

O Problema de Corte de Estoque com Datas de entrega, *Setup* e Estoque

Elisama de Araújo Silva Oliveira, Elisângela Martins de Sá,
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Avenida Amazonas, 7675, Nova Gameleira, Belo Horizonte, MG, 30510-000
elisamaoliveirasophia@gmail.com, elisangelamartins@cefetmg.br

Sérgio Ricardo de Souza, Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Avenida Amazonas, 7675, Nova Gameleira, Belo Horizonte, MG, 30510-000
sergio@cefetmg.br, efwanner@gmail.com

RESUMO

Este trabalho aborda o problema multiperíodo de corte de estoque unidimensional com datas de entrega, *setup* independente da sequência e estoque (1D-PMCE-DE-SI-E). Este problema é uma extensão do Problema de Estoque de Corte unidimensional (1D-PCE) clássico, consistindo em determinar a forma ótima de cortar unidades de um material de tamanho fixo, com o objetivo de atender a uma demanda de peças menores. O 1D-PMCE-DE-SI-E busca minimizar atrasos e antecipações dos pedidos, quantidade de materiais cortados, *setup* e estoque de componentes. Este trabalho apresenta um modelo de programação matemática inteira e uma abordagem de solução matheurística para o 1D-PMCE-DE-SI-E, empregando geração de colunas, procedimento de arredondamento e a matheurística Local Branching. Experimentos computacionais mostram que o método de solução proposto produz soluções viáveis de alta qualidade comparado à resolução do problema com as colunas geradas usando o solucionador CPLEX.

PALAVRAS CHAVE. Problema multiperíodo de corte de estoque, Datas de entrega, *setup*, Geração de coluna, Local Branching.

Tópicos: MH – Meta-heurísticas, OC – Otimização Combinatória.

ABSTRACT

This work addresses the one-dimensional multi-period cutting stocking problem with due dates, sequence-independent setup, and inventory (1D-MCSP-DD-SIS-I). This problem is an extension of the classic one-dimensional cutting stocking problem (1D-CSP), consisting of determining the optimal way to cut units of a fixed-size material to meet a demand for smaller parts. 1D-MCSP-DD-SIS-I seeks to minimize order delays and early orders, quantity of cut materials, setup, and component inventory. This work presents an integer mathematical programming model and a matheuristic solution approach for the 1D-PMCE-DE-SI-E problem, employing column generation, rounding procedure, and the Local Branching matheuristic. The computational experiments allow us to conclude that the proposed solution method provides high-quality feasible solutions when compared to solving the problem with columns generated using the CPLEX solver.

KEYWORDS. Multi-Period Cutting Stock Problem, Due Dates, *setup*, Column Generation, Local branching.

Paper topics: MH - Metaheuristics, OC- Combinatorial optimization.

1. Introdução

O Problema de Corte de Estoque (PCE) clássico envolve o corte de objetos de estoque a partir das solicitações de pedidos para produzir itens, atender à demanda e minimizar uma função objetivo. Neste trabalho, abordamos especificamente a variante unidimensional do problema de corte, em que apenas uma única dimensão do objeto é relevante durante o processo de corte.

Os primeiros estudos sobre o Problema de Corte de Estoque (PCE) focaram na minimização das perdas no processo de corte. A formulação inicial do problema foi proposta por Kantorovich [1960]. As modelagens e métodos mais influentes na literatura são devido a Gilmore e Gomory [1961, 1963, 1965].

O primeiro estudo sobre o PCE com Datas de Entrega (PCE-DE) foi proposto por Li [1996], que abordou o caso bidimensional. Nesse trabalho, foram considerados objetos com comprimentos e larguras diferentes e foram introduzidos modelos matemáticos com restrições de datas de entrega para minimizar uma função objetivo. Em Hendry et al. [1996], os autores aplicaram este modelo a um caso prático na indústria de cobre, propondo a solução sequencial do problema de corte e de *scheduling*. Reinertsen e Vossen [2010] focaram no PCE-DE unidimensional, apresentando modelos de otimização combinatória e técnicas de geração de colunas. Arbib e Marinelli [2014] aprimoraram o modelo de Reinertsen e Vossen [2010], agregando novas ideias. Braga et al. [2016] propuseram uma variante compacta do modelo, incluindo desigualdades válidas baseadas no problema da mochila. Arbib e Marinelli [2017] estudaram uma variação do problema com *Bin Packing* unidimensional e datas de entrega (BP-DE), buscando minimizar uma função objetivo combinando *makespan*, atraso e adiantamento.

Este trabalho aborda o problema multiperíodo de corte de estoque com datas de entrega e *setups* independente da sequência e estoque (PMCE-DE-SI-E), que é um problema de corte de estoque unidimensional multiperíodo, com o objetivo de atender às demandas de itens em diferentes horizontes de planejamento. Vale ressaltar que, no contexto considerado, as restrições de capacidade da máquina são estabelecidas com base no tempo disponível para sua operação dentro de cada período do horizonte de planejamento. Assim, mesmo que existam objetos suficientes em estoque, o fator determinante para a viabilidade do atendimento da demanda em determinado período é o tempo total requerido pelas operações. Em função disso, pode não ser viável atender todos os pedidos exatamente nas datas previstas de entrega.

O problema também considera a existência de *setup*. Assim, assume-se que, na prática, muitos processos de corte envolvem operações preparatórias, como a regulagem da máquina, troca de ferramentas, ajustes de posicionamento, dentre outras. Em muitos casos, essas atividades precisam ser realizadas toda vez que se inicia o corte de uma nova unidade de matéria-prima, independentemente do padrão de corte utilizado. Esse tempo deve ser contabilizado no planejamento, especialmente quando existem restrições de tempo de produção.

Vários trabalhos consideram a minimização do número de *setups* ou restrições de *setup*, como, por exemplo, Vanderbeck [2000], Foerster e Wascher [2000], Umetani et al. [2003], Umetani et al. [2006] e Araujo et al. [2014]. No entanto, Oliveira et al. [2025] foi o único trabalho que considerou data de entrega, minimizando o atraso e adiantamento, e *setup* independente da sequência, como proposto neste trabalho.

A grande contribuição deste trabalho, em relação a Oliveira et al. [2025], é a inclusão da minimização do custo de estoque. Apesar de Oliveira et al. [2025] considerar que os itens podem ser armazenados de um período para outro, o problema abordado por eles não considerou nenhum componente de limitação ou minimização de estoque, o que pode gerar soluções com alto nível de estoque. Neste trabalho, é considerado que o custo de estoque de itens pode ser bastante relevante. Desta forma, o problema considera a minimização do custo de objetos cortados, de

atrasos e antecipações, de *setup* e de estoque.

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

2. Problema proposto

Esta seção apresenta, na Subseção 2.1, a descrição geral do problema abordado. Em seguida, a Subseção 2.2 detalha a notação utilizada e introduz a formulação matemática do problema abordado.

2.1. Descrição do problema

O PMCE-DE-SI-E consiste em determinar a melhor maneira de se cortar objetos em estoque em itens menores, otimizando uma função objetivo. O PCE inteiro pode ser definido da seguinte maneira. Considere que temos disponível em estoque um número suficientemente grande de objetos (barras, bobinas, etc.) de comprimento L e um conjunto de n tipos de itens. O item i , $i = 1, \dots, n$, tem tamanho ℓ_i e demanda conhecida b_i . Os comprimentos dos itens são tais que $\ell_i < L, \forall i = 1, \dots, n$. Além disso, o modelo recebe o tempo de *setup* da máquina e a data de entrega dos pedidos, sendo que cada uma das entregas deve ser realizada em um período de tempo específico.

Note que, em cada período t , a máquina de corte dispõe de um tempo de processamento equivalente ao intervalo entre as datas de entrega consecutivas. Como cada *setup* e operação de corte consomem uma quantidade específica de tempo, essa disponibilidade impõe uma limitação à capacidade operacional da máquina em cada período. Nessa perspectiva, o atraso pode ser interpretado como uma medida do tempo excedido em relação à capacidade disponível para atender à demanda do respectivo período.

A Figura 1 exemplifica o objeto que se tem em estoque (na cor cinza) e os itens demandados (nas cores vermelho, azul, amarelo).

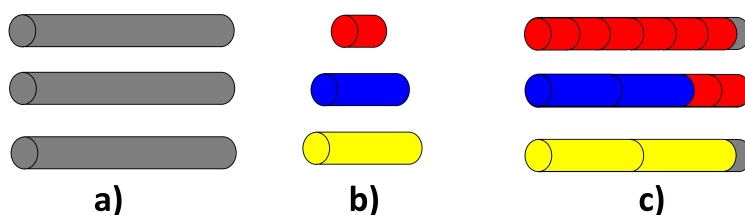


Figura 1: Exemplo de um PCE - Objeto em estoque e um conjunto de tipos de itens. Fonte: Própria dos autores.

O PMCE-DE-SI-E consiste em determinar, para cada período, o conjunto ótimo de padrões de corte viáveis e a quantidade a ser produzida de cada padrão. O objetivo é minimizar, de forma integrada, o custo associado ao uso das matérias-primas, as penalidades por entregas antecipadas ou atrasadas, os custos de estocar itens e os custos de *setup* decorrentes da alternância entre padrões de corte ao longo do horizonte de planejamento.

2.2. Notação e formulação

As Tabelas 1 e 2 apresentam as variáveis de decisão do modelo e os parâmetros utilizados na formulação do PMCE-DE-SI-E.

Tabela 1: Definições das variáveis de decisão para o PMCE-DE-SI-E.

Variáveis de decisão	definição
x_j^t	número de objetos cortados no período t segundo o padrão de corte j
E^t	tempo de adiantamento na entrega do pedido no período t
T^t	tempo de atraso na entrega do pedido no período t
e_i^t	quantidade de itens i restantes em estoque no final do período t
S_j^t	indica se no período t é utilizado o padrão j ($S_j^t = 1$) ou não ($S_j^t = 0$)
$start^t$	tempo de início do processo de corte no período t
q^t	tempo necessário para cortar todos os itens no período t

Tabela 2: Definições de parâmetros para o PMCE-DE-SI-E.

Parâmetro	definição
B	valor suficientemente grande
L	comprimento dos objetos em estoque
ℓ_i	comprimento do item i
M	número total de períodos no horizonte de planejamento
N^t	número de padrões de corte possíveis de objetos no período t
a_{ij}^t	quantidade de itens do tipo i produzidos por um padrão de corte j referente ao período t
b_i^t	quantidade de demanda do tipo de item i com data de entrega d^t
c_j^t	custo de corte de um objeto com padrão j para cada período t
ce_i^t	custo de estocar um tipo de item i para cada período t
d^t	data de entrega no período t
f_j^t	custo de <i>setup</i> para produzir um padrão de corte j no período t
β^t	penalidade por entregar pedidos antes da data de entrega (penalização por antecipação)
γ^t	penalidade por entregar pedidos após a data de entrega (penalização por atraso)
r_j^t	tempo de <i>setup</i> do padrão de corte j no período t
p_j^t	tempo de corte do padrão de corte j no período t

A formulação matemática para o PMCE-DE-SI-E é dada por:

$$\min \quad \sum_{t=1}^M \sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^M \beta^t E^t + \sum_{t=1}^M \gamma^t T^t + \sum_{t=1}^M \sum_{i=1}^n ce_i^t e_i^t + \sum_{t=1}^M \sum_{j=1}^{N^t} f_j^t S_j^t \quad (1)$$

$$\text{subject to:} \quad \sum_{j=1}^N a_{ji}^t x_j^t - e_i^t + e_i^{(t-1)} = b_i^t \quad \forall t, i \quad (2)$$

$$x_j^t \leq B S_j^t \quad \forall t, j \quad (3)$$

$$start^t \geq d^{t-1} + T^{t-1} - E^{t-1} \quad \forall t \quad (4)$$

$$q^t = \sum_{j=1}^N p_j^t x_j^t + \sum_{j=1}^N r_j^t S_j^t \quad \forall t \quad (5)$$

$$start^t + q^t + E^t - T^t = d^t \quad \forall t \quad (6)$$

$$E^0 = 0 \quad (7)$$

$$T^0 = 0 \quad (8)$$

$$d^0 = 0 \quad (9)$$

$$e_i^0 = 0 \quad \forall i \quad (10)$$

$$x_j^t \in \mathbb{Z}_+ \quad \forall j, t \quad (11)$$

$$E^t, T^t, q^t, start^t \in \mathbb{Z}_+ \quad \forall t \quad (12)$$

$$S_j^t \in \{0, 1\} \quad \forall t, j \quad (13)$$

$$e_i^t \in \mathbb{Z}_+ \quad \forall t, i \quad (14)$$

O objetivo deste problema é, portanto, a minimização da função objetivo (1), composta pela soma ponderada de cinco parcelas: (i) o custo total dos objetos cortados (primeira parcela); (ii) penalização por atrasos nas entregas (segunda parcela); (iii) penalização por antecipação nas entregas (terceira parcela), (iv) penalização por estocar itens (quarta parcela) e (v) os custos de *setup* (quinta parcela). O conjunto de restrições (2) garante que a demanda do período deve ser satisfeita usando itens cortados no período ou itens em estoque, além de contabilizar o número de itens que serão armazenados no final do período. O conjunto de restrições (3) garante que, em um determinado período, um determinado padrão de corte só pode ser cortado se houver uma mudança de padrão na máquina. O conjunto de restrições (4) calcula o início do processo de corte, permitindo tempo ocioso na máquina. O conjunto de restrições (5) calcula o tempo a ser gasto na produção de pedidos em cada um dos períodos, adicionando os tempos de corte aos tempos de *setup*. O conjunto de restrições (6) garante o cálculo correto dos tempos de atraso e de antecipação. As restrições (7) e (8) representam as condições iniciais, garantindo que não há atraso ou adiantamento de pedidos no início do processo de corte. A restrição (9) determina a condição de data de entrega inicial para $t = 0$ e as restrições (10) inicializam o estoque de itens. As restrições (11)–(14) representam as condições de não negatividade e integralidade das variáveis de decisão.

3. Método de Solução

A solução do PMCE-DE-SI-E é abordada em três fases distintas, cada uma com um papel crucial na obtenção de uma solução: geração de colunas, arredondamento e *local branching*. Este é o mesmo procedimento usando por Oliveira et al. [2025].

Ao combinar as três fases, a estratégia permite que o problema de corte de estoque seja resolvido de maneira eficiente, com a geração de soluções inteiras. Essa abordagem progressiva garante uma exploração do espaço de soluções, reduzindo o custo computacional e aumentando a viabilidade prática das soluções encontradas.

3.1. Fase 1 (Geração de Colunas)

A primeira fase consiste na resolução do problema relaxado do PMCE-DE-SI-E, utilizando a abordagem de geração de colunas, conforme proposto por Gilmore e Gomory [1961], uma vez que, devido ao grande número de padrões de corte possíveis para o PCE, torna-se impraticável encontrar todos os possíveis padrões de corte para ter uma solução ótima. Para lidar com essa complexidade, uma abordagem amplamente adotada consiste em resolver o relaxamento linear do problema utilizando o método Simplex com geração de colunas, conforme proposto por Gilmore e Gomory [1961]. Essa abordagem permite reduzir significativamente o número de padrões de corte a serem considerados na solução. O objetivo aqui é encontrar uma solução inicial que considere apenas os padrões de corte viáveis para cada período. A técnica de geração de colunas introduz um problema auxiliar, conhecido como problema de precificação, que visa identificar e adicionar novas colunas que possam melhorar a solução corrente. Essa fase reduz significativamente o número de padrões de corte a serem considerados, tornando o processo mais eficiente. No final desse processo, encontramos uma solução inicial para o problema relaxado.

3.2. Fase 2 (Arredondamento)

Uma vez que uma solução relaxada tenha sido obtida através da geração de colunas, a próxima etapa é transformar essa solução em uma solução inteira. Isso pode ser feito através do procedimento de arredondamento para cima, em que as variáveis contínuas são arredondadas para os valores inteiros superiores mais próximos. Sejam x^* e S^* os valores das variáveis x e S na solução ótima do problema relaxado, em que pelo menos um componente de x^* é fracionário. Para obter valores inteiros $\tilde{x}$ e $\tilde{S}$ para as variáveis x e S , adotamos um procedimento de arredondamento. Neste procedimento, componentes não inteiros da variável de corte x_j^t são arredondados para cima, visando garantir que as demandas sejam atendidas. Além disso, a variável binária de *setup* S_j^t é

arredondada para o valor 1 se o padrão de corte j for selecionado para ser cortado no período t . Embora essa técnica seja rápida e simples, ela pode resultar em soluções inviáveis, que podem ultrapassar a largura do objeto, já que arredondamos sua solução para cima.

3.3. Fase 3 (Local Branching)

Para melhorar a solução encontrada na fase anterior (arredondamento), a terceira fase envolve o uso da matheurística *Local Branching*. Este procedimento permite explorar soluções vizinhas da solução inicial, buscando por alternativas melhores, dentro de uma região limitada do espaço de soluções.

De acordo com Fischetti e Lodi [2003], uma restrição de *Local Branching* pode ser definida como:

$$\Delta(x, x^t) := \sum_{t=1}^M \sum_{j \in J_l(x^t)} \mu_j^t (x_j^t - l_j^t) + \sum_{t=1}^M \sum_{j \in J_u(x^t)} \mu_j^t (u_j^t - x_j^t) \leq k, \quad (15)$$

em que x representa a variável do problema e x^t o valor da variável na iteração t . O valor de μ_j^t , como sugerido por Fischetti e Lodi [2003], é dado por:

$$\mu_j^t = \frac{1}{u_j^t - l_j^t} \quad \forall j, t. \quad (16)$$

Os valores l_j^t e u_j^t para os limites inferior e superior, respectivamente, foram determinados como o valor 0 e o maior valor de x_j^t encontrado na fase de solução inicial, respectivamente.

Portanto, dado x_j^t , o espaço de solução pode ser particionado por meio da disjunção:

$$\Delta(x, x^t) \leq k \quad (17)$$

or

$$\Delta(x, x^t) \geq k + 1. \quad (18)$$

Essa fase de busca ajuda a refinar a solução, garantindo que ela seja mais próxima da ótima. O algoritmo consiste em inicialmente explorar a região definida pela expressão (17). Após encontrar uma nova solução de referência x^{t+1} , a busca continua a partir da região definida pela expressão (18). A busca continua, iterativamente, seguindo a estratégia de explorar as duas regiões usando a nova solução como base.

4. Experimentos Computacionais

Foram realizados diversos experimentos utilizando instâncias de diferentes tamanhos e complexidades, a fim de testar o método em uma variedade de cenários. Além disso, os resultados obtidos foram comparados com soluções exatas, quando possível, bem como foram discutidas as vantagens e limitações da abordagem em termos de precisão, tempo de execução e capacidade de lidar com a escalabilidade do problema. Esta seção está organizada como segue. A Subseção 4.1 descreve os dados e parâmetros usados para a definição do problema; a Subseção 4.2 apresenta os resultados encontrados e os analisa.

4.1. Geração de dados

Os testes computacionais foram realizados em um computador com sistema operacional Linux-Ubuntu, processador Intel Core i7 - 7ª geração, 8 GB de RAM e 2 TB de disco rígido. Os modelos foram implementados em C++ usando o pacote Concert Technology e resolvidos utilizando o *solver* CPLEX (IBM [2023]) 12.7.1, com um limite de tempo de execução de 3600 segundos.

Foi usado um gerador de instâncias baseado no gerador CUTGEN 1, que se concentra na geração de problemas de corte unidimensional, proposto por Gau e Wäscher [1995]. Os valores e intervalos que foram atribuídos aos parâmetros descritos no gerador de instâncias foram:

- (i) Comprimento do objeto: $L = 1000$;
- (ii) Número de tipos de itens: $n = 20$;
- (iii) Número de períodos: $M = 3$ e 6 ;
- (iv) A demanda pelos itens foi escolhida aleatoriamente no intervalo $[0, 125]$ (com uma distribuição uniforme);
- (v) O comprimento dos itens foi escolhido aleatoriamente no intervalo $Peq = [0, 01L, 0, 02L]$; $Med = [0, 02L, 0, 8L]$; e $Gr = [0, 2L, 0, 8L]$ (com uma distribuição uniforme);
- (vi) As datas de entrega foram geradas de acordo com uma distribuição uniforme:

$$d^t \in U[0, 8 CAP^t, 1, 0 CAP^t]. \quad (19)$$

O valor CAP^t é definido como a capacidade de corte da máquina em um determinado período, dado por:

$$CAP^t = \frac{\sum_{i=1}^n l_i b_i^t}{L}; \quad (20)$$

- (vii) penalizações: β^t , γ^t e f_j^t são todos iguais 1 ;
- (viii) custos de estocagem $ce_i^t = 1$ e $ce_i^t = 0$.

A Tabela 3 apresenta uma classificação das instâncias do problema em diferentes classes, com base em três características principais: o tipo de item e o número de períodos considerados.

Tabela 3: Classe de Instâncias

Classes	Itens	Períodos
C1	P	3
C2	M	3
C3	G	3
C4	P	6
C5	M	6
C6	G	6

4.2. Resultados computacionais

Esta seção detalha os resultados computacionais obtidos, apresentando o custo total, a viabilidade das soluções e o tempo computacional necessário para encontrar as soluções inteiras. Além disso, a seção também apresenta uma análise das soluções obtidas e as implicações práticas do uso das técnicas propostas no contexto real do problema de corte de estoque.

A análise dos resultados apresentados na Tabela 4, que detalha os valores de cada termo da função objetivo, permite uma compreensão mais aprofundada da composição do custo total nas diferentes classes de instâncias (C1 a C6). Os termos considerados são: objetos cortados, estoque, *setup*, atraso e adiantamento. Esses componentes refletem diferentes aspectos operacionais e logísticos do problema, e sua variação entre as classes revela como as características das instâncias (número de períodos e tamanho dos itens) influenciam o comportamento da solução.

Tabela 4: Resultados para cada um dos termos da função objetivo.

Custo	Classe	Termos da Função Objetivo				
		Objetos	Estoque	Setup	Atraso	Adiantamento
$(ce_i^t = 1)$	C1	449,8	403,2	56,6	23,7	1,2
	C2	2039,9	158,9	57,1	42,7	0,1
	C3	2101,6	83,5	58,4	98,1	0
	Média	1530,43	215,2	57,36	54,83	0,43
$(ce_i^t = 0)$	C1	448,8	7055,8	43,5	16,7	0
	C2	2094,9	7102,5	40	33,6	0
	C3	2075	7186,9	39,6	67,5	0
	Média	1539,56	7115,06	41,03	579,36	0
$(ce_i^t = 1)$	C4	842	4480,3	112,6	36,9	3,5
	C5	3995,9	1553,8	112,9	107,1	0
	C6	4126,8	234,4	112,7	155,9	1,3
	Média	2988,23	2089,5	112,73	99,96	1,6
$(ce_i^t = 0)$	C4	832,9	32509,4	71,1	51	0
	C5	4018,9	21247,6	69,4	33,8	0
	C6	4076	20272,6	68,5	86,7	0
	Média	2975,93	24676,53	69,66	57,16	0

Nas classes C1 a C3, que possuem 3 períodos, observa-se que o termo “objetos” representa a maior parcela da função objetivo, especialmente em C3 (2101,6), refletindo um alto volume de produção. O estoque nestas classes decresce à medida que o tamanho dos itens aumenta: C1 apresenta 403,2, enquanto C3 cai para 83,5. Esse comportamento sugere que a estratégia adotada nas soluções opta por estoques menores à medida que os itens crescem. Já o *setup* se mantém relativamente estável em torno de 56 a 58, sem grande variação entre as classes. Em relação ao atraso, C3 se destaca negativamente com 98,1, indicando maior dificuldade de atendimento dentro dos prazos. O adiantamento é praticamente irrelevante em todas essas classes, mantendo valores próximos de zero.

Nas classes C4 a C6, que envolvem 6 períodos, há um aumento expressivo no estoque, principalmente na classe C4 (4480,3), o maior valor da tabela. Isso reflete o impacto direto do aumento no número de períodos, que favorece a formação de estoques mais altos. Em contraste, a classe C6 apresenta um valor relativamente baixo para estoque (234,4), o que pode estar relacionado a uma estratégia mais ajustada de produção para itens grandes, evitando acúmulo. O termo objetos também se destaca nessas classes, especialmente em C6 (4126,8), mostrando que a demanda ou volume de produção nessas instâncias é significativamente maior. O *setup* praticamente dobra nas classes com 6 períodos em comparação às de 3 períodos, com valores acima de 112, evidenciando o impacto do maior número de períodos e possíveis trocas de configuração mais frequentes. O atraso também cresce em C5 e C6, atingindo 107,1 e 155,9, respectivamente, sugerindo maior dificuldade de atender a demanda a tempo quando os itens são médios ou grandes. O adiantamento volta a ter valores muito baixos em comparação aos outros termos.

Para mais uma comparação, verificamos o que acontece com os resultados quando os custos de estoques são iguais a zero na função objetivo, como assumido por Oliveira et al. [2025]. Neste caso, esse termo (estoque) aumenta significativamente em todas as classes e os outros termos diminuem. Assim, não ocorreu adiantamento em nenhuma das instâncias.

A Figura 2 apresenta a quantidade de colunas que foram geradas na fase 1 (geração de colunas). A análise dos dados da fase de geração de colunas revela que tanto o número de períodos quanto o tamanho dos itens têm um impacto significativo na quantidade de colunas geradas. Itens pequenos, representados pelas classes C1 e C4, não geraram colunas. Já as classes com itens médios

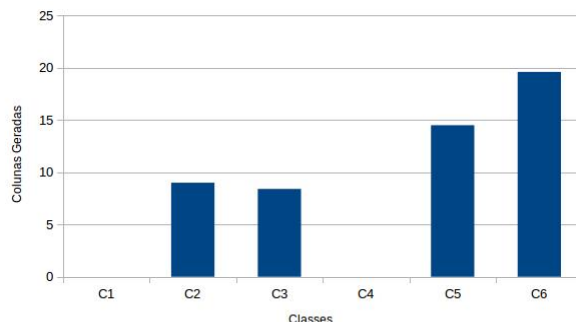


Figura 2: Número de colunas geradas na fase 1 (Geração de Colunas) para os custos de estoque 1.

Tabela 5: Resultados entre a abordagem proposta e o solver CPLEX

Custo	Classe	Fase 1	Fase 2	Fase 3: Local Branching		CPLEX	
		Função Objetivo	Função Objetivo	Função Objetivo	Tempo	Função Objetivo	Tempo
$(ce_i^t = 1)$	C1	446,21	1416	934,5	0,38	934,5	8,53
	C2	2120,72	2452	2298,7	248,06	2298,7	282,61
	C3	2174,16	2445	2341,6	4,73	2341,6	56,47
	Média	1580,36	2104,33	1858,26	84,39	1858,26	115,87
$(ce_i^t = 0)$	C1	444,41	577	509	290,13	509	247,35
	C2	2112,69	2227,9	2168,5	754,77	2168,5	263,51
	C3	2124,54	2226,9	2182,1	181,21	2182,1	176,51
	Média	1560,54	1677,26	1619,86	408,70	1619,86	229,12
$(ce_i^t = 1)$	C4	839,83	4046	1821	116,31	1821	18,10
	C5	4164,30	5233	4501,9	236,28	4502,1	274,65
	C6	4332,60	5310	4639,1	431,27	4639,1	456,98
	Média	3112,24	4863	3654	261,28	3654	249,91
$(ce_i^t = 0)$	C4	835,71	1170,7	933,1	925,84	933,1	1194,42
	C5	4026,63	4349,9	4122,1	848,91	4122,3	1030,59
	C6	4127,66	4439,5	4231,2	880,02	4231,2	772,91
	Média	2996,66	3320,03	3095,46	884,92	3095,53	999,30

e grandes mostram uma tendência crescente de geração de colunas à medida que o número de períodos aumenta, especialmente nas classes C5 e C6, que possuem 6 períodos.

A Tabela 5 apresenta uma comparação detalhada entre a abordagem proposta, que inclui três fases (Fase 1, Fase 2 e Fase 3 com Local Branching), e o solver CPLEX, considerando tanto os valores da função objetivo quanto os tempos de execução. Essa estrutura permite avaliar não apenas a qualidade da solução final, mas também a evolução ao longo das fases e a eficiência computacional de cada abordagem para as diferentes classes de instância (C1 a C6).

Nas classes com 3 períodos (C1 a C3), observa-se uma progressão natural entre as fases. A Fase 1 apresenta os menores valores da função objetivo (por exemplo, 446,21 em C1), uma vez que ainda se trata de uma solução inicial e aproximada. Na Fase 2, esses valores aumentam substancialmente (1416 em C1), o que é esperado. Na Fase 3, com o uso do Local Branching, os valores da função objetivo melhoram significativamente, convergindo para os mesmos valores encontrados pelo CPLEX. No que se refere ao tempo, a abordagem com Local Branching é mais eficiente em C1 (0,38 segundos) e C3 (4,73 segundos) do que o CPLEX (8,53 e 56,47 segundos, respectivamente). Já em C2, os tempos são semelhantes, com leve vantagem para o CPLEX (282,61 segundos contra 248,06).

Nas classes com 6 períodos (C4 a C6), o comportamento da função objetivo ao longo

das fases segue o mesmo padrão observado nas classes anteriores: a Fase 1 apresenta os menores valores, a Fase 2 expande o escopo do problema e aumenta os custos e a Fase 3 busca a solução refinada. Em C4, por exemplo, o valor da função objetivo salta de 839,837 na Fase 1 para 4046 na Fase 2, e é refinado para 1821 na Fase 3, valor idêntico ao do CPLEX. No entanto, o tempo de execução do Local Branching em C4 (116,31 segundos) é consideravelmente maior que o do CPLEX (18,10 segundos), o que mostra que a abordagem proposta nem sempre é superior em tempo. Já nas classes C5 e C6, os tempos do Local Branching e do CPLEX são bastante próximos (com pequena vantagem para o CPLEX) e os valores da função objetivo são praticamente iguais. Apenas na classe C5 a fase 3 foi melhor que o CPLEX, uma vez que o *solver* atingiu o tempo limite em uma instância.

De forma geral, a análise mostra que a abordagem proposta é eficaz em encontrar soluções de mesma qualidade que o CPLEX, com vantagens de tempo em instâncias menores, como C1 e C3. O fato de o Local Branching alcançar os mesmos valores da função objetivo indica sua capacidade de explorar bem o espaço de soluções, enquanto a estrutura em fases permite um controle sobre a evolução da solução. Assim, a abordagem proposta demonstra ser uma alternativa competitiva, principalmente quando se busca um equilíbrio entre qualidade da solução e tempo computacional.

Quando o custo de estoque foi zero, o tempo aumentou tanto para o Local Branching quanto para o CPLEX. Os valores da função objetivo para cada uma das fases diminuiu, porém, devemos lembrar que estamos usando estoque de itens e não estamos contabilizando eles.

5. Considerações finais e propostas futuras

Com base nos resultados apresentados e analisados ao longo deste trabalho, conclui-se que a abordagem proposta, estruturada em múltiplas fases e complementada pelo uso da metaheurística Local Branching, mostrou-se eficaz e competitiva frente ao *solver* comercial CPLEX. Em todas as classes de instância, a abordagem foi capaz de alcançar valores de função objetivo iguais ou melhores aos obtidos pelo CPLEX, validando a qualidade das soluções geradas.

A análise dos componentes da função objetivo permitiu compreender melhor os fatores que mais impactam os custos totais em diferentes configurações do problema. Ficou claro que o número de períodos e o tamanho dos itens influenciam diretamente tanto a geração de colunas quanto a complexidade da solução final, com destaque para o aumento significativo do estoque e do atraso em instâncias mais longas e com itens maiores.

Como sugestão para trabalhos futuros, pode-se ter a aplicação e avaliação da abordagem proposta em outros variantes do problema, incluindo a variação dos valores dos parâmetros analisados, a consideração de cenários com múltiplos objetivos, janelas de tempo mais restritivas e demandas incertas. Essas extensões permitiriam testar a adaptabilidade do método em contextos mais realistas e desafiadores. Por fim, investigações envolvendo o *setup* dependente da sequência podem ser analisadas para verificar o impacto e a diferença que ocorre entre a dependência ou independência da sequência no corte.

Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Fundação de Apoio à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) pelo apoio ao desenvolvimento deste trabalho.

Referências

Araujo, S. A., Poldi, K. C., e Smith, J. (2014). A genetic algorithm for the one-dimensional cutting stock problem with setups. *Pesquisa Operacional*, 34(2):165–187.

- Arbib, C. e Marinelli, F. (2014). On cutting stock with due dates. *Omega*, 46:11–20.
- Arbib, C. e Marinelli, F. (2017). Maximum lateness minimization in one-dimensional bin packing. *Omega*, 68:76–84.
- Braga, N., Alves, C., Macedo, R., e Valério de Carvalho, J. M. (2016). Combined cutting stock and scheduling: a matheuristic approach. *International Journal of Innovative Computing and Applications*, 7(3):135–146.
- Fischetti, M. e Lodi, A. (2003). Local branching. *Mathematical programming*, 98(1-3):23–47.
- Foerster, H. e Wascher, G. (2000). Pattern reduction in one-dimensional cutting stock problems. *International journal of production research*, 38(7):1657–1676.
- Gau, T. e Wäscher, G. (1995). CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, 84(3):572–579.
- Gilmore, P. C. e Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859.
- Gilmore, P. C. e Gomory, R. E. (1963). A linear programming approach to the cutting stock problem – Part II. *Operations research*, 11(6):863–888.
- Gilmore, P. C. e Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1):94–120.
- Hendry, L. C., Fok, K. K., e Shek, K. W. (1996). A cutting stock and scheduling problem in the copper industry. *Journal of the Operational Research Society*, 47(1):38–47. ISSN 0160-5682.
- IBM. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*. IBM Corporation, Armonk, NY, 2023. URL <https://www.ibm.com/docs/en/icos>. Version 22.1.2.
- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.
- Li, S. (1996). Multi-job cutting stock problem with due dates and release dates. *Journal of the Operational Research Society*, 47(4):490–510.
- Oliveira, E. d. A. S., Wanner, E., Martins de Sá, E., e de Souza, S. R. (2025). A local branching-based solution for the multi-period cutting stock problem with tardiness, earliness, and setup costs. *Journal of Heuristics*, 31(1):16.
- Reinertsen, H. e Vossen, T. W. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research*, 201(3):701–711.
- Umetani, S., Yagiura, M., e Ibaraki, T. (2003). An lp-based local search to the one dimensional cutting stock problem using a given number of cutting patterns. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 86(5):1093–1102.
- Umetani, S., Yagiura, M., e Ibaraki, T. (2006). One-dimensional cutting stock problem with a given number of setups: a hybrid approach of metaheuristics and linear programming. *Journal of Mathematical Modelling and Algorithms*, 5:43–64.
- Vanderbeck, F. (2000). Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 48(6):915–926.