Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 3, N. 1, 2015.
Trabalho apresentado no XXXV CNMAC, Natal-RN, 2014.

Heurísticas para o problema de corte de estoque unidimensional inteiro: novas contribuições

Gonçalo Renildo L. Cerqueira, Marlos Marques,
Depto de Ciências Exatas e Tecnológicas, DCET, UESB,
45083-900, Vitória da Conquista, BA
E-mail: goncalorenildo@gmail.com, marlos.uesb@gmail.com

Resumo: *Uma solução para o problema de corte de estoque unidimensional inteiro pode ser determinada por heurísticas construtivas ou residuais. A heurística construtiva após gerar um bom padrão de corte, utiliza-o o maior número de vezes possível, sem que haja excesso na demanda de algum item, já a heurística residual, aproxima a solução relaxada do problema para uma solução inteira e a seguir resolve um problema residual (com menor demanda). Neste trabalho duas heurísticas construtivas bem conhecidas na literatura e uma proposta de modificação numa delas serão apresentadas. Os resultados dos testes computacionais realizados e sua análise serão mostrados ao final deste trabalho.*

Palavras-chave: *Problema de Corte, Programação Inteira, Heurística*

1 Introdução

Problemas de corte consistem em cortar diferentes tipos de itens a partir de peças maiores em estoque (objetos) a fim de otimizar alguma função objetivo (minimização da perda ou dos custos de produção). São muitos os setores da indústria onde estes problemas aparecem, com destaque para empresas que lidam com papel, madeira, vidro, aço, plástico, tecido etc. A variedade no tamanho dos itens envolvidos e as suas demandas, permitem diferentes maneiras para cortar o objeto em estoque. Esta maneira de cortá-lo para produzir os diferentes itens é o que chamamos de padrão de corte. Uma solução para o problema consiste em determinar um conjunto de n-padrões e suas frequências, o número de vezes que cada um deve ser cortado a fim de que toda a demanda seja atendida. Seja m a quantidade de diferentes itens de comprimento l_i e demandas d_i que devem ser cortados de objetos em estoque de comprimento L (em quantidade suficiente para atender o pedido dos itens tal que $l_i \leq L$, $\forall i = 1, ..., m$) e custo igual a c_j, para $j \in J$, em que J é o conjunto de índices de todos os padrões viáveis. Supõe-se sem perda de generalidade que $c_j = 1$, $\forall j \in J$. Se a_{ij} é a variável de decisão que indica a quantidade de itens tipo i no padrão j e x_j representa a frequência do padrão de corte j o problema pode ser formulado como o problema de programação linear inteira seguinte:

$$minimize: \quad \sum_{j \in J} x_j \tag{1}$$

$$sujeito\ a: \quad \sum_{j \in J} a_{ij} x_j = d_i, \quad i = 1, ..., m \tag{2}$$

$$\sum_{i=1}^{m} a_{ij} l_i \leq L, \quad \forall j \in J \tag{3}$$

$$x_j; a_{ij} \geq 0 \text{ e inteiros}, \quad \forall j \in J \ \ i = 1, ..., m. \tag{4}$$

Seja $A = [\mathbf{a_1} \;\; \mathbf{a_2} \;\; \cdots \;\; \mathbf{a_n}]$ a matriz cujas colunas representam os n-possíveis padrões de corte na solução. Cada vetor coluna de A, $\mathbf{a_j} = (a_{1j}, ..., a_{ij}, ..., a_{mj})^T$, representa um padrão de corte viável, e o elemento a_{ij} indica a quantidade de itens de comprimento l_i obtidos do padrão de corte $\mathbf{a_j}$, para $i = 1, ..., m$ e $j = 1, ..., n$. Se $\mathbf{d} = (d_1, ..., d_i, ..., d_m)^T$ é o vetor de demanda dos itens e $\mathbf{x} = (x_1, ..., x_j, ..., x_n)^T$ é o vetor das frequências dos padrões o problema (1-4) toma a forma matricial seguinte:

$$minimize : \sum_{j \in J} x_j \tag{5}$$

$$sujeito\ a : \quad \mathbf{Ax} = \mathbf{d} \tag{6}$$

$$\mathbf{x} \geq \mathbf{0} \text{ e inteiro,} \tag{7}$$

O vetor $\mathbf{a_j} = (a_{1j}, ..., a_{ij}, ..., a_{mj})^T$ corresponde a um padrão de corte viável, o que significa que todos os itens de comprimento l_i a compor o padrão, devem ser obtidos a partir do corte do objeto L em estoque, portanto devem satisfazer as seguintes restrições:

$$\begin{array}{l} l_1 a_{1j} + l_2 a_{2j} + ... + l_m a_{mj} \leq L \\ 0 \leq a_{ij} \leq d_i \text{ e inteiros, } \ i = 1, ..., m \quad j = 1, ..., n. \end{array} \tag{8}$$

No modelo matemático (1-4), inicialmente proposto por Gilmore e Gomory [7], temos um problema não linear devido à restrição (2). Para evitar a não linearidade, pode-se enumerar todos os padrões de corte que satisfazem a restrição (8) e o problema passa a ser linear composto por (1),(2) e $x_j \geq 0$ e inteiro para $j = 1, ..., n$. As dificuldades em se trabalhar com esta formulação estão no fato do número de padrões a explicitar (colunas $\mathbf{a_j}$, $j = 1, ..., n$) aumentar muito à medida que aumenta o número de diferentes itens e suas demandas, e na dificuldade em resolver o problema devido às condições de integralidade impostas às variávies de decisão. Para solucionar o problema, Gilmore e Gomory [7],[8] propuseram relaxar as condições de integralidade das variáveis de decisão e aplicar o método Simplex com Geração de Colunas. A cada iteração do simplex, gera-se uma coluna do modelo.

Na maioria das vezes, a solução relaxada do problema de corte, obtida pelo método simplex com geração de colunas (ver [1]), não é inteira. Para determinar uma solução inteira são aplicadas heurísticas de arredondamento. Marcotte [10] conjecturou que o problema de corte unidimensional com objetivo de minimizar objetos tem a propriedade IRUP (*Integer Round-Up Property*), ou seja, o *gap* de integralidade, a diferença entre o valor ótimo da solução inteira e o valor ótimo da solução relaxada arredondada para o inteiro mais próximo superior é menor ou igual a 1. Posteriormente a mesma autora Marcotte [11] apresentou um exemplo em que a conjectura que ela propôs não se aplica. Scheithauer e Terno [13] conjecturaram então que o problema de corte unidimensional tem a propriedade MIRUP (*Modified Integer Round-Up Property*), em que o *gap* de integralidade é menor ou igual que 2.

2 Heurísticas Construtivas

Alguns métodos simples sugeridos na literatura para determinar um solução do problema de corte de estoque unidimensional inteiro consistem em, a cada iteração, construir um bom padrão de corte, e utilizá-lo exaustivamente, o número máximo possível de vezes, sem que a demanda dos itens seja excedida. Após atualizar a demanda, o processo é repetido até que toda ela tenha sido atendida. Ao final do processo uma solução inteira para o problema de corte é gerada. Estes métodos são referenciados na literatura como heurísticas contrutivas conforme Hinxman [9]. Duas destas heurísticas muito conhecidas que seguem esta idéia são a FFD (*First Fit Decreasing*) e Gulosa. Outra abordagem para determinar tais soluções caracterizam as heurísticas residuais, para mais detalhes recomendamos consultar ([2], [12], [14] e [15])

DOI: 10.5540/03.2015.003.01.0411

2.1 Heurística FFD

A Heurística FFD, consiste em colocar o maior item no padrão, o máximo de vezes possível, sem que haja excesso na demanda. Se o item selecionado não couber mais no padrão, o segundo maior item é selecionado e assim por diante. O algoritmo completo é apresentado a seguir:

INÍCIO
P.1. Ordene os itens em ordem decrescente de tamanho.
Suponha sem perda de generalidade que: $l_1 \geq l_2 \geq ... \geq l_m$
P.2. Seja dr_i a demanda residual do item $i \in I$.
$I = \{1, ..., m\}$. {conjunto de índices dos itens}
Inicialmente: $dr_i = d_i$, $\forall i \in I$.
Faça $k = 1$ {Primeiro padrão de corte }
PARE=Falso {variável lógica que indica demanda não-nula}
Enquanto PARE=Falso)
P.3. Faça: $dem_i = dr_i$, $Sobra = L$ e $a_{ik} = 0$, $\forall i \in I$
Seja $i = 1$ {comece colocando o primeiro item no padrão }
Enquanto ($i \leq m$ e $Sobra \geq l_i$) faça:

$$a_{ik} = \min\left\{\left\lfloor \frac{Sobra}{l_i} \right\rfloor, dr_i\right\}$$

(a_{ik} é a quantidade de itens tipo i no padrão k)
Faça: $Sobra = Sobra - (a_{ik} l_i)$
$dr_i = dr_i - a_{ik}$
$i = i + 1$
Fim do Enquanto
P.4. Determine a frequência do padrão k:

$$x_k = \left\{ \begin{array}{l} \min \left\lfloor \frac{dem_i}{a_{ik}} \right\rfloor, \\ \forall i \in I \ tal \ que \ a_{ik} > 0 \end{array} \right\}$$

P.5. (Critério de Parada)
Se $dr_i = 0$ $\forall i \in I$ **então**
PARE=Verdade.
Senão Faça $k = k + 1$ e volte para **P.3**
Fim do Enquanto.
FIM

2.2 Heurística Gulosa

A Heurística Gulosa difere da Heurística FFD somente na maneira de como o padrão de corte é gerado (passo **P.3.** do algoritmo anterior). Ao invés de construí-lo com prioridade para os itens maiores, a geração do padrão consiste na resolução de um problema da mochila restrito conforme (9). O padrão gerado neste caso, tem a menor perda possível e deve ser utilizado com frequência máxima, sem que haja excesso na produção dos itens.

$$\begin{array}{ll} maximize: & l_1 a_{1j} + l_2 a_{2j} + ... + l_m a_{mj} \\ sujeito\ a: & \left\{ \begin{array}{l} l_1 a_{1j} + l_2 a_{2j} + ... + l_m a_{mj} \leq L \\ 0 \leq a_{ij} \leq d_i \text{ e inteiros}, \\ i = 1, ..., m \quad j = 1, ..., n. \end{array} \right. \end{array} \tag{9}$$

DOI: 10.5540/03.2015.003.01.0411

2.3 Heurística Gulosa Modificada

Propomos aqui modificação na heurística construtiva Gulosa apresentada anteriormente. Existem na literatura alguns poucos trabalhos que propuseram modificações nas heurísticas FFD ou Gulosa, embora todos estes trabalhos modifiquem a maneira de como o padrão de corte é gerado, nenhum deles altera a ordem de prioridade dos itens a serem colocados no padrão, sempre em ordem decrescente de tamanho. Em Cherri e Arenales[4] por exemplo, a heurística FFD foi modificada com a finalidade de resolver o problema de corte de estoque com reaproveitamento de sobras. Basicamente a heurística consiste em aplicar a heurística FFD para obter padrões de corte e, após gerado cada padrão, a perda/sobra é analisada. Se estiver dentro de limitantes aceitáveis (definidos previamente), o próximo padrão de corte é considerado, caso contrário, um item do padrão (o maior) é retirado. Assim, para a sobra gerada com a retirada do item é aplicado o problema da mochila, cuja capacidade é a perda no padrão adicionada ao tamanho do item retirado. Depois de resolvida a mochila, a perda/sobra gerada é analisada e, se ainda não estiver dentro de limitantes aceitáveis, outro item do padrão (segundo maior) é retirado. Novamente para a sobra gerada é resolvido o problema da mochila. Este procedimento é repetido até que a perda/sobra esteja dentro dos limitantes definido como aceitáveis ou a demanda seja totalmente atendida.

Apresentamos agora uma heurística Gulosa que coloca no padrão primeiro os itens de tamanho par em ordem decrescente de tamanho, seguidos dos ímpares também em ordem decrescente de tamanho, se o objeto em estoque é de comprimento par, isto é, $L = 2 \cdot W$, $W \in N$ e a prioridade é para os itens pares, ou seja, $l_i = 2 \cdot w_i$, $i \in \mathbf{P}$, $w_i \in N$. Em que $\mathbf{P}$ é um conjunto de índices i tal que l_i é par, para $i = 1, 2, ..., m$ e N é conjunto dos números naturais. Neste caso denominamos o procedimento de $Gulosa_{par}$. Analogamente, se o objeto em estoque é de comprimento ímpar ($L = 2 \cdot W + 1$, $W \in N$), denominamos a heurística de $Gulosa_{impar}$, neste caso os itens l_i tal que $l_i = 2 \cdot w_i + 1$, $i \in \mathbf{I}$, $w_i \in N$ são alocados primeiro no padrão seguido dos pares também em ordem decrescente de tamanho. Em que $\mathbf{I}$ é um conjunto de índices i tal que l_i é ímpar, para $i = 1, 2, ..., m$. Neste caso, o padrão só será aceito na solução se a soma das quantidades dos itens tipo i, $i \in \mathbf{I}$, $i = 1, 2, ..., m$ de comprimento ímpar produzidos pelo padrão é ímpar. Se isto não ocorre, a lista de prioridade dos itens de comprimento ímpar é alterada, modificando o valor de utilidade do item no problema da mochila restrito (9) passando para o próximo item ímpar, até que um padrão viável com tais características seja gerado. Esta mudança se faz necessária a fim de garantir que a porção do objeto a ser cortada neste caso tenha comprimento ímpar, visto que a soma de dois números naturais ímpares é um número par. Se após passar por todos os itens ímpares, o padrão gerado não é aceito, os itens são colocados em ordem decrescente de ímpares e pares, e segue-se a mesma metodologia adotada no caso da heurística $Gulosa_{par}$. A heurística proposta aposta que cada padrão construído tenha a menor perda possível, pois cortar uma porção ímpar de objeto de comprimento par($Gulosa_{par}$) ou cortar uma porção par de objeto de comprimento ímpar($Gulosa_{impar}$) sempre acarreta em perda.

3 Experimentos Computacionais

Para os experimentos computacionais os problemas testes foram obtidos através do Cutgen1, um gerador aleatório de problemas de corte de estoque unidimensional desenvolvido por Gau e Wascher [6]. Consideramos 18 classes de problemas cada uma com 100 instâncias e adotamos no Cutgen1 a semente, 1994. As classes estão divididas conforme a quantidade de itens m, a média d_M das demandas dos itens d_i para $i = 1, ..., m$ e combinações diferentes de valores v_1 e v_2 para a determinação do comprimento dos itens que são gerados no intervalo $[v_1 L, v_2 L]$, como mostrado na Tabela 1. O comprimento do objeto em estoque é $L = 1000$ para todas as instâncias testadas. As heurísticas FFD, Gulosa e $Gulosa_{par}$ foram implementadas em C_{++} e todos os testes computacionais para este trabalho foram realizados em um um computador Intel

Core i5-2450M com 4GB de memória (Ram).

Classe	m	v_1	v_2	d_M
1	10	0,01	0,2	10
2	10	0,01	0,2	100
3	20	0,01	0,2	10
4	20	0,01	0,2	100
5	40	0,01	0,2	10
6	40	0,01	0,2	100
7	10	0,01	0,8	10
8	10	0,01	0,8	100
9	20	0,01	0,8	10
10	20	0,01	0,8	100
11	40	0,01	0,8	10
12	40	0,01	0,8	100
13	10	0,2	0,8	10
14	10	0,2	0,8	100
15	20	0,2	0,8	10
16	20	0,2	0,8	100
17	40	0,2	0,8	10
18	40	0,2	0,8	100

Tabela 1: Problemas Testes Utilizados

Tabela 2: Testes Computacionais

Classe	**FFD**		**Gulosa**		**GulosaPar**	
	objs	pads	objs	pads	objs	pads
1	11,61	9,50	**11,41**	**8,98**	**11,40**	**9,47**
2	111,68	16,80	**89,86**	**10,07**	**76,23**	**10,32**
3	22,07	18,32	**21,83**	**17,86**	**20,05**	**17,98**
4	217.70	32,36	**168,47**	**18,85**	**147,44**	**18,99**
5	43,13	34,50	**42,78**	**33,61**	42,80	33,82
6	426,53	62,05	**327,32**	**34,66**	**276,06**	**35,29**
7	50,35	10,89	**50,57**	**10,51**	**48,97**	**10,54**
8	501,15	12,12	**187,79**	**7,03**	**197,99**	**6,64**
9	94,06	20,94	**94,55**	**20,21**	**92,25**	**20,23**
10	936,30	23,50	**364,36**	**12,78**	**380,39**	**12,23**
11	177,56	39,84	**183,6**	**39,05**	**176,86**	**39,35**
12	1770,26	44,65	**731,69**	**23,63**	**750,17**	**22,78**
13	63,59	10,55	**64,38**	**10,28**	**63,49**	**10,39**
14	633,70	11,05	**226,74**	**5,35**	**232,60**	**5,17**
15	119,93	20,64	**122,66**	**20,06**	**119,82**	**20,22**
16	1195,29	21,72	**441,65**	**9,73**	**437,77**	**9,79**
17	225,30	39,36	**238,47**	**38,65**	**227,16**	**38,88**
18	2247,42	41,58	**832,70**	**18,69**	**850,21**	**17,72**

4 Análise dos Resultados

Para Análise dos resultados, foram consideradas as soluções que apresentam o menor número de objetos e de padrões na solução, pois na prática, muitas empresas que utilizam máquinas de corte em suas linhas de produção, assumem custos adicionais cada vez que um novo padrão de corte necessita ser cortado, para mais detalhes ver([3], [5], [17], [16]). Na Tabela 2 temos a média de objetos (coluna objs) e média de padrões (coluna pads) correspondentes às soluções obtidas com as três heurísticas. As soluções em negrito correspondem às melhores de cada heurística, aquelas que em pelo menos uma das médias de objetos (objs) ou de padrões (pads), a heurística não é superada pelas correspondentes soluções das outras duas. Como mostra a Tabela 2,

Tabela 3: Desempenho em relação a FFD (%)

Classe	**Gulosa**		**GulosaPar**	
	objs	pads	objs	pads
1	$1,72\%$	$5,47\%$	$1,81\%$	$0,31\%$
2	$19,54\%$	$40,00\%$	$31,74\%$	$38,57\%$
3	$1,09\%$	$2,51\%$	$9,15\%$	$1,86\%$
4	$22,61\%$	$41,74\%$	$32,27\%$	$41,32\%$
5	$0,81\%$	$2,58\%$	$0,76\%$	$1,97\%$
6	$23,26\%$	$44,14\%$	$32,28\%$	$43,12\%$
7	$0,00\%$	$3,45\%$	$2,74\%$	$3,21\%$
8	$62,52\%$	$42,00\%$	$60,49\%$	$45,21\%$
9	$0,00\%$	$3,49\%$	$1,92\%$	$3,39\%$
10	$61,08\%$	$45,61\%$	$59,37\%$	$47,96\%$
11	$0,00\%$	$1,98\%$	$0,39\%$	$1,23\%$
12	$58,67\%$	$47,08\%$	$57,62\%$	$48,98\%$
13	$0,00\%$	$2,56\%$	$0,16\%$	$1,52\%$
14	$64,22\%$	$51,58\%$	$63,29\%$	$53,21\%$
15	$0,00\%$	$2,81\%$	$0,10\%$	$2,03\%$
16	$63,06\%$	$55,20\%$	$63,38\%$	$54,92\%$
17	$0,00\%$	$1,80\%$	$0,00\%$	$1,22\%$
18	$62,95\%$	$51,34\%$	$62,17\%$	$57,38\%$

as soluções da heurística $Gulosa_{par}$ são inferiores às soluções da heurística Gulosa apenas na Classe 5. Observamos que naquelas soluções em que a $Gulosa_{par}$ é superior, o seu desempenho em relação à Gulosa é melhor, tendo como referência as soluções geradas pela FFD, como mostra a Tabela 3. Na maioria das classes testadas houve uma significante redução no número médio de objetos utilizados, isto ocorre porque a prioridade para os itens de comprimento par em ordem decrescente, conforme explicado na subseção 2.3, contribui para que o padrão gerado pela mochila apresente menor perda, que o gerado pela heurística Gulosa. Se considerarmos a classe 2, percebe-se que enquanto a heurística Gulosa reduz a média de objetos cortados em $19,54\%$ a $Gulosa_{par}$ reduz em $31,74\%$, mantendo-se próximos a redução na média de padrões gerados, 40% e $38,57\%$ respectivamente como mostra a Tabela 3.

5 Conclusão e Trabalhos Futuros

Apresentamos neste trabalho modificação de uma heurística da literatura que gera uma solução inteira para o problema de corte de estoque unidimensional. Os testes computacionais mostraram que as soluções geradas pelo método proposto tendem a ser de melhor qualidade quando comparadas com outros métodos da literatura para o problema. Para trabalhos futuros, pretende-se implementar a heurística $Gulosa_{impar}$, e em seguida testar o método para problemas de corte unidimensional com objetos de diferentes comprimento em estoque, a fim de compararmos os resultados com outros trabalhos da literatura sobre o tema.

6 Reconhecimento

Os autores agradecem à UESB, CAPES e CNPq e pelo apoio financeiro e aos revisores anônimos pelos comentários que contribuíram para a melhora deste trabalho.

Referências

[1] Arenales, M. N.; Morabito, R. & Yanasse, H. H. Problemas de corte e empacotamento. In: Simpósio Brasileiro de Pesquisa Operacional. Anais Sobrapo. São João Del Rei-Rio de Janeiro, 2004.

[2] Belov, G. & Scheithauer, G. Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. *European Journal of Operational Research* v. 141, p. 295-312, 2002.

[3] Cerqueira, G.R.L & Yanasse, H.H. A pattern reduction procedure in a one-dimensional cutting stock problem by grouping items according to their demands. *Journal of Computational Interdisciplinary Sciences*, v. 2, p. 159-164, 2009.

[4] Cherri, A.C. & Arenales, M.N. O problema de corte de estoque com reaproveitamento das sobras de material. Heurística FFD Modificada *XXXVIII Simpósio Brasileiro de Pesquisa Operacional*. Gramado-RS, 2005.

[5] Foerster, H. & Wascher, G. Pattern reduction in one-dimensional cutting stock problems. *International Journal of Production Research*, v. 38, p. 1657-676, 1999.

[6] Gau, T. & Wascher, G. Cutgen1: a problem generator for one-dimensional cutting stock problem. *European Journal of Operational Research*, v. 84, p. 572-579, 1995.

[7] Gilmore, P. & Gomory, R. A linear programming approach to the cutting stock problem. *Operations Research*, v. 9, p. 849-859, 1961.

[8] Gilmore, P. & Gomory, R. A linear programming approach to the cutting stock problem-part II. *Operations Research*, v. 6, p. 863-888, 1963.

[9] Hinxman, A. I. The trim-loss and assortment problems: a survey. *European Journal of Operational Research*, v. 5, p. 8-18, 1980.

[10] Marcotte, O. The cutting stock problem and integer rounding. *Mathematical Programming*, v. 33, p. 82-92, 1985.

[11] Marcotte, O. An instance of the cutting stock problem for which the rounding property does not hold. *Operations Research Letters*, v. 4, p. 239-243, 1986.

[12] Poldi, K.C & Arenales, M.N. Heurísticas para o problema de corte de estoque unidimensional inteiro *Pesquisa Operacional*, v.26, n.3, p.473-492, 2006.

[13] Scheithauer, G. & Terno, J. The modified integer round-up property of the one-dimensional cutting stock problem. *European Journal of Operational Research*, v. 84, p. 562-571, 1995.

[14] Stadtler, H. A one-dimensional cutting stock problem in the aluminium industry and its solution. *European Journal of Operational Research*, v. 44, p. 209-223, 1990.

[15] Wascher, G. & Gau, T. Heuristics for the integer one-dimensional cutting stock problem: a computational study. *OR Spektrum*, v. 18, p. 131-144, 1996.

[16] Umetani, S.; Yagiura, M. & Ibaraki, T. One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research*, v. 146, p. 388-402, 2003.

[17] Yanasse, H. H. & Limeira, M. S. A hybrid heuristic to reduce the number of different patterns in cutting stock problems. *Computers & Operations Research*, v. 33, p. 2744-2756, 2006.

DOI: 10.5540/03.2015.003.01.0411