EMA- Encontro de Matemática Aplicada

O Desenvolvimento de Novas Heurísticas para o Problema da Mochila Compartimentada

Matheus Henrique Pimenta Zanon
Departamento de Matemática
Universidade Estadual de Londrina

1 Introdução

O desenvolvimento industrial e tecnológico está em crescente ampliação no cenário nacional [9]. Desta forma as indústrias tem buscado métodos mais eficazes na resolução de problemas, como por exemplo, otimizar recursos e obter a melhor solução, podendo ser em termos de tempo de execução mais rápidos ou em valores obtidos. Buscar alternativas de métodos que apresentem soluções em um espaço de tempo cada vez menor é um dos objetivos dos estudos da otimização matemática e pesquisa operacional.

O Problema da Mochila é um tipo clássico de problemas de otimização que exige uma tomada de decisão. O Problema da Mochila Clássico consiste na escolha de um subconjunto de itens aos quais estão associados uma utilidade e largura (ou peso) que definirá a quantidade de espaço da mochila este utilizará. O objetivo de tal escolha é proporcionar a melhor utilidade ao final do preenchimento da mochila [11]. O Problema da Mochila é uma classe de Problema de Programação Linear inteira e sua classificação é baseada na complexidade de resolução como problemas *NP-Dificéis* [8]. Este é um dos motivos que motivam o desenvolvimento de heurísticas para a resolução dos problemas em busca de soluções "boas" em sacrifício da otimalidade.

Uma variação do Problema da Mochila clássico é o Problema da Mochila Compartimentada, que consiste na criação de compartimentos de capacidades limitadas por um intervalo, dentro da mochila de capacidade conhecida.[5].

Com esta motivação, o objetivo deste trabalho é propor três novas heurísticas para a resolução do Problema da Mochila Compartimentada, utilizando uma adaptação do método exato de resolução proposto por Martello e Toth [11], e a particularidade do modelo linear do Problema da Mochila Compartimentada que é proposto por Inarejos [6]. Tais heurísticas buscam ser ágeis, utilizando poucos recursos computacionais e com resultados próximos ao ótimo para o Problema da Mochila Compartimentada.

2 O Problema da Mochila Compartimentada

Considere uma mochila de capacidade L e um conjunto de índices N, particionado em subconjuntos N_k, onde $k = 1, 2, \ldots, q$, $N = \bigcup_{k=1}^{q} N_k$ e $N_j \cap N_k = \varnothing$, para $k \neq j$ e $k, j \in (1, 2, \ldots, q)$. Para cada índice em N está associado uma largura (ou peso) l_i, uma utilidade u_i e uma demanda d_i. O interior da mochila é organizado de modo que apenas itens da mesma classe podem ser inseridos dentro do mesmo compartimento.

Cada compartimento possui um tamanho variável, porém limitados inferiormente por (L^k_{min}) e superiormente por (L^k_{max}). Cada compartimento possui um custo fixo c_k e uma perda fixa S_k, por serem incluídos, que variam de acordo com cada classe. Em cada classe pode ser construído mais de um compartimento para a alocação dos itens. Desta maneira o *Problema da Mochila Compartimentada* visa definir quais compartimentos irão compor a mochila de maneira que subtraindo os custos dos compartimentos a utilidade seja máxima.

A abordagem linear do Problema da Mochila Compartimentada é trabalhada por [4] na elaboração de novas heurísticas para a solução do problema, após isto, [1] elabora heurísticas para o modelo linear, fortalecendo a conjectura da linearidade como verdade. Por fim, Inarejos [6] propõe um modelo linear para o Problema da Mochila Compartimentada Restrito juntamente com a prova da linearidade do proposto.

Para o tratamento do modelo linear são feitas algumas observações: a primeira é sobre a quantidade de compartimentos, denotados como p_k, a serem criados para cada classe N_k, que é definida como $p_k = min\left\{F_1, \left\lfloor \frac{L}{L^k_{min}} \right\rfloor\right\}$. Eliminando-se a não-linearidade do modelo proposto por Hoto [5]. Embora sejam construídos todos os p_k compartimentos, alguns não serão utilizados, isto é, não terão itens em seu interior, sendo considerados compartimentos nulos. Outra mudança é na questão da indexação dos itens, anteriormente era considerado a indexação global dos itens, nesta nova abordagem os itens são indexados localmente, isto é, os compartimentos são indexados para cada classe, enquanto a indexação das demandas, por exemplo, continua a nível global. O quesito indexação é teoricamente irrelevante, pois a alteração é somente nos índices, alterando os valores de N_k, os itens ainda continuam conectados às suas respectivas classes.

A variável é dada por a^k_{ij}, os índices significam a quantidade de itens i da classe k no compartimento j, onde $i \in N_k$ e j também refere-se a classe k. A variável binária δ também tem o índice modificado, representada como

δ_j^k, recebendo o valor 1 caso o compartimento j da classe k for não nulo e recebendo o valor 0 caso contrário.

Em cada classe $k = 1, \ldots, q$ a quantidade de compartimentos a serem incluídos na mochila não pode ser superior a $\left\lfloor \frac{L}{L_{min}^k} \right\rfloor$. Considerando-se as facas, o número máximo de compartimentos a serem construídos para cada classe será $p_k = min\{F_1, \left\lfloor \frac{L}{L_{min}^k} \right\rfloor\}$. Para a criação não são consideradas a quantidade de itens que devem ser inseridos em cada compartimento, logo, a variável de interesse será o número de itens a serem considerados em cada compartimento para compor a mochila. Disso, em teoria são criados $\sum_{k=1}^{q} p_k$ compartimentos ao todo, alguns compartimentos serão nulos, isto é, não terão itens em seu interior, e outros compartimentos possuirão o mesmo número de itens em seu interior, denominados compartimentos implícitos.

O modelo linear para o Problema da Mochila Compartimentada Restrita proposto por Inarejos [6] é apresentado pelas equações (1) - (7):

$$\text{Maximizar: } z = \sum_{k=1}^{q} \sum_{i \in N_k} \sum_{j=1}^{p_k} u_i a_{ij}^k \tag{1}$$

$$\text{sujeito a: } \sum_{k=1}^{q} \sum_{i \in N_k} \sum_{j=1}^{p_k} l_i a_{ij}^k \leq L \tag{2}$$

$$\delta_j^k L_{min}^k \leq \sum_{i \in N_k} l_i a_{ij}^k \leq \delta_j^k L_{max}^k \ , j = 1, 2, \ldots, p_k \ , \ k = 1, 2, \ldots, q \tag{3}$$

$$\sum_{j=1}^{p_k} a_{ij}^k \leq d_i \ , \ i \in N_k, \ k = 1, 2, \ldots, q \tag{4}$$

$$\sum_{k=1}^{q} \sum_{j=1}^{p_k} \delta_j^k \leq F_1 \tag{5}$$

$$\sum_{i \in N_k} a_{ij}^k \leq F_2 \ , \ j = 1, 2, \ldots, p_k \ , \ k = 1, 2, \ldots, q \tag{6}$$

$$\delta_j^k \in \{0, 1\} \text{ e } a_{ij}^k \geq 0 \text{ e inteiros}, \ i \in N_k, \ j = 1, 2, \ldots, p_k, \ k = 1, 2, \ldots, q \tag{7}$$

A função objetivo 1 busca maximizar a utilidade total dos itens, u_i só será somado se $i \in N_k$, a restrição 2 é a restrição física da mochila, l_i só será somado se $i \in N_k$, só será incluído o item i do compartimento j se são referentes a mesma classe N_k. A restrição 3 fornece os limitantes inferiores e superiores para os compartimentos de cada classe, só serão incluídos compartimentos não nulos, isto é $\delta_j^k = 1$, a restrição 4 limita a quantidade disponível de cada item i, as restrições 5 e 6 são respectivamente a faca 01 e faca 02 do processo de corte, que informam a quantidade de compartimentos não nulos que serão inseridos na mochila e a quantidade de itens que serão inseridos em cada compartimento, respectivamente, por fim a restrição 7 apresenta o domínio das variáveis.

A demonstração da linearidade do modelo (1) - (7) pode ser acessada em [6] e [7].

3 Heurísticas para o Problema da Mochila Compartimentada

As heurísticas são fundamentadas na inserção de compartimentos construtivos de maior eficiência no preenchimento da mochila.

As heurísticas propostas compartilham um mesmo estágio inicial, definido como processo inicial.

O Processo Inicial consiste na determinação dos compartimentos construtivos e na ordenação das classes, compartimentos e itens em ordem decrescente de eficiência.

O Algoritmo que apresenta a obtenção dos compartimentos construtivos também inclui um método para obtenção das utilidades associadas a cada compartimento construtivo.

Após a geração de todos os compartimentos viáveis, será formulado um problema de mochila restrito para a obtenção das melhores utilidades associadas a cada compartimento construído. A forma de obtenção destes valores ótimos das utilidades é obtida através da resolução do problema (8) - (11):

$$\text{Maximizar } U_j = \sum_{i \in N_k} u_i a_{ij} \tag{8}$$

$$\text{Sujeito a: } \sum_{i \in N_k} l_i a_{ij} \leq w_j \tag{9}$$

$$\sum_{i \in N_k} a_{ij} \leq F_2 \tag{10}$$

$$a_{ij} \geq 0 \text{ e inteiro}, i \in N_k \tag{11}$$

Após a obtenção das melhores utilidades associadas a cada um dos p_k compartimentos construtivos de cada classe k, U_j e W_j serão os novos valores para a resolução do problema de programação inteira (12) - (15), o que resultará na solução viável para o problema da mochila compartimentada. W_k são os índices associados a cada compartimento construtivo.

$$\text{Maximizar Z} = \sum_{i \in W_k} U_i y_{ij} \tag{12}$$

$$\text{Sujeito a: } \sum_{i \in W_k} W_i y_{ij} \leq L \tag{13}$$

$$\sum_{i \in W_k} y_{ij} \leq F_1 \tag{14}$$

$$y_{ij} \geq 0 \text{ e inteiro}, i \in W_k \tag{15}$$

3.1 A Heurística p_kX

A heurística denominada p_kX utiliza-se do *software* FICO® Xpress com seu pacote de otimização FICO® Xpress *Optimizer*.

O pacote de otimização é do tipo embarcado, não sendo possível o detalhamento dos procedimentos adotados pela função *maximize* do pacote de otimização.

3.2 A Heurística $p_kGULOSO$

Para a resolução dos problemas (8) - (11) e (12) - (15) a heurística denominada como $p_kGULOSO$ insere o maior número permitido de itens de maior eficiência no interior da mochila, caso ainda seja possível a inserção de outro item, é disposto o próximo item de maior eficiência, até o preenchimento da capacidade da mochila, isto é, apenas um ramo da árvore de enumeração é gerado.

3.3 A Heurística $p_kMTComp$

Esta heurística consiste na resolução dos problemas (8) - (11) e (12) - (15) utilizando o procedimento MTU2 que é proposto por Martello e Toth [11].

Para a resolução dos problemas (8) - (11) e (12) - (15) é necessário realizar uma modificação do algoritmo MTU2, apresentado por [11], para a inclusão das facas no modelo (restrições (10) e (14)).

Para a implementação do método MTU1 na linguagem de programação C, foi proposto a utilização do modelo de máquina de estados finitos, eliminando-se as chamadas *goto* do código original. O uso de *multithreading* também foi possível, onde cada classe na resolução do problema (8) - (11) é atribuida a uma *thread*, proporcionando uma melhor utilização do recurso computacional.

4 Simulações Numéricas

Os dados para realizar as simulações foram organizados em cinco classes com tamanhos definidos em $q = 5, 10, 20, 50$ e 100, e para cada categoria foram criadas mais cinco sub-categorias, que representam a quantidade de itens disponíveis para cada classe, sendo $n = 10, 50, 100, 1000$ e 10000. A Tabela 1 apresenta um resumo da organização das classes e itens.

		Categoria dos Exemplares				
	q	5	10	20	50	100
Sub-Categorias	q/n	5/10	10/10	20/10	50/10	100/10
		5/50	10/50	20/50	50/50	100/50
		5/100	10/100	20/100	50/100	100/100
		5/1000	10/1000	20/1000	50/1000	100/1000
		5/10000	10/10000	20/10000	50/10000	100/10000

Tabela 1: Categorias e sub-categorias definidas para os exemplares.

Para a criação dos exemplares foram utilizados os seguintes dados, baseados em problemas reais de corte de bobinas de aço em duas etapas [5]. Para a capacidade total da mochila, foi designado $L = 1100mm$, os valores para as facas 01 e 02 foram respectivamente $F1 = 7$ e $F2 = 12$, a capacidade dos compartimentos de cada classe estão relacionados entre os valores $L^k_{min} = 154mm$ e $L^k_{max} = 456mm$, as larguras dos itens serão geradas com valores distribuídas entre $53mm$ e $230mm$, já as utilidades dos itens serão valores estritamente positivos, tendo como limitante superior o valor de 100.

Para a geração dos exemplares, foi utilizado um gerador aleatório, baseado em [2], onde para cada categoria q/n os valores da largura e utilidade são gerados de forma aleatória. Dessa maneira, foram gerados 100 exemplares para cada categoria, resultando em 2500 exemplares.

A Figura 1 mostra um gráfico que permite a visualização do comportamento das heurísticas em relação ao $\overline{gap}$.

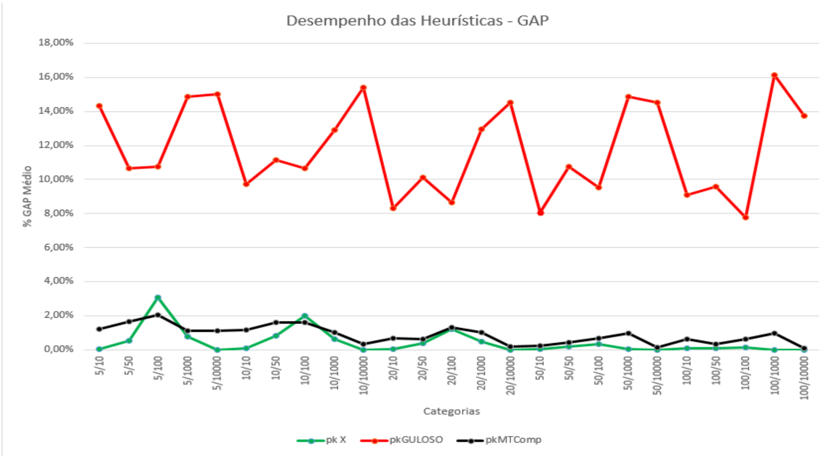

Figura 1: Desempenho das Heurísticas - $\overline{gap}$

A Figura 2 apresenta as comparações entre a heurística $p_k MTComp$ e a heurística das $w-capacidades$ proposta por Marques [10]. Para as categorias com $n = 1000$ e $n = 10000$ a heurística $w-capacidades$ não obteve soluções, isto ocorre devido sua elaboração com a utilização do método de Gilmore e Gomory [3] para a resolução dos subproblemas internos da heurística, que resolve problemas com até centenas de itens.

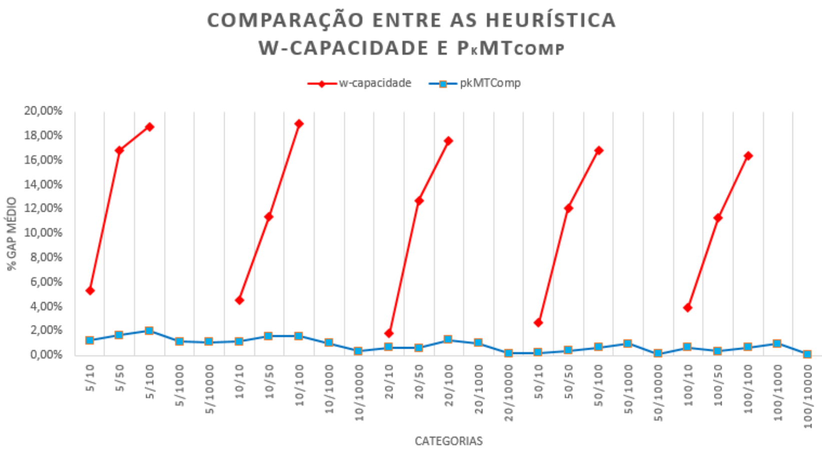

Figura 2: Comparação entre as Heurísticas $w - capacidades$ e $p_k MTComp$ - $\overline{gap}$

5 Conclusão

Como contribuições, pode-se destacar a heurística $p_k MTComp$, onde os experimentos preliminares indicam que o método é promissor. O indicativo em relação ao tempo mostra que quando comparada com as heurísticas p_kX e $p_kGULOSO$ o método com a utilização do algoritmo de Martello e Toth [11], provém resultados com o *gap* menor, resultando em soluções mais próximas ao ótimo.

Quando comparada com a heurística $w-capacidades$, os testes realizados indicam que a heurística $p_kMTComp$ produz resultados mais próximos ao ótimo, isto é *gap* inferiores. A heurística $p_kMTComp$ é capaz de solucionar categorias com um número superior de itens em cada classe.

A continuação deste trabalho envolve o desenvolvimento da heurística para o caso restrito da mochila, quando ocorre a limitação de itens disponíveis para o preenchimento da mochila.

Referências

[1] E. P. d. Cruz. Uma abordagem heurística linear para mochilas compartimentadas restritas. Dissertação de mestrado, UEL, Londrina, 2010.

[2] T. Gau and G. Wäsher. Cutgen1: A problem generator for the standar one-dimensional cutting stock problem. *European Jornal of Operational Research*, 84(3):572–579, 1995.

[3] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem - part ii. *Operations Research*, May 1963.

[4] R. Hoto, A. Fenato, H. Yannasse, N. Maculan, and F. Spolador. Uma nova abordagem para o problema da mochila compartimentada. In *Anel do XXXVIII Simpósio brasileiro de Pesquisa Operacional*, 2006.

[5] R. S. V. Hoto. *O Problema da Mochila Compartimentada Aplicado no Corte de Bobinas de Aço.* Tese de doutorado, UFRJ, Rio de Janeiro, 2001.

[6] Osvaldo Inarejos. Sobre a não-linearidade do problema da mochila compartimentada. Dissertação de mestrado, UEL, Londrina, 2015.

[7] Osvaldo Inarejos, Robinson Hoto, and Nelson Maculan. A integer linear optimization model to the compartimentalized knapsack problem. *International Transactions in Operational Research*, 00(1):1–20, out 2017.

[8] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems.* 01 2004.

[9] Larissa Daiana de Macêdo. Análise do comportamento inovativo da indústria nacional no período de 2003-2014 sob a influência das políticas industriais de inovação. Dissertação, Universidade Federal de Alagoas, Maceio, June 2018.

[10] F.P. Marques and M. N. Arenales. The constrained compartmentalised knapsack problem. *Computers & Operations Research*, 34(7):2109 – 2129, 2007.

[11] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.