

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Relax-and-Fix Aplicado ao Problema de Corte de Estoque com Data de EntregaElisama de Araújo Silva Oliveira¹

CEFET-MG

Elisângela Martins de Sa²

CEFET-MG

Sérgio Ricardo de Souza³

CEFET-MG

Elizabeth Fialho Wanner⁴

CEFET-MG

Resumo. O Problema de Corte de Estoque (PCE) consiste em cortar um conjunto de objetos disponíveis em estoque para produzir um conjunto de itens em quantidades e comprimentos especificados, otimizando, em geral, um objetivo relacionado com as perdas de materiais, como a minimização do número de objetos cortados. O presente artigo propõe uma extensão deste problema, que acrescenta, aos pedidos, uma data de entrega. A abordagem mais utilizada na literatura para a resolução do PCE é o método simplex com geração de colunas proposto por Gilmore e Gomory [5], [6]. Devido à complexidade computacional para resolução de problemas de grande dimensão usando esta abordagem, neste trabalho é proposta uma heurística *Relax-and-Fix* para sua resolução. Resultados computacionais mostram que a fase de geração de colunas gera bons padrões de corte que podem ser usados na fase da solução inteira e que a heurística *Relax-and-Fix* gera soluções com a mesma qualidade das soluções obtidas pelo CPLEX, em um menor tempo computacional médio.

Palavras-chave Geração de colunas, Padrões de corte, Data de entrega, *Relax-and-fix*.

1 Introdução

O Problema de Corte de Estoque (PCE) consiste em cortar um conjunto de objetos idênticos com dimensões fixas para produzir um conjunto de itens em quantidades e dimensões especificadas, de modo a minimizar as perdas de materiais e atender toda a demanda de itens. O PCE é essencial no planejamento da produção em várias indústrias, como indústrias de papel, vidro, móveis, plástico, chapas de madeira, aço, têxtil, etc. Os

¹elisamamatematica.licenciatura@hotmail.com.

²elisangela.martinss@gmail.com.

³sergio@dppg.cefetmg.br.

⁴efwanner@gmail.com.

objetos a serem cortados podem ser bobinas de aço, bobinas de papel, placas de alumínio, chapas de metal, placas de madeira, dentre outros.

De acordo com Dyckhoff [3], um PCE pode ser classificado de acordo com a dimensão do objeto a ser cortado. O problema pode ser unidimensional (1D), bidimensional (2D) ou tridimensional (3D), quando uma, duas ou três dimensões dos objetos, respectivamente, são relevantes no processo de corte. A tipologia mais recente para este problema é, contudo, a proposta por Wäscher et al. [9], sendo parcialmente baseada nas ideias originais de Dyckhoff [3], mas introduzindo novos critérios de categorização. De acordo com esta classificação, o presente artigo aborda uma extensão do problema 1D - *Single Stock Size Cutting Stock Problem* - SSSCSP: Problema de Corte de Estoque com um único comprimento, uma vez que considera-se a data de entrega dos itens cortados.

A primeira modelagem matemática para o PCE foi introduzida por Kantorovich [7]. Algumas versões mais recentes de problemas de corte consideram, além das decisões relativas a como cortar os itens, decisões relacionadas às datas de entrega dos itens que serão cortados. Reinertsen e Vossen [8] abordam o PCE unidimensional com Data de Entrega (PCEDE) aplicado em uma indústria de bobinas de aço. Arbib e Marinelli [1] estudaram o PCE unidimensional com Data de Entrega, propondo novas formulações e métodos de resolução. Arbib e Marinelli [2] estudam o problema de *bin packing* unidimensional com data de entrega, em que o objetivo é minimizar uma combinação convexa do *makespan*, atraso e adiantamento.

O presente artigo propõe um modelo matemático de programação linear inteira para o Problema de Corte de Estoque com Data de Entrega. Além de penalizar o não cumprimento da data dos pedidos, considera-se também o adiantamento na entrega dos itens. Para a resolução do problema, propõe-se um método de resolução em duas fases. Na primeira fase, o problema relaxado, usando a técnica de geração de colunas, é resolvido. Usando as colunas geradas na primeira fase, a segunda fase consiste na aplicação da heurística *Relax-and-Fix* para a resolução do problema em sua versão inteira.

Este artigo está organizado da seguinte forma. A Seção 2 descreve o problema proposto e apresenta uma formulação matemática para resolvê-lo. A Seção 3 apresenta os métodos de resolução utilizados. Os resultados computacionais são apresentados na Seção 4. A Seção 5 apresenta as considerações finais.

2 Problema de Corte de Estoque com Data de Entrega

Uma abordagem para solucionar um PCE se baseia em encontrar padrões de corte e a frequência destes padrões, ou seja, a quantidade de vezes que esses padrões serão executados (cortados). O PCEDE é uma extensão do PCE em que, além de se considerar o planejamento dos padrões de corte e suas respectivas frequências, considera-se também o atraso e o adiantamento na entrega dos itens.

Considere que exista em estoque um conjunto de objetos (grandes) idênticos de comprimento L , em uma quantidade suficiente para que seja possível atender a toda a demanda. Seja, então, um conjunto de n tipos de itens menores de comprimento l_i ($i = 1, 2, \dots, n$), a serem programados em uma operação de corte para pedidos em um horizonte de pla-

nejamento dividido em períodos de tempo $t = 1, 2, \dots, T$, no qual se deve produzir uma determinada demanda b_{ti} a ser entregue em uma data d_t uma quantidade do item tipo i . Seja tp_t o tempo que se leva para cortar o pedido em um período t . Como, em algumas operações de corte, nem sempre é possível atender a todos os pedidos na sua respectiva data de entrega, há uma penalização α_t por unidade de término de atraso e, caso no processo de corte haja pedidos entregues antes de sua data de entrega, também há uma penalização β_t por unidade de tempo de término de adiantamento. Considere, além disso, que não seja possível receber pedidos depois que o processo de corte se iniciou. O objetivo do problema será agrupar os n itens a serem processados de acordo com a capacidade do objeto, de forma a minimizar a soma de três termos: as penalidades por adiantamentos, as penalidades por atrasos nas entregas e os custos do uso de matéria-prima.

Chama-se padrão de corte a como um objeto em estoque é cortado para a produção dos itens demandados. A um padrão de corte, associa-se um vetor m -dimensional, que contabiliza os itens produzidos, de modo que o padrão de corte a_{ij} é a quantidade de itens do tipo i produzidos pelo padrão de corte j para $j = 1, \dots, N$. Sejam, então, as variáveis de decisão do problema dadas por: x_{tj} , que é o número de objetos cortados no período t segundo o padrão de corte j ; T_t e E_t que são, respectivamente, o tempo de atraso e o tempo de adiantamento na entrega do pedido no período t . O modelo matemático para o PCEDE é dado por:

$$\min \quad \sum_{t=1}^T \sum_{j=1}^N x_{tj} + \sum_{t=1}^T \beta_t E_t + \sum_{t=1}^T \alpha_t T_t \quad (1)$$

$$\text{sujeito a:} \quad T_t \geq tp_t + T_{t-1} - E_{t-1} - (d_t - d_{t-1}), \quad \forall t, \quad (2)$$

$$E_t \geq d_t - d_{t-1} - T_{t-1} + E_{t-1} - tp_t, \quad \forall t, \quad (3)$$

$$tp_t = \sum_{j=1}^N x_{jt}, \quad \forall t, \quad (4)$$

$$\sum_{j=1}^N a_{ij} x_{tj} \geq b_{ti}, \quad \forall t, \forall i, \quad (5)$$

$$E_0 = 0, \text{ e } T_0 = 0, \quad (6)$$

$$x_{tj} \in \mathbb{Z}, \quad \forall j, \quad \forall t, \quad (7)$$

$$E_t, T_t, tp_t \in \mathbb{Z}, \quad \forall t. \quad (8)$$

O objetivo desse problema é a minimização da função objetivo (1), composta por três parcelas: (i) o número de objetos cortados x_{jt} ; (ii) o atraso T_i ; e (iii) o adiantamento E_i na entrega dos pedidos. O conjunto de restrições (2)-(3) contabiliza os tempos de atrasos e de adiantamentos. O conjunto de restrições (4) mostra o tempo que será gasto na produção dos pedidos em cada um dos períodos, assumindo que se gasta uma unidade de tempo para cortar um objeto, independente do padrão de corte que será cortado. O conjunto de restrições (5) garante que a demanda dos períodos deve ser atendida. O conjunto de restrições (6) representa as condições de contorno iniciais, que afirmam que não existe atraso ou adiantamento de pedidos no início do processo de corte. As restrições (7)-(8)

representam as condições de não-negatividade e integralidade das variáveis de decisão.

3 Método de Solução

3.1 Método de Geração de Colunas

O Método de Geração de Colunas proposto por Gilmore e Gomory [5],[6], é uma modificação do método simplex, que consiste em resolver o problema relaxado com apenas um subconjunto de padrões de corte, conhecido como problema mestre. Para isso, a cada iteração, são gerados padrões de corte que melhoram a solução do problema relaxado até a obtenção da solução ótima.

No presente artigo, o problema mestre é dado pelo modelo (1)-(8) do PCEDE, tendo apenas alguns padrões de cortes adicionados e considerando, como relaxadas, as restrições de integralidade (7)-(8). Para gerar os padrões de corte, a cada iteração deve ser resolvido um subproblema com a variável de decisão α_{ti} , que contabiliza, no período t , quantas vezes o item i deve ser usado. O subproblema é definido pelo seguinte problema da mochila, para cada período t :

$$\min \quad c_j - \sum_{i=1}^n \pi_{ti} \alpha_{ti} \tag{9}$$

$$\text{sujeito a:} \quad \sum_{i=1}^n \ell_i \alpha_{ti} \leq L, \quad \forall t. \tag{10}$$

$$\alpha_{ti} \geq 0 \quad \text{inteiros.} \tag{11}$$

no qual as variáveis π_{ti} são variáveis duais do problema mestre, relacionadas às restrições de atendimento da demanda, dadas pelas restrições (5) e c_j é o custo de cortar um padrão de corte j , definido como 1.

3.2 Heurística *Relax-and-Fix*

A heurística *Relax-and-Fix*, utilizada para construir uma solução para problemas de programação linear inteira misto [10], baseia-se em dividir o problema, com variáveis inteiras, em subproblemas mais simples. Para isso, o conjunto de variáveis inteiras é dividido em P subconjuntos distintos $Q_i, i = 1, \dots, P$. Em cada iteração n , resolve-se um subproblema em que as variáveis do conjunto Q_n são definidas como inteiras; as variáveis dos conjuntos Q_i ($i = 1, \dots, n - 1$) são fixadas; as variáveis dos conjuntos Q_i ($i = n + 1, \dots, P$) são relaxadas. Se o subproblema corrente for infactível, o processo é interrompido, pois não é possível encontrar uma solução para o problema com as variáveis pertencentes aos conjuntos Q_i ($i = 1, \dots, n - 1$) fixadas. Caso contrário, as variáveis do conjunto Q_n são fixadas. Então, o processo se repete, até que todas as variáveis tenham solução inteira.

Em um primeiro momento, aplicou-se o método de geração de colunas. Para iniciar o método de geração de colunas, foi utilizado um conjunto de padrões de corte homogêneos.

Tabela 1: Descrição das Classes de problemas-teste.

Classe	Períodos	Itens	Intervalo dos itens	Intervalo da demanda
1	3	10	[266,60]	[3 , 44]
2	6	20	[292,60]	[1 , 44]
3	3	10	[299,56]	[3 , 44]
4	6	20	[311,54]	[1 , 44]
5	3	10	[288,60]	[3 , 44]
6	6	20	[285,60]	[1 , 44]

Em seguida, com os padrões de corte determinados, foi aplicada a heurística *Relax-and-Fix*, com o intuito de obter uma solução para as variáveis x_{tj} . Nesta fase, a partição dos conjuntos foi feita sobre os períodos de tempo. Ao final desse passo, obteve-se uma solução para todas as variáveis inteiras x_{tj} do modelo.

4 Resultados Computacionais

Para a realização do estudo computacional, o modelo foi resolvido pelo C++ Concert Technology/CPLEX 12.8, em um notebook Intel Core i7 com 3.60 GHz e 16 Gbyte de memória e o sistema operacional Linux-Ubuntu. Os experimentos computacionais foram realizados usando seis classes de instâncias, contendo 20 problemas-testes com 10 itens e 20 itens, geradas pelo pacote CUTGEN1 [4], que é um gerador de problemas-testes de corte de estoque.

A Tabela 4 apresenta uma descrição das classes de instâncias analisadas para a resolução do problema. Para todas as instâncias, o objeto tem $1000u$ de comprimento. Nesta tabela, a primeira coluna indica o número da classe analisada; a segunda coluna, o número de períodos de cada problema; a terceira coluna, a quantidade de tipos de itens a serem cortados; e a quarta coluna os valores do maior e do menor comprimento do item.

Para os testes computacionais, a data de entrega d_t foi calculada como:

$$d_t = \left[0, 1 \times \frac{\sum_{i=1}^n l_i b_{ti}}{L} \right], \quad \forall t. \quad (12)$$

Em todos os problemas, assume-se que $\alpha = 1$ (penalizações por atraso) e $\beta = 1$ (penalizações por adiantamento) na entrega dos itens, e define-se o tempo limite de 600 s. As Tabelas 2, 3 e 4 mostram os resultados médios para as 120 instâncias analisadas. Os resultados estão divididos na fase de Geração de Colunas (Relaxada) e na fase de *Relax-and-Fix* (Inteira). Para testar a qualidade da solução inteira obtida, os resultados são comparados com a solução obtida pelo CPLEX, sendo o *gap* na fase da aplicação da heurística *Relax-and-Fix* dado por:

$$gap = 100 \times \frac{F_{Relax-and-Fix} - F_{Geração-de-Colunas}}{F_{Relax-and-Fix}}. \quad (13)$$

A Tabela 2 apresenta a média de colunas geradas na fase de Geração de Colunas das Classes. Percebe-se que é um número pequeno de colunas, o que é bom, pois, assim, reduz-se o tamanho do problema a ser resolvido. Note-se que, para resolver o problema

Tabela 2: Média de colunas geradas na geração de colunas nas Classes.

Problema	Colunas
1	16
2	26
3	16
4	29
5	19
6	29
Média	22,5

Tabela 3: Resultados dos termos da função objetivo de todas as Classes.

Classe	Relaxada			<i>Relax-and-Fix</i>			Inteira via CPLEX		
	ATR	ADI	Corte	ATR	ADI	Corte	ATR	ADI	Corte
1	84,44	88,02	128,86	90,65	94,45	138,30	90,65	94,45	138,30
2	174,31	189,15	273,14	189,85	205,10	296,85	189,85	205,10	296,85
3	86,01	90,04	131,36	92,85	96,70	141,45	92,85	96,70	141,45
4	189,47	457,29	552,71	202,30	490,90	593,75	202,30	490,90	593,75
5	81,07	233,94	273,98	85,90	248,80	289,30	85,90	248,80	289,30
6	178,87	430,69	520,86	191,95	464,40	560,85	191,95	464,40	560,85
Média	132,36	248,18	313,48	142,25	266,72	336,75	142,25	266,72	336,75

Tabela 4: Resultado de todas as Classes para a função objetivo e tempo.

Classe	Relaxada		<i>Relax-and-Fix</i>			Inteira via CPLEX	
	F.Objetivo	Tempo(s)	F.Objetivo	Tempo(s)	gap(%)	F.Objetivo	Tempo(s)
1	300,44	0,18	323,40	0,20	7,09	323,40	0,16
2	636,62	0,23	691,80	0,26	7,97	691,80	0,18
3	307,41	0,19	331,00	0,22	7,12	331,00	0,77
4	1200,04	1,57	1286,70	1,86	6,73	1286,70	105,57
5	589,00	0,98	624,00	1,12	5,60	624,00	190,55
6	1130,35	1,66	1217,20	1,71	7,13	1217,20	175,89
Média	693,97	0,80	745,68	0,89	6,94	745,68	78,85

inteiro, a quantidade de colunas pode ser muito grande, e, conseqüentemente, o número de variáveis, tornando-se inviável analisar todas as possíveis colunas.

A Tabela 3 mostra a comparação das médias de cada um dos termos da função objetivo (ATR=tempo de atraso, ADI=tempo de adiantamento e Corte=objetos cortados). Em todas as instâncias, verifica-se que o valor de tempo de atraso é sempre menor que o valor de tempo de adiantamento. No pior caso a fase de geração de colunas demorou 2,29 s, a fase do *Relax-and-Fix* demorou 2,64 s e o CPLEX demorou 552,02 s. Observa-se que para as classes 1 e 2 o tempo médio gasto foi maior que o tempo gasto pelo CPLEX, o que nos leva a crer que são problemas não tão desafiadores.

A Tabela 4 mostra os valores para a solução relaxada e inteira. Observa-se que, para as instâncias tratadas, a resolução ocorre, em média, em um tempo computacional pequeno. O menor valor de *gap* para essas classes foi de 5,60%. Observa-se também que um fator complicante para a solução de instâncias do problema é a quantidade de tipos de itens, pois, à medida que a quantidade de itens cresce, o tempo despendido para a solução também aumenta. A partir destes resultados obtidos, verifica-se que a técnica *Relax-and-Fix* gera o mesmo resultado que o obtido pelo CPLEX, em um menor tempo computacional médio.

5 Conclusões e Propostas Futuras

Este artigo apresentou uma modelagem matemática para o PCEDE e um método de solução inteira usando a heurística *Relax-and-Fix*, baseado nas colunas geradas na fase de geração de colunas. Os resultados mostraram que esta solução gerou baixos valores de *gap*, em tempos computacionais pequenos. O uso da técnica de Geração de Colunas determinou boas colunas (padrões de corte) para a solução do problema, usadas como entrada para a solução do problema inteiro, sem a necessidade de gerar todas as colunas possíveis para o problema. A heurística *Relax-and-Fix* gerou boas soluções para o problema inteiro, em um menor tempo computacional, encontrando a mesma solução que a obtida pelo *solver* CPLEX. Como trabalho futuro, propõe-se resolver o PCDE usando a heurística *Local Branching*, tendo, como solução inicial, a solução gerada pela heurística *Relax-and-Fix*.

Referências

- [1] C. Arbib and F. Marinelli. On cutting stock with due dates, *Omega*, 46:11–20, 2014. DOI: 10.1016/j.omega.2014.01.004.
- [2] C. Arbib and F. Marinelli. Maximum lateness minimization in one-dimensional bin packing, *Omega*, 68:76–84, 2017. DOI: 10.1016/j.omega.2016.06.003.
- [3] H. Dyckhoff. A typology of cutting and packing problems, *European Journal of Operational Research*, 44(2):145–159, 1990. DOI: 10.1016/j.ejor.2005.12.047.
- [4] T. Gau and G. Wäscher. CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, 84(3):572–579, 1995. DOI: 10.1016/0377-2217(95)00023-J.
- [5] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem, *Operations Research*, 9(6):849–859, 1961. DOI: 10.1287/opre.9.6.849.
- [6] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem—Part II, *Operations Research*, 11(6):863–888, 1963. DOI: 10.1287/opre.11.6.863.
- [7] L. V. Kantorovich. Mathematical methods of organizing and planning production, *Management Science*, 6:366–422, 1960. DOI: 10.1287/mnsc.6.4.366.
- [8] H. Reinertsen and T. W. M. Vossen, The one-dimensional cutting stock problem with due dates, *European Journal of Operational Research*, 201(3):701–771, 2010. DOI: 10.1016/j.ejor.2009.03.042.
- [9] G. Wäscher, H. Haußner, and H. Schumann, An improved typology of cutting and packing problems, *European Journal of Operational Research*, 183(3):1109–1130, 2007. DOI: 10.1016/j.ejor.2005.12.047.
- [10] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.