11	510	509	86	86	86	0,0317	0,0366	0,0316
101	150	93	58	73	65	0,0938	0,0497	0,0258
101	200	109	38	54	55	0,0447	0,0241	0,0117
101	600	714	91	94	98	0,0208	0,0116	0,0050
101	1100	2.298	191	199	204	0,0153	0,0034	0,0024
101	5100	41.092	1.193	1.204	1.212	0,0113	0,0047	0,0023
Média		5.608	210	217	218	0,0346	0,0205	0,0123

Fonte: O próprio autor

7. Conclusão

A partir da análise dos resultados parciais apresentados na seção anterior, a eficácia da heurística é evidenciada em uma ampla gama de categorias de mochilas. Apesar de $\eta = 5$ ser um parâmetro suficientemente para a grande maioria das mochilas avaliadas, a determinação de um valor mínimo de η para uma dada taxa de erro ainda é incerta.

De modo à ampliarmos a validade da heurística, pretende-se realizar testes com mochilas de maiores escalas de peso e utilidade. Contudo, devido a natureza *NP-Hard* dos Problemas da Mochila Ilimitada e aos elevados tempos de execução das mochilas completas, atualmente tal processo se mostra inviável. Por esse motivo, voltamos nosso interesse à abordagem de Yanasse e Soma (1987) para Problemas da Mochila Ilimitada, considerada mais eficiente em comparação à de Martello e Toth (1990).



Referências

CHVÁTAL, Vašek. **Linear Programming**. [S.l.]: W. H. Freeman, 1983.

GILMORE, P.C.; GOMORY, Ralph. A Linear Programming Approach to the Cutting Stock Problem I. **Operations Research**, v. 9, p. 771–925, 1961.

MARTELLO, Silvano; TOTH, Paolo. **Knapsack problems: algorithms and computer implementations**. USA: John Wiley & Sons, Inc., 1990.

YANASSE, Horacio Hideki; SOMA, Nei Yoshihiro. A new enumeration scheme for the knapsack problem. **Discrete Applied Mathematics**, v. 18, p. 235–245, 1987.