CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Modelagem Matemática e Computacional

O Problema de Corte de Estoque Unidimensional com Datas de Entrega e Setup

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, como parte dos requisitos necessários para a obtenção do título de Doutora em Modelagem Matemática e Computacional.

Área de Concentração: Modelagem Matemática e Computacional
Linha de Pesquisa: Sistemas Inteligentes

Aluno : Elisama de Araújo Silva Oliveira

Orientadora : Profa. Dra. Elisangela Martins de Sá (CEFET-MG)

Co-orientadora : Profa. Dra. Elizabeth Wanner Fialho (CEFET-MG)

Co-orientador : Prof. Dr. Sérgio Ricardo de Souza (CEFET-MG)

Belo Horizonte - MG
Maio - 2025

Oliveira, Elisama de Araújo Silva
O48p O problema de corte de estoque unidimensional com datas de entrega e setup / Elisama de Araujo Silva Oliveira. – 2025.
146 f.

Tese (doutorado) apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientadora: Elisangela Martins de Sá.
Coorientadora: Elizabeth Warner Fialho.
Coorientador: Sérgio Ricardo de Souza.
Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Problema de corte de estoque – Teses. 2. Programação (Matheurística) – Teses. 3. Programação de produção – Teses. 4. Espaço de configuração – Teses. 5. Algoritmos – Teses. 6. Processos de ramificação – Teses. I. Sá, Elisangela Martins de. II. Fialho, Elizabeth Warner. III. Souza, Sérgio Ricardo de. IV. Centro Federal de Educação Tecnológica de Minas Gerais. V. Título.

CDD 519.622

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6ª 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"PROBLEMA DE CORTE DE ESTOQUE UNIDIMENSIONAL COM DATAS DE ENTREGA E SETUP".

Tese de Doutorado apresentada por **Elisama de Araújo Silva Oliveira,** em 30 de maio de 2025, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

gov.br Documento assinado digitalmente
ELISANGELA MARTINS DE SA
Data: 06/06/2025 17:52:02-0300
Verifique em https://validar.iti.gov.br

Profª. Drª. Elisangela Martins de Sá (Orientadora)
Centro Federal de Educação Tecnológica de Minas Gerais

gov.br Documento assinado digitalmente
ELIZABETH FIALHO WANNER
Data: 06/06/2025 10:50:21-0300
Verifique em https://validar.iti.gov.br

Profª. Drª. Elizabeth Fialho Wanner (Coorientadora)
Centro Federal de Educação Tecnológica de Minas Gerais

gov.br Documento assinado digitalmente
SERGIO RICARDO DE SOUZA
Data: 06/06/2025 16:26:05-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Sérgio Ricardo de Souza (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

gov.br Documento assinado digitalmente
SILVIO ALEXANDRE DE ARAUJO
Data: 04/06/2025 12:14:22-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Silvio Alexandre de Araújo
Universidade Estadual Paulista

gov.br Documento assinado digitalmente
KELLY CRISTINA POLDI
Data: 04/06/2025 13:49:13-0300
Verifique em https://validar.iti.gov.br

Profª. Drª. Kelly Cristina Poldi
Universidade Estadual de Campinas

gov.br Documento assinado digitalmente
GUSTAVO CAMPOS MENEZES
Data: 05/06/2025 08:39:58-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Gustavo Campos Menezes
Centro Federal de Educação Tecnológica de Minas Gerais

gov.br Documento assinado digitalmente
MARCONE JAMILSON FREITAS SOUZA
Data: 04/06/2025 17:49:45-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Marcone Jamilson Freitas Souza
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida à impressão,

Prof. Dr. Alisson Marques da Silva
Coordenador do Programa de Pós-Graduação Stricto Sensu em Modelagem Matemática e Computacional

Ao meu amado esposo, Jônathas Douglas, meu maior incentivador e companheiro em todos os momentos. Você é a prova viva de que "o amor tudo sofre, tudo crê, tudo espera, tudo suporta"(1 Coríntios 13:7). Esta conquista é tanto minha quanto sua. E a minha amada e querida filha, Sophia Ellen, minha luz e minha maior inspiração.

Agradecimentos

Em primeiro lugar, agradeço a Deus, que me sustentou em cada passo desta jornada. Foi Nele que encontrei forças nos momentos de dificuldade, paciência nos momentos de incerteza e gratidão em cada conquista. Sem Sua graça e amor, nada disso seria possível. Agradeço, em especial, ao meu amado esposo, Jônathas Douglas, meu maior incentivador. Se não fosse por ele, eu sequer teria pensado em seguir o caminho do Doutorado. Sua fé inabalável em mim, seu amor, paciência e apoio constante foram essenciais em cada etapa desta jornada. Nos momentos mais desafiadores, foi ele quem me fortaleceu e me lembrou do meu potencial. Esta conquista é tanto minha quanto dele. À minha filha, Sophia Ellen, a razão da minha maior felicidade, agradeço por ser minha luz e minha inspiração diária. Seu sorriso e carinho foram meu combustível para seguir adiante, mesmo nos momentos mais difíceis. Tudo o que faço é também por você e para você. Aos meus orientadores, expresso minha imensa gratidão pelo suporte acadêmico, pelas valiosas orientações e pelo incentivo contínuo. Mais do que mestres, foram guias nessa caminhada, compartilhando conhecimento, oferecendo direcionamento e me desafiando a alcançar sempre o melhor. Sua paciência, dedicação e generosidade intelectual foram fundamentais para o desenvolvimento deste trabalho. Cada conselho, cada revisão cuidadosa e cada palavra de incentivo fizeram a diferença e moldaram não apenas esta pesquisa, mas também meu crescimento acadêmico e profissional. Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) pelo apoio financeiro e institucional que tornou possível a realização desta pesquisa. Por fim, sou grata a todos que, direta ou indiretamente, contribuíram para esta trajetória, seja por meio de apoio, palavras de encorajamento ou simples gestos de incentivo. Cada pessoa que esteve ao meu lado fez parte desta conquista, e levo comigo toda a gratidão por esse apoio inestimável.

"Porque eu, o Senhor teu Deus, te tomo pela tua mão direita; e te digo: Não temas, eu te ajudo." **Isaías 41:13**, Bíblia Sagrada.

Resumo

Esta tese aborda o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega (1D-PCE-DE), uma extensão do problema clássico de corte de estoque que incorpora restrições de tempo associadas a pedidos. Essa adaptação torna o problema mais alinhado aos desafios industriais reais, em que o atendimento oportuno da demanda é crucial. Inicialmente, formulamos o 1D-PCE-DE e desenvolvemos um método de solução trifásico, avaliando sua eficácia por meio de testes computacionais. A partir dessa análise inicial, o modelo foi expandido para considerar *setup* independente da sequência, resultando no 1D-PCE-DE-SI. Incluir *setup* na modelagem é essencial para capturar os custos e tempos envolvidos na reconfiguração da máquina ao alternar entre diferentes tipos de corte ou materiais. Essa consideração impacta diretamente a eficiência da produção, os custos operacionais e os níveis de estoque, fornecendo uma solução mais realista aplicável ao ambiente industrial. Além disso, propomos uma modelagem ainda mais abrangente, 1D-PCE-DE-SD, que incorpora *setup* dependente da sequência. Nesse caso, os custos e tempos de configuração não são fixos, mas variam de acordo com a ordem dos padrões de cortes. Essa característica torna o problema significativamente mais complexo, pois exige a consideração do sequenciamento da produção, fator crítico em setores industriais nos quais a troca entre determinados materiais ou formatos requer ajustes específicos. Para avaliar a eficácia dos modelos propostos, comparamos seus desempenhos, analisando-os em relação a variáveis críticas, como atraso, *lead time*, estoque, *setup* e custo total. Os resultados de testes computacionais foram obtidos utilizando o solver CPLEX, garantindo a validação da qualidade das soluções obtidas. Para resolução dos problemas propostos, foram desenvolvidos algoritmos matheurísticos para cada versão do problema. Especificamente, os algoritmos empregados foram baseados nas matheurísticas *Local Branching*, *Relax-and-Fix*, *Fix-and-Optimize* e de Arredondamento, cada um com características específicas para otimizar o desempenho, dependendo das variáveis e restrições de cada problema. O desempenho desses algoritmos foi comparado com o do *solver* CPLEX, observando-se não apenas a qualidade das soluções obtidas, mas também o tempo de execução. No geral, os resultados indicaram que os algoritmos matheurísticos, como o *Local Branching*, apresentaram uma capacidade superior de escapar de ótimos locais, enquanto o *Relax-and-Fix* mostrou eficácia na construção de soluções viáveis. Os resultados desta pesquisa não apenas demonstram a relevância dos modelos propostos, mas também as suas possibilidades de aplicação.

Palavras-chave Problema de Corte de Estoque. Matheurística. Geração de Colunas. Datas de entrega. *Setup*. *Relax-and-Fix*. *Local Branching*.

Abstract

This thesis addresses the One-dimensional Cutting Stock Problem with Due Dates (1D-CSP-DE), an extension of the classical inventory cutting problem that incorporates time constraints associated with orders. This adaptation makes the problem more aligned with real industrial challenges, where timely demand fulfillment is crucial. Initially, we formulated CSP-DE and developed a three-phase solution method, evaluating its effectiveness through computational tests. From this initial analysis, the model was expanded to consider the sequence-independent cutting setup, resulting in CSP-DE-SI. Including setup in the modeling is essential to capture the costs and times involved in machine reconfiguration when switching between different types of cutting or materials. This consideration directly impacts production efficiency, operating costs, and inventory levels, providing a more realistic solution applicable to the industrial environment. In addition, we propose an even more comprehensive modeling, CSP-DE-SD, which incorporates the sequence-dependent setup. In this case, setup costs and times are not fixed, but vary according to the order of the cuts. This feature makes the problem significantly more complex, as it requires consideration of production sequencing, a critical factor in industrial sectors where switching between certain materials or formats requires specific adjustments. To evaluate the effectiveness of the proposed models, we analyzed their performance in relation to critical variables, such as delay, lead time, inventory, setup, and total cost. The results of computational tests were obtained using the CPLEX solver, guaranteeing validation of the quality of the solutions obtained. To solve the proposed problems, matheuristic algorithms were developed for each version of the problem. Specifically, these algorithms used were based on the Local Branching, Relax-and-Fix, Fix-and-Optimize, and Rounding matheuristics, each with specific characteristics to optimize performance, depending on the variables and constraints of each problem. The performance of these algorithms was compared with that of solver CPLEX, observing not only the quality of the solutions obtained, but also the runtime. Overall, the results indicated that the matheuristic algorithms, such as Local Branching, showed a superior ability to escape from local optima, while Relax-and-Fix showed effectiveness in constructing feasible solutions. The results of this research not only demonstrate the relevance of the proposed models but also their applicability.

Keywords: Cutting-Stock problem. Matheuristics. Column generation. Due date. Setup. Relax-and-Fix. Local Branching.

Sumário

Lista de Tabelas xiii
Lista de Figuras xv
Lista de Algoritmos xvi

1 Introdução **1**
1.1 Apresentação Geral 1
1.2 Objetivos Gerais e Específicos 4
1.2.1 Objetivos Gerais 4
1.2.2 Objetivos Específicos 4
1.3 Justificativa 5
1.4 Publicações Derivadas desta pesquisa 5
1.4.1 Artigos publicados em anais de congresso 5
1.4.2 Artigos em periódicos (publicados e aceitos para publicação) . 6
1.5 Organização da tese 6

2 Fundamentação Teórica **7**
2.1 Problema de Corte de Estoque Unidimensional 7
2.1.1 Modelo clássico do 1D-PCE 9
2.2 Tipologias para o Problema de Corte e Empacotamento 10
2.2.1 Tipologia de Wäscher et al. (2007) 10
2.3 Revisão Bibliográfica para o 1D-PCE 13
2.4 Métodos de Solução para o 1D-PCE 16
2.4.1 Método Simplex com Geração de Colunas 16
2.4.2 *Relax-and-Fix* 18
2.4.3 *Fix-and-Optimize* 19
2.4.4 Matheurística *Local Branching*(LB) 19
2.4.5 Comparação entre as matheurísticas apresentadas 23

3 Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega **25**
3.1 Revisão dos Problemas de Corte de Estoque Unidimensional com Datas de Entrega 25
3.2 Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega 26
3.2.1 Contribuições relativas ao 1D-PMCE-DE 27
3.2.2 Descrição geral do 1D-PMCE-DE 27
3.2.3 Modelagem Proposta 28
3.2.4 Exemplo numérico 28

3.3 Abordagem proposta para solução do 1D-PMCE-DE 30
3.3.1 Fase 1 – Encontrando soluções iniciais: Método de Geração de Colunas 31
3.3.2 Fase 2 – Obtenção de soluções inteiras: heurística *Relax-and-Fix* 32
3.3.3 Fase 3: Melhorando a solução encontrada – Matheurística *Local Branching* 34
3.4 Experimentos Computacionais para o 1D-PMCE-DE 36
3.4.1 Resultados para o primeiro conjunto de instâncias 38
3.4.2 Resultados para o segundo conjunto de instâncias 40
3.5 Conclusões e discussões parciais 42

4 Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Independente da Sequência 44
4.1 Revisão de Problemas de Corte de Estoque com *Setup* 44
4.1.1 Análise comparativa: trabalhos de referência e o presente estudo 46
4.2 Modelagem proposta 47
4.2.1 Descrição do problema 47
4.2.2 Formulações matemáticas para o 1D-PMCE-DE-SI 49
4.3 Abordagem de solução proposta 51
4.3.1 Fase 1: Método de geração de colunas 52
4.3.2 Fase 2: Encontrando uma solução inicial 56
4.3.3 Fase 3: Melhorando a solução inicial 58
4.4 Experimentos Computacionais para o 1D-PMCE-DE-SI 60
4.4.1 Geração das Instâncias 60
4.4.2 Ajuste de parâmetros usando IRACE 63
4.4.3 Escolha de heurísticas para as Fases 2 e 3 64
4.4.4 Resultados dos Testes Computacionais 65
4.4.5 Resultados para diferentes valores de custo 76
4.5 Conclusões e discussões parciais 82

5 Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Dependente da Sequência 84
5.1 Descrição do Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Dependente da Sequência 84
5.2 Revisão da literatura para o 1D-PMCE-DE-SD 86
5.3 Modelagem matemática do 1D-PMCE-DE-SD 88
5.4 Método de Solução para o 1D-PMCE-DE-SD 92
5.4.1 Fase 1: Processo de Geração de Padrões de Corte com a heurística *First-Fit-Decreasing*-Modificada (FFD-M) 93
5.4.2 Fase 2: Geração da Solução inicial 95
5.4.3 Fase 3: Melhorando a solução inicial com a matheurística *Local Branching* 97
5.5 Geração de Instâncias para o 1D-PMCE-DE-SD 97
5.6 Experimentos Computacionais para o 1D-PMCE-DE-SD 100
5.6.1 Solução para as classes de instâncias com 3 períodos de tempo 101
5.6.2 Solução para as classes de instâncias com 6 períodos de tempo 105
5.7 Conclusões e Discussões Parciais 110

6 Comparação entre as Modelagens Propostas **112**
6.1 Comparação entre as modelagens com *setup* 113
6.1.1 Exemplo Numérico 113
6.1.2 Análise da influência dos itens P, M e G nas modelagens propostas 116
6.2 Dados para a comparação entre as três modelagens propostas: análise da inclusão de termos 118
6.3 Resultados Computacionais para as instâncias para custos iguais a 1 . 119
6.4 Resultados Computacionais para as instâncias para custos variados . 122
6.5 Conclusões Parciais 126

7 Conclusões finais e Direções futuras **128**
7.1 Conclusões finais 128
7.2 Trabalhos futuros 129

A Formulações da literatura para o PCE com data de entrega **131**
A.1 Proposta de modelagem de Reinertsen e Vossen (2010) 131
A.2 Proposta de Arbib e Marinelli (2014) 133
A.3 Proposta de Arbib e Marinelli (2017) 136

Referências Bibliográficas **138**

Lista de Siglas

1BP - *Bin Packing* Unidimensional
BP - *Bin Packing*
C & E - Problema de Corte e Empacotamento
CSP - *Cutting Stock Problem*
GC - Geração de Colunas
IRUP - *Integer Round-Up Property*
KP - *Knapsack Problem*
LB - *Local Branching*
MIP - *Mixed Integer Program*
MIRUP - *Modified Integer Round-Up Property*
ODP - *Open Dimension Problem*
PI - Programação Inteira
PCE - Problema de Corte de Estoque
PCE-DE - Problema de Corte de Estoque com Data de Entrega
PCE-DE-SI - Problema de Corte de Estoque com Data de Entrega e *Setup* Independente da Sequência
PCE-DE-SD - Problema de Corte de Estoque com Data de Entrega e *Setup* Dependente da Sequência
PDL - Problema de Dimensionamento de Lotes
PPLI - Problema de Programação Linear Inteira
PL - Problema Linear
RF - *Relax-and-Fix*
SSSCSP - *Single Stock Size Cutting Stock Problem*

Lista de Tabelas

2.1 Uma Visão Geral Comparativa das Características dos Problemas Abordados. 15
2.2 Comparação entre *Relax-and-Fix*, *Fix-and-Optimize* e *Local Branching* 24

3.1 Exemplo Numérico: demanda de itens em cada período. 29
3.2 Soluções de padrões de corte e seus períodos associados para o exemplo numérico usando o 1D-PMCE-DE. 29
3.3 Decomposição do valor da função objetivo para cada período do exemplo numérico. 30
3.4 Características do primeiro conjunto de instâncias com comprimento de objeto de $L = 1000$. 38
3.5 Valores médios da função objetivo, tempo médio de execução e valor médio de *gap* para a solução obtida usando a abordagem proposta para o primeiro conjunto de instâncias. 39
3.6 Número de soluções melhoradas pelo *Local Branching* quando comparado à solução encontrada pelo *Relax-and-Fix*, usando o primeiro conjunto de instâncias. 39
3.7 Cada termo de valor médio da função objetivo da solução obtida pela metodologia proposta usando o primeiro conjunto de instâncias. . . . 40
3.8 Características do segundo conjunto de instâncias com comprimento de objeto de $L = 500$. 40
3.9 Valor médio da função objetivo, tempo de execução médio e valor médio de *gap* para a solução obtida usando a abordagem proposta para o segundo conjunto de instâncias. 41
3.10 Número de soluções melhoradas pelo *Local Branching* quando comparado à solução encontrada pelo *Relax-and-Fix*, usando o segundo conjunto de instâncias com datas de Entrega D3. 42
3.11 Valores médios de cada um dos termos da função objetivo encontrados na solução obtida pela metodologia proposta usando o segundo conjunto de instâncias 42

4.1 Uma Visão Geral Comparativa das Características dos Problemas Abordados (1D-PMCE-DE-SI) 46
4.2 Definições de parâmetros para o 1D-PMCE-DE-SI. 49
4.3 Tipos de Itens 54
4.4 Solicitações da empresa 54
4.5 Decomposição do valor da função objetivo para cada período para o exemplo numérico do 1D-PMCE-DE-SI com e sem tempo ocioso. . . . 55

4.6 Características do conjunto de instâncias com comprimento de objeto de $L = 1000$. 62
4.7 Modelo sem ociosidade: comparando os valores médios da função objetivo para instâncias da classe com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e os valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ para $p = 1$, e $\beta,\ \gamma,\ f \in U[1,p]$ para $p = 10$ e $p = 100$. 64
4.8 Modelo com ociosidade: comparando os valores médios da função objetivo para instâncias da classe com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e os valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ para $p = 1$, e $\beta,\ \gamma,\ f \in U[1,p]$ para $p = 10$ e $p = 100$. 65
4.9 Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ ($p = 1$). 67
4.10 Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 10$. 68
4.11 Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 100$. 68
4.12 Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ ($p = 1$). 70
4.13 Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 10$. 71
4.14 Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1, D2), os dois valores dos números de período (3, 6) e valores de penalização para atraso, antecipação e *setup* dado por β, γ, e $f \in U[1,p]$ para $p = 100$. 72
4.15 Modelo sem ociosidade: análise do comportamento do algoritmo proposto 74

4.16 Modelo com ociosidade: análise do comportamento do algoritmo proposto . . . 75
4.17 Conjunto de parâmetros de custo, em que os valores do tipo 1 foram usados nos experimentos anteriores e os valores do tipo 2 são baseados em Cui et al. (2015). . . . 77
4.18 Valores médios para 20 instâncias para cada uma das classes com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), 3 períodos para os modelos sem ociosidade (SO) e com ociosidade (CO). . . . 77

5.1 Uma Visão Geral Comparativa das Características do 1D-PMCE-DE-SD 88
5.2 Exemplo Numérico: demanda de item em cada período . . . 94
5.3 Plano de Corte Inicial. . . . 96
5.4 Intervalo de Valores para as datas de entrega . . . 99
5.5 Classes de Instâncias . . . 100
5.6 Resultados dos componentes da função objetivo para o *Local Branching* (LB) para as classes com 3 períodos. . . . 101
5.7 Resultados com termos da função objetivo para o *Local Branching* em comparação com o valor do CPLEX para as classes com 3 períodos . 104
5.8 Resultados da função objetivo da Solução Referência e do *Local Branching* para as classes com 6 períodos . . . 106
5.9 Resultados da Solução de Referência com a do *Local Branching* (LB) em comparação com o valor do CPLEX para as classes com 6 períodos. 109

6.1 Dados para análise de *setup* e data de entrega igual a 26 dias. . . . 113
6.2 Resultados para o exemplo numérico para os itens P. . . . 114
6.3 Resultados para o exemplo numérico para os itens G. . . . 115
6.4 Tipos de Itens e Comprimentos . . . 116
6.5 Comparação das soluções para os modelos SD e SI usando os dados da Tabela 6.4: valores da função objetivo (FO), de termos da função objetivo e do tempo computacional. . . . 117
6.6 Geração de Dados para a comparação das Modelagens . . . 119
6.7 Comparação entre a evolução das modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD para a classe com 20 tipos de itens e penalizações de custos iguais a 1. . . . 120
6.8 Comparação da evolução das modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD para a classe com 20 tipos de itens e penalizações de custos variados. . . . 123

Lista de Figuras

2.1 Exemplo de um 1D-PCE - Objeto em estoque e um conjunto de tipos de itens. Fonte: Própria do autor. 8
2.2 Exemplo de dois padrões de corte. O segundo padrão de corte é homogêneo. Fonte: Própria do autor. 9
2.3 Exemplo de um 1D-PCE (a) Irregular e (b) Regular. Fonte: Própria do Autor. 12

4.1 Esquema dos componentes fornecidos pelo usuário exigidos pelo pacote IRACE (López-Ibánez et al., 2016). 63
4.2 Tempos de execução para resolver cada instância do problema com ociosidade para as duas datas de entrega, considerando os diferentes tipos de custos. 78
4.3 Tempos de execução para resolver cada instância do problema sem ociosidade para as duas datas de entrega, considerando os diferentes tipos de custos. 79
4.4 Número de *setups* para cada instância do problema com ociosidade para os diferentes tipos de custos. 80
4.5 Número de *setups* para cada instância do problema sem ociosidade para os diferentes tipos de custos. 80
4.6 Número de objetos cortados para cada instância do problema com ociosidade para os diferentes tipos de custos. 81
4.7 Número de objetos cortados para cada instância do problema sem ociosidade para os diferentes tipos de custos. 81

5.1 Exemplo de um planejamento com a divisão dos períodos de corte e um conjunto de planos de corte sequenciados. Fonte: Própria da Autora. 85
5.2 Padrão de corte para o primeiro período do Exemplo Ilustrativo 2. Fonte: Própria do Autor. 94
5.3 Padrão de corte para o segundo período do Exemplo Ilustrativo 2. Fonte: Própria do Autor. 95
5.4 Padrões de corte para o terceiro período do Exemplo 2. Fonte: Própria do Autor. 95
5.5 Apresentação dos padrões de corte homogêneos para o Exemplo Ilustrativo 2. Fonte: Própria da Autora. 96
5.6 Matriz dos custos de *setup* para o Exemplo ilustrativo 2. 99
5.7 Comparação dos valores das componentes da função objetivo para as classes C1-P, C1-M e C1-G. Fonte: Própria da Autora. 102

5.8 Comparação dos valores das componentes da função objetivo para a classe C6-P, C6-M e C6-G. Fonte: Própria do Autor. 103
5.9 Comparação dos valores das componentes da função obejtivo para as classe C7-P, C7-M e C7-G. Fonte: Própria do Autor. 107
5.10 Comparação dos valores das componentes da função objetivo para as classe C12-P, C12-M e C12-G. Fonte: Própria do Autor. 108

6.1 Comparação entre os valores das funções objetivo para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD com custos iguais a 1. 121
6.2 Comparação entre os tempos computacionais para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD com custos iguais a 1. 121
6.3 Comparação entre o valores de função objetivo para as modelagens com custos variados. 125
6.4 Comparação entre os valores de tempo de processamento para as modelagens com custos variados. 125

Lista de Algoritmos

1 *Relax-and-Fix* - RF 18
2 *Fix-and-Optimize* – FOp 19
3 *Local Branching* 23
4 Abordagem de três fases proposta para resolver o 1D-PMCE-DE. . . 31
5 Método de geração de colunas para o 1D-PMCE-DE. 32
6 *Relax-and-Fix* - RF 34
7 *Local Branching* aplicada ao 1D-PMCE-DE 36
8 Método de geração de colunas para o 1D-PMCE-DE-SI. 54
9 Heurística de arredondamento aplicada ao 1D-PMCE-DE-SI. 57
10 *Relax-and-Fix* - RF 58
11 *Fix-and-Optimize* – FOp 60
12 Processo de Refinamento - RF 136

Capítulo 1

Introdução

1.1 Apresentação Geral

O Problema de Corte de Estoque (PCE) refere-se à otimização do corte de objetos (maiores) disponíveis em estoque, para atender a pedidos de corte de itens menores, cujos comprimentos são inferiores ao dos objetos originais. O objetivo é garantir que a demanda seja plenamente atendida, ao mesmo tempo em que se busca a minimização de uma função objetivo, frequentemente relacionada à redução de desperdícios ou ao aproveitamento eficiente dos materiais.

Os primeiros estudos sobre esse problema focaram na minimização das perdas resultantes do processo de corte. Com o tempo, abordagens mais abrangentes foram desenvolvidas para melhor representar as necessidades do planejamento da produção industrial. Os problemas de corte de estoque começaram a ser estudados por volta de 1940, embora as principais publicações tenham surgido apenas por volta de 1960. Historicamente, sua primeira formulação é devida a Kantorovich (1960). Esse trabalho teve sua versão original em russo publicada em 1939, mas sua divulgação em língua inglesa e consequente maior difusão se deu somente a partir de 1960. Porém, as modelagens e métodos de solução de maior repercussão na literatura são devido a Gilmore e Gomory (1961, 1963, 1965).

O PCE está presente em diversos setores produtivos, como o corte de bobinas de aço na indústria siderúrgica, o corte de madeira na indústria moveleira, o corte de rolos de tecido e peças na indústria têxtil, e o corte de blocos de espuma para a fabricação de colchões, este último mais comumente associado ao problema de empacotamento. Esses desafios refletem a importância do problema no contexto industrial, exigindo soluções eficientes para a alocação e utilização dos materiais disponíveis.

Em Gilmore e Gomory (1961), os autores apresentam uma formulação matemática para o PCE unidimensional, na qual propõem uma extensão do método simplex (Dantzig, 1963; Arenales et al., 2007; Bazaraa et al., 2011), usando geração de colunas para a solução da relaxação linear do problema de corte. Esta formulação permitiu, pela primeira vez, solucionar instâncias de grande porte para o caso unidimensional do PCE. Em Gilmore e Gomory (1963), uma formulação e um novo método de solução do PCE são propostos, e um novo método de solução para o problema da mochila. Além disso, é realizado um estudo de caso para o corte de papel. Os autores também consideraram uma nova restrição, qual seja, o número limite de lâminas de facas na máquina. Em Gilmore e Gomory (1965), os métodos descritos nos trabalhos

anteriores foram adaptados para o PCE bidimensional, acrescentando-se restrições como o corte guilhotinado, estagiado e irrestrito.

A solução de um PCE fornece um plano de corte, que especifica tanto o conjunto de padrões de corte que devem ser cortados, quanto o número de vezes em que tais padrões de corte devem ser utilizados. No entanto, o PCE clássico não agrega qualquer informação quanto à formulação temporal para a realização dessas tarefas. Assim, por exemplo, o plano de corte não aborda como a produção dos padrões resultantes é agendada, como também não aborda a respeito de prazos de conclusão dos pedidos de corte. Porém, em aplicações práticas, as empresas, ao receber seus pedidos, têm, definido, um determinado prazo para que eles sejam entregues.

Esta situação prática exige que se acrescente, ao PCE, novas restrições, que tratem as datas de entrega dos pedidos. Sendo assim, nesta tese, estudamos o Problema Multiperíodo de Corte de Estoque Unidimensional Unidimensional com Datas de Entrega (1D-PMCE-DE) dos pedidos. Trata-se de uma extensão do clássico PCE, ao qual se agregam restrições associadas às datas de entregas, definidas nos pedidos de corte. Neste sentido, é formulado um modelo para o problema que venha representar a integração de características de tempo ao problema de corte de estoque clássico.

O 1D-PMCE-DE consiste basicamente em resolver, a cada período de tempo, um problema de corte de estoque, para atender à demanda de itens nos diversos períodos do horizonte de planejamento, tendo claro que, devido às restrições de capacidade da máquina, pode ser que não seja possível atender aos pedidos na sua respectiva data de entrega. O estudo de problemas de corte de estoque com restrições de datas de entrega possui uma ampla variedade de aplicações no mundo real, tornando-se um tema de grande relevância para a comunidade científica. A necessidade de coordenar a otimização do corte de materiais com prazos específicos de entrega reflete desafios práticos enfrentados em diversos setores industriais, fortalecendo o interesse em abordagens que conciliem eficiência produtiva e atendimento à demanda dentro das restrições temporais estabelecidas.

O primeiro estudo do PCE-DE é devido a Li (1996). Neste artigo, é estudado o caso bidimensional, no qual os objetos em estoque têm comprimentos e larguras diferentes. O artigo introduz modelos matemáticos para o PCE-DE envolvendo restrições referentes às de data de entrega dos pedidos, para minimizar uma função objetivo. Em Hendry et al. (1996), os autores exploraram o problema estudado por Li (1996) para um caso prático em uma indústria de cobre. Para a resolução, é proposta uma abordagem em duas fases distintas: inicialmente, é tratado o problema de corte, seguido pela resolução do problema de *scheduling*.

Reinertsen e Vossen (2010) estudam o 1D-PCE-DE. Neste artigo, propõem modelos de otimização combinatória e procedimentos de solução baseados em técnica de geração de colunas. Arbib e Marinelli (2014) aprimoram o modelo matemático proposto por Reinertsen e Vossen (2010) e agregam novas ideias ao modelo. Braga et al. (2016a), por outro lado, abordam uma variante do problema que foi estudado por Reinertsen e Vossen (2010) e Arbib e Marinelli (2014), descrevendo uma formulação compacta de atribuição. Neste artigo, eles propõem diferentes abordagens que fortalecem o modelo proposto, incluindo a inserção de desigualdades válidas com base no problema da mochila usando funções dual-factíveis.

Arbib e Marinelli (2017) estudam o problema de *Bin Packing* unidimensional com Datas de Entrega (1D-BP-DE). Neste problema, os itens têm diferentes comprimentos

e devem ser atribuídos a um número mínimo de caixas (*bins*) de comprimento unitário. O problema é generalizado para o caso em que cada item deve ser entregue em uma certa data. O objetivo é a minimização de uma função, que envolve a combinação convexa de *makespan*, atraso e adiantamento.

Sezer e Muter (2018) apresentam uma extensão do 1D-PCE-DE, em que o corte é feito em dois estágios. Esta variação do problema surge quando restrições técnicas impedem que os itens que foram demandados possam ser cortados diretamente dos objetos que estão em estoque e, portanto, exige que o processo de corte seja realizado em dois estágios subsequentes. O modelo matemático proposto pelos autores visa determinar um plano de corte que não apenas minimize o número de objetos em estoque que foram utilizados, mas, também, reduza os custos de atraso e antecipação dos pedidos.

Bennell et al. (2013) propõem um algoritmo genético para resolver o problema *Bin Packing* com datas de entrega. O objetivo é minimizar o atraso máximo, usando o menor número possível de caixas. Polyakovskiy e M'Hallah (2018) estudam o problema *Bin Packing* não orientado, que pode ser girado em noventa graus. O corte é feito em caixas retangulares idênticas, e, além disso, acrescenta, as datas de entrega, admitindo que os tempos de processamentos são iguais, independentemente da caixa.

Estes problemas pertencem à classe NP-difícil, uma vez que não se conhece algoritmos que os resolvam em tempo polinomial (Johnson e Garey, 1979). Estes problemas podem ser resolvidos de maneira exata, para instâncias de pequena dimensão; contudo, como as instâncias de interesse possuem dimensões maiores, sua solução usando técnicas exatas resulta em um custo computacional muito alto (Valério de Carvalho, 1996). Essa dificuldade no tempo de execução de algoritmos exatos justifica a existência de uma grande quantidade de métodos heurísticos na literatura, como Wäscher e Gau (1996), Stadtler (1990) Poldi e Arenales (2006), Belov e Scheithauer (2002), Poldi e de Araujo (2016). Estes métodos não garantem a qualificação de otimalidade para a solução encontrada, porém, garantem soluções sub-ótimas em tempo computacional razoável. Esta mesma dificuldade quanto à qualificação da solução obtida é encontrada ao se adotar metaheurísticas para a solução dessa classe de problemas.

Nas operações industriais de corte, a formulação de um modelo eficiente para o Problema Multiperíodo de Corte de Estoque Unidimensional Unidimensional (1D-PMCE) deve levar em conta um fator crítico: a variação nos padrões de corte. A necessidade de realizar ajustes frequentes na configuração (*setup*) da máquina pode resultar em tempos ociosos e desperdícios de recursos, tornando essencial a incorporação desses aspectos no modelo de otimização.

O *setup* de corte de uma máquina pode ser classificado como dependente ou independente da sequência de padrões cortados, com base no impacto da preparação sobre a operação da máquina. O *setup* dependente da sequência ocorre quando a preparação da máquina depende da última produção realizada na máquina. Já o *setup* independente da sequência não ocorre essa dependência. A otimização do *setup* independente é uma estratégia essencial para reduzir o tempo de *setup* total (ver Haessler (1975) , Martin et al. (2018), Mobasher e Ekici (2013), Cui et al. (2015), Ma et al. (2018), Ma et al. (2019a) e Henn e Wäscher (2013)).

O processo de *setup* dependente da sequência gera custos e tempos adicionais que variam conforme a sequência dos cortes, sendo que padrões de corte mais semelhantes podem demandar um tempo de ajuste menor em comparação com aqueles que

apresentam diferenças mais significativas. O custo associado à mudança de padrão de corte depende diretamente do último corte realizado na máquina, introduzindo um nível considerável de complexidade ao problema. Como o tempo de *setup* e a eficiência operacional são fortemente influenciados pela ordem em que os cortes são executados, surge uma dependência sequencial que torna tanto a modelagem matemática quanto a busca por soluções eficientes um desafio significativo. Para uma revisão da importância de *setup* em problemas de *scheduling*, veja Allahverdi et al. (1999), Allahverdi et al. (2008) e Allahverdi (2015).

Dessa forma, a consideração explícita dos custos e do tempo associados à troca de padrões de corte tem se consolidado como um elemento fundamental nas formulações do 1D-PMCE-DE, permitindo uma representação mais realista do problema e aprimorando a eficiência das soluções implementadas na prática industrial. Esta tese propõe um estudo aprofundado do Problema Multiperíodo de Corte de Estoque Unidimensional Unidimensional com Datas de Entrega (1D-PMCE-DE) e suas extensões considerando *setup* com e sem dependência da sequência, com o objetivo de desenvolver novos modelos matemáticos e métodos eficientes de resolução baseados em matheurísticas. Abordagens híbridas, como as matheurísticas, que combinam técnicas de solução exata com métodos heurísticos, têm ganhado destaque. Os problemas de corte são clássicos problemas combinatórios e NP-difíceis. Isso significa que o número de combinações possíveis cresce rapidamente com o tamanho da instância. Resolver exatamente com métodos exatos se torna muito lento ou inviável para instâncias grandes. Então, as matheurísticas podem equilibrar qualidade de solução e tempo de processamento.

1.2 Objetivos Gerais e Específicos

1.2.1 Objetivos Gerais

O principal objetivo desta tese é a formulação e o desenvolvimento de modelos matemáticos e métodos para a resolução do Problema Multiperíodo de Corte de Estoque Unidimensional Unidimensional com Datas de Entrega, considerando cenários operacionais com *setups* independentes e *setups* dependentes da sequência de corte, elementos que impactam diretamente a eficiência da produção. Além disso, questões relacionadas com tempo ocioso ou não na máquina são abordados.

1.2.2 Objetivos Específicos

Para que este objetivo geral seja alcançado, é necessário atingir os seguintes objetivos específicos:

(i) Reformular e/ou desenvolver modelos matemáticos para gerar uma representação do Problema de Corte de Estoque Unidimensional com Datas de Entrega (1D-PMCE-DE);

(ii) Reformular e/ou desenvolver modelos matemáticos para gerar uma representação do Problema de Corte de Estoque Unidimensional com Datas de Entrega com *setup* Independente da Sequência (1D-PMCEDE-SI);

(iii) Reformular e/ou desenvolver modelos matemáticos para gerar uma representação do Problema de Corte de Estoque Unidimensional com Datas de Entrega com *setup* Dependente da Sequência (1D-PMCEDE-SE);

(iv) Resolver instâncias de grandes dimensões de itens e períodos; e

(v) Comparar os resultados com o obtidos usando o *solver* CPLEX;

(vi) Realizar uma avaliação aprofundada dos resultados obtidos.

1.3 Justificativa

O 1D-PMCE-DE, e suas extensões com questões de *scheduling*, é um problema ainda pouco estudado e que tem muito a ser investigado. Sendo assim, o desafio proposto nesta tese é estudar esses problemas buscando avançar no entendimento da integração do Problema de Corte de Estoque com o Problema de *Scheduling*.

A justificativa para estudar é solucionar estas variantes do PCE, principalmente, devido ao grande número de aplicações deste problema em ambientes industriais. Desde a revolução industrial, a competitividade entre as indústrias desencadeou a procura pela otimização dos processos de produção com o intuito de aumentar a produtividade, evitar o desperdício e diminuir custos. Principalmente em tempos de crise, a procura por procedimentos mais eficientes torna-se ainda maior, uma vez que elevar a produtividade e diminuir custos tende a aumentar o lucro e, consequentemente, a capacidade de permanecer e crescer no mercado.

1.4 Publicações Derivadas desta pesquisa

A seguir são listadas as publicações geradas pela presente pesquisa.

1.4.1 Artigos publicados em anais de congresso

(1) (Oliveira et al., 2020) Oliveira, Elisama de Araújo Silva e de Sá, Elisângela Martins e de Souza, Sérgio Ricardo e Wanner, Elizabeth Fialho (2020). *Relax-and-Fix* Aplicado ao Problema de Corte de Estoque com Data de Entrega. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 7, n. 1, pág. 010426-1 – 010426-7. DOI: `https://doi.org/10.5540/03.2020.007.01.0426`.

(2) (Oliveira et al., 2022) Oliveira, Elisama de Araújo Silva e de Sá, Elisângela Martins e de Souza, Sérgio Ricardo and Wanner, Elizabeth Fialho (2022). Uma proposta de solução via *Local Branching* do Problema de Corte de Estoque com Data de Entrega e *setup* de troca de Padrão de corte. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 9, n. 1, pág. 010426-1 – 010426-7. pág. 010325-1 – 010325-7. DOI: `https://doi.org/10.5540/03.2022.009.01.0325`.

1.4.2 Artigos em periódicos (publicados e aceitos para publicação)

(1) (Oliveira et al., 2025) Oliveira, Elisama de Araújo Silva e de Sá, Elisângela Martins e de Souza, Sérgio Ricardo e Wanner, Elizabeth Fialho (2025). A local branching-based solution for the multi-period cutting stock problem with tardiness, earliness, and *setup* costs. Journal of Heuristics, v. 31, n. 16, DOI: `https://doi.org/10.1007/s10732-025-09547-4`.

(2) (Oliveira et al., 2024) Oliveira, Elisama de Araújo Silva e de Sá, Elisângela Martins and de Souza, Sérgio Ricardo e Wanner, Elizabeth Fialho (2024). Matheuristic Applied to the Cutting Stock Problem with Due Dates. International Journal of Services and Operations Management. Aceito para publicação.

1.5 Organização da tese

O restante desta tese está organizado como segue.

O Capítulo 2 apresenta uma revisão de fundamentos teóricos dos assuntos e temas necessários para o desenvolvimento desta pesquisa. No Capítulo 3, apresentamos a modelagem e a proposta de solução para o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega (1D-PMCE-DE), assim como os testes realizados e os resultados computacionais obtidos. Além disso, apresenta-se as conclusões derivadas dos resultados alcançados. No Capítulo 4, apresentamos a modelagem e a proposta de solução para o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *setup* Independente da Sequência (1D-PMCE-DE-SI), juntamente com os testes realizados e os resultados computacionais obtidos. Adicionalmente, são apresentadas as conclusões baseadas nos resultados alcançados. No Capítulo 5 descrevemos a modelagem e a proposta de solução para o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *setup* Dependente da sequência (1D-PMCE-DE-SD), bem como os testes e resultados computacionais. Além de apresentamos uma conclusão dos resultados obtidos. No Capítulo 6 é feita uma comparação entre as modelagens descritas nos Capítulos 3, 4 e 5 para 10 instâncias definidas para estes fins. No Capítulo 7 foram sintetizados os principais achados deste estudo, reforçando a relevância e os objetivos inicialmente propostos. Além disso, são discutidas as contribuições da pesquisa, suas limitações e possíveis direcionamentos para teses futuros. O objetivo é reforçar a relevância do estudo e destacar como os conhecimentos gerados podem ser aplicados e aprofundados em novas investigações.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta uma revisão do problema de corte de estoque (PCE) clássico. A Seção 2.1 descreve detalhadamente sua formulação e principais características. Em seguida, na Seção 2.2, explora a tipologia do problema, classificando suas diferentes variantes de acordo com critérios relevantes, como dimensões dos materiais, restrições operacionais e objetivos de otimização. Além disso, apresenta uma revisão abrangente da literatura sobre o 1D-PCE na Seção 2.3, analisando os principais estudos, abordagens metodológicas e algoritmos propostos ao longo do tempo. Essa revisão permite não apenas compreender o estado da arte no tema, mas também identificar lacunas e desafios ainda não solucionados, reforçando a relevância desta pesquisa. Dessa forma, o embasamento teórico aqui construído serve como fundamento para o desenvolvimento das contribuições deste trabalho. A Seção 2.4 apresenta alguns dos métodos mais utilizados na resolução desses problemas, que são: a Geração de Colunas (Seção 2.4.1), os métodos *Relax-and-Fix* (Seção 2.4.2) e *Fix-and-Optimize*(Seção 2.4.3) e a matheurística *Local Branching* (Seção 2.4.4).

2.1 Problema de Corte de Estoque Unidimensional

O 1D-PCE consiste, de um modo geral, em determinar a melhor maneira de se cortar objetos, em estoque, em itens menores, otimizando uma função objetivo. O 1D-PCE inteiro pode ser definido da seguinte maneira. No caso unidimensional, 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, $i = 1, \ldots, n$, tem tamanho ℓ_i e demanda conhecida b_i. Os comprimentos dos itens são tais que $\ell_i < L, \forall i = 1, \ldots, n$. O problema consiste em produzir os itens demandados a partir do corte dos objetos em estoque, atendendo à demanda, de modo que o número de objetos em estoque necessários para satisfazê-la seja minimizado. A Figura 2.1 exemplifica o objeto que se tem em estoque (na cor cinza) e os itens demandados (nas cores vermelho, azul, amarelo).

Definição 1. *Padrão de corte é a maneira como um objeto em estoque é cortado para a produção dos itens demandados. A um padrão de corte associamos um vetor*

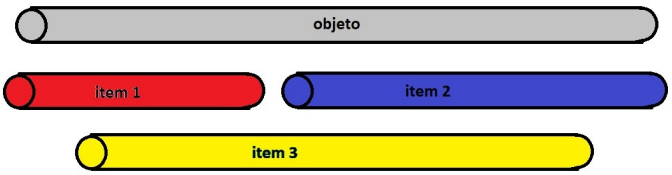


Figura 2.1: Exemplo de um 1D-PCE - Objeto em estoque e um conjunto de tipos de itens. Fonte: Própria do autor.

n-dimensional que contabiliza os itens produzidos, na forma:

$$a = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix},$$

em que α_i é o número de vezes que o item tipo i está presente no padrão de corte.

Definição 2. *Um padrão de corte homogêneo produz um único tipo de item:*

$$a = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Em particular, para o 1D-PCE , podemos escrever um padrão de corte homogêneo da seguinte forma:

$$a = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \lfloor \frac{L}{\ell_i} \rfloor \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

A Figura 2.2 ilustra dois padrões de corte para o 1D-PCE. Observe que o segundo padrão de corte é um padrão de corte homogêneo, ou seja, fornece apenas um tipo de item demandado.

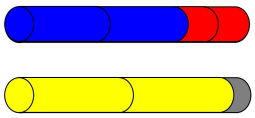


Figura 2.2: Exemplo de dois padrões de corte. O segundo padrão de corte é homogêneo. Fonte: Própria do autor.

2.1.1 Modelo clássico do 1D-PCE

Essa seção apresenta a formulação clássica de Gilmore e Gomory (Gilmore e Gomory, 1961, 1963, 1965) para o 1D-PCE. É importante ressaltar que existem outras modelagens, como, por exemplo, as propostas em Kantorovich (1960), Valério de Carvalho (1996), Valério de Carvalho (2002), Martinovic et al. (2018), e Dyckhoff (1981), que, porém, não são abordadas na presente tese.

Esta formulação assume que existe um conjunto de padrões de corte. Sendo assim, o primeiro passo é definir este conjunto padrões de corte, e em seguida determinar o número de vezes que cada padrão de corte será utilizado, para resolver o problema. Assim, a modelagem matemática de um1D-PCE pode ser feitas em duas etapas:

(i) Definir todos os possíveis padrões de corte (supondo n o número de padrões de corte obtidos);

$$a_1 = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{m1} \end{bmatrix}, \quad a_2 = \begin{bmatrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{m2} \end{bmatrix}, \ldots, \quad a_n = \begin{bmatrix} \alpha_{1n} \\ \alpha_{1n} \\ \vdots \\ \alpha_{mn} \end{bmatrix},$$

sendo que α_{ij} é o número de itens do tipo i no padrão de corte j.

(ii) Definir o número x_j de vezes que cada padrão de corte a_j será utilizado para atender à demanda, sendo x_j um número inteiro e não-negativo.

Considere que o 1D-PCE tenha, como objetivo, minimizar o número de objetos cortados. Portanto, pode ser formulado como um problema de programação linear inteira, na forma:

$$\begin{aligned} \min \quad & x_1 + x_2 + \ldots + x_n && (2.1) \\ \text{sujeito a:} \quad & a_1x_1 + a_2x_2 + \ldots + a_nx_n = b && (2.2) \\ & x_j \in \mathbb{Z}_+. && (2.3) \end{aligned}$$

A função objetivo na expressão (2.1), a ser minimizada, representa o número total de objetos cortados. A restrição (2.2) garante o atendimento da demanda de cada item, em que b representa o vetor de demandas. A restrição (2.3) garante que a frequência de cada padrão de corte j seja um número inteiro e não negativo.

2.2 Tipologias para o Problema de Corte e Empacotamento

Uma tipologia é uma organização sistemática de problemas em categorias homogêneas, com base em um conjunto de critérios e características. Um Problema de Corte e Empacotamento (C&E) consiste em alocar um conjunto de peças dentro de um objeto. Uma tipologia fornece uma visão concisa de objetos relevantes e importantes e ajuda a unificar definições e notações e, ao fazer isso, facilita a comunicação entre pesquisadores na área.

Uma tipologia para problemas de C&E busca caracterizar os problemas existentes, levando em conta propriedades como, por exemplo, se há somente um tipo de objeto em estoque ou se há diferentes tipos de objetos; se os itens possuem um determinado padrão de corte; se os itens são retangulares, dentre outras propriedades. Um dos objetivos da tipologia é facilitar a busca de trabalhos já realizados, podendo os futuros pesquisadores da mesma classe de problemas realizarem aprimoramentos nos métodos ou comparações com novos métodos.

A estrutura dos problemas de C&E pode ser resumida como se segue. Seja:

- um conjunto de objetos (de entrada, estoque);
- um conjunto de itens de tamanhos menores que o tamanho do objeto (de saída, demanda).

Devemos selecionar alguns (ou todos os) itens, agrupá-los em um ou mais subconjuntos e atribuir, a cada um dos subconjuntos, um dos objetos em estoque, de tal modo que a condição geométrica seja satisfeita, ou seja, os itens de cada subconjunto têm que ser colocados no objeto correspondente, de tal forma que:

- todos os itens do subconjunto fiquem totalmente dentro do objeto;
- os itens não fiquem sobrepostos.

Existe uma grande diversidade de problemas que podem ser definidos sob uma formulação geral de problemas de C&E e, para que não ocorram problemas semelhantes com nomes diferentes, Dyckhoff (1990) e Wäscher et al. (2007) propuseram tipologias com o objetivo de padronizar e classificar essa diversidade. As subseções a seguir descrevem, em linhas gerais, estas tipologias.

2.2.1 Tipologia de Wäscher et al. (2007)

A tipologia proposta inicialmente por Dyckhoff (1990) foi um excelente instrumento para a organização e caracterização da literatura. Contudo, ao longo dos anos surgiram algumas deficiências, pois problemas semelhantes poderiam receber nomes diferentes, o que gerou dificuldades em lidar com os desenvolvimentos recentes e impediu-a de ser aceita de forma geral.

Wäscher et al. (2007) apresentam uma reformulação da tipologia de Dyckhoff (1990). Esta nova tipologia é parcialmente baseada nas ideias originais de Dyckhoff (1990), mas introduz novos critérios de categorização, que definem categorias de problemas diferentes das tratadas em Dyckhoff (1990).

Esta nova tipologia classifica os Problemas de Corte de Estoque de acordo com cinco critérios:

Dimensionalidade

Este critério não se altera e permanece, portanto, análogo ao de Dyckhoff (1990).

Tipo de atribuição

Como em Dyckhoff (1990):

- maximização da saída: no qual a quantidade de objetos disponíveis é limitada e, dessa forma, não é suficiente para acomodar todos os itens. Deste modo, é necessário selecionar os itens de modo a maximizar a utilização dos objetos;
- minimização da entrada: um determinado conjunto de itens deve ser alocado em um conjunto de objetos disponíveis em quantidade suficiente para alocar todos os itens, sendo necessário alocar os itens buscando minimizar a quantidade de objetos utilizada.

Variedade dos itens

Podemos distinguir os problemas em três casos:

- itens idênticos: existe um único tipo de item, ou seja, todos os itens possuem as mesmas dimensões e aparência;
- variedade fracamente heterogênea: a classificação dos itens é dita fracamente heterogênea quando há muitos itens iguais, ou seja, é possível agrupar os itens em poucos grupos e, em cada grupo, há vários itens idênticos;
- variedade fortemente heterogênea: a classificação dos itens é dita fortemente heterogênea quando poucos itens são iguais, ou seja, há vários grupos de itens com poucos itens em cada grupo.

Variedade dos objetos

Com respeito aos objetos, temos as seguintes classificações:

- um objeto: um único objeto está disponível para a alocação de itens;
- vários objetos: temos vários objetos disponíveis para alocação. Porém, para determinarmos se todos os objetos são de dimensões iguais ou diferentes, usamos as sub-classificações apresentadas para os itens, e, portanto, temos os casos:
  - idênticos;
  - variedade fracamente heterogênea; e
  - variedade fortemente heterogênea.

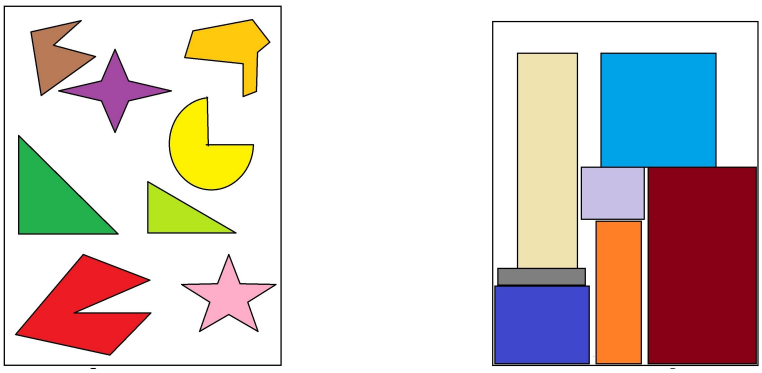
  Esta sub-classificação segue os mesmos significados apresentados anteriormente.

Forma dos itens

Podemos classificar os problemas que possuem duas ou três dimensões de acordo com a forma de seus itens:

- regulares: retângulos, círculos, caixas, cilindros, esferas, etc;
- irregulares.

A Figura 2.3 ilustra um exemplo de forma dos itens, de modo que Figura 2.3(a) mostra itens irregulares e Figura 2.3(b) mostra itens regulares. A grande maioria dos problemas da literatura trata de formas regulares, especialmente de formas retangulares.



(a) Irregulares (b) Regulares

Figura 2.3: Exemplo de um 1D-PCE (a) Irregular e (b) Regular. Fonte: Própria do Autor.

Problemas que permitem *layouts* não ortogonais e/ou misturas de itens regulares e irregulares são definidos como problemas variantes.

Com base no objetivo do problema e nos critérios descritos anteriormente, a estrutura da classe dos Problemas de Corte de Estoque proposta por Wäscher et al. (2007) é definida em três etapas:

- Tipo básico: os critérios Tipo de atribuição e Variedade dos itens são combinados para definir a estrutura básica dos problemas de corte e empacotamento. Esta etapa inicial é subdividida em duas categorias:
  - maximização da saída: por exemplo, Problema de Empacotamento de Itens Idênticos (IIPP - *Identical Item Packing Problem*), Problema de Alocação (PP - *Placement Problem*), Problema da Mochila (KP - *Knapsack Problem*);
  - minimização da entrada: por exemplo, Problema de Dimensão Aberta (ODP - *Open Dimension Problem*), Problema de Corte de Estoque (CSP - *Cutting Stock Problem*), Problema de Empacotamento em Bins (BPP - *Bin Packing Problem*).

- Tipo intermediário: o critério Tipo de objetos é combinado com problemas do tipo apresentados anteriormente.
- Tipo refinado: os critérios de Dimensionalidade e Forma dos itens são acrescentados à estrutura intermediária do problema, sendo definida, assim, a classe do mesmo.

Portanto, o resultado final da estrutura proposta por Wäscher et al. (2007) é obtido através da seguinte regra:

{1,2,3}-D {retangular, circular,...,irregular} {classificação intermediária}.

De acordo com esta classificação, a presente tese de doutorado aborda uma extensão do seguinte problema:

- 1D - *Single Stock Size Cutting Stock Problem* - SSSCSP:1D-PCE com um único comprimento,

uma vez que estamos considerando a data de entrega dos itens cortados.

2.3 Revisão Bibliográfica para o 1D-PCE

Kantorovich (1960) introduziu a primeira modelagem matemática conhecida para o problema. Com base nisso, Gilmore e Gomory (1961) apresentaram uma formulação matemática para o 1D-PCE unidimensional, juntamente com uma extensão inovadora do método Simplex usando geração de colunas. Subsequentemente, Gilmore e Gomory (1963) propuseram uma nova formulação e solução para o 1D-PCE, incluindo um novo método para resolver o problema da mochila e ilustrando sua aplicação por meio de um estudo de caso de recorte de papel. Vários outros trabalhos desenvolveram formulações para o 1D-PCE, incluindo o modelo de um corte (Dyckhoff, 1981) e o modelo baseados em *arc flow* (Valério de Carvalho, 2002). Uma comparação entre formulações1D-PCE foi feita por Martinovic et al. (2018), que mostrou a equivalência entre o modelo de Gilmore e Gomory e as versões reduzidas dos modelos de *arc flow* e de um corte, o que também é verdadeiro para os relaxações contínuos desses modelos.

No artigo Mellouli e Dammak (2008), os autores tratam do corte de um conjunto de peças retangulares, solicitadas em grandes quantidades e obtidas com cortes de guilhotina orientados de um conjunto de rolos longos de material com larguras padrão de corte de forma.

Para o caso 3D, podemos citar De Queiroz et al. (2012), no qual os autores apresentam algoritmos para os seguintes problemas de corte de guilhotina tridimensionais (3D): mochila sem limites, corte de estoque e empacotamento em tiras. Eles consideram o caso em que os itens têm orientação fixa e o caso em que rotações ortogonais em torno de todos os eixos são permitidas.

Um problema originário de uma fábrica de móveis, em que os produtos são fabricados a partir de operações de corte e junção, foi estudado por Hao et al. (2023). Os autores formularam o problema usando um modelo de atribuição baseado em *cluster*, que estabeleceu limites inferiores e superiores para o modelo baseado em atribuição. Eles demonstraram que o modelo de atribuição baseado em *cluster* pode atingir o

cronograma ideal para o problema original. Dados industriais foram empregados para avaliar o desempenho da heurística iterativa.

Os Problemas de Corte de Estoque (PCEs), incluindo o problema clássico de corte unidimensional e o problema de empacotamento binário, pertencem à classe dos problemas NP-difíceis, conforme demonstrado em trabalhos fundamentais como Garey e Johnson (1979) e amplamente discutido por Martello e Toth (1990), para esses problemas não há algoritmos que os resolvam exatamente em tempo polinomial. Esse fato justifica a aplicação de heurísticas e metaheurísticas para resolver esses problemas. Recentemente, procedimentos que utilizam metaheurísticas têm sido propostos para resolver problemas de corte de estoque. Bertolini et al. (2024) consideraram a possibilidade de que as objetos possam possuir diferentes valores econômicos e prioridades para processamento ou envio. Para enfrentar esse desafio, eles introduziram uma heurística baseada em *Simulated Annealing*. Essa heurística passou por testes rigorosos em várias instâncias do problema e foi comparada com três métodos estabelecidos.

Um modelo matemático eficiente empregando perda sustentável foi proposto em Vishwakarma e Powar (2021). A abordagem de perda sustentável abordou o problema de corte de estoque unidimensional (1D-PCE) com foco na sustentabilidade, considerando tanto a eficiência econômica quanto o impacto ambiental. Este problema tem como objetivo reduzir o desperdício e promover práticas mais sustentáveis na produção industrial. Um destaque importante deste trabalho é o fornecimento de soluções viáveis do mundo real para aplicações industriais. Além disso, este modelo pode ser estendido para lidar com problemas de corte de material bidimensional e tridimensional.

Luo et al. (2022) examinaram um problema de corte de material especializado para a indústria de fabricação de transformadores, no qual um conjunto de tiras é cortado de materiais laminados de vários tamanhos. Os autores desenvolveram uma heurística construtiva de tempo polinomial, que incorpora três estratégias: camada dupla, busca e randomização. Para avaliar o desempenho dos algoritmos propostos, experimentos computacionais foram conduzidos em instâncias geradas a partir de dados do mundo real fornecidos pela empresa de fabricação de transformadores.

Em Pitombeira-Neto e Murta (2022), uma formulação para o 2D-PCE estocástico foi proposta. Os autores desenvolveram uma abordagem de solução heurística baseada em aprendizado por reforço. Em cada época de decisão, dado o estoque atual de itens, um agente selecionava quais padrões de corte usar em antecipação à demanda incerta. Uma solução ótima é dada por uma política que vincula cada estado a uma decisão, visando minimizar o custo total esperado.

Sá Santos e Nepomuceno (2022) abordaram grandes instâncias do 1D-PCE empregando um procedimento de geração de padrões de corte e propondo uma estratégia para limitar o número de padrões de corte gerados. Eles implementaram uma técnica de geração de colunas e aplicaram a estrutura *Generate-and-Solve* para obter soluções para instâncias de *benchmark*. Esse método exato resolveu com sucesso instâncias pequenas e médias do problema.

Quando componentes industriais são produzidos cortando barras de matéria-prima (estoques), defeitos aleatórios nos estoques podem afetar os volumes e valores da produção. Para abordar esse problema, Arbib et al. (2023) propuseram projetar padrões de corte reconfiguráveis que podem ser ajustados para reduzir o máximo possível o desperdício. Essa abordagem introduziu um *trade-off* entre a quantidade

de sucata gerada e a probabilidade de obter uma reconfiguração sem perder nenhum item.

Em Montiel-Arrieta et al. (2023), os autores abordaram o 1D-PCE, em que a minimização de desperdícios é o objetivo principal. A Otimização do Búfalo Africano (ABO) foi empregada neste estudo. Segundo os autores, o método desenvolvido compete efetivamente na minimização de desperdícios quando comparado a outras heurísticas, metaheurísticas e abordagens híbridas.

Vários recursos de problemas do mundo real foram explorados para acomodar necessidades distintas de planejamento de produção. Notavelmente, certos modelos integraram efetivamente desafios como otimizar o corte de estoque junto com o dimensionamento de lote em processos industriais. Na área de problemas de corte de estoque unidimensional, Poldi e Arenales (2009) conceberam heurísticas adaptadas para cenários envolvendo quantidades limitadas e baixa demanda por comprimentos de estoque. Além disso, Poldi e de Araujo (2016) estenderam a discussão para abranger uma variante multiperíodo do problema de corte de estoque unidimensional, generalizando modelos originalmente projetados para o 1D-PCE clássico de período único.

A Tabela 2.1 apresenta uma análise comparativa entre trabalhos de referência da literatura e a abordagem proposta nesta tese. São considerados artigos que tratam da minimização da função objetivo, analisando o número de objetos cortados, além da dimensionalidade do problema, seja ele unidimensional (1D), bidimensional (2D) ou tridimensional (3D). Observa-se um foco significativo nos estudos voltados para o caso 1D, que é também o escopo desta pesquisa. Nos próximos capítulos, aprofundaremos a discussão sobre os trabalhos analisados e a desta tese, todos voltados para o contexto unidimensional

Tabela 2.1: Uma Visão Geral Comparativa das Características dos Problemas Abordados.

		Características abordadas			
Tipo	Artigo de Referência	Objetos cortados ou Desperdício	1D	2D	3D
1	Kantorovich (1960), Gilmore e Gomory (1961), Mellouli e Dammak (2008), Dyckhoff (1981), Valério de Carvalho (2002), Martinovic et al. (2018), Hao et al. (2023), Bertolini et al. (2024), Vishwakarma e Powar (2021), Luo et al. (2022), Pitombeira-Neto e Murta (2022), Sá Santos e Nepomuceno (2022), Poldi e Arenales (2009),Arbib et al. (2023), Poldi e Arenales (2009), Montiel-Arrieta et al. (2023),Poldi e de Araujo (2016)	sim	sim	–	–
2	Gilmore e Gomory (1963), Gilmore e Gomory (1965),	sim	–	sim	–
3	De Queiroz et al. (2012)	sim	–	–	sim
4	**Este estudo**	sim	sim	–	

2.4 Métodos de Solução para o 1D-PCE

No contexto da programação inteira e da otimização combinatória, diversos procedimentos são utilizados para desenvolver algoritmos de resolução. Entre esses procedimentos, essa seção cita o método de Geração de Colunas (Seção 2.4.1), as matheurísticas *Relax-and-Fix* (Seção 2.4.2), *Fix-and-Optimize* (Seção 2.4.3) e *Local Branching* (Seção 2.4.4). Essas abordagens têm se mostrado eficazes na resolução de problemas complexos, como os de corte de estoque, de planejamento de produção, roteirização e escalonamento, por permitirem o tratamento eficiente de grandes volumes de variáveis e restrições.

2.4.1 Método Simplex com Geração de Colunas

Considere um problema de otimização linear no qual o número de variáveis (colunas) é significativamente maior que o número de restrições (linhas). Antes de discutir a geração de padrões de corte, é fundamental apresentar a formulação clássica do problema de corte de estoque, conforme proposto por Gilmore e Gomory (1961). Esta formulação é essencial para contextualizar a utilização do método de geração de colunas, uma vez que a cada iteração novas colunas (padrões de corte) são adicionadas a um modelo mestre relaxado.

Considere o seguinte Problema de Corte de Estoque Unidimensional (1D-PCE). Seja L o comprimento do objeto a ser cortado e ℓ_i o comprimento da peça de tipo i, com uma demanda b_i, para $i = 1, \ldots, m$. Um *padrão de corte* é uma combinação de peças que pode ser obtida a partir de um objeto de comprimento L, respeitando a restrição de não ultrapassá-lo.

A formulação do problema é dada por:

$$\min \quad \sum_{j=1}^{n} x_j \tag{2.4}$$

$$\text{sujeito a} \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad \forall i = 1, \ldots, m \tag{2.5}$$

$$x_j \in \mathbb{Z}_+, \quad \forall j = 1, \ldots, n, \tag{2.6}$$

na qual temos o seguinte:

- x_j representa o número de vezes que o padrão de corte j é utilizado.
- a_{ij} representa o número de peças do tipo i no padrão de corte j.
- n é o número total de padrões de corte possíveis (geralmente exponencial em relação a m).

Durante as iterações do método Simplex, é necessário determinar uma nova coluna a ser inserida na base, o que é impraticável em casos em que há uma grande quantidade de colunas que precisam ser analisadas. Para contornar essa dificuldade, Gilmore e Gomory (1961) propuseram uma técnica de geração de colunas altamente eficiente para resolver esse tipo de problema. Esse procedimento consiste na geração de uma

nova coluna k, utilizando o critério de Dantzig, que seleciona a variável com o menor custo reduzido. Isso leva à formulação do seguinte problema:

$$c_k - \pi^T a_k = \min\left\{c_j - \pi^T a_j, j = 1,2,\ldots,n\right\}, \tag{2.7}$$

em que π é o vetor de multiplicadores do método Simplex de uma determinada iteração e n é o número de variáveis de decisão.

No caso específico em que buscamos minimizar o total de objetos em estoque, algumas adaptações no Método Simplex com Geração de Colunas são necessárias. Para essa abordagem, utilizamos um vetor de custos uniformemente definido como $(1,1,\ldots,1)$, refletindo a minimização direta da quantidade total de itens armazenados. A seguir, é apresentada a formulação correspondente e os passos para sua resolução.

Observe que

$$\min\left\{c_j - \pi^T a_j\right\} = 1 - \max\{\pi^T a_j\}. \tag{2.8}$$

Seja L o comprimento do objeto que se tem em estoque para realizar os cortes e b_i a demanda do item do tipo i. Então, encontrar uma coluna não-básica, denotada genericamente por $(\alpha_1, \alpha_2, \ldots, \alpha_n)^T$, que substituirá uma coluna básica, corresponde a resolver o problema:

$$g(a) = \max \quad \pi_1 a_1 + \pi_2 a_2 + \ldots + \pi_m a_m \tag{2.9}$$

$$\text{sujeito a:} \quad \ell_i \alpha_1 + \ell_2 \alpha_2 + \ldots + \ell_m \alpha_m \leq L \tag{2.10}$$

$$0 \leq \alpha_i \leq b_i,\ \alpha_i \in \mathbb{Z},\ i = 1,\ldots,m. \tag{2.11}$$

A função objetivo (2.9) representa a maximização no lado direito da expressão (2.8), para cálculo do custo reduzido mínimo. A restrição (2.10) garante que não seja possível alocar no padrão de corte uma quantidade de itens maior que o comprimento do objeto. As restrições (2.11) representam o domínio das variáveis.

O problema (2.9)-(2.11) é uma formulação para o problema da mochila (*knapsack problem*) inteiro. Como existem diversos métodos para resolver esse problema, a escolha da abordagem mais adequada depende do objetivo de otimização adotado. Após a resolução deste problema, é obtido o valor de $g(a)$. Se o menor dos custos reduzidos é não-negativo, isto é, se $1 - g(a) \geq 0$, podemos garantir que a solução do problema relaxado é ótima. Caso contrário, o padrão de corte gerado é adicionado à formulação do problema de corte.

Assim, adota-se a técnica de geração de colunas, na qual se resolve iterativamente um *problema mestre relaxado*, em que:

- Apenas um subconjunto dos padrões de corte é inicialmente considerado;
- O problema mestre é resolvido de forma relaxada (permitindo $x_j \in \mathbb{R}_+$);
- A seguir, utiliza-se as informações do dual para resolver um *problema de geração de colunas* (um problema de mochila), que busca encontrar novos padrões de corte com custo reduzido negativo;
- Se tal padrão de corte é encontrado, ele é adicionado ao problema mestre relaxado.

Esse processo se repete até que não existam mais colunas com custo reduzido negativo, momento em que se atinge a solução ótima do problema relaxado. Opcionalmente, pode-se então aplicar um procedimento de arredondamento.

2.4.2 *Relax-and-Fix*

Segundo, Ngoo et al. (2024), uma matheurística (termo em inglês: *matheuristic*) é um algoritmo de otimização que surge pela integração de heurísticas e Programação Matemática (PM). Uma característica essencial é a exploração em partes dos algoritmos de recursos provenientes a partir do modelo matemático dos problemas de interesse.

Dentre as matheurísticas, estão as heurísticas de horizonte rolante, que são estratégias usadas principalmente em planejamento e tomada de decisão ao longo do tempo. A ideia central é tomar decisões com base em uma janela de tempo que se move (ou desliza) ao longo do tempo, em vez de planejar tudo de uma vez desde o início até o final. Elas partem do princípio que apesar de que o horizonte de planejamento seja finito, existe uma alta complexidade para resolvê-lo considerando todo o horizonte. Particionando o horizonte de planejamento, as heurísticas de horizonte rolante trabalham com essas partições. Nessa categoria de heurísticas surge a matheurística *Relax-and-Fix*, proposta por Wolsey (1998).

A matheurística *Relax-and-Fix* consiste na decomposição pelo particionamento das variáveis inteiras do modelo dentro do horizonte de planejamento. Ela se baseia na partição das variáveis inteiras do problema criando P conjuntos distintos, $Q_i, i = 1, \ldots, P$. O número P de conjuntos determina o número de iterações da heurística. Em uma iteração n, apenas as variáveis que pertencem ao conjunto Q_n são definidas como inteiras, as demais variáveis que precisam ser inteiras serão relaxadas ou fixadas em algum valor. O submodelo resultante será então resolvido. Se o submodelo é inviável, a heurística para, uma vez que, não é possível encontrar uma solução viável para o problema original com as variáveis do conjunto $Q_i, i = 1, \ldots, (n-1)$, fixadas nos valores definidos atuais. Se o submodelo for factível, as variáveis do conjunto Q_n, ou parte delas, são fixadas em seu valor atual, e o processo se repete para os demais partições. Quando o subconjunto das variáveis relaxadas fica vazio, uma solução viável inteira é encontrada e o procedimento termina. O Algoritmo 1 mostra o pseudocódigo para o *Relax-and-Fix*.

Algoritmo 1 *Relax-and-Fix* - RF

procedure RELAX-AND-FIX – RF
    Escolha uma decomposição disjuntiva para o conjunto de variáveis inteiras $Q = Q^1 \cup Q^2 \cup \ldots \cup Q^P$.
    Relaxe a integralidade de todas as variáveis inteiras.
    for $r \leftarrow 1$ **ate** P **do**
        Torne as variáveis associadas ao subconjunto Q^r inteiras.
        Resolver o problema resultante.
        Fixe as variáveis associadas ao subconjunto Q^r, de forma que essas variáveis não devem ser alteradas nos próximos subproblemas.
    end for
end procedure

Exemplos práticos de aplicação da heurística *Relax-and-Fix*: Planejamento de Produção em Horizontes de Tempo, Escalonamento de Funcionários (*Workforce Scheduling*), Problema de Roteamento de Veículos (VRP), Problemas de Corte e Empacotamento (*Cutting and Packing*), Problemas de Alocação de Tarefas, Escalonamento

de Produção em Ambientes Multiproduto, Problemas de Tripulação Aérea (*Crew Scheduling*), Gestão de Projetos com Restrições de Recursos. Todos esses problemas têm uma característica comum: grandes quantidades de variáveis inteiras, horizonte temporal, e decisões interdependentes.

2.4.3 *Fix-and-Optimize*

Semelhante ao *Relax-and-Fix*, a heurística *Fix-and-Optimize* se baseia na decomposição do problema em subproblemas mais fáceis de serem resolvidos. A heurística *Fix-and-Optimize* consiste na decomposição pelo particionamento das variáveis do modelo dentro do horizonte de planejamento em dois subconjuntos. A cada iteração parte das variáveis são fixadas e uma parte das variáveis permanecem como variáveis inteiras. O subconjunto das variáveis inteiras é deslocado no horizonte de planejamento. As variáveis correspondente a todos os demais períodos do horizonte que esteja fora da janela corrente fará parte do subconjunto de variáveis fixadas. Quando o subconjunto das variáveis restritas se desloca até o fim do horizonte de planejamento, uma solução viável inteira é encontrada e o procedimento termina.

A heurística se inicia recebendo uma solução viável e a cada nova iteração é verificado se com a nova solução ocorreu uma melhoria em relação a última solução. Caso ocorra uma melhora a solução inicial e substituída tendo uma nova solução inicial. O Algoritmo 2 detalha a aplicação da heurística de melhoramento *Fix-and-Optimize*. Inicialmente, determina-se o particionamento das variáveis (linha 2) e a solução incumbente inicial (linha 3). Em seguida, para cada subconjunto, as variáveis são fixadas (linha 5), o subproblema é resolvido (linha 6) e a solução incumbente é atualizada, a partir da solução do subproblema corrente (linha 7).

Algoritmo 2 *Fix-and-Optimize* – FOp

1: **procedure** FIX-AND-OPTIMZE – FOP
2: Escolha uma decomposição disjuntiva para o conjunto de variáveis inteiras $Q = Q^1 \cup Q^2 \cup \ldots \cup Q^P$
3: Determine uma solução inicial incumbente.
4: **for all** $r \in \{1, \ldots, P\}$ **do**
5: Fixe as variáveis em $Q \setminus Q^r$ de acordo com a solução incumbente.
6: Otimize o subproblema resultante.
7: Atualize a solução incumbente com a solução ótima do subproblema.
8: **end for**
9: **end procedure**

2.4.4 Matheurística *Local Branching* (*LB*)

A matheurística *Local Branching (LB)* foi proposta em Fischetti e Lodi (2003) para solucionar Problemas Inteiros Mistos (PIM). Para tal, define uma estratégia de exploração da árvore de *branching*, semelhante ao procedimento realizado por metaheurísticas de busca local, porém as vizinhanças são obtidas através dos chamados pseudos-cortes *LB*. Essa estratégia atua em dois níveis: em alto nível, define a vizinhança de uma solução, enquanto que, em baixo nível, explora as ramificações em si.

Desse modo, o método favorece atualizações na solução incumbente em seus passos iniciais, bem como a produção de soluções melhoradas na fase final de exploração.

O método *LB* explora, para uma dada solução, a vizinhança k-OPT (conjunto de soluções obtidas através da permutação de até k arestas) como sendo o conjunto de soluções que satisfazem a uma restrição adicional de *branching*. Esta restrição define que a distância entre duas soluções seja no máximo k, parâmetro conhecido como raio da vizinhança. É importante que o valor de k permita a geração de sub-problemas que possam ser resolvidos rapidamente, mas, ao mesmo tempo, também contenha diversidade para que boas soluções sejam encontradas. Para melhorar o desempenho do *LB*, são adicionados parâmetros relacionados ao tempo limite de exploração e número de nós limite explorados na árvore *LB*.

Conforme Fischetti e Lodi (2003), a matheurística *LB* controla algum outro algoritmo de solução de PIM exato, priorizando a exploração parcial ou completa da vizinhança de soluções viáveis já identificadas. A *LB* pode ser descrita como segue.

Considere, então, o seguinte problema (P):

$$\begin{aligned} (P)\ \min \quad & c^t x \\ \text{subjeito a:} \quad & Ax \geq b \\ & x_j \in \{0,1\}, \forall j \in \mathbb{I}, \end{aligned}$$

em que $\mathbb{I}$ é o conjunto de índices das variáveis $\mathbf{x_j}$ que devem ser binárias.

Se $\hat{x}$ é uma solução factível de referência de (P), defina o conjunto $\overline{S} = \{j \in \mathbb{I}, \hat{x} = 1\}$ como o conjunto suporte de $\hat{x}$. Para um dado parâmetro inteiro positivo k, definimos a vizinhança $N^k(\hat{x})$ como o conjunto das soluções factíveis de (P), satisfazendo uma restrição adicional *Local Branching*.

Uma restrição de ramificação local em relação a uma solução $\hat{x}$ inteira de referência para o problema (P) pode ser definida como:

$$\Delta(x, \hat{x}) = \sum_{j \in \overline{\mathbf{S}}} (1 - x_j) + \sum_{j \in \mathbb{I} \setminus \overline{\mathbf{S}}} x_j \leq k, \tag{2.12}$$

em que os dois termos do lado esquerdo contam o número de variáveis binárias que mudam de 1 para 0 e de 0 para 1, respectivamente.

Essa restrição pode ser mais convenientemente escrita da seguinte maneira:

$$\Delta(x, \hat{x}) = \sum_{j \in \overline{\mathbf{S}}} (1 - x_j) \leq k' = \left(\frac{k}{2}\right). \tag{2.13}$$

O espaço da solução associado ao nó de ramificação atual pode ser particionado por meio da disjunção:

$$\Delta(x, \hat{x}) \leq k \quad \text{ou}$$

$$\Delta(x, \hat{x}) \geq k + 1.$$

2.4.4.1 Trabalhando com soluções de referência inviáveis

A estrutura de ramificação local requer uma solução de referência inicial (viável), que assumimos ser fornecida pelo solucionador. No entanto, para problemas difíceis, até mesmo a definição desta solução pode exigir um tempo de computação excessivo. Neste caso, Fischetti e Lodi (2003) afirmam que é possível considerar a possibilidade de trabalhar com soluções de referência inviáveis. É então aconselhável adotar o mecanismo amplamente usado em otimização que consiste em modificar o modelo PIM adicionando variáveis de folga a (algumas das) restrições, enquanto as penaliza na função objetivo.

2.4.4.2 Trabalhando com variáveis inteiras gerais

A matheurística *Local Branching* é baseada na hipótese de que há um conjunto de variáveis binárias. Porém, Fischetti e Lodi (2003) afirmam que é possível definir restrições de *LB* usando variáveis inteiras.

Considere, então, o seguinte problema (P2):

$$\begin{aligned} (P2)\ \min \quad & c^t x \\ \text{sujeito a:} \quad & Ax \geq b \\ & x_j \in \mathbb{Z}_+,\ \forall j \in \mathbb{I} \\ & r_j \leq x_j \leq u_j,\ \forall j \in \mathbb{I}. \end{aligned}$$

Seja $\hat{x}$ é uma solução factível de referência de $(P2)$. Para um dado parâmetro inteiro positivo k, definimos a vizinhança $N^k(\hat{x})$ como o conjunto das soluções factíveis de (P2), satisfazendo uma restrição adicional de *LB* modificada, definida pela expressão (2.12).

$$\Delta_1(x,\hat{x}) = \sum_{j\in I:\hat{x}_j=r_j} \mu_j(x_j - r_j) + \sum_{j\in I:\hat{x}_j=u_j} \mu_j(u_j - x_j) + \sum_{j\in I:r_j<\hat{x}_j<u_j} \mu_j(x_j^+ + x_j^-) \leq k. \tag{2.14}$$

Nesta expressão, observe que r_j e u_j são valores limitantes para a variável x_j, de modo que $r_j \leq \hat{x}_j \leq u_j$. No qual os valores μ_j representam penalizações, sendo definidos, por exemplo, como:

$$\mu_{\mathbf{j}} = \frac{1}{u_j - r_j},\ \forall\ j \in \mathbb{I}. \tag{2.15}$$

As variáveis x_j^+ e x_j^- requerem a introdução, no modelo PIM, de restrições da seguinte forma:

$$\begin{aligned} & x_j = \hat{x}_j + x_j^+ - x_j^- \\ & x_j^+ \geq 0,\quad x_j^- \geq 0,\quad \forall j \in \mathbb{I},\quad r_j < x_j < u_j. \end{aligned}$$

Alternativamente, pode-se evitar o uso dessas variáveis adicionais, acrescentando-se, em seu lugar, uma das seguintes definições relaxadas:

$$\Delta_2(x, \hat{x}) := \sum_{j \in \mathbb{I}: \hat{x}_j = r_j} \mu_j (x_j - r_j) + \sum_{j \in \mathbb{I}: \hat{x}_j = u_j} \mu_j (u_j - x_j) \leq k' \tag{2.16}$$

ou

$$\Delta_3(x, \hat{x}) := \sum_{j \in \mathbb{I}: \hat{x}_j = r_j} \mu_j (x_j - r_j) \leq k''. \tag{2.17}$$

Dada a solução de referência $\hat{x}$, o espaço de solução associado com o nó de *Local Branching* pode ser particionado, portanto, por meio da disjunção:

$$\Delta_1(x, \hat{x}) \leq k \quad \text{ou}$$

$$\Delta_1(x, \hat{x}) \geq k + 1.$$

Quanto ao parâmetro k de dimensão da vizinhança, a idéia é que o vizinhança $N(\hat{x}, k)$ correspondente para o ramo esquerdo deve ser suficientemente pequena para ser otimizada dentro de um curto período de tempo computacional, mas, por outro lado, deve ainda ser grande o suficiente para provavelmente conter soluções melhores que $\hat{x}$. De acordo com experimentos computacionais, a escolha de k raramente é um problema por si só. Normalmente, segundo aponta Fischetti e Lodi (2003), valores de k na faixa $[10, 20]$ são eficazes, na maioria dos casos, para garantir essas propriedades. Observe que o valor $\Delta_1(x, \hat{x})$ corresponde à distância entre a solução de referência $\hat{x}$ e uma solução qualquer x.

O Algoritmo 3 apresenta o pseudocódigo para o procedimento *Local Branching*. Inicialmente é assumido que se tem uma solução incumbente x^{-1} para o nó 1. A otimização dentro da vizinhança $N(x^{-1}, k)$, dada no nó do ramo esquerdo 2, corresponde à que é realizada através de um esquema de ramificação tático para uma solução ideal na vizinhança, digamos x^{-2}. Essa solução se torna a nova solução atual. O esquema é então reaplicado ao nó do ramo direito 3, onde a exploração de $N(x^{-2},k) \setminus N(x^{-1},k)$ no nó 4 produz uma nova solução incumbente x^{-3}. O nó 5 é então endereçado, o que corresponde ao problema inicial (P) alterado pelas duas restrições adicionais $\Delta(x,x^{-1}) \geq k+1$ e $\Delta(x,x^{-2}) \geq k+1$. No exemplo, o nó do ramo esquerdo 6 produz um subproblema que não contém solução de melhoria. Nesta situação, a adição do restrição $\Delta(x,x^{-3}) \geq k+1$ leva ao nó do ramo direito 7, que é explorado pela ramificação. Observe que a solução fracionária do nó 1 não é necessariamente cortada em ambos os nós filho 2 e 3. O mesmo vale para os nós 3 e 5. De fato, a ramificação local é bem diferente do padrão de corte: não queremos forçar o valor de uma variável fracionária, mas instruímos o método de solução a explorar primeiro algumas regiões promissoras do espaço de solução. Em outras palavras, esperamos encontrar rapidamente soluções cada vez melhores até chegarmos a um ponto em que a ramificação local não pode mais ser aplicado (nó 7, no exemplo), portanto, temos que recorrer a práticas táticas de ramificação para concluir a enumeração.

Algoritmo 3 *Local Branching*

procedure LOCAL BRANCHING – $LB(k_0)$
  $x' \leftarrow$ Solução inicial viável
  $k \leftarrow k_0$
  $h \leftarrow 1$
  while critério de parada = falso **do**
    Adicione a restrição de *Local Branching* $\Delta(x, x') \leq k$
    Resolva o problema de otimização resultante
    if uma solução melhor que $x^{'}$ é encontrada **then**
      Remova a última restrição de *Local Branching* adicionada
      Adicione a restrição de *Local Branching* $\Delta(x, x') \geq k + 1$
      $x' \leftarrow$ solução ótima encontrada
    else
      Remova a última restrição de *Local Branching* adicionada
      $k \leftarrow k + \lceil \max(1, \frac{k}{2}) \rceil$
    end if
    $h++$
    if $h > h_{\max}$ **or** $Time > timeLimit$ **then**
      critério de parada = verdadeiro
    end if
  end while
end procedure

2.4.5 Comparação entre as matheurísticas apresentadas

As matheurísticas *Relax-and-Fix*, *Fix-and-Optimize* e *Local Branching* são técnicas poderosas para a resolução de problemas de otimização complexos, cada uma com suas abordagens e características específicas. Como ilustrado na Tabela 2.2, a *Relax-and-Fix* foca na construção de uma solução viável ao longo do tempo ou estrutura do problema, relaxando variáveis progressivamente, o que é útil em cenários onde não há uma solução inicial viável. Em contraste, a *Fix-and-Optimize* parte de uma solução viável já existente e a refina, otimizando localmente variáveis específicas sem a necessidade de reprocessar toda a solução, o que a torna ideal para problemas onde a construção da solução já foi feita. A *Local Branching*, por sua vez, explora soluções locais ao redor de uma solução inicial, oferecendo flexibilidade para diversificar a busca e não ficarem presos em ótimos locais, como mostrado na tabela. Essa técnica é vantajosa em situações onde a busca por soluções globais pode estar limitada pela natureza do espaço de soluções.

Essas estratégias de solução podem ser utilizadas de forma isolada ou combinadas, dependendo da complexidade do problema em questão, e são especialmente valiosas em problemas de grandes dimensões, nos quais soluções exatas podem ser computacionalmente inviáveis.

Tabela 2.2: Comparação entre *Relax-and-Fix*, *Fix-and-Optimize* e *Local Branching*

Critério	*Relax-and-Fix*	*Fix-and-Optimize*	*Local Branching*
Objetivo	Construir uma solução viável progressivamente	Melhorar uma solução existente	Melhorar a solução ao redor de uma vizinhança
Tipo de exploração	Relaxa variáveis futuras e gradualmente fixa as anteriores	Libera variáveis específicas para otimizar	Explora soluções locais ao redor de uma solução inicial
Método de busca	Construção de uma solução progressiva	Otimização local de uma solução viável	Busca local através de vizinhanças de soluções
Uso principal	Quando não se tem uma solução inicial viável	Para refinar soluções viáveis existentes	Quando se quer explorar vizinhanças de soluções já viáveis
Estratégia	Avança no tempo ou na estrutura do problema	Libera blocos de variáveis e otimiza iterativamente	Explora um conjunto de soluções locais e tenta diversificar a busca
Flexibilidade	Menos flexível, fixa variáveis gradualmente	Flexível, permite otimização local em diferentes partes	Muito flexível, permite explorar uma grande variedade de vizinhanças de soluções
Vantagem principal	Garante uma solução viável inicial	Refina e melhora uma solução viável existente	Diversifica a busca e tem mecanismos para não ficar presos em ótimos locais
Limitação	Pode não ser eficiente em problemas altamente complexos	Depende de uma boa solução inicial	Pode ser menos eficiente em encontrar a melhor solução global
Exemplo de aplicação	Planejamento de produção com capacidade limitada	Ajuste de escalonamento de tarefas	Escalonamento de tarefas em janelas de tempo ou otimização de transporte

Capítulo 3

Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega

Neste capítulo introduzimos definições essenciais para a compreensão do problema em estudo, além de uma revisão detalhada sobre os problemas de corte de estoque com restrições de datas de entrega. Em seguida, apresentamos a modelagem matemática proposta, especificando as variáveis, restrições e função objetivo utilizada para representar o problema de forma precisa. Também discutimos o método de solução adotado, destacando suas principais características e justificativas teóricas. Por fim, analisamos os resultados obtidos a partir dos experimentos realizados, identificamos os principais achados do estudo e sugerimos possíveis direções para pesquisas futuras.

O Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega (1D-PMCE-DE) representa uma variante multiperíodo do 1D-PCE clássico, uma vez que busca atender às demandas de itens, ao mesmo tempo em que reconhece que as restrições de capacidade da máquina podem dificultar o atendimento de pedidos dentro das datas de entrega estipuladas. No contexto de problemas de programação, eventos como antecipação e atraso na produção são considerados prejudiciais e precisam ser minimizados. No caso de problemas Multiperíodo de Corte de Estoque Unidimensional, esses eventos se traduzem em perdas diversas, incluindo multas, redução de receita e aumento de custos de transporte e estoque.

A Seção 3.1 apresenta uma revisão da literatura sobre o Problema de Corte de Estoque com Datas de Entrega. O problema foco de estudo dessa tese é descrito na seção 3.2. A Seção 3.3 descreve a abordagem proposta de solução para o 1D-PMCE-DE. Os resultados computacionais são abordados na Seção 3.4. Por fim, a Seção 3.5 apresenta as conclusões e discussões parciais.

3.1 Revisão dos Problemas de Corte de Estoque Unidimensional com Datas de Entrega

O Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entregas (1D-PMCE-DE) ainda não foi amplamente abordado na literatura. Podemos citar alguns trabalhos como apresentados na introdução dessa tese: Haessler e Tal-

bot (1983), Li (1996), Reinertsen e Vossen (2010), Arbib e Marinelli (2014), Braga et al. (2016a), Braga et al. (2016b), Braga et al. (2015), Arbib e Marinelli (2017), Sezer e Muter (2018). Os modelos propostos em Reinertsen e Vossen (2010),Arbib e Marinelli (2014) e Arbib e Marinelli (2017) são apresentados no Apêndice A.

Recentemente, Wu et al. (2024) abordaram um 1D-PCE com Datas de lançamento e Entrega. O problema abordado é chamado de Problema de Envio de Produto (PEP), e os autores caracterizam o problema como um PCE unidimensional. O artigo apresentou algoritmos baseados na geração de colunas para as versões *offline* e *online* do PSP. No entanto, apesar de incluir datas de entrega, o problema considerado não permite atrasos nem impõe penalidades por antecipação.

Na dissertação de mestrado de Oliveira (2016), são apresentados dois modelos de programação linear inteira voltados para o problema de corte unidimensional com data de entrega (1D-PCE-DE). O primeiro modelo trata de um cenário em que existe apenas um único tipo de objeto disponível em estoque, em quantidade infinita. Por outro lado, o segundo modelo amplia essa abordagem ao considerar diversos tipos de objetos, cada um com quantidade limitada em estoque. Para resolver ambos os modelos, os autores empregam o método Simplex associado à técnica de geração de colunas, uma das abordagens mais consolidadas na literatura para lidar com o problema de corte unidimensional com data de entrega (1D-PCE-DE). Os experimentos computacionais foram conduzidos utilizando o ambiente OPL/CPLEX, com o objetivo de validar os modelos propostos.

Para resolver esses modelos, os autores adotam a técnica de geração de colunas acoplada ao método Simplex, uma abordagem amplamente utilizada na literatura para enfrentar a complexidade combinatória do problema de corte unidimensional. Essa técnica permite lidar eficientemente com o grande número de variáveis envolvidas, gerando iterativamente apenas aquelas colunas (ou padrões de corte) que têm potencial de melhorar a solução corrente.

Os experimentos computacionais foram realizados utilizando o ambiente OPL/CPLEX, com o objetivo de validar a eficácia dos modelos propostos tanto em termos de qualidade de solução quanto de viabilidade computacional. Os resultados obtidos reforçam a adequação da abordagem na resolução do 1D-PCE-DE, especialmente diante de restrições operacionais como o controle de estoques e prazos de entrega.

3.2 Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega

Esta tese propõe um modelo de programação inteira e uma abordagem de solução matheurística para abordar o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega (1D-PMCE-DE). O objetivo descrito para o problema é elaborar uma solução que minimize simultaneamente três componentes principais:

(i) a contagem total de objetos cortados;

(ii) a penalidade associada à antecipação ponderada; e

(iii) a penalidade vinculada ao atraso ponderado.

Este problema representa um cenário em que uma empresa requer itens fabricados a partir de objetos disponíveis em estoque em grandes quantidades para atender a todas as demandas de itens, levando em consideração as datas de entrega associadas a esses pedidos. As datas de entrega são consideradas organizadas em uma ordem não decrescente. Além disso, a capacidade da máquina surge como uma consideração crítica, definida pelo número de objetos que podem ser processados pela máquina até ou antes das datas de Entrega. Presume-se que cada unidade de qualquer padrão de corte necessita de uma unidade de tempo para conclusão, sendo esse requisito de tempo consistente em diversos padrões de corte.

3.2.1 Contribuições relativas ao 1D-PMCE-DE

Essa tese apresenta as seguintes contribuições específicas no que diz respeito ao Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entregas:

(i) Até onde sabemos, pela primeira vez, o modelo matemático 1D-PMCE-DE inclui a minimização conjunta da antecipação e do adiantamento por meio de uma abordagem de penalização, para o caso com data de entrega;

(ii) Propomos uma abordagem matheurística de três fases, que integra e explora as vantagens da técnica de geração de colunas, a heurística *Relax-and-Fix* e a matheurística *Local Branching*;

(iii) O procedimento de solução proposto permite encontrar soluções viáveis de alta qualidade com custo computacional significativamente menor em comparação ao uso do solucionador CPLEX.

3.2.2 Descrição geral do 1D-PMCE-DE

Considere um cenário em que um estoque compreende uma coleção de objetos grandes, todos compartilhando o mesmo comprimento L, juntamente com um conjunto diverso de n tipos de itens menores caracterizados por comprimentos ℓ_i, $(i = 1,2,\ldots,n)$, a serem programados em uma operação de corte para pedidos. O horizonte de planejamento se divide em períodos de tempo discretos, denotados como $t = 1,2,\ldots,M$, e, para cada período t, existe uma demanda b_i^t do tipo de item i com uma data de entrega correspondente d^t, no qual deve atender a um conjunto de tipos de itens, diferentemente do que foi apresentado por Arbib e Marinelli (2014) que trabalham com entrega de tipos de itens. Seja a_{ij}^t o número de itens do tipo i produzidos no período t por meio da aplicação do padrão de corte j, $(j = 1,2,\ldots,N)$.

As variáveis de decisão x_j^t informam o número de objetos cortados no período t de acordo com o padrão de corte j. Defina também T^t como as variáveis de decisão que mostram o tempo de atraso do pedido no período t; e E^t como as variáveis de decisão associadas ao tempo de antecipação do pedido no período t. Além disso, considere a variável de decisão q^t que representa o tempo de processamento, ou seja, o tempo gasto para cortar o pedido em um período t.

Em operações de corte, atender todos os pedidos precisamente em suas datas de entrega pode nem sempre ser viável. Portanto, a função objetivo inclui um fator de penalidade por unidade de tempo de atraso (θ^t). Por outro lado, se o processo de

corte resultar em pedidos sendo atendidos antes de suas datas de entrega, um fator de penalidade por unidade de tempo de antecipação (β^t) também é considerado.

3.2.3 Modelagem Proposta

Usando esta notação, o modelo matemático do 1D-PMCE-DE abordado aqui é formulado como:

$$\min \quad \sum_{t=1}^{M}\sum_{j=1}^{N} x_j^t + \sum_{t=1}^{M} \theta^t T^t + \sum_{t=1}^{M} \beta^t E^t \tag{3.1}$$

$$\text{sujeito a:} \quad T^t \geq q^t + T^{t-1} - E^{t-1} - (d^t - d^{t-1}), \quad \forall\, t, \tag{3.2}$$

$$E^t \geq d^t - d^{t-1} - T^{t-1} + E^{t-1} - q^t, \quad \forall t, \tag{3.3}$$

$$q^t = \sum_{j=1}^{N} x_j^t, \quad \forall\, t, \tag{3.4}$$

$$\sum_{j=1}^{N} a_{ji}^t x_j^t \geq b_i^t \qquad \forall t,\, i \tag{3.5}$$

$$d^0 = 0, \tag{3.6}$$

$$E^0 = 0, T^0 = 0, \tag{3.7}$$

$$E^t, T^t, q^t \in \mathbb{Z}_+, \quad \forall t, \tag{3.8}$$

$$x_j^t \in \mathbb{Z}_+, \forall j, t. \tag{3.9}$$

A função objetivo (3.1) minimiza a soma de três termos: o número total de objetos cortados, as penalidades por atraso e as penalidades por entregas antecipadas. As restrições (3.2)-(3.3) indicam os tempos de atraso e antecipação. O conjunto de restrições (3.4) calcula o tempo a ser gasto na produção de pedidos em cada um dos períodos, de acordo com o número de objetos que foi cortado naquele determinado período de tempo t. O conjunto de restrições (3.5) garante que a demanda do período deve ser satisfeita. A restrição (3.6) define a condição de data de entrega inicial para $t = 0$. As restrições (3.7) representam as condições de contorno iniciais, que afirmam que não há atraso ou antecipação de pedidos no início do processo de corte. Expressões (3.8)-(3.9) representam o domínio das variáveis de decisão.

3.2.4 Exemplo numérico

A seguir, ilustramos um exemplo numérico conciso do Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega (1D-PMCE-DE). A solução é obtida usando a seguinte abordagem: (i) empregando o método de geração de colunas introduzido por Gilmore e Gomory (1961) na relaxação de programação linear do problema; e (ii) usando as colunas geradas, um solucionador de propósito geral é aplicado para obter a solução final.

Exemplo 1. *Suponha que há disponível em estoque objetos de comprimento $L = 115$ para atender às demandas de itens fornecidas na Tabela 3.1 que mostra as solicitações da empresa.*

Tabela 3.1: Exemplo Numérico: demanda de itens em cada período.

	Demanda do item (comprimento)					
Período	1(12)	2(16)	3(22)	4(34)	5(56)	Data de Entrega
1	12	34	52	67	54	83
2	45	32	34	87	54	107
3	13	45	67	43	23	123

A solução para o exemplo numérico é detalhada nas Tabelas 3.2 e 3.3. Na Tabela 3.2, os padrões de corte identificados durante a fase de geração de colunas e seus períodos correspondentes são descritos. Cada linha na tabela representa um padrão de corte distinto, com uma observação de que cinco padrões de corte são aplicados em cada período. Nesta solução específica, os fatores de penalidade $\theta = 1$ e $\beta = 1$ são empregados para atraso e antecipação, respectivamente. A coluna # Objetos na Tabela 3.2 especifica o número de objetos designados para corte em cada padrão, e a tabela também fornece o tempo de processamento cumulativo para os pedidos alinhados com cada período respectivo.

Tabela 3.2: Soluções de padrões de corte e seus períodos associados para o exemplo numérico usando o 1D-PMCE-DE.

Períodos	Padrões de Corte					# Objetos
	1	2	3	4	5	
1	0	0	0	0	2	27
	1	0	0	3	0	12
	0	0	2	2	0	15
	0	3	3	0	0	8
	0	5	0	1	0	2
	Tempo de processamento					64
2	0	0	0	0	2	23
	1	0	0	3	0	27
	0	3	3	0	0	11
	3	0	1	0	1	2
	2	0	0	1	1	6
	Tempo de processamento					69
3	0	0	5	0	0	4
	0	0	0	0	2	12
	1	0	0	3	0	13
	0	0	2	2	0	2
	0	3	3	0	0	15
	Tempo de processamento					46

O atraso e a antecipação das solicitações são obtidos abaixo para todos os períodos:

(i) Atraso ou antecipação para o Período 1*: como, inicialmente, não há atraso ($T^0 = 0$) nem antecipação ($E^0 = 0$), apenas o tempo de processamento e a data de vencimento devem ser levados em consideração:*

$$\begin{aligned} T^1 &\geq q^1 + T^0 - E^0 - (d^1 - d^0) = 64 + 0 - 0 - (83 - 0) = -19, \\ E^1 &\geq d^1 - d^0 - T^0 + E^0 - q^1 = 83 - 0 - 0 + 0 - 64 = 19. \end{aligned}$$

Portanto, neste período houve uma antecipação de 19*.*

(ii) Atraso ou antecipação para o Período 2: já que $T^1 = 0$ *e* $E^1 = 19$*:*

$$\begin{aligned} T^2 &\geq q^2 + T^1 - E^1 - (d^2 - d^1) = 69 + 0 - 19 - (107 - 83) = 26, \\ E^2 &\geq d^2 - d^1 - T^1 + E^1 - q^2 = 107 - 83 - 0 + 19 - 69 = -26. \end{aligned}$$

Portanto, o atraso foi igual a 26.

(iii) Atraso ou antecipação para o Período 3: já que $T^2 = 26$ *e* $E^2 = 0$*:*

$$\begin{aligned} T^3 &\geq q^3 + T^2 - E^2 - (d^3 - d^2) = 46 + 26 - 0 - (123 - 107) = 56, \\ E^3 &\geq d^3 - d^2 - T^2 + E^2 - q^3 = 123 - 107 - 26 + 0 - 46 = -56. \end{aligned}$$

Portanto, o atraso foi igual a 56.

Tabela 3.3: Decomposição do valor da função objetivo para cada período do exemplo numérico.

Períodos	Componentes da função objetivo		
	#Objetos	Atraso	Antecipação
1	64	–	19
2	69	26	–
3	46	56	–
Valor da função objetivo			280

A Tabela 3.3 mostra, para cada período, os valores dos componentes da função objetivo. O valor ótimo da função objetivo é 280*. Observando os resultados para este exemplo,* 179 *objetos foram cortados (*$64 + 69 + 46 =$ *tempo de corte), com um atraso total de* 82 *(*$26 + 56$*) e uma antecipação total de* 19*.*

3.3 Abordagem proposta para solução do 1D-PMCE-DE

Devido à complexidade do PCE, uma vez que há um grande número de padrões de corte possíveis, a relaxação linear do problema é geralmente resolvida pelo método de geração de colunas proposto por Gilmore e Gomory (1961) para gerar padrões de corte. Este método cria e resolve um problema auxiliar, chamado problema de precificação, para determinar novas colunas que podem melhorar a solução atual. Nesta abordagem, o problema inicial é relaxado, e a solução relaxada resultante é frequentemente fracionária, exigindo, portanto, um método heurístico para encontrar uma solução inteira viável.

Esta seção descreve em detalhes a abordagem proposta para resolver o 1D-PMCE-DE. O procedimento de solução proposto consiste em três fases: na primeira fase, apresentada na Subseção 3.3.1, o problema relaxado linear é resolvido pelo método Simplex com geração de colunas, obtendo uma solução que pode não ser viável para o 1D-PMCE-DE. A segunda fase, mostrada na Subseção 3.3.2, encontra uma solução inteira para o problema aplicando a heurística *Relax-and-Fix* e usando, como colunas, as encontradas na fase anterior. A terceira fase, descrita na Subseção 3.3.3, visa melhorar, se possível, a solução inteira encontrada na fase anterior usando a matheurística *Local Branching*. O Algoritmo 4 resume a abordagem de três fases proposta para resolver o 1D-PMCE-DE.

Algoritmo 4 Abordagem de três fases proposta para resolver o 1D-PMCE-DE.

procedure ABORDAGEM DE TRÊS FASES PARA RESOLVER O 1D-PMCE-DE

Fase 1 - Encontrando uma solução inicial via método de geração de colunas

1. Relaxe as restrições de integralidade do modelo
2. Aplique o procedimento de geração de colunas

Fase 2 - Obtendo soluções inteiras via heurística *Relax-and-Fix*

1. Retorne a integralidade das variáveis originais do modelo
2. Divida o modelo em subproblemas divididos pelo período de tempo $t = 1, \ldots, M$
3. Aplique a heurística *Relax-and-Fix*

Fase 3 - Melhorando a solução encontrada via matheurística *Local Branching*

1. Obtenha a solução encontrada por *Relax-and-Fix*
2. Aplique *Local Branching* para melhorar a solução

end procedure

3.3.1 Fase 1 – Encontrando soluções iniciais: Método de Geração de Colunas

Em problemas práticos de Corte de Estoque, como cada coluna está associada a um padrão de corte, o número de variáveis (colunas) é muito maior que o número de restrições (linhas). A cada iteração do método Simplex, temos que determinar uma nova coluna para entrar na base. Este problema se torna numericamente impraticável. Assim, para superar esta dificuldade, Gilmore e Gomory (1961, 1963) propuseram uma técnica de geração de colunas, chamada Método Simplex com Geração de Colunas, que é muito eficiente para resolver o problema.

Este método encontra, a cada iteração, um novo padrão de corte, adicionando uma nova coluna à matriz inicial do problema. Inicialmente, temos padrões homogêneos, de modo que a matriz do problema é uma matriz diagonal, na qual elementos não nulos correspondem ao número máximo de vezes que cada item pode ser cortado no objeto. A ideia é simples: no método Simplex, ao procurar uma coluna que melhore a solução atual, em vez de usar métodos do tipo enumeração, resolvemos o problema da mochila como um problema auxiliar. A cada iteração de cada período, um problema da mochila é resolvido: os coeficientes da função objetivo são os coeficientes duais do problema de corte, e os coeficientes de restrição são os comprimentos dos itens.

Nesta tese, o problema mestre do procedimento de geração de colunas é dado pelo modelo matemático 1D-PMCE-DE (3.1)-(3.9). O problema mestre restrito corresponde a uma relaxação de programação linear do problema mestre com um subconjunto de padrões de corte. Inicialmente, o problema mestre restrito tem apenas padrões de corte homogêneos máximos. Para gerar os padrões de corte, definimos o subproblema SP^t, dado pelo seguinte problema da mochila, para cada período $t = 1, \ldots, M$ no horizonte de planejamento:

$$(SP^t) \qquad \min \qquad c^j - \lambda^t - \sum_{i=1}^{n} \pi_i^t \alpha_i^t \tag{3.10}$$

$$\text{sujeito a:} \quad \sum_{i=1}^{n} \ell_i \alpha_i^t \leq L \tag{3.11}$$

$$\alpha_i^t \in \mathbb{Z}_+. \tag{3.12}$$

Neste problema de mochila, $c^j = 1$ é o custo de cortar um objeto em estoque; L é o comprimento do objeto em estoque; λ^t e π_i^t são as variáveis duais do problema mestre relacionadas às restrições (3.4) e (3.5), respectivamente.

O Algoritmo 5 descreve a fase 1 da abordagem proposta. Nesta fase, uma solução da relaxação linear da formulação (3.1)-(3.9) é encontrada por meio do procedimento de geração de colunas.

Algoritmo 5 Método de geração de colunas para o 1D-PMCE-DE.

```
1: procedure GERAÇÃO DE COLUNAS PARA O 1D-PMCE-DE
2:     Relaxe as condições de integralidade do modelo
3:     Definir os padrões de corte homogêneos
4:     while uma nova coluna pode ser adicionada ao problema mestre restrito do
5:         for t ← 1 ate M do
6:             Resolva o problema mestre restrito através do método Simplex
7:             Resolve o SP^t
8:             if ν(SP^t) < 0 then
9:                 Adicione uma nova coluna
10:            else
11:                Nenhuma coluna pode ser adicionada
12:            end if
13:        end for
14:    end while
15: end procedure
16: return valor de x_j^t, S_j^t, e_i^t, E^t, e T^t.
```

No procedimento de geração de colunas, começamos com colunas homogêneas máximas, o que significa que $a = [0, \ldots, 0, \lfloor L/\ell_i \rfloor, 0, \ldots, 0]$. Dessa forma, o processo começa com o número de colunas igual ao número de itens a serem cortados.

3.3.2 Fase 2 – Obtenção de soluções inteiras: heurística *Relax-and-Fix*

O algoritmo proposto em Gilmore e Gomory (1961) é usado para resolver o problema de corte relaxado, mantendo, assim, as colunas geradas neste processo. Como nem todas as soluções geradas serão inteiras, uma heurística é usada para obter contrapartes inteiras. A heurística escolhida é a heurística *Relax-and-Fix* (ver Wolsey (1998)) que se baseia na ideia de dividir o problema com variáveis inteiras em subproblemas mais simples. A cada passo, um submodelo, derivado do problema original, é resolvido.

Nesta heurística, o conjunto de índices de variáveis inteiras, dado por Q, é particionado da seguinte forma: defina P subconjuntos distintos Q^r, $r = 1, \ldots, P$, nos quais o valor P determina o número de iteração da heurística, e $Q = Q^1 \cup Q^2 \cup \ldots \cup Q^P$. Na r-ésima iteração, apenas as variáveis com índices no conjunto Q^r são definidas como inteiras. As variáveis inteiras restantes são relaxadas, e o submodelo resultante é resolvido. Se o submodelo for inviável, então a heurística é encerrada, pois não é possível encontrar uma solução viável para o problema original com as variáveis associadas aos conjuntos Q^i, $i = 1, \ldots,(r-1)$, fixadas nos valores atuais. Se o submodelo for viável, as variáveis associadas ao conjunto Q^r são fixadas em seu valor atual, e o processo se repete para os outros conjuntos. Como há uma diminuição no número de variáveis inteiras, também há uma redução no custo computacional, mesmo com a heurística resolvendo vários subproblemas em vez de apenas um.

Uma questão importante para o algoritmo é como definir as partições de variáveis, pois a partição do conjunto de variáveis afeta o grau de dificuldade da solução dos submodelos. A partição escolhida dos conjuntos está relacionada aos períodos de tempo: cada período t representa um submodelo, e as variáveis x_j^t associadas a esse período são definidas como inteiras. Ao final desta etapa, obteve-se uma solução para todas as x_j^t, variáveis inteiras do modelo. Como as variáveis de atraso T^t e as variáveis de antecipação E^t estão associadas às variáveis de corte, determina-se uma solução inteira para essas variáveis.

Portanto, decompomos as variáveis de decisão x_j^t de modo que Q^1 se relacione com todas as x_j^1 variáveis associadas ao período de tempo 1; Q^2 se relacione com todas as x_j^2 variáveis associadas ao período de tempo 2, e assim por diante. O primeiro problema de Programação Linear Inteira Misto (PLIM) resultante, denominado PRF^1, é dado por:

$$(PRF^1) \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N} x_j^t + \sum_{t=1}^{M} \theta^t T^t + \sum_{t=1}^{M} \beta^t E^t \tag{3.13}$$

$$\text{sujeito a:} \quad (3.2) - (3.7) \tag{3.14}$$

$$E^t, T^t, q^t \geq 0, \forall t, \tag{3.15}$$

$$x_j^t \in \mathbb{Z}_+ \ , \quad \forall t \in Q^1, \forall j \tag{3.16}$$

$$x_j^t \in \mathbb{R}_+ \ , \quad \forall t \in Q - Q^1, \forall j, \tag{3.17}$$

em que a integralidade das variáveis de decisão x_j^t associadas a Q^1 são impostas, e a restrição de integralidade é relaxada em todas as outras partições. A solução do PRF^1 produz valores inteiros para o conjunto de variáveis associadas a Q^1.

Assim, para $2 \leq r \leq P$, no subproblema (PRF^r), as variáveis $x_j^t \in Q^1 \cup Q^2 \cup \ldots \cup Q^{r-1}$ são fixadas ($x_j^t = x_{j,r-1}^t$), de acordo com as soluções obtidas na solução dos subproblemas anteriores; as restrições de integralidade para as variáveis associadas a Q^r são adicionadas; e as variáveis associadas a $Q - (Q^1 \cup Q^2 \cup \ldots \cup Q^r)$ são relaxadas.

Portanto:

$$(PRF^r) \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N} x_j^t + \sum_{t=1}^{M} \theta^t T^t + \sum_{t=1}^{M} \beta^t E^t \tag{3.18}$$

$$\text{sujeito a:} \quad (3.2) - (3.7), (3.15) \tag{3.19}$$

$$x_j^t = x_{j,r-1}^t \ , \quad \forall t \in \left(Q^1 \cup Q^2 \cup \ldots \cup Q^{r-1}\right), \forall j \tag{3.20}$$

$$x_j^t \in \mathbb{Z}_+ \ , \quad \forall t \in Q^r, \forall j \tag{3.21}$$

$$x_j^t \in \mathbb{R}_+ \ , \quad \forall t \in Q - \left(Q^1 \cup Q^2 \cup \ldots \cup Q^r\right), \forall j. \tag{3.22}$$

O Algoritmo 6 mostra o pseudocódigo para a Fase 2. Nesta fase, uma solução inteira é obtida por meio da heurística *Relax-and-Fix*.

Algoritmo 6 *Relax-and-Fix* - RF

1: **procedure** RELAX-AND-FIX – RF
2: Escolha uma decomposição disjuntiva para o conjunto de variáveis inteiras $Q = Q^1 \cup Q^2 \cup \ldots \cup Q^P$.
3: Relaxe a integralidade de todas as variáveis inteiras.
4: **for** $r \leftarrow 1$ **ate** P **do**
5: Torne as variáveis associadas ao subconjunto Q^r inteiras.
6: Resolver o problema (PRF^r).
7: Seja x_j^r e S_j^r a solução ótima do problema (PRF^r).
8: Fixe as variáveis x_j e S_j associadas ao subconjunto Q^r, de forma que essas variáveis não devem ser alteradas nos próximos subproblemas.
9: **end for**
10: **end procedure**

3.3.3 Fase 3: Melhorando a solução encontrada – Matheurística *Local Branching*

Nesta subseção discutimos a aplicação da matheurística *Local Branching* para aprimorar a solução inteira viável inicialmente derivada da heurística *Relax-and-Fix*. A matheurística *Local Branching*, conforme introduzida por Fischetti e Lodi (2003), exerce controle estratégico sobre um solucionador de problema de PLIM, direcionando estrategicamente a exploração da vizinhança ao redor de soluções viáveis já identificadas, seja parcial ou totalmente. A essência da ramificação local está em sua abordagem de considerar uma determinada solução viável como um ponto de referência, visando descobrir uma solução melhorada dentro de uma proximidade definida. Nesta metodologia, uma restrição de ramificação local é introduzida para estabelecer a distância máxima permitida k entre as duas soluções.

Nesta tese, a restrição de ramificação local é aplicada à soma das variáveis de decisão x_j^t, em que t varia de 1 a M e j de 1 a N. Consequentemente, considerando um valor viável $x_j^{'t}$ para as variáveis x_j^t e um parâmetro inteiro positivo especificado k, definimos a vizinhança desta solução, com um tamanho de k, como o conjunto de soluções viáveis que obedecem a uma restrição adicional de ramificação local.

Considere que as variáveis x_j^t são limitadas, sendo $l_j^t \leq x_j^t \leq u_j^t$, para todo j e t. Além disso, definimos $J^l(x') = \{j \in \{1,2,\ldots,N\} : x_j^{'t} = l_j^t\}$ e $J_u(x') = \{j \in \{1,2,\ldots,N\} : x_j^{'t} = u_j^t\}$. Então, de acordo com Fischetti e Lodi (2003), uma restrição de ramificação local pode ser definida como:

$$\Delta(x,x') := \sum_{t=1}^{M} \sum_{j \in J^l(x')} \mu_j^t (x_j^t - l_j^t) + \sum_{t=1}^{M} \sum_{j \in J_u(x')} \mu_j^t (u_j^t - x_j^t) \leq k. \tag{3.23}$$

O valor de μ_j^t, como sugerido por Fischetti e Lodi (2003), é dado por:

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

Os valores para os limites inferior e superior, l_j^t e u_j^t, 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:

$$\Delta(x,x') \leq k \qquad \text{ou} \qquad \Delta(x,x') \geq k+1. \tag{3.24}$$

O Algoritmo 3 descreve o pseudocódigo para o procedimento de ramificação local. Este algoritmo começa com a formulação original, e a heurística *Relax-and-Fix* é usada para obter uma solução inicial viável. Em seguida, uma vizinhança dessa solução é explorada usando um solucionador de propósito geral. Se uma solução melhor for encontrada, uma ramificação é aplicada. Um novo problema é gerado, alterando a restrição que define a vizinhança e adicionando uma nova restrição de ramificação ao redor do novo ponto. Nessa ramificação, o procedimento é repetido até que uma solução melhor seja encontrada. Além disso, a cada iteração em que uma solução melhor não é encontrada, o valor de k é incrementado.

Algoritmo 7 *Local Branching* aplicada ao 1D-PMCE-DE

procedure LOCAL BRANCHING – LB(k^0)
  $x' \leftarrow$ Solução inicial viável
  $k \leftarrow k^0$
  $h \leftarrow 1$
  while critério de parada = falso **do**
    Adicione a restrição de *local branching* $\Delta(x, x') \leq k$ para a formulação 1D-PMCE-DE
    Resolva o problema de otimização resultante
    if uma solução melhor que $x^{'}$ é encontrada **then**
      Remova a última restrição de *local branching* adicionada
      Adicione a restrição de *local branching* $\Delta(x, x') \geq k + 1$
      $x' \leftarrow$ solução ótima encontrada
    else
      Remova a última restrição de *local branching* adicionada
      $k \leftarrow k + \lceil \max(1, \frac{k}{2}) \rceil$
    end if
    $h \leftarrow h + 1$
    if $h > h_{\max}$ **or** $Time > timeLimit$ **then**
      critério de parada = verdadeiro
    end if
  end while
end procedure

3.4 Experimentos Computacionais para o 1D-PMCE-DE

Esta seção apresenta os resultados dos experimentos computacionais realizados para avaliar o modelo matemático proposto do 1D-PMCE-DE, mostrado por Expressões (3.1)-(3.9) na Subseção 3.2.3. Além disso, os resultados abrangem a aplicação do procedimento de solução heurística descrito na Seção 3.3. Os testes computacionais foram executados em um computador com o Sistema Operacional Linux-Ubuntu, equipado com Processador Intel Core i7 - 7ª geração, 8 GB de RAM e Disco Rígido de 2 TB. Foi utilizado o pacote `Concert Technology` CPLEX 12.7.1 com a linguagem `C++`. Nesta tese, incorporamos dois conjuntos de instâncias geradas por meio do CUTGEN1, um gerador de problemas Multiperíodo de Corte de Estoque Unidimensional proposto por Gau e Wäscher (1995).

Para analisar o impacto do fator de penalidade de atraso (θ) e do fator de penalidade de antecipação (β) na função objetivo do 1D-PMCE-DE, dois conjuntos distintos de valores são empregados para esses fatores. No primeiro cenário, assume-se que tanto as penalidades de atraso quanto as de antecipação têm igual significância. Este cenário é denotado por (θ, β) = (1,1). No entanto, reconhecendo o cenário prático em que os custos associados ao atraso na entrega geralmente superam os custos incorridos devido ao estoque prematuro de itens e devido à produção antecipada, o segundo conjunto de valores visa capturar esse desequilíbrio, com fatores de penalidade

definidos como $(\theta,\beta) = (10, 1)$.

Além disso, para aumentar a complexidade dos cenários e facilitar uma avaliação completa da qualidade do modelo 1D-PMCE-DE apresentado e do procedimento de solução proposto, três conjuntos distintos de datas de entrega são levados em consideração para os experimentos computacionais. Esses conjuntos, semelhantes às capacidades de corte, são gerados de acordo com a seguinte regra de construção:

$$d^t = \gamma^t \left(\frac{\sum_{i=1}^{n} l_i b_i^t}{L} \right). \tag{3.25}$$

O conjunto inicial de datas de entrega, denotado como D1, é construído seguindo a regra de construção (3.25) com $\gamma^t = 1{,}1 \ \ \forall\, t$. Passando para o segundo conjunto, chamado D2, ele é formado pela implementação da regra de construção (3.25) com $\gamma^t = 1{,}1$ para $t = 1$ e $\gamma^t = 1$ para $t = 2, \ldots, M$. Finalmente, o terceiro conjunto de datas de entrega, conhecido como D3, é criado usando a regra de construção (3.25) com $\gamma^t = 1{,}1$ para $t = 1$ e $\gamma^t = 1{,}3$ para $t = 2, \ldots, M$. Em todas as tabelas apresentadas a seguir, a coluna "DE" identifica as datas de entregas avaliadas.

Todas as instâncias e os códigos utilizados nos experimentos computacionais estão acessíveis no repositório público `htqs://github.com/Elisama-Oliver/Pcede`.

Para apresentar os resultados ao longo desta seção, a seguinte notação é adotada para representar o procedimento de solução:

- CG: procedimento de Geração de Colunas relacionado à Fase 1 do Algoritmo 4;
- RF: procedimento de *Relax-and-Fix* relacionado à Fase 2 do Algoritmo 4;
- LB: procedimento *Local Branching* relacionada à Fase 3 do Algoritmo 4;
- CP: solução do programa inteiro via aplicação direta do solucionador CPLEX após a Fase 1 do Algoritmo 4.

Estabelecemos como critérios de parada para o procedimento LB, o seguinte: o tempo limite de execução em 900 segundos e definindo o valor máximo para o tamanho da vizinhança k em $0{,}1 \times N \times M$. Para o solver CPLEX limitamos o tempo de execução em 3600 segundos. Vale a pena notar que, nos casos em que o CPLEX identifica a solução dentro de um período de tempo inferior a 3600 segundos, esta solução pertence à solução ótima para o conjunto de padrões de corte gerados por meio da geração de colunas (Fase 1). Isso ocorre porque atingir a solução ótima para todo o problema exigiria a aplicação de um algoritmo *branch-and-price* para gerar todos os padrões de corte necessários, e isso não foi feito.

Seja $\nu(Alg)$ o valor da função objetivo associado ao procedimento Alg. Os valores de *gap* representam a diferença percentual na evolução do valor da função objetivo do procedimento adotado:

$$gap^1 = 100\, \frac{\nu(RF) - \nu(LB)}{\nu(LB)}, \tag{3.26}$$

$$gap^2 = 100\, \frac{\nu(CP) - \nu(LB)}{\nu(LB)}. \tag{3.27}$$

O valor gap_1 ilustra a evolução percentual do valor da função objetivo, comparando a solução inteira produzida por RF com a solução inteira aprimorada gerada por LB. Instâncias em que os valores são zero indicam nenhuma melhoria nos valores da função objetivo após a aplicação da respectiva fase. Por outro lado, valores positivos mais altos significam uma contribuição mais impactante para a fase associada. Simultaneamente, gap_2 representa a diferença relativa entre o valor alcançado pela fase final do procedimento proposto (LB) e o valor determinado pelo solucionador CPLEX.

3.4.1 Resultados para o primeiro conjunto de instâncias

O conjunto inicial de instâncias, foi gerado de acordo com os padrões do pacote CUTGEN1 (Gau e Wäscher, 1995). As instâncias foram geradas pela autora e escrita em linguagem C++, no qual elas estão categorizadas em duas classes, compreendendo 40 problemas cada, somando um total de 80 instâncias. As características deste conjunto de instâncias, distinguidas pelo número de itens e períodos considerados, são apresentadas na Tabela 3.4. Observe que, em todas as instâncias, o comprimento de um objeto de estoque permanece constante em $L = 1000$.

Tabela 3.4: Características do primeiro conjunto de instâncias com comprimento de objeto de $L = 1000$.

Classes	Período	Itens	Instâncias	Comprimento do item	Demanda
C1	3	10	40	[0,03L , 0,6L]	[1,44]
C2	6	20	40	[0,03L , 0,6L]	[1,44]

A Tabela 3.5 apresenta os valores médios da função objetivo resultantes da aplicação do procedimento proposto ao primeiro conjunto de instâncias, juntamente com o tempo de execução médio correspondente e *gaps* médios conforme definido pelas Expressões (3.26) e (3.27). Além disso, gap_{CP} inclui o valor final médio de *gap* fornecido pelo solucionador CPLEX, e $\#Opt$ denota o número de soluções ótimas comprovadas em cada classe. O tempo de execução cumulativo necessário para obter cada solução em segundos é representado pela coluna Tempo(s), com a coluna LB indicando o tempo de execução médio final gasto pelo procedimento proposto em Algorithm 4. O valor da função objetivo em si constitui a soma de três termos: a contagem de objetos cortados e a ponderação dos valores de atraso e antecipação.

Analisando a Tabela 3.5, uma observação notável é que, para todas as instâncias vinculadas às datas de entrega D1 e D2, o solucionador CPLEX esgota o limite de tempo (3600 segundos), produzindo soluções sem otimalidade comprovada. Por outro lado, a abordagem proposta, conforme indicado pela coluna gap^2, atinge precisamente os mesmos valores objetivos que aqueles obtidos pelo solucionador CPLEX, mas dentro de um tempo de execução consideravelmente menor. Além disso, o *Local Branching* não encontra soluções de valor menor que as soluções identificadas pelo método *Relax-and-Fix*. Como é evidente na coluna $\#Opt$, o solucionador CPLEX não encontra nenhuma solução ótima inteira comprovada para instâncias associadas às datas de entrega D1 e D2. A coluna dessa tabela descrita como Colunas apresenta o número de colunas que foram geradas apenas na fase inicial (fase 1) Para instâncias associadas às datas de entrega D3, excluindo D3-(1,1)-C1, a abordagem proposta

Tabela 3.5: Valores médios da função objetivo, tempo médio de execução e valor médio de *gap* para a solução obtida usando a abordagem proposta para o primeiro conjunto de instâncias.

Class				Função Objetivo				Tempo(s)				*Gap*			
DE	(θ,β)	Classes	Colunas	CG	RF	LB	CP	CG	RF	LB	CP	gap^1	gap^2	gap_{CP}	$\#Opt$
D1	(1,1)	C1	21,95	132,70	135,77	135,77	135,77	3,30	4,65	6,75	3600,00	0,00	0,00	1,72	0
		C2	39,51	531,78	542,97	542,97	542,97	31,43	63,66	678,90	3600,00	0,00	0,00	1,96	0
	(10,1)	C1	21,78	138,53	145,07	145,07	145,07	3,28	4,64	6,80	3600,00	0,00	0,00	2,71	0
		C2	39,46	544,31	560,22	560,22	560,22	28,44	50,85	744,32	3600,00	0,00	0,00	2,51	0
D2	(1,1)	C1	23,14	119,02	120,95	120,95	120,95	1,87	5,71	35,50	3600,00	0,00	0,00	1,57	0
		C2	38,31	499,72	504,85	504,85	504,85	7,80	79,37	772,60	3600,00	0,00	0,00	1,89	0
	(10,1)	C1	21,82	127,32	135,45	135,45	135,45	1,95	25,3	56,75	3600,00	0,00	0,00	2,48	0
		C2	40,51	522,32	534,65	534,65	534,65	10,02	88,25	789,20	3600,00	0,00	0,00	2,48	0
D3	(1,1)	C1	4,07	130,18	153,82	153,57	153,42	0,43	1,04	1,22	1523,66	0,16	-0,09	0,75	26
		C2	5,19	599,71	653,22	653,22	653,22	3,79	7,46	11,05	1938,18	0,00	0,00	0,27	20
	(10,1)	C1	6,24	151,22	205,80	201,80	201,80	1,02	1,35	1,69	1427,71	1,98	0	1,75	27
		C2	5,17	611,77	1055,07	1051,85	1051,85	2,98	9,16	12,35	2012,25	0,31	0,00	1,74	18

atinge, como a coluna gap^2 indica, precisamente os mesmos valores objetivos que aqueles obtidos pelo solucionador CPLEX, em um tempo de execução relativamente curto. No entanto, com base na coluna $\#Opt$, o solucionador CPLEX obtém um total de 91 soluções ótimas comprovadas, dentre 160 instâncias. Para D3-(1,1)-C1, o solucionador CPLEX supera a abordagem proposta, produzindo um valor de função objetivo médio superior. Essas descobertas sugerem que lidar com instâncias com uma data de entrega mais relaxada (D3) é comparativamente mais fácil que instâncias nas demais datas de entrega.

Tabela 3.6: Número de soluções melhoradas pelo *Local Branching* quando comparado à solução encontrada pelo *Relax-and-Fix*, usando o primeiro conjunto de instâncias.

Classes			
DE	(θ,β)	Classes	#n. de soluções
D3	(1,1)	C1	5
		C2	0
	(10,1)	C1	11
		C2	3

A Tabela 3.6 mostra o número de soluções melhoradas pelo *Local Branching*. A eficácia do *Local Branching*, ou seja, as instâncias nas quais as soluções encontradas pelo *Local Branching* são melhores que as soluções encontradas pelo *Relax-and-Fix*, pode ser vista em todos os cenários, exceto D3-(1,1)-C2. Para esse conjunto de instâncias, o *Local Branching* é mais eficaz quando um pequeno número de colunas é gerado.

A Tabela 3.7 detalha o número médio de objetos de corte, atraso e antecipação individualmente. Esses valores são apresentados para cada fase da abordagem proposta, bem como para o solucionador CPLEX. Em geral, a comparação dos pares ($\theta = 1$,$\beta = 1$) e ($\theta = 10$,$\beta = 1$) revela um impacto mínimo dos fatores de penalidade de atraso e antecipação em todas as instâncias, independentemente das datas de Entrega e classes. A pequena variação nos valores implica um comportamento de compensação desses termos dentro da função objetivo.

Tabela 3.7: Cada termo de valor médio da função objetivo da solução obtida pela metodologia proposta usando o primeiro conjunto de instâncias.

Classes			Objetos				Atraso				Adiantamento			
DE	(θ, β)	Classes	CG	RF	LB	CP	CG	RF	LB	CP	CG	RF	LB	CP
D1	(1,1)	C1	130,09	131,25	131,25	131,25	1,75	2,15	2,15	2,15	0,89	2,07	2,07	2,07
		C2	521,11	526,55	526,55	526,55	3,20	5,92	5,92	5,92	7,59	10,50	10,50	10,50
	(10,1)	C1	130,10	131,60	131,60	131,60	0,64	1,03	1,03	1,03	1,94	3,10	3,10	3,10
		C2	521,45	526,72	526,72	526,72	1,67	2,13	2,13	2,13	9,12	12,12	12,12	12,12
D2	(1,1)	C1	117,20	117,87	117,87	117,87	1,12	2,15	2,15	2,15	0,70	1,15	1,15	1,15
		C2	495,52	499,30	499,30	499,30	2,65	3,55	3,55	3,55	1,55	2,00	2,00	2,00
	(10,1)	C1	118,17	120,50	120,50	120,50	0,75	1,55	1,55	1,55	1,65	1,95	1,95	1,95
		C2	495,87	499,55	499,55	499,55	2,50	3,30	3,30	3,30	3,30	2,10	2,10	2,10
D3	(1,1)	C1	117,46	147,82	147,82	147,82	0,48	5,77	5,52	5,37	0,91	0,00	0,42	0,42
		C2	586,40	608,77	608,72	608,72	6,62	44,45	44,42	44,45	6,62	0,00	0,07	0,50
	(10,1)	C1	142,81	148,05	147,85	147,62	0,64	5,77	5,37	5,37	1,92	0,02	0,42	0,42
		C2	591,79	608,82	608,75	608,75	0,83	40,12	44,30	44,30	6,57	0,00	0,10	0,10

3.4.2 Resultados para o segundo conjunto de instâncias

Para uma avaliação mais abrangente da metodologia proposta, a inclusão de um segundo conjunto de instâncias se torna necessária. Em uma tentativa de estabelecer um cenário diverso, modificamos as instâncias originalmente definidas por Ma et al. (2019b). Esse conjunto de instâncias é categorizado em 10 grupos com base no número de períodos no horizonte de planejamento e quantidades variáveis de objetos a serem cortados.

A Tabela 3.8 apresenta as características dessas instâncias, juntamente com a configuração de parâmetros empregada em sua geração, seguindo as diretrizes descritas em Gau e Wäscher (1995). Cada classe compreende um total de 20 instâncias geradas aleatoriamente. A metodologia para gerar o comprimento, itens e demanda dos objetos se alinha com o procedimento detalhado em Gau e Wäscher (1995).

Tabela 3.8: Características do segundo conjunto de instâncias com comprimento de objeto de $L = 500$.

Classes	Períodos	Itens	Instâncias	Comprimento do item	Demanda
C1	3	5	20	[0,1L , 0,8L]	[0,20]
C2	3	10	20	[0,1L , 0,8L]	[0,20]
C3	3	15	20	[0,1L , 0,8L]	[0,20]
C4	3	20	20	[0,1L , 0,8L]	[0,20]
C5	6	5	20	[0,1L , 0,8L]	[0,20]
C6	6	10	20	[0,1L , 0,8L]	[0,20]
C7	6	15	20	[0,1L , 0,8L]	[0,20]
C8	6	20	20	[0,1L , 0,8L]	[0,20]
C9	9	20	20	[0,1L , 0,8L]	[0,20]
C10	9	30	20	[0,1L , 0,8L]	[0,20]

Os resultados obtidos ao resolver o 1D-PMCE-DE usando o segundo conjunto de instâncias são mostrados a seguir. Embora os experimentos tenham abrangido todos os três conjuntos de datas de Entrega, as datas D1 e D2 provaram ser menos desafiadores. O solucionador CPLEX obteve soluções com otimalidade comprovada, e a abordagem proposta obteve valores objetivos idênticos, exigindo notavelmente menos tempo de execução. Consequentemente, excluímos D1 e D2 da análise detalhada.

Tabela 3.9: Valor médio da função objetivo, tempo de execução médio e valor médio de *gap* para a solução obtida usando a abordagem proposta para o segundo conjunto de instâncias.

Classes				Função Objeivo				Tempo(s)				Gap			
DE	(θ,β)	Classes	Colunas	CG	RF	LB	CP	CG	RF	LB	CP	gap^1	gap^2	gap_{CP}	$\#Opt$
D3	(1,1)	C1	1,84	114,48	116,85	116,85	116,85	0,17	0,48	0,51	0,19	0,00	0,00	0,00	20
		C2	6,69	195,73	199,00	198,90	198,90	0,12	0,36	0,81	0,45	0,05	0,00	0,00	20
		C3	11,26	244,03	249,45	248,95	248,85	4,71	5,19	10,91	14,36	0,20	-0,04	0,00	20
		C4	15,45	355,42	360,50	359,65	359,55	1,01	1,21	1,26	11,72	0,23	-0,02	0,00	20
		C5	2,15	229,33	237,35	236,95	236,95	0,13	0,19	0,86	1,43	0,16	0,00	0,00	20
		C6	5,61	369,69	375,30	375,15	375,05	3,42	3,86	8,65	4,06	0,03	-0,02	0,00	20
		C7	10,82	521,83	529,40	528,50	528,40	0,81	0,93	1,02	16,92	0,17	-0,01	0,00	20
		C8	17,30	655,44	660,35	659,85	659,70	1,68	1,79	11,01	16,05	0,07	-0,02	0,00	20
		C9	16,67	1024,25	1035,90	1034,55	1034,30	24,24	25,32	44,11	358,07	0,13	-0,02	0,0005	19
		C10	30,86	1410,77	1421,00	1418,20	1418,15	38,46	59,95	106,81	416,45	0,19	-0,03	0,002	19
	(10,1)	C1	1,84	321,21	337,80	337,80	337,80	0,10	0,11	0,14	0,12	0,00	0,00	0,00	20
		C2	6,73	468,77	491,95	490,95	490,95	0,43	0,48	0,51	2,27	0,20	0,00	0,00	20
		C3	11,08	473,84	517,65	511,65	511,65	0,76	0,91	1,02	4,79	1,17	0,00	0,00	20
		C4	15,48	653,58	687,25	678,35	678,35	0,72	0,87	1,02	20,22	1,31	0,00	0,00	20
		C5	2,20	727,89	789,00	789,00	789,00	0,15	0,20	0,23	0,27	0,00	0,00	0,00	20
		C6	5,77	860,71	907,50	906,05	906,05	0,68	0,77	0,81	3,47	0,16	0,00	0,00	20
		C7	11,09	967,88	1026,05	1016,65	1016,65	0,91	1,00	1,09	5,39	0,92	0,00	0,00	20
		C8	17,38	1008,43	1046,90	1039,50	1039,50	2,23	2,45	2,61	22,13	0,71	0,00	0,00	20
		C9	16,76	2013,55	2105,95	2098,05	2098,05	2,63	2,88	3,22	115,41	0,37	0,00	0,00	20
		C10	31,38	1861,85	1930,40	1903,10	1903,10	46,32	47,44	49,04	686,96	1,43	0,00	0,002	19

A Tabela 3.9 apresenta os resultados obtidos pela metodologia proposta utilizando datas de entrega D3, dois valores distintos de θ e β, em 10 classes de instâncias. A tabela inclui valores médios de função objetivo, tempo de execução e *gap* conforme definido pelas expressões (3.26) e (3.27). A coluna $\#Opt$ denota o número de soluções ótimas comprovadas em cada classe, e o valor médio final de *gap* do solucionador CPLEX também é fornecida. De maneira semelhante ao primeiro conjunto de instâncias, o tempo de execução representa o tempo acumulado necessário para obter cada solução; a coluna LB exibe o tempo de execução médio final gasto pelo procedimento proposto pelo Algoritmo 4.

Para instâncias associadas às datas de entrega D3, o solucionador CPLEX obtém, conforme indicado pela coluna $\#Opt$, um total de 1193 soluções ótimas comprovadas dentre as 1200 instâncias avaliadas. Excluindo D3-(1,1)-C3, D3-(1,1)-C4, D3-(1,1)-C6, D3-(1,1)-C7, D3-(1,1)-C8, D3-(1,1)-C9 e D3-(1,1)-C10, a abordagem proposta atinge, conforme visto na coluna gap^2, precisamente os mesmos valores de função objetivo que aqueles obtidos pelo CPLEX, enquanto gasta consideravelmente menor tempo de execução. No entanto, para D3-(1,1)-C3, D3-(1,1)-C4, D3-(1,1)-C6, D3-(1,1)-C7, D3-(1,1)-C8, D3-(1,1)-C9 e D3-(1,1)-C10, o solucionador CPLEX supera a abordagem proposta, produzindo uma solução média superior.

A aplicação do *Local Branching* prova ser eficiente com datas de entrega D3, independentemente das escolhas de penalidade para θ e β, pois melhora consistentemente as soluções encontradas pela matheurística *Relax-and-Fix*. Excluindo as classes D3-(1,1)-C1, D3-(10,1)-C1 e D3-(10,1)-C5, o *Local Branching* demonstra sua capacidade de melhorar as soluções *Relax-and-Fix*, evidente na coluna gap^1 na Tabela 3.9. O número de soluções melhoradas pelo *Local Branching* em comparação com aquelas obtidas pelo *Relax-and-Fix* para cada classe de instância é descrito na Tabela 3.10. Para D3-(1,1)-C10 e D3-(10,1)-C10, o *Local Branching* melhora 15 e 17 de 20 soluções, respectivamente. Vale ressaltar que o *Local Branching* exibe desempenho superior quando aplicado a instâncias com um número maior de períodos e itens. Além disso, considerando que o *Relax-and-Fix* aborda subproblemas com foco no período atual, as variáveis fixas de períodos anteriores e as variáveis relaxadas de

Tabela 3.10: Número de soluções melhoradas pelo *Local Branching* quando comparado à solução encontrada pelo *Relax-and-Fix*, usando o segundo conjunto de instâncias com datas de Entrega D3.

Classes				Classes			
DE	(θ, β)	Classes	# de soluções	DE	(θ, β)	Classes	# de soluções
D3	(1,1)	C1	0	D3	(10,1)	C1	0
		C2	1			C2	1
		C3	5			C3	7
		C4	7			C4	9
		C5	2			C5	0
		C6	1			C6	2
		C7	7			C7	9
		C8	5			C8	8
		C9	5			C9	9
		C10	15			C10	17

Tabela 3.11: Valores médios de cada um dos termos da função objetivo encontrados na solução obtida pela metodologia proposta usando o segundo conjunto de instâncias

			Objetos				Atraso				Adiantamento			
DE	(θ, β)	Classes	CG	RF	LB	CPLEX	CG	RF	LB	CPLEX	CG	RF	LB	CPLEX
D3	(1,1)	C1	91,36	62,25	92,25	92,25	22,97	24,55	24,55	24,55	0,15	0,05	0,05	0,05
		C2	165,39	166,45	166,35	166,35	30,34	32,55	32,45	32,45	0,00	0,00	0,10	0,10
		C3	217,69	219,65	219,15	219,05	26,34	29,80	29,30	29,20	0,00	0,00	0,50	0,60
		C4	322,12	324,05	323,10	323,10	33,12	36,30	35,45	35,35	0,17	0,15	1,10	1,10
		C5	173,71	175,45	175,30	175,30	55,49	61,90	61,45	61,45	0,13	0,00	0,20	0,20
		C6	314,75	315,70	315,65	315,65	54,71	59,35	59,20	59,05	0,23	0,25	0,30	0,45
		C7	471,81	473,85	473,00	473,00	49,56	55,35	54,35	54,25	0,46	0,20	1,15	1,15
		C8	615,64	616,85	616,25	616,25	39,40	43,05	42,45	42,45	0,40	0,45	1,00	1,00
		C9	913,55	915,65	915,15	914,95	110,07	119,75	118,35	117,95	0,62	0,50	1,05	1,40
		C10	1360,37	1363,80	1361,15	1361,10	50,12	57,10	54,30	54,20	0,27	0,10	2,75	2,85
	(10,1)	C1	91,36	92,25	92,25	92,25	22,97	24,55	24,55	24,55	0,15	0,05	0,05	0,05
		C2	165,39	166,45	166,35	166,35	30,33	32,55	32,45	32,45	0,00	0,00	0,10	0,10
		C3	210,44	219,65	219,05	219,05	26,34	29,30	29,20	29,20	1,10	0,00	0,60	0,60
		C4	322,12	324,05	323,15	323,15	33,12	36,30	35,40	35,40	0,17	0,20	1,10	1,10
		C5	173,71	175,30	175,30	175,30	55,38	61,35	61,35	61,35	0,31	0,20	0,20	0,20
		C6	314,75	315,65	315,55	315,55	54,58	59,15	59,00	59,00	0,37	0,35	0,50	0,50
		C7	471,81	473,80	472,95	472,95	49,56	55,20	54,25	54,25	0,46	0,25	1,20	1,20
		C8	615,64	616,85	616,15	616,15	39,22	42,95	42,20	42,20	0,57	0,55	1,35	1,35
		C9	913,45	915,60	916,60	916,60	109,92	118,95	117,98	117,98	0,88	0,85	1,60	1,60
		C10	1360,37	1363,60	1360,85	1360,85	50,12	56,65	53,90	53,90	0,27	0,30	3,25	3,25

períodos futuros podem não ser tão eficazes para instâncias grandes, como sugerido pelos resultados obtidos.

A Tabela 3.11 apresenta o número médio de objetos cortados, atraso e antecipação individualmente. Esses valores são apresentados para cada fase da abordagem proposta, bem como para o solucionador CPLEX. Também de maneira semelhante ao primeiro conjunto de instâncias, as escolhas dos fatores de penalidade, $(\theta = 1,\ \beta = 1)$ e $(\theta = 10,\ \beta = 1)$, não apresentam impacto visível nos resultados, dado que o número de objetos supera atraso e antecipação.

3.5 Conclusões e discussões parciais

Este tese aborda, pela primeira vez, o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entregas (1D-PMCE-DE), que integra a minimização conjunta de custos de material, antecipação ponderada e atraso ponderado usando uma abordagem baseada em penalidades. Este tese também introduziu uma meto-

dologia de solução matheurística para o 1D-PMCE-DE em um método de resolução de três fases. A metodologia proposta compreende a Fase 1, com foco na geração de colunas; Fase 2, dedicada à obtenção de uma solução inteira viável por meio da heurística *Relax-and-Fix* para o problema relaxado; e a Fase 3, que envolve a aplicação da matheurística *Local Branching*.

Além disso, introduzimos um novo conjunto de instâncias para o 1D-PMCE-DE, fornecendo uma plataforma de avaliação abrangente para a metodologia proposta. O primeiro conjunto compreende duas classes, cada uma contendo 40 instâncias, enquanto o segundo conjunto consiste em 10 classes, cada uma com 20 instâncias. Em ambos os conjuntos, três datas de entrega distintas foram geradas para avaliação abrangente.

O primeiro conjunto de instâncias apresentou um desafio considerável para o solucionador CPLEX, pois ele se esforçou para encontrar soluções ótimas comprovadas (para as colunas que foram geradas na fase 1) dentro do limite de tempo predefinido. Em contraste, a abordagem proposta atingiu consistentemente os mesmos valores objetivos que o solucionador CPLEX, gastando tempos de execução significativamente mais curtos. A eficácia do *Local Branching* é evidente dentro deste conjunto, principalmente para as datas de entrega D3; para as datas D1 e D2, o *Local Branching* não melhorou as soluções encontradas pelo *Relax-and-Fix*. Passando para o segundo conjunto de instâncias, as datas de entrega D1 e D2 não apresentaram desafios notáveis para o CPLEX nem para e a abordagem proposta. No entanto, com as datas de entrega D3, enquanto a abordagem proposta obteve soluções quase idênticas às do solucionador CPLEX, seu tempo de execução foi consideravelmente menor. Além disso, a eficácia do *Local Branching* tornou-se aparente neste conjunto.

Ao analisar o impacto dos parâmetros de penalidade $(\theta,\ \beta)$, torna-se claro que, com as escolhas selecionadas $((\theta,\ \beta) = (1,\ 1)$ e $(\theta,\ \beta) = (10,\ 1))$, a influência na função objetivo foi marginal. Um teste experimental cuidadoso precisa ser projetado para avaliar o impacto desses fatores no modelo de forma abrangente e para ajustar os parâmetros de penalidade.

Apesar das contribuições desta modelagem proposta, ela tem limitações que são abordados em capítulos futuros. Questões de *setup*, que é o conjunto de atividades de preparação da máquina de corte ou troca de ferramentas, não foram considerados. A consideração das atividades de preparação merece atenção, pois elas afetam significativamente o tempo de produção e o custo dos itens em cenários práticos.

Nos Capítulos 4 e 5 abordaremos essa questão a modelagem proposta. O Capítulo 4 apresenta o *setup* independente da sequência e no capítulo 4 o *setup* dependente da sequência. No primeiro caso, o tempo necessário para preparar a máquina é fixo, independentemente da ordem em que os padrões são cortados. Já no caso do *setup* dependente da sequência, o tempo de preparação varia conforme a transição entre os cortes específicos, o que pode exigir estratégias mais sofisticadas de planejamento para minimizar atrasos e otimizar a eficiência operacional.

Capítulo 4

Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Independente da Sequência

Este capítulo apresenta o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Independente da Sequência (1D-PMCE-DE-SI). A Seção 4.1 apresenta uma revisão detalhada do problema de corte de estoque com *setup*, incluindo uma análise comparativa entre os trabalhos de referência e o presente estudo. A Seção 4.2 descreve a formulação para a modelagem matemática utilizada. A proposta de procedimento de resolução desenvolvida para abordar o problema é apresentada na Seção 4.3. A Seção 4.4, por sua vez, apresenta os resultados dos experimentos computacionais realizados. Por fim, a Seção 4.5 sintetiza as conclusões desse estudo, propondo possíveis direções para sua continuidade.

4.1 Revisão de Problemas de Corte de Estoque com *Setup*

Um dos principais desafios nos Problemas de Corte de Estoque (PCE) é minimizar o desperdício de material, garantindo uma utilização eficiente dos recursos disponíveis. Entretanto, além da preocupação com a otimização do corte, em muitos cenários industriais há um fator adicional relevante a ser considerado: o *setup*. O *setup* se refere ao tempo e custo necessários para preparar as máquinas ou ajustar os processos de corte antes de iniciar uma nova *setup* de produção. Negligenciar esse aspecto pode levar a planejamentos ineficientes, nos quais o custo operacional aumenta significativamente, mesmo que o desperdício de material seja minimizado.

Dessa forma, a consideração do *setup* no PCE torna-se fundamental para equilibrar a eficiência do corte e os custos operacionais. Portanto, considerar os custos ou o tempo associados à *setup* do padrão torna-se crucial no PCE clássico. Nessa seção são apresentados alguns estudos que abordam *setup* e assumem que, ao se alternar entre diferentes padrões, incorre-se em um custo de *setup* do padrão.

O trabalho pioneiro de Haessler (1975) focou em um Problema de Corte de Esto-

que com Padrões, visando minimizar tanto o custo de *setup* quanto o custo de perda de corte. Para lidar com o problema, o trabalho propôs um procedimento heurístico sequencial baseado na geração de padrões, considerando três critérios de aspiração. Esta abordagem foi posteriormente modificada por Martin et al. (2018), incorporando os mesmos critérios de aspiração e utilizando um Problema de Mochila Limitado Inteiro para gerar padrões de corte. Mobasher e Ekici (2013) propôs um modelo de PLIM e algoritmos heurísticos baseados em busca local e geração de colunas para lidar com o Problema de Corte de Estoque com *Setup* (PCES) e minimizar os custos totais de *setup* e material. Considerando o mesmo objetivo, Cui et al. (2015) introduziu um algoritmo baseado em um procedimento de agrupamento sequencial para gerar padrões de corte e resolver o problema resultante usando um solucionador.

Em outra abordagem, Ma et al. (2018, 2019a) apresentaram uma heurística baseada em programação dinâmica para PCES multiperíodo, minimizando os custos totais que abrangem *setup* de produção, *setup* de padrão, manutenção de estoque e custos de desperdício de material. Mais recentemente, Silva et al. (2023) propôs um PCES multiperíodo com o objetivo de minimizar os custos de material, estoque e *setup* para padrões de corte. Os autores propuseram uma formulação adaptada de Gilmore e Gomory (1961), uma formulação baseada na formulação proposta por Johnston e Sadinlija (2004), uma formulação de reflexão adaptada baseada em Alves e Valério de Carvalho (2008), além de uma reformulação de cada uma das três formulações com base na localização da instalação.

Conforme destacado por Henn e Wäscher (2013), as informações de custo geralmente não estão disponíveis em situações práticas, levando muitos trabalhos a abordar PCESs concentrando-se em minimizar o número de *setups* ou impor restrições de *setup*. Por exemplo, Vanderbeck (2000) abordou um PCES com o objetivo de minimizar a variedade de padrões de corte, limitando a quantidade de cada padrão, elaborando um procedimento *branch-and-price-and-cut* para obter soluções ótimas ou quase ótimas para instâncias reais. Em outra abordagem, Foerster e Wascher (2000) forneceu um método para derivar um novo plano de corte, considerando apenas padrões fornecidos do plano, minimizando a diversidade de padrões e atendendo à demanda com o menor custo total de material. Além disso, Umetani et al. (2003) e Umetani et al. (2006) elaboraram um algoritmo híbrido combinando a meta-heurística *Iterated Local Search* com programação matemática para minimizar o número de objetos de corte, restringindo a diversidade de padrões. O trabalho de de Araujo et al. (2014) propôs um algoritmo genético para resolver instâncias de um PCES bi-objetivo que, além de minimizar o número de diferentes tipos de padrões, também minimizava o número de objetos usados. O trabalho de Martin et al. (2022) abordou um PCES bi-objetivo, inicialmente não linear e não convexo devido a termos bilineares em restrições de atendimento de demanda, analisando *trade-offs* entre o número de objetos e a diversidade de padrões. Este modelo foi transformado em um modelo mono-objetivo por meio da penalização apropriada desses termos dentro de uma única função objetivo.

Wuttke e Heese (2018) abordaram um PCE bidimensional com cortes guilhotinados, particularmente no contexto de uma empresa têxtil. Seu objetivo era minimizar o tempo de produção da tecelagem, o tempo de montagem do eixo e o tempo de *setup* dependente da sequência, representado pelo tempo de movimento da faca. Além disso, outros pesquisadores se aprofundaram na integração de PCE ou problemas de *bin-packing* com problemas de dimensionamento de lote, levando em consideração

os custos e complexidades de *setup*. Oliveira et al. (2021) abordaram uma integração multiobjetivo envolvendo problemas de dimensionamento de lote e estoque de corte, com configurações associadas a fatores como tipo de objeto, matéria-prima, especificações da máquina e período de produção.

Melega et al. (2020) propuseram um problema que integra dimensionamento de lote, programação e decisões de corte de estoque em dois estágios, considerando tempos e custos de *setup* dependentes da sequência. No primeiro estágio, uma máquina de corte é usada para cortar objetos grandes em pedaços menores. Em seguida, as peças cortadas no primeiro estágio são usadas para montar os produtos finais no segundo estágio de produção, onde os produtos finais são programados para atender às demandas do cliente. Para resolver o problema integrado de dois estágios, eles apresentam métodos de solução baseados em uma abordagem *branch-and-price*. Além disso, Melega et al. (2022) abordaram um problema integrado envolvendo *bin-packing* e dimensionamento de lote, também incorporando configurações dependentes da sequência para uma abordagem abrangente.

Tabela 4.1: Uma Visão Geral Comparativa das Características dos Problemas Abordados (1D-PMCE-DE-SI)

Tipo	Artigo de Referência	Características abordadas					
		Objetos cortados	*Setup*	Data de entrega	Atraso	Adiantamento	Dimensão
1	Gilmore e Gomory (1961), Gilmore e Gomory (1963), Gilmore e Gomory (1965), Kantorovich (1960), Valério de Carvalho (2002), Delorme et al. (2016), Delorme e Iori (2020), Rao (1976), Dyckhoff (1981), Martinovic et al. (2018), Cherri et al. (2009), Cherri et al. (2014), Cui e Yang (2010), Harjunkoski et al. (1996), Harjunkoski (1997), Harjunkoski et al. (1998), Harjunkoski et al. (1999), Dyckhoff (1990), Wäscher et al. (2007), Boschetti et al. (2009), Poldi e de Araujo (2016)	Sim	–	–	–	–	1D
2	Henn e Wäscher (2013), Haessler (1975), Martin et al. (2018), Mobasher e Ekici (2013), Cui et al. (2015), Ma et al. (2018), Ma et al. (2019a), Silva et al. (2023), Foerster e Wascher (2000), Umetani et al. (2003), Umetani et al. (2006), de Araujo et al. (2014), Martin et al. (2022), Oliveira et al. (2021), Melega et al. (2020), Melega et al. (2022)	Sim	Sim	–	–	–	1D
3	Li (1996), Hendry et al. (1996)	Sim	–	Sim	–	–	1D
4	Reinertsen e Vossen (2010), Arbib e Marinelli (2014), Braga et al. (2016b), Braga et al. (2015), Braga et al. (2016a), Arbib e Marinelli (2017), Sezer e Muter (2018), Wu et al. (2024)	Sim	–	Sim	Sim	–	1D
5	**Abordagem proposta**	Sim	Sim	Sim	Sim	Sim	1D

4.1.1 Análise comparativa: trabalhos de referência e o presente estudo

A Tabela 4.1 apresenta uma comparação das principais características do problema exploradas nesta tese com aquelas da literatura relacionada. Uma observação significativa é que estudos anteriores focam principalmente no atraso em relação às

datas de entrega, muitas vezes negligenciando o fator igualmente importante da antecipação. Considerar a antecipação é essencial para uma compreensão mais completa dos impactos do cronograma no PCE. Além disso, as pesquisas sobre o PCES normalmente desconsideram tanto o atraso quanto a antecipação.

Curiosamente, a literatura existente sobre o PCES raramente aborda os tempos de *setup* e os custos de *setup* simultaneamente. Esta tese preenche essas lacunas, incorporando um tratamento matemático abrangente que considera o atraso, a antecipação, os custos de *setup*, os tempos de *setup* e a ociosidade dentro da estrutura do PCE, oferecendo uma contribuição distinta e valiosa para este domínio do problema.

Mais especificamente, os artigos de referência do tipo 1 não abordam nenhum problema associado à data de entrega ou *setup*. Esses artigos abordam principalmente a minimização de custos e desperdícios. Os artigos do tipo 2, por sua vez, incluem custos de *setup*, especialmente aqueles que lidam com problemas multiperíodos, como Ma et al. (2018), Ma et al. (2019a) e Silva et al. (2023), dentre outros. Os artigos do tipo 3 são os pioneiros em abordar questões de programação associadas ao problema de corte. Os artigos do tipo 4 estendem a integração entre problemas de programação e problemas de corte, incluindo custos de atraso e avanço em suas formulações. Nossa formulação, classificada como tipo 5, dá um novo passo neste processo ao abordar o problema multiperíodo, com uma data de entrega definida, custos de atraso e antecipação e custos de *setup*, considerando também o tempo gasto para alterar os padrões de corte, ou seja, o tempo de *setup*.

4.2 Modelagem proposta

Esta seção descreve o problema central desta tese: o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* (1D-PMCE-DE). Este problema corresponde à integração do problema clássico de corte de estoque unidimensional com características de problemas de programação e problemas de dimensionamento de lote, como a possibilidade de atraso ou antecipação na entrega de pedidos, a inclusão de tempo de *setup* na máquina de corte, a possibilidade de ociosidade na máquina de corte e o armazenamento de itens de corte para atender à demanda de períodos futuros. Uma descrição detalhada deste problema pode ser encontrada na Subseção 4.2.1. Duas variantes do problema também são apresentadas. A primeira não permite tempo ocioso na máquina de corte, enquanto a segunda variante permite que a máquina de corte permaneça ociosa durante períodos especificados. Cada variante é acompanhada por sua formulação matemática correspondente na Subseção 4.2.2.

4.2.1 Descrição do problema

Suponha que haja um conjunto de objetos de mesmo comprimento L em estoque. Esses objetos são usados em operações de corte para obter itens menores de um conjunto contendo n tipos de itens, em que o tipo de item i tem comprimento ℓ_i, para $i = 1,2,\ldots,n$. O horizonte de planejamento para operações de corte é dividido em M períodos.

Seja N^t o número de padrões de corte possíveis de objetos no período t, para

$t = 1, 2, \ldots, M$, no qual um padrão de corte $j = 1, 2, \ldots, N^t$ referente ao período t produz uma quantidade a_{ij}^t de itens do tipo i. Além disso, em cada período $t = 1, 2, \ldots, M$, há uma quantidade de demanda b_i^t do tipo de item i com data de entrega d^t. Observe que, assumindo que as datas de entrega são classificadas em ordem crescente, o período t representa a t-ésima data de entrega, ou seja, d^t. Assim, todos os pedidos com a mesma data de entrega correspondem ao mesmo período. Apenas para simplificar a notação usada no restante do capítulo, sempre que for referida para todos os t, para todos os j e para todos os i, assumiremos, respectivamente, $t = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N^t$ e $i = 1, 2, \ldots, n$.

As premissas gerais do 1D-PMCE-DE-SI são as seguintes:

(i) O estoque disponível compreende objetos idênticos em quantidade suficiente para atender aos requisitos de produção;

(ii) Um custo de *setup* independente de sequência é incorrido quando a produção começa a partir de um novo padrão de corte;

(iii) Há um tempo de *setup* independente de sequência antes de iniciar as operações de corte a partir de um determinado padrão de corte;

(iv) Os tempos de processamento e *setup* e as demandas de itens são determinísticos;

(v) Os itens podem ser produzidos antes ou depois de suas datas de entrega associadas, incorrendo em uma penalização por antecipação ou atraso, respectivamente, na função objetivo;

(vi) Itens cortados não utilizados em um período específico podem ser estocados e utilizados para atender às demandas em períodos subsequentes, sem incorrer em custo adicional.

(vii) Nenhum custo de estoque para itens é considerado.

Nesta tese, assumimos, para cada período t, que o valor c_j^t representa o custo de corte de um objeto em estoque com padrão j; f_j^t representa o custo de *setup* para produzir um padrão de corte j; β^t representa a penalidade por entregar pedidos antes da data de entrega, e γ^t representa a penalidade por entregar pedidos após a data de entrega. Além disso, r_j^t e p_j^t denotam o tempo de *setup* e o tempo de corte, respectivamente, do padrão de corte j no período t. A Tabela 4.2 resume as definições de parâmetros para o 1D-PMCE-DE-SI.

Note que, em cada período t, a máquina de corte tem um tempo de processamento disponível igual a $d^t - d^{t-1}$. Como cada *setup* e operação de corte leva um certo tempo, consequentemente a máquina tem uma capacidade operacional em cada período. Neste caso, para cada período, o atraso pode representar quanto do tempo disponível, ou seja, esta capacidade, foi excedido para atender à demanda do período.

O 1D-PMCE-DE-SI consiste em encontrar o melhor conjunto de padrões de corte viáveis e a quantidade a ser cortada de cada um desses padrões em cada período. O objetivo é minimizar conjuntamente o custo de uso de matérias-primas, a penalização por qualquer atraso e antecipação nas entregas e o custo total de *setup* para alternar os padrões de corte.

Tabela 4.2: Definições de parâmetros para o 1D-PMCE-DE-SI.

Parâmetro	definição
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 tipos 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 em estoque com padrão j para cada período t
d^t	data de entrega no período t
f^t	custo de *setup* 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 *setup* do padrão de corte j no período t
p_j^t	tempo de corte do padrão de corte j no período t

4.2.2 Formulações matemáticas para o 1D-PMCE-DE-SI

Para modelar o problema, definimos as seguintes variáveis de decisão. A variável contínua q^t significa o tempo necessário para cortar todos os tipos de itens no período t, compreendendo tanto o tempo total de corte quanto o tempo total de *setup*. A variável inteira x_j^t representa o número de objetos cortados no período t seguindo o padrão de corte j. As variáveis T^t e E^t representam atraso e antecipação, respectivamente, na entrega de pedidos para o período t. A variável binária S_j^t indica se no período t é utilizado o padrão j ($S_j^t = 1$) ou não ($S_j^t = 0$). Finalmente, a variável de decisão $start^t$ representa o tempo de início do processo de corte no período t. A variável inteira e_i^t representa o número de tipos de itens i restantes em estoque no final do período t, para todos os $t = 0,1,\ldots,M$. Assumimos um estoque inicial $\bar{e}_i$ para o item i.

Nesta tese, abordamos duas variantes de 1D-PMCE-DE-SI:

(i) Subseção 4.2.2.1 apresenta a primeira variante, que não permite ociosidade na máquina de corte. Isso significa que a máquina de corte deve estar sempre executando alguma operação. Esta variante sem tempo ocioso é denotada como 1D-PMCE-DE-SI-wIT.

(ii) Subseção 4.2.2.2 apresenta a segunda variante, que permite que a máquina de corte fique ociosa quando conveniente. Esta variante com tempo ocioso é denotada como 1D-PMCE-DE-SI-IT.

4.2.2.1 Formulação matemática para o problema sem ociosidade na máquina de corte

A formulação matemática que apresentamos para o problema sem ociosidade na máquina de corte encapsula atraso e antecipação da seguinte forma. Em um período específico t, se a data de conclusão da produção ($start^t + q^t$) for anterior à data de entrega d^t, os pedidos para esse período são adiantados em E^t unidades de tempo, calculadas como $E^t = d^t - (start^t + q^t)$. Por outro lado, se a data de conclusão da produção exceder a data de entrega do pedido, os pedidos para o período t são atrasados em T^t unidades de tempo, dadas por $T^t = (start^t + q^t) - d^t$. Além disso, se $E^t > 0$ para qualquer período t dado, então T^t é igual a 0; inversamente, se $T^t > 0$,

então E^t é igual a 0. Assim, o atraso e a antecipação são modelados como:

$$start^t + q^t + E^t - T^t = d^t \ \forall t. \tag{4.1}$$

Os custos por unidade de tempo são incorridos para atrasos e adiantamentos, que podem ser uniformes em todos os períodos ou variar para cada período. Se um período for adiantado, uma penalidade de $\beta^t E^t$ é imposta, em que $\beta^t > 0$ representa o custo de antecipação por unidade de períodos. Da mesma forma, para períodos com atrasos, uma penalidade de $\gamma^t T^t$ é incorrida, sendo $\gamma^t > 0$ o custo de atraso por unidade de períodos.

A formulação matemática para o 1D-PMCE-DE-SI-wIT é dada por:

$$\min \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M} \beta^t E^t + \sum_{t=1}^{M} \gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \tag{4.2}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{N} a_{ji}^t x_j^t - e_i^t + e_i^{(t-1)} = b_i^t \qquad \forall t,\ i \tag{4.3}$$

$$x_j^t \leq B S_j^t \qquad \forall t,\ j \tag{4.4}$$

$$start^t = d^{t-1} + T^{t-1} - E^{t-1} \qquad \forall t \tag{4.5}$$

$$q^t = \sum_{j=1}^{N} p_j^t x_j^t + \sum_{j=1}^{N} r_j^t S_j^t \qquad \forall t \tag{4.6}$$

$$start^t + q^t + E^t - T^t = d^t \qquad \forall t \tag{4.7}$$

$$E^0 = 0 \tag{4.8}$$

$$T^0 = 0 \tag{4.9}$$

$$d^0 = 0 \tag{4.10}$$

$$e^{0i} = \bar{e}_i \qquad \forall i \tag{4.11}$$

$$x_j^t \in \mathbb{Z}_+ \qquad \forall j,\ t \tag{4.12}$$

$$E^t, T^t, q^t, start^t \in \mathbb{Z}_+ \qquad \forall t \tag{4.13}$$

$$S_j^t \in \{0,1\} \qquad \forall t,j \tag{4.14}$$

$$e_i^t \in \mathbb{Z}_+ \qquad \forall t,i. \tag{4.15}$$

O objetivo deste problema é, portanto, a minimização da função objetivo (4.2), composta pela soma ponderada de quatro parcelas: (*i*) o custo total dos objetos cortados (primeira parcela); (*ii*) penalização por atrasos nas entregas (segunda parcela); (*iii*) penalidades por antecipação nas entregas (terceira parcela); e (*iv*) os custos de *setup* (quarta parcela). O conjunto de restrições (4.3) 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 são armazenados no final do período. O conjunto de restrições (4.4) garante que, em um determinado período, um determinado padrão de corte só pode ser cortado se houver uma mudança de padrão de corte na máquina. O conjunto de restrições (4.5) calcula o início do processo de corte sem permitir tempo ocioso na máquina. O conjunto de restrições (4.6) 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 (4.7) considera os tempos de

atraso e os tempos de antecipação. As restrições (4.8) e (4.9) representam a condição de limite inicial, que afirma que não há atraso ou avanço de pedidos no início do processo de corte. A restrição (4.10) determina a condição de data de entrega inicial para $t = 0$ e as restrições (4.11) inicializam o estoque de itens. As restrições (4.12)–(4.15) representam as condições de não negatividade e integralidade das variáveis de decisão.

A desvantagem dessa variante do 1D-PMCE-DE-SI é que, para reduzir a penalização por antecipação, a solução obtida pode consistir em *setups* desnecessários ou cortes ineficientes para evitar a produção antecipada de todos os itens demandados. Portanto, a próxima seção apresenta a variante do 1D-PMCE-DE-SI que permite tempo ocioso, o que pode evitar essa situação.

4.2.2.2 Formulação matemática para o problema com ociosidade na máquina de corte

Essa variante permite ociosidade na máquina de corte, indicando os intervalos de tempo durante os quais nenhuma tarefa (corte ou *setup*) é realizada na máquina de corte. A formulação matemática para essa versão do problema, denominada 1D-PMCE-DE-SI-IT, que permite ociosidade na máquina de corte e é baseada na formulação 1D-PMCE-DE-SI-wIT, é a seguinte:

$$\text{min} \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M} \beta^t E^t + \sum_{t=1}^{M} \gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \tag{4.16}$$

$$\text{sujeito a:} \quad (4.3) - (4.4), (4.6) - (4.15) \tag{4.17}$$

$$start^t \geq d^{t-1} + T^{t-1} - E^{t-1} \quad \forall\ t. \tag{4.18}$$

A função objetivo e as restrições desta formulação matemática permanecem as mesmas da formulação 1D-PMCE-DE-SI-wIT, com a única diferença sendo o conjunto de restrições (4.18), que substitui as restrições (4.5). As restrições (4.18) permitem a possibilidade de a máquina ser parada em determinados momentos, em contraste com as restrições (4.5), que forçariam uma troca imediata para outra *setup* na máquina de corte assim que a produção fosse concluída.

4.3 Abordagem de solução proposta

Esta seção detalha o método proposto para resolver as formulações introduzidas para o 1D-PMCE-DE-SI. O procedimento proposto consiste em três fases: na primeira fase, descrita na Seção 4.3.1, o problema linear é relaxado e resolvido pelo método simplex com geração de colunas, obtendo uma solução que pode não satisfazer as restrições de integralidade do 1D-PMCE-DE-SI. A segunda fase, mostrada na Seção 4.3.2, encontra uma solução inteira inicial para o problema, usando as colunas encontradas na primeira fase. São propostas duas heurísticas que podem ser usadas nesta fase: heurística de arredondamento e heurística *Relax-and-Fix*. A terceira fase, apresentada na Seção 4.3.3, visa melhorar, se possível, a solução inteira encontrada na fase anterior. Propõe-se o uso de duas matheurísticas nesta fase: matheurística *Local Branching* e matheurística *Fix-and-optimize*.

4.3.1 Fase 1: Método de geração de colunas

Devido à complexidade do Problema de Corte de Estoque (PCE), atribuída ao vasto número de padrões de corte potenciais, a abordagem padrão envolve resolver seu relaxamento linear usando o método simplex com geração de colunas, conforme proposto por Gilmore e Gomory (1961). Essa abordagem introduz um problema auxiliar conhecido como problema de precificação, visando identificar e incorporar novas colunas que melhorem a solução existente.

Neste procedimento, é necessário fornecer padrões de corte iniciais para compor a base inicial do método simplex, que é então incrementalmente melhorado pela geração iterativa de padrões de corte superiores. Para isto, foi utilizada a mesma abordagem usada no Capítulo 3, que utiliza padrões de corte homogêneos e/ou unitários como base inicial.

Assim, assumimos que, em uma dada iteração do método de geração de colunas, temos um conjunto de cortes indexados $j = 1, \ldots, N^t$, onde N^t corresponde ao conjunto de padrões de corte gerados até aquele momento para o período $t = 1, \ldots, M$. Então, o programa linear mestre restrito para o 1D-PMCE-DE-SI-wIT, denotado por RMLP-wIT, é dado por:

$$\text{(RMLP-wIT)} \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M} \beta^t E^t + \sum_{t=1}^{M} \gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \tag{4.19}$$

$$\text{sujeito a:} \quad (4.3) - (4.11) \tag{4.20}$$

$$x_j^t \in \mathbb{R}_+, \ \forall j, \ \forall t; \tag{4.21}$$

$$E^t, T^t, q^t, z^t \in \mathbb{R}_+, \ \forall t; \tag{4.22}$$

$$S_j^t \in \{0,1\}, \quad \forall t,j; \tag{4.23}$$

$$e_i^t \in \mathbb{R}_+, \quad \forall t,i. \tag{4.24}$$

O programa linear mestre restrito para o 1D-PMCE-DE-SI-IT, denotado por RMLP-IT, é dado por:

$$\text{(RMLP-IT)} \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M} \beta^t E^t + \sum_{t=1}^{M} \gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \tag{4.25}$$

$$\text{sujeito a:} \quad (4.3) - (4.4), (4.6) - (4.11), (4.18), (4.21) - (4.24), \tag{4.26}$$

em que as restrições e variáveis nos modelos se referem apenas aos padrões de corte gerados até a iteração atual do método de geração de colunas.

Para gerar um novo padrão de corte capaz de melhorar o valor da função objetivo para RMLP-wIT ou RMLP-IT, precisamos identificar, para um dado período $t = 1, \ldots, M$, um padrão de corte viável. Este padrão de corte, quando adicionado ao conjunto existente de padrões de corte, tem o potencial de melhorar o valor da função objetivo do problema. A seleção do padrão de corte a ser adicionado é baseada nos custos reduzidos das colunas que são adicionadas, com preferência dada às colunas com os menores custos reduzidos em um problema de minimização. Observe que, no problema em questão, adicionar um padrão de corte j ao modelo para um certo período t envolve introduzir as variáveis x_j^t e S_j^t, bem como incluir uma restrição no conjunto de restrições (4.4). Portanto, identificar um padrão de corte promissor

se traduz em gerar duas colunas e introduzir uma linha na matriz de coeficientes. A restrição adicionada à matriz (relacionada ao conjunto (4.4)) introduz uma nova variável dual, que está associada ao "preço sombra"dessa nova restrição. Essa variável dual geralmente representa o custo marginal de atender à restrição com um novo padrão.

A seleção dos padrões de corte a serem incorporados ao modelo é orientada pela análise dos custos reduzidos associados às colunas candidatas, sendo priorizadas aquelas com os menores custos reduzidos, conforme a natureza de minimização do problema. No caso específico tratado, a inclusão de um padrão de corte j referente a um período t demanda a inserção simultânea de duas variáveis de decisão x_j^t , que representa a frequência de utilização do padrão, e S_j^t, associada ao estoque gerado. Adicionalmente, uma nova restrição deve ser acrescida ao conjunto (4.4), de forma a garantir a consistência entre o uso do padrão e os níveis de estoque. Dessa forma, a escolha de um padrão de corte promissor implica, a geração de duas colunas e a adição de uma linha à matriz de coeficientes do modelo, o que afeta diretamente a estrutura e o desempenho do método de geração de colunas adotado. No problema dual, isso se traduz em uma nova variável dual associada à nova restrição; e um ajuste no custo reduzido das novas variáveis, a partir dos valores das variáveis duais existentes.

Ao calcular os custos reduzidos, precisamos distinguir entre dois cenários. O primeiro cenário ocorre quando o custo e o tempo de produção associados a um padrão de corte são independentes do próprio padrão de corte, ou seja, $c_j^t = c^t$ e $p_j^t = p^t$. O segundo cenário surge quando pelo menos um desses parâmetros varia com base no padrão de corte. Um exemplo disso é quando o custo das perdas do processo de corte, referente a sobras inutilizáveis, é fatorado no custo de produção. Nesses casos, o custo de corte depende do padrão de corte específico. Nesta tese, o procedimento de geração de colunas considera apenas o primeiro cenário. Em outras palavras, o procedimento de geração de colunas não determina os padrões de corte e seus custos ou tempos de processamento específicos. No entanto, se os possíveis padrões de corte e seus parâmetros específicos forem conhecidos (segundo cenário), o procedimento matemático proposto neste trabalho também pode ser aplicado.

Considerando que em uma solução ótima para o RMLP, as restrições (4.4) não terão folgas, implicando $x_j^t = BS_j^t$, a variável S_j^t pode ser eliminada do RMLP e substituída por $\frac{x_j^t}{B}$. Além disso, ao associar as variáveis duais π_{ti} e σ^t com as restrições (4.3) e (4.6), respectivamente, podemos determinar a variável x_j^t associada ao menor custo reduzido buscando o padrão de corte viável $\alpha = (\alpha_i^t)_{i=1}^n$, para cada $t = 1, \ldots, M$, que minimiza o subproblema de precificação:

$$(SP^t) \quad \min \left(c^t + \frac{f^t}{B} - \sum_i \alpha_i^t \pi_i^t + \left(p^t + \frac{r^t}{B} \right) \sigma^t \right) \tag{4.27}$$

$$\sum_i \alpha_i^t \leq L \tag{4.28}$$

$$\alpha_i^t \in \mathbb{Z}_+. \tag{4.29}$$

O Algoritmo 8 descreve a Fase 1 da abordagem proposta. Nesta fase, uma solução da relaxação linear da formulação proposta é encontrada por meio do procedimento de geração de colunas. O pseudocódigo de geração de colunas é descrito por este

Algoritmo 8 Método de geração de colunas para o 1D-PMCE-DE-SI.

```
1: procedure Geração de colunas para o 1D-PMCE-DE-SI
2:     Relaxe as condições de integralidade do modelo
3:     Defina os padrões de corte homogêneos
4:     while uma nova coluna pode ser adicionada ao problema mestre restrito do
5:         for t ← 1 ate M do
6:             Resolva o problema mestre restrito por meio do método simplex
7:             Resolva o SP^t
8:             if ν(SP^t) < 0 then
9:                 Adicione uma nova coluna
10:            else
11:                Nenhuma coluna pode ser adicionada
12:            end if
13:        end for
14:    end while
15: end procedure
16: return x_j^t, S_j^t, e_ti, E^t e T^t.
```

algoritmo, no qual $\nu(SP^t)$ representa o valor ótima da função objetivo para o subproblema de precificação SP^t.

4.3.1.1 Um exemplo numérico

Esta seção apresenta um exemplo numérico do 1D-PMCE-DE-SI. Suponha que temos objetos em estoque disponíveis de comprimento $L = 100$ para atender aos tipos de itens dada na Tabela 4.3. A Tabela 4.4 mostra as solicitações da empresa. Além disso, todos os custos, tempos e penalizações são iguais a 1.

Tabela 4.3: Tipos de Itens

Item	1	2	3
Comprimento	65	24	15

Tabela 4.4: Solicitações da empresa

	Demanda de Item			
Período	1	2	3	Data D1
1	80	24	71	75
2	40	70	2	140
3	41	6	26	200

A solução é obtida usando o método de geração de colunas no Algoritmo 8 para 1D-PMCE-DE-SI-IT e 1D-PMCE-DE-SI-wIT, seguido pela aplicação de um *solver* para resolver o programa inteiro com as colunas geradas. Além disso, o procedimento que gera todas as colunas possíveis (padrões de corte) *a priori* também foi considerado. Note que essa abordagem é impraticável para grandes instâncias do problema.

Tabela 4.5: Decomposição do valor da função objetivo para cada período para o exemplo numérico do 1D-PMCE-DE-SI com e sem tempo ocioso.

			Componentes da função objetivo						
Versão	Períodos	Padrões	Objetos	*Setup*	Atraso	Adiantamento	Total	Estoque	DE
1D-PMCE-DE-SI-IT-CG	1	(1 0 2)	37	1	7	0	89	(0,19,3)	172
		(1 1 0)	43	1					
	2	(1 1 0)	57	1	0	0	58	(17,6,1)	
	3	(1 0 2)	24	1	0	0	25	(0,0,3)	
1D-PMCE-DE-SI-IT-CE	1	(1 0 2)	37	1	7	0	89	(0,19,3)	172
		(1 1 0)	43	1					
	2	(1 1 0)	57	1	0	0	58	(17,6,1)	
	3	(1 2 0)	24	1	0	0	25	(0,0,23)	
1D-PMCE-DE-SI-wIT-CG	1	(1 0 2)	54	1	7	0	89	(0,2,37)	207
		(1 1 0)	26	1					
	2	(0 4 0)	7	1	0	0	58	(0,0,71)	
		(0 0 6)	6	1					
		(1 1 0)	40	1					
		(0 0 1)	0	1					
		(1 0 2)	0	1					
	3	(1 0 0)	41	1	0	0	60	(0,2,129)	
		(0 4 0)	2	1					
		(0 0 6)	14	1					
1D-PMCE-DE-SI-wIT-CE	1	(1 0 2)	50	1	7	0	89	(0,6,29)	207
		(1 1 0)	30	1	0	0			
	2	(0 4 0)	16	1	0	0	58	(0,0,27)	
		(1 0 0)	40	1	0	0			
	3	(1 0 0)	41	1	0	14	60	(0, 0, 1)	
		(0 4 0)	1	1	0				
		(0 2 0)	1	1	0				

A Tabela 4.5 apresenta o valor de cada termo da função objetivo (FO) para as soluções obtidas, juntamente com os padrões de corte usados em cada período, o custo total para cada período e o estoque final de cada item no final de cada período. Nesta tabela, 1D-PMCE-DE-SI-wIT-GC e 1D-PMCE-DE-SI-IT-GC correspondem às soluções usando geração de colunas, enquanto 1D-PMCE-DE-SI-wIT-CE e 1D-PMCE-DE-SI-IT-CE correspondem à solução que enumera completamente as colunas.

Ambos os procedimentos de solução encontram os mesmos valores de função objetivo para as duas versões do problema, diferindo apenas na escolha dos padrões de corte. Isso demonstra a eficácia do procedimento de geração de colunas para esta instância. Como a função objetivo não considera os custos de estoque, as soluções encontradas pelos dois procedimentos de solução também diferem no estoque para cada item. Além disso, todos os resultados foram obtidos com um atraso de 7 unidades de tempo no primeiro período, pois a data de entrega é 75, e 82 unidades de tempo foram utilizadas em operações para atendimento da demanda (80 unidades para corte de padrão e 2 unidades para *setup*).

Observe que a solução para o problema sem tempo ocioso tem um valor de função objetivo maior do que com tempo ocioso. Neste caso, para o 1D-PMCE-DE-SI-wIT-CG, para evitar a penalidade de antecipação da entrega, a solução obtida resulta em um número maior de *setups* e objetos de corte, gastando 200 unidades de tempo para atender a mesma demanda que pode ser atendida com 165. Portanto, a versão do problema com tempo ocioso gera um plano de corte mais eficiente, exigindo menos tempo de operação.

4.3.2 Fase 2: Encontrando uma solução inicial

Em geral, a solução ótima para o problema relaxado é tipicamente fracionária. Esta seção apresenta duas técnicas para obter uma solução inteira. A Seção 4.3.2.1 discute a heurística *Round-Up*, um método direto para gerar uma solução inteira inicial, conforme explorado por Poldi e Arenales (2009) no contexto do problema de corte de estoque unidimensional com comprimentos de estoque múltiplos limitados. Em seguida, a Seção 4.3.2.2 apresenta a heurística *Relax-and-Fix*. Além disso, a Seção 4.4.3 compara o desempenho dessas heurísticas para determinar qual é mais adequada para resolver grandes instâncias do 1D-PMCE-DE-SI.

4.3.2.1 Fase 2: Heurística *Round-up*

Sejam x^* e S^* os valores das variáveis x e S na solução ótima do problema relaxado, onde pelo menos um componente de x^* é fracionário. Para obter valores inteiros $\widetilde{x}$ e $\widetilde{S}$ 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, como segue:

$$\widetilde{x_j}^t = \lceil x_j^{*t} \rceil \qquad j \in N \tag{4.30}$$

$$\widetilde{S}_j^t = 1 \qquad \text{if } x_j^{*t} \neq 0,\ \forall j \in \mathbb{N}. \tag{4.31}$$

A partir dos valores $\widetilde{x}$ e $\widetilde{S}$, e dos valores iniciais definidos nas restrições (4.8)-(4.11), os valores inteiros para as variáveis restantes podem ser determinados usando o seguinte sistema de equações:

$$\widetilde{e}_i^t = \sum_{j=1}^{N} a_{ji}^t \widetilde{x}_j^t + \widetilde{e}_i^{(t-1)} - b_i^t \qquad \forall t,\ i \tag{4.32}$$

$$\widetilde{z}^t = d^{t-1} + \widetilde{T}_{t-1} - \widetilde{E}^{t-1} \qquad \forall t \tag{4.33}$$

$$\widetilde{q}^t = \sum_{j=1}^{N} p_j^t \widetilde{x}_{jt} + \sum_{j=1}^{N} r_j^t \widetilde{S}_j^t \qquad \forall t \tag{4.34}$$

$$\widetilde{T}^t = \max\{0, (\widetilde{q}^t + \widetilde{z}^t) - d^t\} \qquad \forall t \tag{4.35}$$

$$\widetilde{E}^t = \max 0, d^t - (\widetilde{q}^t + \widetilde{z}^t) \qquad \forall t, \tag{4.36}$$

$$\tag{4.37}$$

em que as equações (4.32), (4.33) e (4.34) são derivadas das expressões (4.3), (4.5) e (4.6), respectivamente.

Entretanto, o foco principal do método de solução proposto é ter apenas uma solução inicial para melhoria subsequente na terceira fase. Caso, na Fase 3, seja utilizada a matheurística *Local Branching*, basta uma solução inicial compreendendo apenas uma parte de uma solução viável. Portanto, ao aplicar esta matheurística, consideramos apenas a construção de uma solução parcial, composta de valores inteiros para as variáveis x e S. Essa solução parcial serve como um ponto de partida, permitindo a geração de uma solução viável que seja pelo menos tão boa quanto a solução viável correspondente a essa solução parcial.

O procedimento *Round-Up* é altamente eficiente, produzindo uma solução (completa ou parcial) para o 1D-PMCE-DE-SI que adere às restrições de demanda, desde

que haja itens disponíveis para armazenamento e utilização em períodos subsequentes. No entanto, uma desvantagem é que este procedimento normalmente produz soluções inteiras com valores de função objetivo maiores que o valor ótimo. No entanto, uma vez que o objetivo é obter uma solução inteira (completa ou parcial) para o problema restrito, esta técnica fornece com sucesso tal solução, que então passará por melhorias na Fase 3 do método, conforme descrito a seguir. Quando as demandas de itens são grandes, o arredondamento normalmente produz uma solução de alta qualidade, pois o impacto no valor da função objetivo é relativamente menor.

O Algoritmo 9 descreve a heurística *Round-Up*, proposta como Fase 2 para a solução do procedimento trifásico do 1D-PMCE-DE-SI. Na Fase 2, a solução relaxada gerada a partir da Fase 1 é, neste caso, arredondada para cima.

Algoritmo 9 Heurística de arredondamento aplicada ao 1D-PMCE-DE-SI.

procedure HEURÍSTICA DE ARREDONDAMENTO
  Entrada: x^* e S^* a solução viável para o problema relaxado;
  Saída: $\widetilde{x}$ e $\widetilde{S}$ são soluções viáveis para o problema inteiro;
  for all $t \in T$ **do**
    for all $j \in I$ **do**
      $\widetilde{x}_j^t \leftarrow \lceil x_j^{*t} \rceil$
      $\widetilde{S}_j^t \leftarrow \lceil S_j^t \rceil$
    end for
  end for
end procedure

4.3.2.2 Fase 2: Heurística *Relax-and-Fix*

Existem várias estratégias para particionar as variáveis em subconjuntos ao implementar a heurística *Relax-and-Fix*. Nesta tese, adotamos a estratégia de particionamento de período. Seja Q^t o conjunto de variáveis de decisão inteiras associadas ao período t. Supõe-se que apenas as variáveis de decisão de corte x_j^t e as variáveis de *setup* S_j^t sejam inteiras, pois, ao fixar essas variáveis, se garantirá que as variáveis restantes também sejam inteiras. Consequentemente, Q^t contém todas as variáveis x_j^t e S_j^t associadas ao período t.

Assim, para cada período r, temos o seguinte subproblema $(IPRF_r)$ para o 1D-PMCE-DE-SI-wIT:

$$(IPRF_r) \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M} \beta^t E^t + \sum_{t=1}^{M} \gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \quad (4.38)$$

$$\text{sujeito a:} \quad (4.3) - (4.11), (4.12) - (4.13) \quad (4.39)$$

$$x_j^t = x_j^{t,r-1}\ , \quad \forall t \in \left(Q^1 \cup Q^2 \cup \ldots \cup Q^{r-1}\right), \forall j \quad (4.40)$$

$$S_j^t = S_j^{t,r-1}\ , \quad \forall t \in \left(Q^1 \cup Q^2 \cup \ldots \cup Q^{r-1}\right), \forall j \quad (4.41)$$

$$x_j^t \in \mathbb{Z}_+\ , \quad \forall t \in Q^r, \forall j \quad (4.42)$$

$$S_j^t \in \{0,1\}\ , \quad \forall t \in Q^r, \forall j \quad (4.43)$$

$$x_j^t \in \mathbb{R}_+\ , \quad \forall t \in Q - \left(Q^1 \cup Q^2 \cup \ldots \cup Q^r\right), \forall j \quad (4.44)$$

$$S_j^t \in [0,1]\ , \quad \forall t \in Q - \left(Q^1 \cup Q^2 \cup \ldots \cup Q^r\right), \forall j. \quad (4.45)$$

A formulação do subproblema para o 1D-PMCE-DE-SI-IT é definida pelas equações (4.3)-(4.4), (4.6)-(4.11), (4.18), (4.12)-(4.13) e (4.40)-(4.45).

O Algoritmo 10 detalha o pseudocódigo para a Fase 2 com a heurística *Relax-and-Fix*.

Algoritmo 10 *Relax-and-Fix* - RF

procedure RELAX-AND-FIX – RF
  Escolha uma decomposição disjuntiva para o conjunto de variáveis inteiras $Q = Q_1 \cup Q_2 \cup \ldots \cup Q_P$.
  Relaxe a integralidade de todas as variáveis inteiras.
  for all $r \in \{1, \ldots, P\}$ **do**
    Altere as variáveis associadas ao subconjunto Q_r para se tornarem inteiros novamente.
    Resolva o problema $(IPRF_r)$.
    Sejam x_j^r e S_j^r a solução ótima do problema $(IPRF_r)$.
    Fixe as variáveis associadas ao subconjunto Q_r em x_j^r e S_j^r, de forma que essas variáveis não devam ser alteradas nos próximos subproblemas.
  end for
end procedure

4.3.3 Fase 3: Melhorando a solução inicial

Essa fase recebe uma solução inicial, vinda da Fase 2, e tenta melhorá-la no decorrer do percurso. A seguir apresentaremos duas propostas para obtermos uma solução inicial: (i) implementação da matheurística *Local Branching*, descrita na Subseção 4.3.3.1; (ii) implementação da matheurística *Fix-and-Optimize*, conforme Subseção 4.3.3.2.

4.3.3.1 Matheurística *Local Branching*

Para a fase de melhoria a partir de uma solução inicial inteira, a implementação realizada da matheurística *Local Branching* é semelhante à realizada na Seção 3.3.3. Dessa maneira, as mesmas restrições e limitantes foram estabelecidos para a resolução desse problema, como já descritos no Algoritmo 3.

Então, de acordo com Fischetti e Lodi (2003), uma restrição de *Local Branching* pode ser definida como:

$$\Delta(x, x') := \sum_{t=1}^{M} \sum_{j \in J_l(x')} \mu_j^t (x_j^t - l_j^t) + \sum_{t=1}^{M} \sum_{j \in J_u(x')} \mu_j^t (u_j^t - x_j^t) \leq k. \tag{4.46}$$

O valor de μ_j^t, como sugerido por Fischetti e Lodi (2003), é dado por:

$$\mu_j^t = \frac{1}{u_j^t - l_j^t} \quad \forall j, t. \tag{4.47}$$

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 $x_j^{t\prime}$, o espaço de solução pode ser particionado por meio da disjunção:

$$\Delta(x,x') \leq k, \tag{4.48}$$

or

$$\Delta(x,x') \geq k+1. \tag{4.49}$$

O algoritmo utilizado aqui é o mesmo que foi descrito na Seção 3.3.3 com os mesmos parâmetros e estratégias.

4.3.3.2 Fase 3: Matheurística *Fix-and-Optimize*

Há outros métodos de melhoria de solução na literatura para problemas semelhantes que podem ser ajustados para o problema específico que foi estudado. Uma heurística para melhorar a solução inicial inteira é a matheurística *Fix-and-Optimize*.

Essa abordagem é útil em situações nas quais o espaço de busca é muito grande, permitindo que o problema seja resolvido em partes menores. Ao final das iterações, obtém-se uma solução factível otimizada para o problema inteiro. A solução final pode não ser a solução ótima global, já que a heurística não garante isso, mas em muitos casos fornece uma solução de alta qualidade em tempo computacional muito menor do que os métodos exatos aplicados diretamente ao problema completo. Esta tese utilizou a mesma estratégia de particionamento usada para particionar as variáveis para a matheurística *Relax-and-Fix*. Dessa maneira, as variáveis particionadas são as variáveis de corte x_j^t e as variáveis de *setup* S_j^t.

Assim, para cada período r, temos o seguinte subproblema $(IPFO_r)$ para o 1D-PMCE-DE-SI-wIT:

$$(IPFO_r) \quad \min \quad \sum_{t=1}^{M}\sum_{j=1}^{N^t} c_j^t x_j^t + \sum_{t=1}^{M}\beta^t E^t + \sum_{t=1}^{M}\gamma^t T^t + \sum_{t=1}^{M}\sum_{j=1}^{N^t} f_j^t S_j^t \tag{4.50}$$

$$\text{sujeito a:} \quad (4.3)-(4.11), \tag{4.51}$$

$$(4.12)-(4.15), \tag{4.52}$$

$$(4.42)-(4.43) \tag{4.53}$$

$$x_j^t = x_{j,r-1}^t\ , \quad \forall t \in (Q \setminus Q^r)\,, \forall j \tag{4.54}$$

$$S_j^t = S_{j,r-1}^t\ , \quad \forall t \in (Q \setminus Q^r)\,, \forall j. \tag{4.55}$$

A formulação do subproblema para o 1D-PMCE-DE-SI-IT é definida pelas expressões (4.3)-(4.4), (4.6)-(4.11), (4.18), (4.12)-(4.13),(4.42)-(4.43) e (4.54)-(4.55).

O Algoritmo 11 detalha a aplicação da heurística de melhoramento *Fix-and-Optimize* para a Fase 3 do procedimento proposto. Inicialmente, determina-se o particionamento das variáveis (linha 2) e a solução incumbente inicial (linha 3). Em seguida, para cada subconjunto, as variáveis são fixadas (linha 5), o subproblema é resolvido (linha 6) e a solução incumbente é atualizada, a partir da solução do subproblema corrente (linha 7).

Algoritmo 11 *Fix-and-Optimize* – FOp

procedure FIX-AND-OPTIMZE – FOP
  Escolha uma decomposição disjuntiva para o conjunto de variáveis inteiras $Q = Q^1 \cup Q^2 \cup \ldots \cup Q^P$
  Determine uma solução inicial incumbente.
  for all $r \in \{1, \ldots, P\}$ **do**
    Fixe as variáveis em $Q \setminus Q^r$ de acordo com a solução incumbente.
    Otimize o subproblema ($IPFO^r$).
    Atualize a solução incumbente com a solução ótima do subproblema.
  end for
end procedure

4.4 Experimentos Computacionais para o 1D-PMCE-DE-SI

Esta seção fornece uma análise aprofundada dos experimentos computacionais realizados para avaliar os modelos matemáticos propostos para o 1D-PMCE-DE-SI, introduzidos na Seção 4.2.2. Os testes computacionais foram realizados em um computador com um 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 12.7.1, com um limite de tempo de execução de 1200 segundos. A organização desta seção é estruturada da seguinte forma. A Subseção 4.4.1 delineia a metodologia por trás da geração das instâncias de teste usadas nos experimentos. A Subseção 4.4.2 descreve o processo de ajuste dos parâmetros associados ao algoritmo proposto. A Subseção 4.4.3 descreve o procedimento para escolher as heurísticas a serem usadas nas Fases 2 e 3 do algoritmo trifásico proposto para resolver o 1D-PMCE-DE-SI. A Subseção 4.4.4 é a parte principal desta seção, apresentando os resultados computacionais obtidos a partir dos experimentos, abrangendo tanto a versão do problema sem ociosidade quanto a com ociosidade.

4.4.1 Geração das Instâncias

Os testes computacionais foram projetados para avaliar de forma abrangente o desempenho do método de solução proposto. Para executar esses testes, desenvolvemos um gerador de instâncias para o 1D-PMCE-DE-SI usando a linguagem `C++`. Este gerador se inspirou nos conceitos encontrados no gerador CUTGEN 1, que se concentra na geração de problemas de corte unidimensional, proposto por Gau e Wäscher (1995). Utilizando este gerador de instâncias, criamos 36 classes de instâncias, cada uma caracterizada por parâmetros de entrada variáveis, para acessar completamente a eficácia do método proposto. Três parâmetros foram definidos:

(i) comprimento do objeto;

(ii) número de tipos de itens; e

(iii) número de períodos.

Posteriormente, seis parâmetros foram gerados aleatoriamente dentro de intervalos predefinidos:

(i) demanda do item;

(ii) comprimento do item;

(iii) datas de entrega;

(iv) penalidade por atraso;

(v) penalidade por antecipação; e

(vi) penalidade por *setup* associada à alteração do padrão de corte.

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: $n = 50$, 100 e 200;

(iii) Número de períodos: $t = 3$ e 6;

(iv) A demanda pelos itens foi escolhida aleatoriamente no intervalo $[1; 100]$;

(v) O comprimento dos itens foi escolhido aleatoriamente no intervalo $[0{,}02L; 0{,}8L]$;

(vi) As datas de entrega foram geradas de acordo com uma distribuição uniforme. Dois tipos de instâncias foram geradas: instâncias com datas largas (D1) e datas estreitas (D2) como segue:

(a) $d^t \in U[1{,}0\ CAP^t;\ 2{,}0\ CAP^t]$ para datas de entrega D1 (data larga);

(b) $d^t \in U[0{,}8\ CAP^t;\ 1{,}0\ CAP^t]$ para datas de entrega D2 (data estreita).

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

$$CAP^t = \frac{\sum_{i=1}^{n} \ell_i b_i^t}{L}; \tag{4.56}$$

(vii) β^t: penalização por atraso de pedidos, dada por:

(a) $\beta^t = 1$; ou

(b) $\beta^t \in U[1,p]$;

(viii) γ^t: penalização por antecipação de pedidos, dada por:

(a) $\gamma^t = 1$; ou

(b) $\gamma^t \in U[1,p]$;

(ix) f_j^t: penalização por *setup*, dada por:

(a) $f_j^t = 1$; ou

(b) $f_j^t \in U[1,p]$.

Os valores do parâmetro p foram definidos como $p \in \{1; 10; 100\}$. Para cada resultado apresentado, o valor usado para p é destacado.

Os valores de parâmetros escolhidos para o conjunto de instâncias do 1D-PMCE-DE-SI estão, em parte, alinhados com as definições apresentadas em Poldi e de Araujo (2016) e Ma et al. (2019a) para seus respectivos conjuntos de instâncias. No entanto, as instâncias em Poldi e de Araujo (2016) apresentaram objetos de comprimentos variados em estoque, um número limitado de tipos de itens e incluíram custos de manutenção associados, mas não levaram em conta atrasos ou antecipação em datas de entrega, tempos de *setup* ou ociosidade. Em contraste, as instâncias em Ma et al. (2019a) consideraram apenas um único comprimento de objeto, um pequeno número de tipos de itens e impuseram penalidades para alterar padrões de corte, mas também não trabalham com os tempos de *setup* e atrasos ou antecipação. Consequentemente, os parâmetros que selecionamos refletem o foco específico deste estudo e destacam suas distinções de trabalhos anteriores, conforme discutido na Seção 4.1.1 de revisão da literatura.

A Tabela 4.6 mostra as doze classes de instâncias que foram geradas usando o gerador de instâncias proposta nesta tese, com base nas definições acima. Cada classe é representada com a notação X-Y-Z, em que X indica a data de entrega; Y indica o número de períodos; e Z indica o número de tipos de itens.

Tabela 4.6: Características do conjunto de instâncias com comprimento de objeto de $L = 1000$.

Classes	Data de Entrega	Períodos	#Items	Classes	Data de Entrega	Períodos	#Items
C1	D1	3	50	C7	D1	6	50
C2	D1	3	100	C8	D1	6	100
C3	D1	3	200	C9	D1	6	200
C4	D2	3	50	C10	D2	6	50
C5	D2	3	100	C11	D2	6	100
C6	D2	3	200	C12	D2	6	200

Além dos parâmetros descritos na Tabela 4.6 e de acordo com as formulações matemáticas incluídas na Seção 4.2.2, resta definir os valores de c_j^t (custo para cortar um padrão de corte); r_j^t (tempo de *setup* do padrão de corte); p_j^t (tempo de corte do padrão de corte); e B (um valor constante grande o suficiente), também apresentados no modelo matemático. Esses parâmetros foram definidos como $c_j^t = 1$, $r_j^t = 1$, $p_j^t = 1$ e $B = 1000$. Em relação à fase de *Local Branching* (Algoritmo 3) e Expressões (4.46) e (4.47), devemos definir os parâmetros μ_j^t, dados por:

$$\mu_j^t = \frac{1}{u_j^t - l_j^t} \quad \forall j, t. \tag{4.57}$$

Esses valores foram baseados na definição introduzida por Fischetti e Lodi (2003). Na presente tese, l_j^t é definido como $l_j^t = 0$ e u_j^t é definido como o maior elemento da solução inteira inicial.

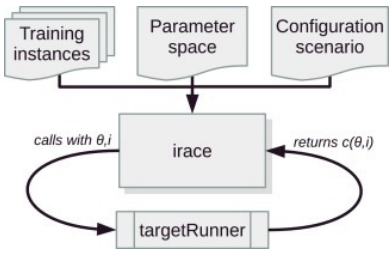


Figura 4.1: Esquema dos componentes fornecidos pelo usuário exigidos pelo pacote IRACE (López-Ibánez et al., 2016).

O algoritmo matheurístico *Local Branching* implementado requer a definição dos valores de dois parâmetros: k, que representa o tamanho das vizinhanças, e $h_{\max}$, que é o número máximo de ramificações. Nesta tese, definimos esses valores de acordo com o procedimento de calibração realizado usando o pacote IRACE, que está detalhado na Seção 4.4.2

4.4.2 Ajuste de parâmetros usando IRACE

O pacote IRACE, apresentado em López-Ibánez et al. (2016), serve para fazer a calibração de parâmetros para serem utilizados nos algoritmos. O pacote usa instâncias de treinamento e configurações de parâmetros geradas de acordo com as possibilidades fornecidas pelo usuário. Ele executa um algoritmo *iterative racing*, que busca determinar as melhores configurações para aquele grupo específico de instâncias.

A Figura 4.1 mostra como os vários componentes do pacote IRACE interagem entre si. O componente *Instâncias de treinamento* compreende um conjunto finito que representam as instâncias do problema em questão. Para ajustar os parâmetros do algoritmo, escolhemos 10% de instâncias de classes específicas, com cada classe compreendendo 20 instâncias. Portanto, 2 instâncias foram escolhidas para cada classe. Essa seleção envolveu variações na contagem de número de itens (instâncias com 50 e 100 itens), variações no número de períodos (3 e 6 períodos) e variações nas datas de entrega (instâncias representando as datas D1 e D2). Assim, escolhemos 24 instâncias de teste para essa calibração de parâmetros. No componente *Cenário de configurações*, as configurações padrão de corte para ajuste de parâmetros foram adicionadas. O componente *Espaço de parâmetros* definiu os parâmetros a serem ajustados, seus tipos, intervalos de valores e constantes. No caso presente, k e $h_{\max}$ foram os parâmetros sob calibração. Para k, o intervalo fornecido foi [5,40], e para $h_{\max}$, foi [1,30]. O tempo de *setup* foi realizado com valores fixos para penalização por atraso ($\beta^t = 1$), penalização por antecipação ($\gamma^t = 1$) e penalização por *setup* ($f_j^t = 1$). Após o ajuste de parâmetros, o pacote IRACE apresentou três configurações estatisticamente equivalentes. Para os experimentos computacionais, optamos por utilizar a *setup* com $k_0 = 20$ e $h_{\max} = 16$, pois demonstrou equivalência estatística às outras configurações.

4.4.3 Escolha de heurísticas para as Fases 2 e 3

Esta seção apresenta os testes computacionais realizados para determinar (i) a heurística para gerar a solução inteira inicial, e (ii) a heurística para refinar a solução inicial, ambas a serem adotadas pelo algoritmo de três fases proposto para a solução do 1D-PMCE-DE-SI. Para a geração da solução inteira inicial, as heurísticas *Round-Up* (Algoritmo 9, aqui denominada RU) e *Relax-and-Fix* (Algoritmo 10, aqui denominada RF) são avaliadas. Além disso, os procedimentos *Local Branching* (Algoritmo 3, aqui denominado LB) e *Fix-and-Optimize* (Algoritmo 11, aqui denominado FOp) também são testados quanto ao seu desempenho como matheurísticas de refinamento. Os experimentos utilizam a classe de instância com 50 tipos de itens, duas datas de entrega (D1 e D2), dois períodos de tempo (3 e 6) e valores do parâmetro $p = 1$, 10 e 100. Para $p = 1$, foram utilizados $\beta = 1$, $\gamma = 1$ e $f = 1$; para $p = 10$ e $p = 100$, foram utilizados β, γ, $f \in U[1,p]$.

A metodologia de teste adotada foi a seguinte: experimentos foram realizados para quatro estruturas de solução: RU→LB, RU→FOp, RF→LB e RF→FOp. A Tabela 4.7 (problema sem ociosidade) e a Tabela 4.8 (problema com ociosidade) resumem os resultados. É importante notar que todos os casos avaliados usaram a heurística de geração de colunas mostrada no Algoritmo 8 como Fase 1. Essas tabelas são organizadas da seguinte forma. As colunas 1 a 4 mostram os dados das instâncias; as colunas "RU" e "RF" fornecem os valores médios da função objetivo gerados pela solução inteira inicial da heurística *Round-Up* (Algoritmo 9) e da heurística *Relax-and-Fix* Algoritmo 10), respectivamente; a coluna "RU→LB" apresenta os valores médios da função objetivo gerados após a aplicação da matheurística *Local Branching* na solução inteira inicial gerada pela heurística *Round-Up*; as colunas "RF→LB", "RU→FOp" e "RF→FOp" representam situações semelhantes, *mutatis mutandis*, para as heurísticas *Relax-and-Fix* (RF) e *Fix-and-Optimize* (FOp).

Tabela 4.7: Modelo sem ociosidade: comparando os valores médios da função objetivo para instâncias da classe com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e os valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ para $p = 1$, e β, γ, $f \in U[1,p]$ para $p = 10$ e $p = 100$.

Dados da Instância				Int. inicial. Sol.		*Local Branching*		*Fix-and-Optimize*	
Itens	Data	Períodos	(α, β, f)	RU	RF	RU→LB	RF→LB	RU→FOp	RF→FOp
50	D1	3	p=1	46203,00	46197,20	46196,35	46197,20	46200,90	46197,20
			p=10	47589,00	46705,55	46625,60	46705,55	46971,00	46705,55
			p=100	59115,00	51011,20	50353,00	51011,20	53299,20	51011,20
50	D2	3	p=1	37001,00	36984,00	36984,00	36984,00	36991,90	36984,0
			p=10	38493,00	37503,40	37426,00	37503,40	37779,55	37503,40
			p=100	54134,00	45073,00	44275,30	45073,00	47547,50	45073,00
50	D1	6	p=1	90609,00	90594,50	90587,05	90594,50	90612,70	90594,50
			p=10	94278,00	91776,10	91650,00	91776,10	92730,10	91776,10
			p=100	128327,00	104103,10	103040,00	104103,10	113113,00	104103,10
50	D2	6	p=1	71205,00	71156,95	71149,00	71156,95	71196,20	71156,95
			p=10	77882,00	75041,65	74888,00	75041,65	76040,40	75041,65
			p=100	144991,00	118123,00	116600,00	118123,00	128017,80	118123,00
			Valores totais	889827,00	814269,65	809774,3	814269,65	840500,25	814269,65

Após uma análise completa das Tabelas 4.7 e 4.8, é evidente que o procedimento RF produziu consistentemente soluções iniciais de maior qualidade (ou seja, valores de

Tabela 4.8: Modelo com ociosidade: comparando os valores médios da função objetivo para instâncias da classe com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e os valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ para $p = 1$, e $\beta,\ \gamma,\ f \in U[1,p]$ para $p = 10$ e $p = 100$.

Dados da Instância				Int. inicial. Sol.		*Local Branching*		*Fix-and-Optimize*	
Itens	Data	Períodos	(α, β, f)	RU	RF	RU→LB	RF→LB	RU→FOp	RF→FOp
50	D1	3	p=1	4361,30	4349,30	4344,70	4349,30	4352,30	4349,30
			p=10	4729,20	4657,95	4620,30	4657,95	4668,50	4657,95
			p=100	7694,50	7256,76	6619,50	7256,76	7291,10	7256,76
50	D2	3	p=1	4441,50	4427,85	4425,10	4427,85	4433,50	4427,85
			p=10	4805,90	4746,80	4705,05	4746,80	4770,20	4746,80
			p=100	12050,00	10487,90	9886,70	10487,90	10578,85	10487,90
50	D1	6	p=1	8325,10	8294,90	8281,90	8294,90	8305,25	8294,90
			p=10	9202,40	9059,75	8968,80	9059,75	9161,05	9059,75
			p=100	18855,00	17213,85	16152,00	17213,85	17991,40	17213,85
50	D2	6	p=1	8662,60	8639,80	8619,90	8639,80	8659,70	8639,80
			p=10	12253,00	12120,25	12026,00	12120,25	12212,15	12120,25
			p=100	52335,00	50626,45	49695,00	50626,45	51477,20	50626,45
			Valores totais	147715,50	141881,56	138344,95	141881,56	143901,20	141881,56

função objetivo mais baixos) que o procedimento RU em todas as instâncias avaliadas. No entanto, quando os procedimentos de refinamento da solução foram aplicados, nem a matheurística LB nem a FOp produziram quaisquer melhorias sobre as soluções fornecidas por RF. Em contraste, embora as soluções geradas por RU fossem de qualidade inferior, elas permitiram que os procedimentos de refinamento alcançassem resultados de melhor qualidade.

Em relação ao desempenho dos dois procedimentos de refinamento, descobrimos que o LB supera consistentemente o FOp em todas as instâncias usando soluções iniciais fornecidas por RU. Consequentemente, a sequência RU→LB produziu soluções de melhor qualidade para todas as instâncias avaliadas em comparação aos outros três procedimentos. Esta sequência foi selecionada para servir como procedimentos de Fase 2 e Fase 3 para os experimentos computacionais subsequentes apresentados mais adiante nesta tese.

4.4.4 Resultados dos Testes Computacionais

Esta seção apresenta os resultados dos testes computacionais, realizados com diferentes valores das penalizações $[\beta, \gamma, f]$. A seguinte notação é adotada ao longo desta seção para representar o procedimento de solução:

(i) LB: representa os resultados após a aplicação da Fase 3 (Algoritmo 3) do algoritmo proposto, ou seja, após a aplicação do algoritmo de *Local Braching*;

(ii) CPLEX: representa os resultados da resolução do programa inteiro através do *solver* CPLEX usando as mesmas colunas da Fase 1 (Algoritmo 8) do algoritmo proposto.

Para avaliar os resultados dos testes computacionais referentes ao valor da função objetivo, definimos:

$$Gap_F = 100 \times \frac{\nu(\text{LB}) - \nu(\text{CPLEX})}{\nu(\text{LB})}. \tag{4.58}$$

O valor Gap_F representa a diferença relativa entre o valor da função objetivo obtido pela fase final do procedimento proposto ($\nu(LB)$) e o valor encontrado pelo procedimento CPLEX ($\nu(CPLEX)$). É importante notar que foi reduzido o limite de tolerância do *gap* padrão de corte usado pelo *solver* CPLEX. Ao usar o valor padrão de corte, o procedimento matemático proposto encontra soluções melhores que a obtida pelo CPLEX.

Além disso, para avaliar os resultados obtidos com respeito ao tempo de execução necessário para resolver as instâncias de teste, definimos:

$$Gap_{RT} = 100 \times \frac{\tau(\text{LB}) - \tau(\text{CPLEX})}{\tau(\text{LB})}. \tag{4.59}$$

O valor Gap_{RT} representa a diferença relativa entre o valor de tempo de execução obtido pela fase final do procedimento proposto ($\tau(LB)$) e o valor encontrado pelo procedimento CPLEX ($\tau(CPLEX)$).

A seção é organizada da seguinte forma. A Subseção 4.4.4.1 mostra os resultados envolvendo a formulação sem ociosidade na máquina de corte, conforme descrito na Subseção 4.2.2.1. A Subseção 4.4.4.2 apresenta os resultados para a formulação permitindo ociosidade na máquina de corte, conforme mostrado na Subseção 4.2.2.1. Em relação ao Algoritmo 3 proposto, esses resultados mostram o número de colunas não homogêneas geradas na Fase 1; o valor total da função objetivo encontrada na última fase do algoritmo proposto (Fase 3 – matheurística *Local Branching* – Algoritmo 3), bem como os valores de cada um dos seus quatro termos; e o tempo de execução computacional. Os resultados da função objetivo e do tempo de execução são comparados com aqueles referentes ao procedimento CGPX.

4.4.4.1 Resultados para a formulação sem ociosidade

Esta seção apresenta os resultados para os testes computacionais com a formulação, mostrada na Subseção 4.2.2.1, que não permite ociosidade na máquina de corte. Ela também analisa o comportamento do modelo em relação a duas propostas para penalizar os termos da função objetivo: primeiro, avalia o caso quando $\beta = \gamma = f = 1$; segundo, avalia o caso quando β, γ e $f \in U[1,p]$ para diferentes valores de p. As Tabelas 4.9, 4.10 e 4.11 apresentam esses resultados.

Resultados para a formulação sem ociosidade: caso $\beta = 1$, $\gamma = 1$ e $f = 1$

A Tabela 4.9 mostra os valores encontrados quando os modelos sem ociosidade foram penalizados com valores de $\beta = 1$, $\gamma = 1$ e $f = 1$, ou seja, $p = 1$, para classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2) e os dois valores de números de período (3 e 6). Vale lembrar que valores negativos nas colunas $Gap(\%)$ identificam a melhor qualidade da solução LB em comparação ao CPLEX. Para essas instâncias, o procedimento LB teve desempenho semelhante ao procedimento CPLEX em relação aos valores encontrados para a função objetivo. Essa afirmação é apoiada pelos valores médios de *gap* percentual mostrados na coluna Gap_F, pois o valor médio do *gap* percentual para todas as classes é 0,0000%. Além disso, a diferença percentual entre os valores totais da função objetivo encontrados pelo CPLEX e pelo LB também é +0,0000%. Assim, o valor da função objetivo

encontrado pelo procedimento LB foi melhor ou igual à solução do CPLEX em 9 das 12 classes, com um valor médio de *gap* negativo ou igual a 0,0%. Por outro lado, em relação aos valores de tempo de execução, o procedimento LB superou o procedimento CPLEX por uma ampla margem. Para a coluna Gap_{RT}, o valor médio para todas as classes é $-294{,}6383\%$. A diferença percentual entre o tempo de execução total encontrado pelo CPLEX e pelo LB é $-38{,}9473\%$. Além disso, em 10 das 12 classes, o procedimento LB teve desempenho superior, obtendo um tempo de execução menor que o encontrado pelo CPLEX. Vale ressaltar que a coluna "*Setup*", que informa o número de alterações de *setup*, contabiliza também, consequentemente, as alterações de padrão de corte que foram feitas e, portanto, o número de colunas que foram utilizadas pelo algoritmo proposto. Observa-se que poucas colunas foram geradas ao final da Fase 1 (coluna "Colunas"), sendo utilizadas, na maioria dos casos, colunas homogêneas geradas inicialmente para obtenção da solução inicial. A título de exemplo, para a classe com 50 tipos de itens e 3 períodos, seria possível utilizar 150 colunas homogêneas, 20 em cada um dos períodos; no entanto, foram utilizadas no máximo 114 colunas para esta classe, para a data D2.

Ao analisar a tabela, uma observação significativa é a notável disparidade nos valores da função objetivo, de Atraso e de Adiantamento para a mesma classe de instância. Esta disparidade está ligada ao número limitado de colunas geradas na Fase 1 e utilizadas na solução. Essa escassez de colunas resultou no uso de vários padrões de corte homogêneos, cada um contendo apenas um único item a ser cortado, consequentemente levando a um avanço nas ordens de corte. Além disso, observe que a razão Adiantamento/Atraso (razão E/T) varia significativamente com a largura da janela de tempo, sendo maior para janelas de tempo mais largas (D1) e menor para janelas de tempo mais estreitas (D2). Janelas de tempo estreitas produzem valores de atraso mais altos e um número menor de objetos de corte, enquanto janelas de tempo mais largas produzem valores de adiantamento maiores e um número maior de objetos de corte. Além disso, o comportamento dos valores de *setup* é interessante. Os valores de *setup* permanecem relativamente consistentes para várias janelas de tempo com o mesmo número de itens e períodos. No entanto, um aumento no número de itens a serem cortados (de 50 para 200 itens) leva a um aumento no valor de configuração, o que é esperado, considerando que envolve uma mudança mais significativa nos padrões de corte. Um efeito semelhante é observado ao considerar um número maior

Tabela 4.9: Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ ($p = 1$).

Dados da Instância			Colunas	Termos da Função Objetivo para LB				LB		CPLEX		$Gap(\%)$	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	2,75	16,70	14698,85	109,25	31371,55	46196,35	6,89	46196,35	60,02	0,00	-771,11
50	D2	3	3,13	97,70	11294,05	114,00	25478,00	36983,75	2,14	36983,75	55,67	0,00	-2501,40
50	D1	6	2,56	54,80	14536,60	250,30	75745,35	90587,05	33,82	90587,05	58,26	0,00	-72,26
50	D2	6	2,93	398,20	10892,95	249,60	59608,05	71148,80	116,45	71148,80	108,46	0,00	6,86
100	D1	3	5,11	21,50	19842,50	233,00	69258,25	89355,25	58,81	89355,50	60,05	-0,00	-2,10
100	D2	3	8,18	155,45	11999,60	228,05	58855,90	71239,00	58,98	71238,80	63,77	0,00	-8,12
100	D1	6	5,42	77,55	372,05	72,85	149525,85	179442,40	56,18	179443,05	52,84	-0,00	5,94
100	D2	6	7,84	707,9	22089,35	501,8	118086,85	141385,90	61,69	141385,95	116,96	0,00	-89,59
200	D1	3	11,26	46,55	25804,35	470,35	150975,75	177297,00	56,44	177297,40	60,01	-0,00	-6,32
200	D2	3	20,75	336,10	13873,95	431,45	122001,05	139091,95	54,05	139092,60	73,50	-0,00	-35,98
200	D1	6	10,30	122,30	35898,10	938,30	315963,05	352921,75	51,49	352920,10	55,56	0,00	-7,90
200	D2	6	18,78	988,05	17202,20	1003,20	259303,70	278497,15	59,56	278496,70	91,51	0,00	-53,64
						Total		1674146,35	616,50	1674146,05	856,61	0,00	-38,94

de períodos (de 3 para 6 períodos).

Resultados para a Formulação sem Ociosidade: caso β, γ, e $f \in U[1,p]$ para $p \in \{10,100\}$

Esta seção avalia o impacto da utilização de valores de penalização (β, γ, f) de uma distribuição uniforme, em que β, γ e $f \in U[1,p]$ e $p >> 1$. Notavelmente, nesses casos, os valores de penalização para vários termos na função objetivo provavelmente são diferentes. Além disso, pretendemos avaliar como o modelo sem ociosidade se comporta com valores variáveis de p. As Tabelas 4.10 e 4.11 mostram os resultados para os valores de β, γ e $f \in U[1,p]$ para $p = 10$ e $p = 100$, respectivamente. Essas tabelas incluem classes com 50, 100 e 200 tipos de itens para corte, considerando dois tipos de datas de entrega (D1 e D2) e dois números diferentes de períodos (3 e 6). É importante observar que as Tabelas 4.10 e 4.11 seguem uma estrutura semelhante à da Tabela 4.9.

Tabela 4.10: Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 10$.

Dados da Instância			Colunas	Termos da Função Objetivo para LB				LB		CPLEX		*Gap*(%)	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	2,90	22,60	3502,25	74,90	42586,60	46625,60	6,03	46625,50	949,33	0,00	-15643,44
50	D2	3	2,98	103,95	2859,40	75,05	33935,60	37426,35	44,02	37426,15	951,44	0,00	-2061,38
50	D1	6	2,45	52,70	1724,50	132,80	88659,55	91650,30	151,33	91649,15	1137,49	0,00	-651,66
50	D2	6	2,79	400,95	1389,55	125,90	69221,55	74888,45	262,81	74888,30	1131,11	0,00	-330,39
100	D1	3	5,20	21,45	3642,70	171,05	85521,00	90193,40	169,39	90191,45	1199,49	0,00	-608,12
100	D2	3	8,06	156,90	2916,55	167,05	67999,15	72484,70	179,67	72483,40	1142,3	0,00	-535,77
100	D1	6	5,45	52,30	3170,22	276,55	176199,45	180743,95	241,37	180743,00	1070,83	0,00	-343,64
100	D2	6	7,34	703,70	2279,40	252,60	138116,70	147779,00	194,73	147780,35	1130,36	-0,00	-480,47
200	D1	3	15,48	0,30	10864,80	340,20	166023,95	179143,80	116,61	179149,20	1137,50	-0,00	-875,47
200	D2	3	20,75	333,25	9852,50	333,70	128570,15	142659,50	74,95	142661,80	1135,58	-0,00	-1415,11
200	D1	6	10,01	117,80	9377,39	557,30	342872,75	356142,60	573,73	356134,30	1171,49	0,00	-104,18
200	D2	6	18,86	1015,25	7861,95	516,75	269131,60	287637,45	322,50	287636,00	1130,60	0,00	-250,57
							Total	1707375,10	2337,14	1707368,60	13287,52	0,00	-468,54

Tabela 4.11: Modelo sem ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 100$.

Dados da Instância			Colunas	Termos da Função Objetivo para LB				LB		CPLEX		*Gap*(%)	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	2,90	16,45	0,00	83,50	46088,95	50353,00	9,92	50353,00	1075,70	0,00	-10743,75
50	D2	3	3,18	98,05	0,10	81,10	36797,25	44275,30	81,48	44275,30	1074,49	0,00	-1218,71
50	D1	6	2,51	54,50	0,00	141,90	90390,20	103040,05	258,72	103033,35	1197,92	0,00	-363,01
50	D2	6	2,78	399,40	0,00	134,00	70617,70	116600,40	410,59	116595,55	1189,79	0,00	-189,77
100	D1	3	4,96	27,35	2412,45	170,00	86752,15	98556,40	180,60	98557,05	1136,09	-0,00	-529,06
100	D2	3	8,01	164,20	2026,50	164,90	68888,40	81162,60	239,65	81165,30	1140,82	-0,00	-376,03
100	D1	6	5,44	60,59	730,95	283,00	178345,60	194081,35	305,28	194113,25	1191,24	-0,01	-290,21
100	D2	6	7,59	704,95	525,15	265,55	139886,85	173941,30	414,84	173963,10	1186,93	-0,012	-186,11
200	D1	3	11,56	48,95	54,32	346,65	176904,25	201958,85	235,53	202014,80	1196,68	-0,02	-408,07
200	D2	3	19,76	337,45	0,00	336,60	138419,95	183582,35	174,85	183673,15	1190,46	-0,04	-580,84
200	D1	6	9,89	195,80	0,00	561,25	352245,05	389343,35	254,17	389365,75	1170,46	-0,00	-360,50
200	D2	6	17,82	1251,75	0,00	513,70	276996,00	353272,65	388,11	353185,95	1188,30	0,02	-206,17
							Total	1990167,60	2953,74	1990295,55	13938,88	-0,00	-371,91

Essas tabelas merecem diferentes camadas de análise. A primeira camada diz respeito aos valores da função objetivo. Na Tabela 4.10, o procedimento LB demonstra um comportamento semelhante ao caso $p = 1$ na Tabela 4.9 em relação aos valores

da função objetivo, alinhando-se estreitamente com o procedimento CPLEX. Para 3 das 12 classes, o valor da função objetivo LB foi até superior à solução CPLEX; a diferença percentual entre o valor total da função objetivo encontrado pelo procedimento LB e o procedimento CPLEX é de meros +0,0004%. Por outro lado, na Tabela 4.11, o procedimento LB supera notavelmente o CPLEX: para 2 das classes exibidas, o valor da função objetivo é idêntico para ambos os procedimentos e para 8 classes, o procedimento LB atinge um valor de função objetivo superior. A diferença percentual entre os valores totais da função objetivo LB e CPLEX é −0,0064%.

A segunda camada está relacionada ao tempo de execução exigido pelo procedimento LB e pelo procedimento CPLEX ao lidar com essas instâncias. Notavelmente, as instâncias se tornam cada vez mais desafiadoras com valores mais altos de p em comparação com $p = 1$. O tempo de execução total para o procedimento LB aumenta de 616,5000 segundos ($p = 1$) para 2337,1400 segundos ($p = 10$) e 2953,7400 segundos ($p = 100$). Quanto ao procedimento CPLEX, o tempo de execução total aumenta para 856,6100 segundos ($p = 1$), 13287,5200 segundos ($p = 10$) e 13938,8800 segundos ($p = 100$). O procedimento LB supera o procedimento CPLEX para ambos os cenários avaliados $p = 10$ e $p = 100$. Em todas as classes avaliadas, o tempo de execução do procedimento LB supera consistentemente o do procedimento CPLEX. Esses valores de tempo de execução ressaltam o desempenho superior do procedimento LB no tratamento das instâncias de teste em consideração.

A terceira camada de análise envolve uma comparação entre as Tabelas 4.9 ($\beta = 1, \gamma = 1$, e $f = 1$), 4.10 (β, γ, e $f \in U[1,p]$, para $p = 10$) e 4.11 (β, γ, e $f \in U[1,p]$, para $p = 100$), esclarecendo os impactos dessa alteração nas instâncias de teste ao abordar o modelo sem ociosidade. Em todos os casos, a fase de geração de colunas é notavelmente rápida, pois a maioria das instâncias não produz novas colunas e utiliza predominantemente as colunas iniciais (homogêneas). Além disso, exceto para a classe com 200 tipos de itens, data de entrega D1 e 3 períodos, o número de colunas geradas permanece relativamente estável à medida que o valor de p aumenta de $p = 1$ para $p = 10$ e $p = 100$. Por outro lado, os valores de *setup* diminuem de $p = 1$ para $p = 10$, excluindo a classe com 100 tipos de itens, data de entrega D1 e 6 períodos; no entanto, eles permanecem relativamente consistentes à medida que p aumenta de $p = 10$ para $p = 100$. Por outro lado, os valores de antecipação são significativamente impactados pelo aumento no valor da penalidade. Os valores totais caem de 198504,55 ($p = 1$) para 59441,21 ($p = 10$) e 5749,47 ($p = 100$). Para $p = 100$, em 6 das 12 classes abordadas, o valor de antecipação é zero. Por outro lado, o número de objetos cortados cresce com o aumento do valor de p, indo de 1436173,35 ($p = 1$) para 1608838,05 ($p = 10$) e 1662332,35 ($p = 100$). Em essência, com a elevação do valor da penalidade, a solução do modelo corresponde à cessação da antecipação da produção, alinhando a produção dentro do intervalo especificado pela data de entrega.

4.4.4.2 Resultados para a formulação com ociosidade

Esta seção mostra os resultados para os testes computacionais com a formulação que permite ociosidade na máquina de corte, conforme introduzido na Subseção 4.2.2.2. As Tabelas 4.12, 4.13 e 4.14 apresentam esses resultados. A estrutura desta seção espelha a da Subseção 4.4.4.1, analisando o comportamento do modelo

com relação às penalizações dos termos da função objetivo. Primeiro, examinamos o cenário em que $\beta = 1$, $\gamma = 1$ e $f = 1$, seguido pelo cenário em que β, γ e $f \in U[1,p]$ para dois valores de p.

Resultados para a formulação com ociosidade: caso $\beta = 1$, $\gamma = 1$ e $f = 1$

A Tabela 4.12 apresenta os resultados computacionais para os modelos com ociosidade, penalizados com $\beta = 1$, $\gamma = 1$ e $f = 1$ (ou seja, $p = 1$), considerando classes com 50, 100 e 200 tipos de itens, dois tipos de datas de entrega (D1 e D2) e dois valores de números de período (3 e 6). Conforme observado na formulação sem ociosidade, o procedimento LB proposto corresponde de perto ao desempenho do procedimento CPLEX em termos dos valores da função objetivo. O *gap* entre os valores totais da função objetivo para ambos os métodos é de apenas −0,0131, enfatizando a eficácia do procedimento LB. O valor médio de Gap_F é −0,0084. Notavelmente, na classe com 100 tipos de itens, data de entrega D2 e 6 períodos, o procedimento CPLEX supera ligeiramente o procedimento LB com um *gap* de 0,0006 no valor da função objetivo. No entanto, essa pequena diferença não diminui a eficácia geral do procedimento LB.

Em relação ao tempo de execução, o procedimento LB não exibiu um padrão de corte consistente em comparação com a formulação sem ociosidade. Na Tabela 4.12, a coluna Gap_{RT} mostra que para seis das doze classes, o procedimento LB tem um tempo de execução significativamente menor que o procedimento CPLEX. No entanto, para as seis classes restantes, o tempo de execução do procedimento LB é pior que o do procedimento CPLEX. No entanto, o tempo de execução médio total para o procedimento LB é 16,6206 segundos a menos do que o do solucionador.

Note que o número de colunas geradas (mostrado na coluna "Colunas") é consistentemente maior que o valor encontrado para *setup* (coluna "*Setup*"). Este fato caracteriza fortemente a formulação com ociosidade. Portanto, se o valor de *setup* representa o número de colunas utilizadas pelo procedimento de solução proposto, pode-se concluir que há colunas geradas que são descartadas e não utilizadas na solução. Este descarte varia de 3,38% (para o caso com 200 itens, entrega D1 e 6 períodos) a 38,80% (para o caso com 200 itens, entrega D1 e 3 períodos). Esta porcentagem de descarte está associada ao número de períodos considerados, sendo 3,85 vezes maior para 3 períodos em comparação com 6 períodos. Uma característica proeminente dos

Tabela 4.12: Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por $\beta = 1$, $\gamma = 1$ e $f = 1$ ($p = 1$).

Dados da Instância			Colunas	Termos da Função Objetivo para LB				LB		CPLEX		*Gap*(%))	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	65,96	16,35	0,00	53,40	4274,95	4344,70	138,44	4344,70	952,07	0,00	-85,45
50	D2	3	66,45	96,20	0,00	53,95	4274,95	4425,10	244,43	4425,10	892,00	0,00	-72,59
50	D1	6	67,32	55,05	0,00	59,95	8166,85	8281,85	937,02	8281,90	1175,14	0,00	-20,26
50	D2	6	65,92	392,90	0,00	60,20	8166,75	8619,85	836,75	8620,90	1127,00	-0,01	-25,75
100	D1	3	153,81	1,20	0,00	107,75	8020,40	8129,35	1069,62	8129,60	1171,62	0,00	-8,70
100	D1	6	135,56	24,40	0,00	122,90	16055,90	16203,20	1150,28	16203,70	1147,70	-0,00	0,22
100	D2	6	135,97	693,65	0,00	128,05	16053,60	16875,30	1135,52	16875,20	1134,36	0,00	0,10
200	D1	3	359,00	0,00	0,00	219,70	15579,65	15799,35	1090,95	15800,95	1078,46	-0,01	1,15
200	D2	3	361,24	318,65	0,00	224,50	15576,20	16119,35	1075,25	16120,75	1032,05	-0,00	4,18
200	D1	6	271,83	1,95	0,00	262,65	31162,40	31427,00	947,78	31442,20	934,61	-0,04	1,40
200	D2	6	288,45	975,45	0,00	268,75	31179,55	32423,75	941,70	32425,40	907,76	-0,00	3,73
							Total	170914,00	10580,75	170936,45	12689,88	-0,01	-16,62

resultados na Tabela 4.12 é a presença consistente de um valor zero para antecipação em todas as classes de instância. Isso indica que a formulação com ociosidade efetivamente impede qualquer avanço na produção. Por outro lado, os valores de atraso apresentam variação significativa. Quando a data de entrega é D1 (janela de tempo ampla), os valores de atraso diminuem conforme o número de itens aumenta, eventualmente chegando a zero para a classe com 200 itens e três períodos de tempo. No entanto, para uma data de entrega de D2 (janela de tempo estreita), os valores de atraso aumentam com o número de itens. Esse efeito não se estende ao número de objetos cortados, que varia principalmente com base no número de itens e períodos. Especificamente, para um determinado número de itens e a mesma data de entrega, alterar o número de períodos pode efetivamente dobrar o número de objetos cortados.

Resultados para a Formulação com Ociosidade: caso β, γ e $f \in U[1,p]$ para $p \in \{10,100\}$

Esta seção apresenta os resultados para o modelo com ociosidade usando valores de penalização (β, γ, f) extraídos de uma distribuição uniforme, em que $\beta, \gamma, f \in U[1,p]$ para um grande p ($p >> 1$). A Tabela 4.13 exibe os resultados para $p = 10$, enquanto a Tabela 4.14 mostra os resultados para $p = 100$. Essas tabelas contém classes com 50, 100 e 200 tipos de itens a serem cortados, considerando dois tipos de datas de entrega (D1 e D2) e dois valores para o número de períodos (3 e 6). Além disso, a estrutura dessas tabelas se assemelha à da Tabela 4.12.

Tabela 4.13: Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), os dois valores dos números de período (3 e 6) e valores de penalização para atraso, antecipação e *setup* dados por β, γ, e $f \in U[1,p]$ para $p = 10$.

Dados da Instância			Colunas	Termos da Função Objetivo para LB				LB		CPLEX		*Gap*(%)	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	66,13	21,25	0,00	48,65	4290,89	4620,30	437,88	4620,50	1149,17	-0,00	-61,89
50	D2	3	65,99	100,45	0,00	48,80	4293,30	4705,05	478,43	4705,30	1143,88	-0,00	-58,17
50	D1	6	271,83	52,85	0,00	57,20	8177,10	8968,80	944,63	8969,30	1175,33	-0,00	-19,62
50	D2	6	288,45	391,95	0,00	58,75	8173,75	12025,55	1041,45	12026,25	1165,59	-0,00	-10,6504
100	D1	3	148,98	0,50	0,00	98,80	8053,50	8580,90	1168,75	8587,00	1164,53	-0,07	0,36
100	D2	3	148,46	137,85	0,00	100,05	8031,00	9129,35	1169,43	9137,80	1168,40	-0,09	0,08
100	D1	6	271,83	15,15	0,00	120,05	16069,80	16666,85	1124,68	16682,95	1138,04	-0,09	-1,17
100	D2	6	288,45	691,30	0,00	124,70	16066,00	22776,95	1091,81	22781,20	1055,38	-0,01	3,45
200	D1	3	346,69	0,00	0,00	190,00	15687,30	16950,65	1101,09	17018,90	1103,92	-0,40	-0,25
200	D2	3	353,31	317,05	0,00	195,30	15683,95	18931,20	1092,72	18950,35	1035,75	-0,10	5,50
200	D1	6	271,83	0,00	0,00	239,85	31264,50	32491,45	877,01	32504,85	905,71	-0,04	-3,16
200	D2	6	288,45	979,75	0,00	250,30	31253,25	40397,05	956,41	40408,10	899,90	-0,02	6,27
							Total	196244,10	11484,29	196392,50	13105,60	-0,0756	-12,3711

Essas tabelas ilustram a influência da penalização no modelo com ociosidade. O primeiro impacto avaliado é observado nos valores da função objetivo. Em ambas as Tabelas, as colunas "Gap_F" demonstram a superioridade do procedimento LB sobre o procedimento CPLEX, exceto para a classe com 200 itens, data de entrega D2 e 6 períodos, ao considerar um fator de penalização de $p = 100$. Essa superioridade cresce de $-0{,}0131$ ($p = 1$) para $-0{,}0756$ ($p = 10$) e $-0{,}0893$ ($p = 100$). À medida que o fator de penalização aumenta, o procedimento LB se adapta melhor para fornecer uma solução para a instância em termos do valor da função objetivo.

O segundo impacto avaliado diz respeito ao tempo de execução, revelando padrões intrigantes. O tempo de execução total do procedimento LB é melhor que o alcançado pelo procedimento CPLEX em 12,37% ($p = 10$) e 29,94% ($p = 100$). No

Tabela 4.14: Modelo com ociosidade: valores médios para 20 instâncias para cada uma das classes com 50, 100 e 200 tipos de itens, considerando os dois tipos de datas de entrega (D1, D2), os dois valores dos números de período (3, 6) e valores de penalização para atraso, antecipação e *setup* dado por β, γ, e $f \in U[1,p]$ para $p = 100$.

Dados da Instância			Colunas	Função Objetivo terms for LB				LB		CPLEX		*Gap*(%)	
Items	Data de entrega	Períodos		Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)	Gap_F	Gap_{RT}
50	D1	3	65,70	16,45	0,00	41,50	4449,05	6619,45	144,21	6619,65	1168,20	-0,00	-87,65
50	D2	3	66,18	95,85	0,00	41,60	4462,50	9886,70	204,76	9888,10	1160,31	-0,01	-82,35
50	D1	6	271,83	55,10	0,00	48,35	8418,20	16152,00	864,28	16162,15	1180,87	-0,06	-26,80
50	D2	6	288,45	393,45	0,00	49,70	8387,00	49695,30	738,11	49699,20	1125,81	-0,00	-34,43
100	D1	3	150,20	4,05	0,00	74,45	8497,90	12190,70	703,75	12198,00	1169,13	-0,05	-39,80
100	D2	3	153,80	145,45	0,00	76,95	8447,80	12852,80	761,30	12857,45	1175,50	-0,03	-35,23
100	D1	6	271,83	23,45	0,00	98,55	16376,10	22239,15	1137,93	22214,10	1117,97	0,11	1,78
100	D2	6	288,45	695,75	0,00	104,70	16445,25	41988,15	1063,35	42068,70	1097,00	-0,19	-3,06
200	D1	3	347,01	0,00	0,00	140,40	16666,45	25095,20	872,04	25181,10	1076,81	-0,34	-19,01
200	D2	3	356,18	319,40	0,00	148,40	16644,05	49934,50	911,09	49988,55	1049,98	-0,10	-13,22
200	D1	6	271,83	0,00	0,00	183,60	32500,40	42956,25	869,98	43084,50	917,01	-0,29	-5,12
200	D2	6	288,45	979,90	0,00	198,10	32276,40	91055,00	944,54	91043,85	915,00	0,01	3,22
							Total	380665,20	9215,34	381005,35	13153,59	-0,08	-29,94

entanto, para $p = 10$ e classes com 100 tipos de itens, a situação é invertida, com o procedimento CPLEX superando o procedimento LB em 2,73% (100 itens) e 8,35% (200 itens). Assim, para $p = 10$, o procedimento LB supera o tempo de execução do procedimento CPLEX em 7 de 12 classes.Em outras palavras, para instâncias com um número maior de itens, o tempo de execução do procedimento CPLEX foi menor que o tempo computacional exigido pelo procedimento LB. Por outro lado, para $p = 100$, o procedimento LB demonstra um tempo de execução significativamente melhor, superando o procedimento CPLEX em 76,32% para 100 itens e 34,14% para 200 tipos de itens em 10 de 12 instâncias. Curiosamente, o procedimento CPLEX supera o tempo de execução do procedimento LB para a classe com 200 tipos de itens, data de entrega D2 e seis períodos de tempo em todas as três situações de penalização analisadas.

O aumento no valor da penalidade impacta o número de colunas geradas. De $p = 1$ para $p = 10$, o número total de colunas geradas cresce de 2122,96 para 2810,4 ($p = 10$) e, ao contrário do esperado, há poucas mudanças para $p = 100$ (2819,91). Analisando apenas a transição de $p = 1$ para $p = 10$, se considerarmos o mesmo número de objetos e períodos e alterarmos a data de entrega, há um baixo impacto no número de colunas geradas. Ao variar o número de períodos de 3 para 6 para o mesmo número de objetos e a mesma data de entrega, há um aumento no número de colunas se o número de objetos for 50 ou 100. No entanto, para os casos com 200 tipos de itens, há uma redução no número de colunas geradas. O comportamento do modelo ocioso inclui vários fatores que afetam o número de colunas geradas e a qualidade da resposta. Os valores de atraso não mudam com o aumento do valor da penalidade, seja para $p = 10$ (2708.1) ou $p = 100$ (2728.85). A falta de antecipação (já que a antecipação é zero para todas as classes), característica deste modelo, não implica um aumento nos valores totais de atraso. Esta observação é verificada ao considerar que a classe com 200 tipos de itens, data de entrega D1 e 3 períodos consistentemente mostra atraso zero para todos os três p valores avaliados. Da mesma forma, a classe com 200 tipos de itens, data de entrega D1 e 6 períodos também demonstra atraso zero para $p = 10$ e $p = 100$. Os valores de configuração, por outro lado, são afetados pelo aumento do valor de p. Os valores de *setup* diminuem para todas as classes, considerando os valores de referência para $p = 1$. Os valores também diminuem para

$p = 100$, da referência $p = 10$. Os valores totais de *setup* diminuem de 1669,65 ($p = 1$) para 1532,45 ($p = 10$) e 1206,3 ($p = 100$). Em relação ao número de objetos cortados mostrados na coluna "Obj", os valores aumentam em todas as classes com o aumento de p. Esse crescimento é mais intenso na transição de $p = 10$ para $p = 100$. Os valores totais vão de 166530,65 ($p = 1$) para 167044,25 ($p = 10$) e 173571,1 ($p = 100$). No entanto, essas características observadas para $p = 1$ permanecem consistentes para $p = 10$ e $p = 100$, destacando que, ao considerar o mesmo número de itens e o mesmo número de períodos, a quantidade de objetos cortados permanece quase inalterada em resposta a variações na data de entrega.

4.4.4.3 Análise do comportamento do algoritmo

Nesta seção, analisamos o comportamento do algoritmo proposto para resolução do problema, considerando suas três fases distintas. Para recapitular, a Fase 1 gera um solução e conduz a geração de colunas para as fases subsequentes. A Fase 2 envolve uma heurística de arredondamento aplicada às colunas geradas na Fase 1, fornecendo um limite superior. Finalmente, a Fase 3 utiliza a matheurística *Local Branching* para aprimorar a solução inicial obtida na Fase 2. A Fase 1 produz um limite inferior para a solução do problema, permitindo-nos avaliar a evolução da função objetivo nessas fases. As Tabelas 4.15 e 4.16 apresentam os resultados que ilustram essa evolução.

Para esses experimentos, definimos:

$$gap_{21} = 100 \times \frac{\nu(Fase\ 2) - \nu(Fase\ 1)}{\nu(Fase\ 3)}, \tag{4.60}$$

$$gap_{31} = 100 \times \frac{\nu(Fase\ 3) - \nu(Fase\ 1)}{\nu(Fase\ 3)}, \tag{4.61}$$

$$gap_{32} = 100 \times \frac{\nu(Fase\ 3) - \nu(Fase\ 2)}{\nu(Fase\ 3)}, \tag{4.62}$$

$$\tag{4.63}$$

sendo $\nu(Fase\ i)$ o valor da função objetivo no final da Fase i e gap_{ij} é o *gap* entre a Fase i e a Fase j, em relação ao valor da função objetivo no final da Fase 3, ou seja, no final do algoritmo proposto. Observe, das Tabelas 4.15 e 4.16, que $\nu(Fase\ 2) \geq \nu(Fase\ 3) \geq \nu(Fase\ 1)$.

Portanto, gap_{21} representa a diferença relativa entre o limite superior derivado da Fase 2 e o limite inferior produzido pela Fase 1. Para o modelo sem ociosidade, os resultados na Tabela 4.15 demonstram que, para $p = 1$, essa diferença relativa foi consistentemente mantida em menos de 1% em todas as instâncias, independentemente do número de períodos considerados. Para $p = 10$, esse valor variou entre 2-5%, e, para $p = 100$, aumentou para a faixa de 20-30%. A influência dos valores do fator de penalidade na função objetivo (4.2) se torna mais pronunciada, acentuando a diferença dentro dessas faixas. Instâncias com 200 tipos de itens e data de entrega D2 permaneceram como exceções, exibindo valores de gap_{21} de 49,92%, 52,85% e 78,79% para $p = 1$, $p = 10$ e $p = 100$, respectivamente. É interessante notar que, em apenas 8 das 36 instâncias avaliadas, o valor obtido pela Fase 3 (resultado da *Local Branching*) estava mais próximo do limite superior de Fase 2 que do limite inferior de Fase 1. Dessas 8 instâncias, 6 tinham 6 períodos, e 3 delas eram instâncias com

Tabela 4.15: Modelo sem ociosidade: análise do comportamento do algoritmo proposto

Dados da Instância					Função Objetivo			*Gap*(%)		
Items	Data de entrega	Períodos	p	Colunas	Fase 1	Fase 2	Fase 3	gap_{21}	gap_{31}	gap_{32}
50	D1	3	1	2,75	46177,75	46202,75	46196,35	0,05	0,04	-0,01
100	D1	3	1	5,11	89322,80	89405,35	89355,25	0,09	0,04	-0,06
200	D1	3	1	11,26	177228,95	177371,60	177297	0,08	0,04	-0,04
50	D2	3	1	3,13	36963,73	37000,50	36983,75	0,10	0,05	-0,05
100	D2	3	1	8,18	71175,62	71311,55	71239,00	0,19	0,09	-0,10
200	D2	3	1	20,75	139005,45	139237,25	139091,95	0,17	0,06	-0,10
50	D1	3	10	2,90	46371,68	47589,05	46625,60	2,61	0,54	-2,07
100	D1	3	10	5,20	89741,26	92121,70	90193,40	2,64	0,50	-2,14
200	D1	3	10	15,48	178201,08	182797,85	179143,80	2,57	0,53	-2,04
50	D2	3	10	2,98	37118,13	38493,15	37426,35	3,67	0,82	-2,85
100	D2	3	10	8,06	71850,01	74576,05	72484,70	3,76	0,88	-2,89
200	D2	3	10	20,75	141121,68	146648,90	142659,50	3,87	1,08	-2,80
50	D1	3	100	2,90	48240,93	59114,70	50353,00	21,60	4,19	-17,40
100	D1	3	100	4,96	93870,67	117497,30	98556,40	23,97	4,75	-19,22
200	D1	3	100	11,56	188220,60	237656,05	201958,85	24,48	6,80	-17,68
50	D2	3	100	3,18	41820,92	54134,15	44275,30	27,81	5,54	-22,27
100	D2	3	100	8,01	77594,64	102220,15	81162,60	30,34	4,40	-25,94
200	D2	3	100	19,76	167991,34	220658,10	183582,35	28,69	8,49	-20,20
50	D1	6	1	2,56	90537,87	90609,15	90587,05	0,08	0,05	-0,02
100	D1	6	1	5,42	179348,45	179523,50	179442,40	0,10	0,05	-0,05
200	D1	6	1	10,30	352756,35	353261,40	352921,75	0,14	0,05	-0,10
50	D2	6	1	2,93	71088,18	71204,95	71148,80	0,16	0,09	-0,08
100	D2	6	1	7,84	141267,71	141504,60	141385,90	0,17	0,08	-0,08
200	D2	6	1	18,78	141267,71	278892,10	278497,15	49,42	49,27	-0,14
50	D1	6	10	2,45	91089,92	94277,55	91650,30	3,48	0,61	-2,87
100	D1	6	10	5,45	179984,74	185237,70	180743,95	2,91	0,42	-2,49
200	D1	6	10	10,01	354180,36	366246,50	356142,60	3,39	0,55	-2,84
50	D2	6	10	2,79	74246,94	77882,40	74888,45	4,85	0,86	-4,00
100	D2	6	10	7,34	146659,89	152657,05	147779,00	4,06	0,76	-3,30
200	D2	6	10	18,86	146659,89	298689,70	287637,45	52,85	49,01	-3,84
50	D1	6	100	2,51	96707,89	128327,30	103040,05	30,69	6,15	-24,54
100	D1	6	100	5,44	186439,33	234883,95	194081,35	24,96	3,94	-21,02
200	D1	6	100	9,89	371164,72	470332,25	389343,35	25,47	4,67	-20,80
50	D2	6	100	2,78	109320,44	144991,40	116600,40	30,59	6,24	-24,35
100	D2	6	100	7,59	163974,99	218646,90	173941,30	31,43	5,73	-25,70
200	D2	6	100	17,82	163974,99	442315,25	353272,65	78,79	53,58	-25,21

200 itens, como mencionado anteriormente. Nas 28 instâncias restantes, o valor determinado por Fase 3 estava mais próximo do limite inferior gerado por Fase 1. Os valores médios de gap_{21}, gap_{31} e gap_{23} aumentaram com o número de períodos.

No entanto, conforme descrito na Tabela 4.16, a introdução de ociosidade no modelo, conforme descrito na Seção 4.2.2.2, leva a alterações significativas nessas avaliações. Os valores de gap_{21} persistem no intervalo de 1-2,2% para $p = 1$, aumentam para o intervalo de 6,8-11,45% para $p = 10$ e aumenta ainda mais para o intervalo de 17,5-56,5% para $p = 100$. Consequentemente, esse intervalo de variação dos valores de gap_{21} permanece compilado para os dois modelos em consideração, contabilizando os números de dois períodos (3 e 6). No entanto, apenas em 6 das 36 instâncias, o valor obtido na Fase 3 é mais próximo do limite inferior da Fase 1. Nas instâncias 30 restantes, esse valor se alinha mais de perto com o limite superior vindo da Fase 2 do método de resolução. Além disso, os valores médios de gap_{21}, gap_{31} e gap_{32} para o número de períodos igual a 3 superam os valores médios para o número de períodos igual a 6.

Além disso, os valores de gap_{21} para o modelo considerando a ociosidade são

Tabela 4.16: Modelo com ociosidade: análise do comportamento do algoritmo proposto

Dados da Instância					Função Objetivo			$Gap(\%)$		
Items	Data de entrega	Períodos	p	Colunas	Fase 1	Fase 2	Fase 3	gap_{21}	gap_{31}	gap_{32}
50	D1	3	1	65,96	4281,93	4361,25	4344,70	1,83	1,44	-0,38
100	D1	3	1	153,81	8025,37	8173,30	8129,35	1,82	1,28	-0,54
200	D1	3	1	359,00	15722,74	15880,15	15799,35	1,00	0,48	-0,51
50	D2	3	1	66,45	4361,76	4441,50	4425,10	1,80	1,43	-0,37
100	D2	3	1	151,45	8138,37	8312,70	8265,20	2,11	1,53	-0,57
200	D2	3	1	361,24	15858,61	16206,05	16119,35	2,16	1,62	-0,54
50	D1	3	10	66,13	4304,33	4729,15	4620,30	9,19	6,84	-2,36
100	D1	3	10	148,98	8062,15	8789,20	8580,90	8,47	6,05	-2,43
200	D1	3	10	346,69	15586,58	17453,55	16950,65	11,01	8,05	-2,97
50	D2	3	10	65,99	4384,2	4805,9	4705,05	8,96	6,82	-2,14
100	D2	3	10	148,46	8514,16	9329,85	9129,35	8,93	6,74	-2,20
200	D2	3	10	353,31	17309,85	19476,85	18931,20	11,45	8,56	-2,88
50	D1	3	100	65,70	4606,69	7694,5	6619,45	46,65	30,41	-16,24
100	D1	3	100	150,20	8451,28	14829,85	12190,70	52,32	30,67	-21,65
200	D1	3	100	347,01	16504,46	30691,35	25095,20	56,53	34,23	-22,30
50	D2	3	100	66,18	7879,59	12049,70	9886,70	42,18	20,30	-21,88
100	D2	3	100	153,80	8903,29	16042,00	12852,80	55,54	30,73	-24,81
200	D2	3	100	356,18	37178,51	61286,55	49934,50	48,28	25,55	-22,73
50	D1	6	1	67,32	8189,9	8325,05	8281,85	1,63	1,11	-0,52
100	D1	6	1	135,56	16066,75	16293,40	16203,20	1,40	0,84	-0,56
200	D1	6	1	271,83	31369,35	31820,45	31427,00	1,44	0,18	-1,25
50	D2	6	1	65,92	8527,31	8662,60	8619,85	1,57	1,07	-0,50
100	D2	6	1	135,97	16682,26	16964,55	16875,30	1,67	1,14	-0,53
200	D2	6	1	288,45	32051,35	32581,85	32423,75	1,64	1,15	-0,49
50	D1	6	10	62,68	8358,40	9202,40	8968,80	9,41	6,81	-2,60
100	D1	6	10	136,95	16121,23	17328,55	16666,85	7,24	3,27	-3,97
200	D1	6	10	295,23	31336,51	33994,8	32491,45	8,18	3,55	-4,63
50	D2	6	10	65,20	11395,06	12253,30	12025,55	7,14	5,24	-1,89
100	D2	6	10	131,24	21625,30	23181,80	22776,95	6,83	5,06	-1,78
200	D2	6	10	309,61	37934,01	41272,30	40397,05	8,26	6,10	-2,17
50	D1	6	100	63,52	10205,71	18854,95	16152,00	53,55	36,81	-16,73
100	D1	6	100	128,07	16789,54	26211,65	22239,15	42,37	24,50	-17,86
200	D1	6	100	289,23	32931,62	56162,6	42956,25	54,08	23,34	-30,74
50	D2	6	100	62,97	43608,98	52334,75	49695,3	17,56	12,25	-5,31
100	D2	6	100	127,10	34523,37	46130,55	41988,15	27,64	17,78	-9,87
200	D2	6	100	287,24	72895,74	99951,35	91055	29,71	19,94	-9,77

consistentemente maiores do que aqueles obtidos para o modelo sem ociosidade em 33 das 36 instâncias avaliadas. O mesmo comportamento é observado para gap_{31}. Em essência, a amplitude das janelas delimitadoras se expande quando a ociosidade é considerada, causando uma diferença maior entre o limite inferior e o valor fornecido pela Fase 3. As 3 instâncias que se desviam desse padrão são aquelas com 200 itens, 6 períodos e dados de entrega D2. O comportamento atípico encontrado nesses 3 casos para o modelo sem ociosidade não é replicado para o modelo considerando ociosidade. Por outro lado, estabelecer um padrão definitivo para os valores de gap_{32} ao comparar os modelos não é viável.

As observações sobre os valores de gap_{21} são correlacionadas com o número de colunas geradas em cada um dos modelos. No modelo sem ociosidade, o número de colunas geradas permanece no interior do intervalo $[2{,}45; 20{,}75]$, enquanto, para o modelo com tempo ocioso, esse valor se expande significativamente para o intervalo $[62{,}68; 361{,}24]$. Em ambos os modelos, o número de colunas geradas aumenta proporcionalmente com o número de tipos de itens na instância. Essa diferença no número

de colunas geradas entre os dois modelos influencia a largura da janela delimitadora, conforme explicado anteriormente.

4.4.4.4 Uma discussão sobre as formulações com e sem ociosidade

Nesta seção, analisamos os resultados computacionais para traçar comparações que esclarecem as implicações das duas formulações do problema. Discutimos a função objetivo, o tempo de execução, os termos constituintes da função objetivo e a técnica de geração de colunas. A formulação sem ociosidade produz valores de função objetivo aproximadamente 9,80 vezes maiores que aqueles obtidos com ociosidade em $p = 1$; 8,70 vezes em $p = 10$; e 5,23 vezes em $p = 100$. Curiosamente, o número de objetos cortados exibe uma relação de ordem de magnitude semelhante entre as formulações, mantendo uma faixa consistente de valores apesar do aumento de p (8,62 para $p = 1$; 9,63 para $p = 10$; e 9,58 para $p = 100$). Esse padrão também é apresentado para valores de atraso e configuração, embora em escalas diferentes.

No entanto, as disparidades mais significativas são observadas no número de colunas geradas e no tempo de execução. A formulação com ociosidade produz um número maior de colunas geradas em comparação à formulação sem ociosidade. Para $p = 1$, o número de colunas geradas na formulação com ociosidade excede o da formulação sem ociosidade em 2122,96. Essa disparidade aumenta para 2810,40 para $p = 10$ e ainda para 2819,91 para $p = 100$.

No entanto, a formulação com ociosidade representa um desafio computacional maior para encontrar uma solução em comparação à formulação sem ociosidade, particularmente pronunciado para $p = 1$. Neste cenário, o tempo de execução para a formulação com ociosidade supera o da formulação sem ociosidade para o procedimento LB e o procedimento CPLEX em todas as 12 classes avaliadas. A taxa de tempo de execução atinge o limite máximo em 17,16 vezes para o procedimento LB e 14,81 vezes para o procedimento CPLEX.

Para $p = 10$ e $p = 100$, esse nível aumentado de desafio não é tão proeminente. Para $p = 10$ e considerando o procedimento CPLEX, em seis das 12 classes, a formulação sem ociosidade exibe um tempo de execução maior que a formulação com ociosidade. Essas seis classes incluem todas aquelas com 200 itens. No entanto, o tempo de execução total para a formulação com ociosidade é menor. Para $p = 100$ e também considerando o procedimento CPLEX, a mesma tendência ocorre em sete das 12 classes, mais uma vez abrangendo todas as classes com dimensões maiores. Entretanto, ao considerar o procedimento LB, a formulação com ociosidade continua sendo mais desafiadora, com o tempo de execução total sendo 4,91 vezes maior para $p = 10$ e 3,12 vezes para $p = 100$. Em todas as classes avaliadas, para $p = 10$ ou $p = 100$, o tempo de execução da formulação com ociosidade é menor do que o da formulação sem ociosidade. Assim, esta análise leva à conclusão de que o procedimento LB se adapta melhor à solução destas formulações que o procedimento CPLEX

4.4.5 Resultados para diferentes valores de custo

Um conjunto final de experimentos foi conduzido para avaliar diferentes parâmetros de custo, focando exclusivamente em instâncias com 3 períodos, 50 tipos de itens e datas de entrega D1 e D2. Os valores de custo usados são detalhados na Tabela 4.17.

Tabela 4.17: Conjunto de parâmetros de custo, em que os valores do tipo 1 foram usados nos experimentos anteriores e os valores do tipo 2 são baseados em Cui et al. (2015).

Parâmetros	Valores		Significado
	Grupo 1	Grupo 2	
c_j^t	$L = 1$	$L = 1000$	Custo de corte de um objeto
f_j^t	$[1,p]$, $p = 1; 10; 100$	$0{,}1L = 100$	Custo de *setup*
β^t	$[1,p]$, $p = 1; 10; 100$	$0{,}1L = 100$	Custo de Adiantamento
α^t	$[1,p]$, $p = 1; 10; 100$	$0{,}1L = 100$	Custo de Atraso

Nesta tabela, os valores de parâmetro para o Grupo 1 refletem aqueles utilizados nos experimentos da seção anterior, enquanto os valores de parâmetro para o Grupo 2 são baseados nos custos de corte e *setup* mostrados em Cui et al. (2015). Esses parâmetros foram selecionados porque Cui et al. (2015) apresenta valores para comprimento, quantidade e demanda do item que se alinham estreitamente com aqueles neste estudo. No entanto, como o problema abordado no Cui et al. (2015) não considera atraso ou antecipação, os custos associados ao atraso e antecipação no Grupo 2 foram definidos como iguais ao custo de configuração. Assim, a principal distinção entre os dois grupos de custos é que os custos de corte no Grupo 1 são menores ou iguais aos custos de configuração, atraso e antecipação, enquanto no Grupo 2, eles excedem esses custos.

A Tabela 4.18 mostra o tempo médio total de atraso e antecipação e o número médio de *setups* e objetos de corte na solução obtida pelo LB (Fase 3 do algoritmo proposto). Adicionalmente, apresenta o valor da função objetivo da solução obtida pelo LB e CPLEX, bem como os tempos de execução para cada uma delas.

Tabela 4.18: Valores médios para 20 instâncias para cada uma das classes com 50 tipos de itens, considerando os dois tipos de datas de entrega (D1 e D2), 3 períodos para os modelos sem ociosidade (SO) e com ociosidade (CO).

Dados da Instância			Termos da Função Objetivo para LB				LB		CPLEX	
Modelo	Data de entrega	(α, β, f, c)	Atraso	Adiantamento	*Setup*	Objetos	FO	Tempo(s)	FO	Tempo(s)
SO	D1	(1,1,1,1)	16,70	14698,85	109,25	31371,55	46196,35	6,89	46196,35	60,02
		$(\alpha, \beta, f) \in [1,10]$; $c = 1$	22,60	3502,25	74,9	42586,60	46625,60	6,03	46625,50	949,33
		$(\alpha, \beta, f) \in [1,100]$; $c = 1$	16,45	0,00	83,50	46088,95	50353,00	9,92	50353,00	1075,70
		(100,100,100,1000)	16,55	726663,90	231,80	4275,20	$1{,}23 \times 10^7$	1,09	$1{,}23 \times 10^7$	193,64
SO	D2	(1,1,1,1)	97,70	11294,05	114,00	25478,00	36983,75	2,14	36983,75	55,67
		$(\alpha, \beta, f) \in [1,10]$; $c = 1$	103,95	2859,40	75,05	33935,60	37426,35	44,02	37426,15	951,44
		$(\alpha, \beta, f) \in [1,100]$; $c = 1$	98,05	0,10	81,10	36797,25	44275,30	81,48	44275,30	1074,49
		(100,100,100,1000)	95,50	60791,30	347,85	4274,95	$1{,}04 \times 10^7$	2,23	$4{,}41 \times 10^8$	194,08
CO	D1	(1,1,1,1)	16,35	0,00	53,40	4274,95	4344,70	138,44	4344,70	952,07
		$(\alpha, \beta, f) \in [1,10]$; $c = 1$	21,25	0,00	48,65	4290,8	4620,30	437,88	4620,50	1149,17
		$(\alpha, \beta, f) \in [1,100]$; $c = 1$	16,45	0,00	41,50	4449,05	6619,45	144,21	6619,65	1168,20
		(100,100,100,1000)	15,85	0,00	52,20	4274,85	42765855	697,38	42765855,00	1106,90
CO	D2	(1,1,1,1)	96,2	0,00	53,95	4274,95	4425,10	244,43	4425,10	892,00
		$(\alpha, \beta, f) \in [1,10]$; $c = 1$	100,45	0,00	48,80	4293,30	4705,05	478,43	4705,30	1143,88
		$(\alpha, \beta, f) \in [1,100]$; $c = 1$	95,85	0,00	41,60	4462,50	9886,7	204,76	9888,10	1160,31
		(100,100,100,1000)	95,60	0,00	53,20	4274,75	$4{,}29 \times 10^6$	633,96	$6{,}02 \times 10^8$	1041,50

De acordo com essa tabela, o tempo médio de atraso, o número de configurações e o número de objetos cortados são muito semelhantes entre os dois grupos de instâncias. No entanto, o modelo sem ociosidade resultou em maior antecipação para as instâncias no Grupo 2 em comparação com aquelas no Grupo 1. Em termos de

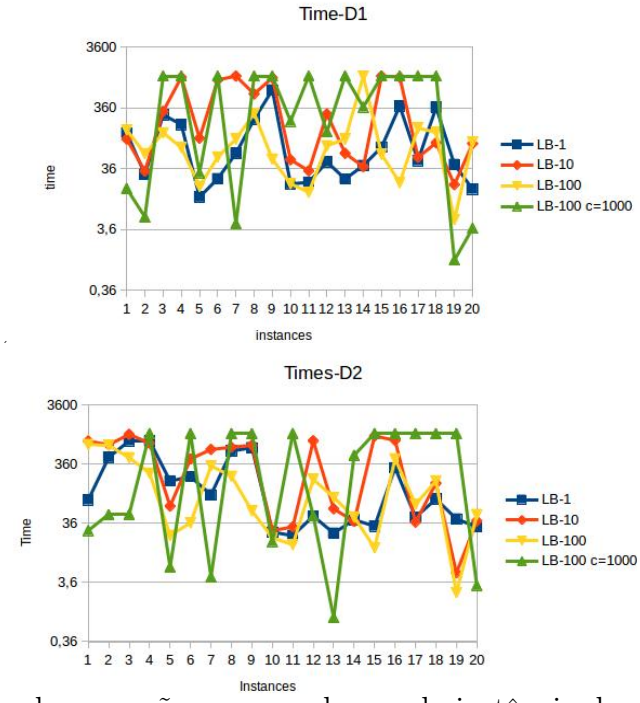


Figura 4.2: Tempos de execução para resolver cada instância do problema com ociosidade para as duas datas de entrega, considerando os diferentes tipos de custos.

desempenho, o LB produziu consistentemente soluções com qualidade média comparável à do solverCPLEX para ambos os grupos de instâncias. Notavelmente, para instâncias do Grupo 2, o LB obteve soluções significativamente melhores para datas de entrega D2. Além disso, os tempos de execução do LB foram substancialmente mais curtos do que os do solverCPLEX, particularmente para instâncias do Grupo 2.

As Figuras 4.2 e 4.3 ilustram os tempos de execução para resolver cada uma das 20 instâncias dos modelos com e sem ociosidade, respectivamente, em ambas as datas de entrega e considerando vários tipos de parâmetros de custo. Cada cor nos gráficos corresponde a resultados derivados de diferentes conjuntos de parâmetros. Nessas figuras, LB-1, LB-10 e LB-100 denotam os resultados para instâncias do Grupo 1 com $p = 1$, 10 e 100, respectivamente, enquanto $LB - 100\ c = 1000$ se refere aos resultados para instâncias do Grupo 2.

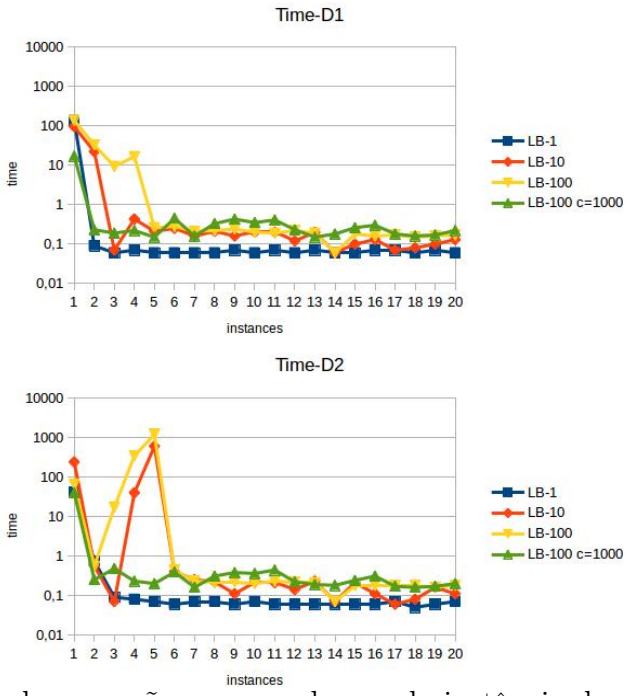


Figura 4.3: Tempos de execução para resolver cada instância do problema sem ociosidade para as duas datas de entrega, considerando os diferentes tipos de custos.

Esses números revelam que, para o problema com ociosidade, muitas instâncias atingem o limite de tempo de 1200 segundos, enquanto outras são resolvidas relativamente rápido. Em contraste, os tempos de execução para o problema sem ociosidade são geralmente menores, e apenas algumas instâncias atingiram o limite de tempo. Além disso, há uma variação mínima nos tempos de execução para as instâncias analisadas.

As Figuras 4.4 e 4.5 exibem o número de *setups* para cada uma das 20 instâncias dos modelos com e sem ociosidade, respectivamente, em ambas as datas de entrega e vários tipos de parâmetros de custo. De acordo com a Figura 4.4, o número de *setups* no modelo com ociosidade é bastante semelhante em diferentes valores de parâmetros de custo. No entanto, a Figura 4.5 revela que, no modelo sem ociosidade, as instâncias do Grupo 2 exibem a maior variação no número de *setups*, bem como uma contagem geral maior de *setups* em comparação aos outros grupos.

As Figuras 4.6 e 4.7 ilustram o número de objetos cortados para cada uma das 20 instâncias dos modelos com e sem ociosidade, respectivamente, em ambas as datas de entrega e diferentes tipos de parâmetros de custo. De acordo com a Figura 4.6, no modelo com ociosidade, o menor número de objetos cortados corresponde aos maiores custos de corte, especificamente aqueles no Grupo 2. Além disso, há maior variação no número de objetos cortados entre esses valores. Na Figura 4.7, que representa o modelo sem ociosidade, o número de objetos cortados para ambas as datas de entrega aumenta de maneira semelhante. No entanto, para o conjunto de parâmetros com os maiores custos de corte, o número permanece consistentemente menor e mais estável em comparação aos outros parâmetros.

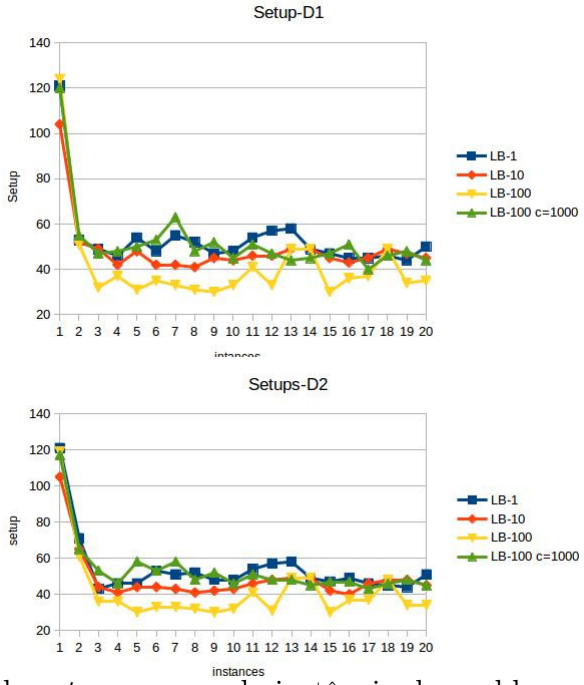


Figura 4.4: Número de *setups* para cada instância do problema com ociosidade para os diferentes tipos de custos.

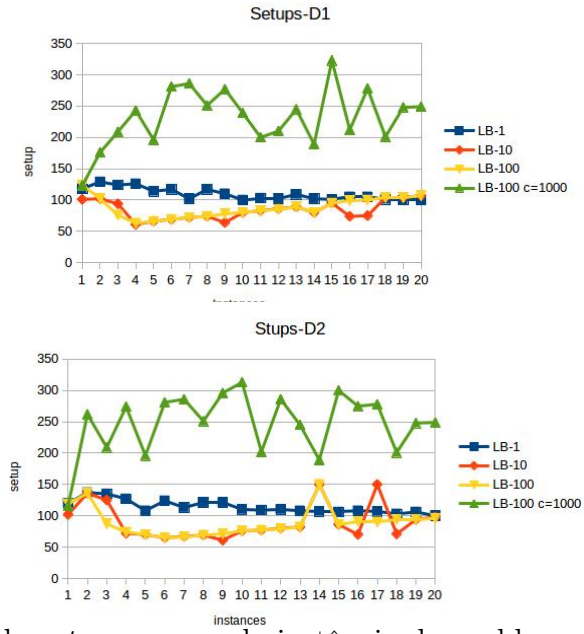


Figura 4.5: Número de *setups* para cada instância do problema sem ociosidade para os diferentes tipos de custos.

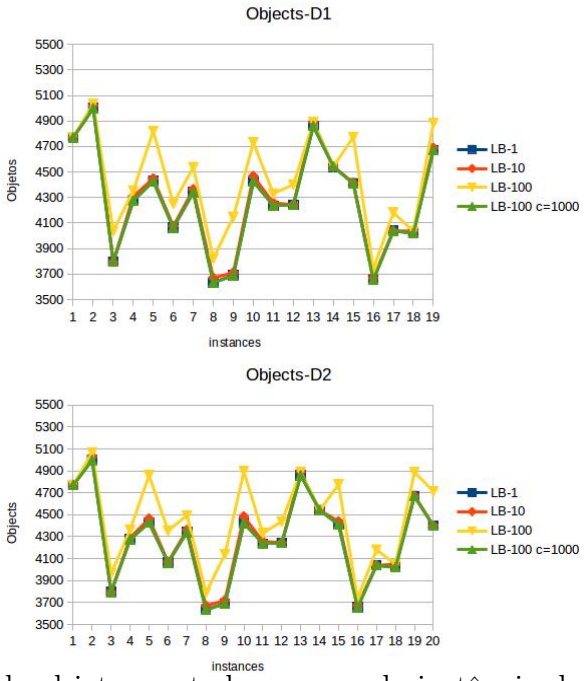


Figura 4.6: Número de objetos cortados para cada instância do problema com ociosidade para os diferentes tipos de custos.

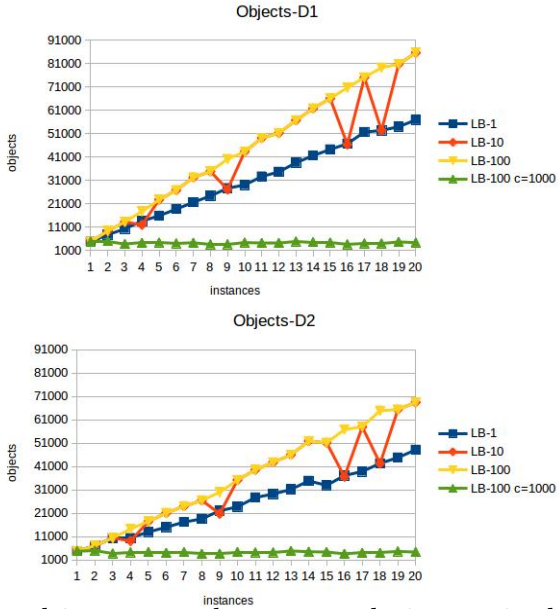


Figura 4.7: Número de objetos cortados para cada instância do problema sem ociosidade para os diferentes tipos de custos.

4.5 Conclusões e discussões parciais

Este capítulo abordou o Problema de Corte de Estoque Unidimensional Multiperíodo com Datas de Entrega e *Setup* Independente (1D-PMCE-DE-SI), que incorpora tempos de *setup* para alternar padrões de corte e considera penalidades para atraso e antecipação. Duas versões do problema são exploradas: a primeira versão não permite tempos ociosos na máquina de corte, enquanto a segunda versão permite períodos ociosos nas operações da máquina de corte. Apresentamos uma abordagem de modelagem matemática para ambas as versões do problema e uma metodologia de resolução de três fases. O método proposto envolve três fases: a Fase 1 se concentra na geração de colunas para resolver o problema relaxado; a Fase 2 gera uma solução parcial inteira viável e a Fase 3 aplica a matheurística *Local Branching* para refinar a solução.

Na Fase 1 do método de solução, calculamos os custos mínimos reduzidos. Esta fase apresenta dois cenários potenciais. O primeiro cenário surge quando o custo e o tempo de produção vinculados a um padrão de corte são independentes, enquanto o segundo cenário ocorre quando, pelo menos, um parâmetro - custo ou tempo de produção associado a um padrão de corte - é interdependente. Para esta tese o cenário escolhido para focar foi o Cenário 1, no qual os custos permanecem independentes dos padrões de corte. A Fase 2 gera um limite superior na melhor solução identificada pela Fase 3, e esse limite superior deve ser definido para permitir que a Fase 3 melhore o valor fornecido. Este cenário foi observado com a heurística *Round-Up*, que permitiu que a Fase 3 superasse os resultados obtidos pela heurística *Relax-and-Fix*. Passando para a Fase 3, a matheurística *Local Branching* implementada necessita definir dois parâmetros: k e h_{max}. Esses parâmetros, vitais para resolver esta proposta, foram ajustados usando o pacote IRACE.

Para conduzir os testes computacionais, criamos uma série de instâncias ajustando parâmetros específicos. Utilizamos os conceitos descritos no gerador proposto por Gau e Wäscher (1995) e implementamos um gerador usando a linguagem C++. Dentro desta coleção de testes, definimos o comprimento do objeto a ser cortado, o número de itens para corte e os períodos de tempo para valores específicos. Também selecionamos valores variados para os parâmetros restantes dentro de um intervalo definido. Além disso, parâmetros adicionais foram essenciais para a análise do teste computacional, e a seção de resultados mostra tabelas apresentando cada um desses parâmetros específicos.

Também apresentamos um conjunto desafiador de instâncias para o 1D-PMCE-DE-SI avaliar nossa abordagem proposta, representando dois cenários distintos de data de entrega. Optamos por nos envolver com dois cenários de data de entrega, que foram gerados usando uma distribuição uniforme. Esses cenários abrangem instâncias com datas amplas (D1) e datas mais estreitas (D2). Para criar esses cenários de data, usamos um valor aproximado como parâmetro, refletindo o número de objetos necessários para cortar todos os itens demandados dentro de um determinado período de tempo. As instâncias provaram ser exigentes, com o solver frequentemente incapaz de encontrar soluções ótimas comprovadas dentro do limite de tempo especificado. No entanto, nossa abordagem proposta consistentemente alcança soluções com valores objetivos médios comparáveis ou aprimorados em comparação ao procedimento CPLEX. Além disso, a abordagem proposta geralmente exibe tempos de

execução mais curtos que o solucionador. Notavelmente, para grupos específicos de instâncias, os tempos de execução heurísticos foram significativamente mais eficientes do que os do solucionador.

Ao analisar os resultados, nosso foco foi ajustar as penalidades associadas aos termos da função objetivo, mantendo um custo constante para cortar um objeto, que foi definido como um valor de 1. Os resultados também destacam o impacto das penalizações nas soluções e a eficiência dos procedimentos de resolução. Especificamente, quando as penalizações por atraso, antecipação e configurações são consistentes, o valor de *gap* entre a solução heurística e o limite inferior do procedimento de geração de colunas permanece pequeno. No entanto, quando as penalidades diferem, esse valor de *gap* tende a aumentar, enfatizando a influência dos valores de penalidade na qualidade geral da solução.

Comparando as versões com e sem tempo ocioso, os resultados demonstram claramente que as instâncias na versão sem tempo ocioso são menos desafiadoras de resolver que aquelas na versão que permite a ociosidade da máquina. A inclusão do tempo ocioso aumenta significativamente a complexidade e as demandas computacionais do problema, apresentando um obstáculo maior para a resolução efetiva. Os resultados obtidos da formulação envolvendo ociosidade revelam que, em todo o conjunto de instâncias de teste, o método de solução gera consistentemente resultados em que tanto o atraso quanto o avanço permanecem zero. Nesses casos específicos, a data das instâncias foi denotada como D1, representando a data de entrega mais branda ou relaxada.

Uma questão futura significativa a ser abordada como trabalho futuro envolve a suposição de que a troca de padrões de corte é dependente da sequência, tanto em termos de tempo de *setup* quanto do próprio custo de *setup* associado. Além disso, as questões associadas aos custos de estoque também devem ser abordadas. Esses pontos são trabalhados no Capítulo 5 dessa tese.

Capítulo 5

Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Dependente da Sequência

Este capítulo apresenta o Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *D*ependente da Sequência (1D-PMCE-DE-SD). A Seção 5.1 descreve o 1D-PMCE-DE-SI, incluindo justificativas para sua proposição. A Seção 5.2 apresenta uma revisão de literatura relativa ao problema tratado. Em seguida, a Seção 5.3 propõe uma formulação matemática que busca representar as restrições e objetivos envolvidos, definindo as variáveis de decisão, as funções objetivo e as restrições do modelo. Já a Seção 5.4 detalha o método de solução adotado. A Seção 5.5 apresenta como foi realizada a geração de instâncias para o 1D-PMCE-DE-SD. Logo após, a Seção 5.6 descreve os experimentos computacionais realizados para avaliar o desempenho da abordagem proposta, discutindo os parâmetros utilizados, os cenários testados, bem como a análise dos resultados obtidos. Por fim, a Seção 5.7 sintetiza as principais conclusões extraídas a partir dos testes realizados.

5.1 Descrição do Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entrega e *Setup* Dependente da Sequência

Até o momento trabalhamos com o problema de corte de estoque com datas de entrega e *setup* independente da sequência. Porém, na prática, a máquina de corte pode requer um tempo para trocar a sua configuração, resultando em tempos e custos adicionais que dependem da ordem na qual os cortes são realizados. Padrões de corte mais similares podem levar um tempo de troca menor que aqueles que são mais diferentes. Dessa maneira, como o tempo da troca de padrões (ou seja, o custo, em termos de tempo, associado à troca) depende do corte do último padrão de corte na máquina. Isso introduz um nível significativo de complexidade, pois o tempo de *setup* e a eficiência da operação são influenciados diretamente pela ordem de ocorrência das

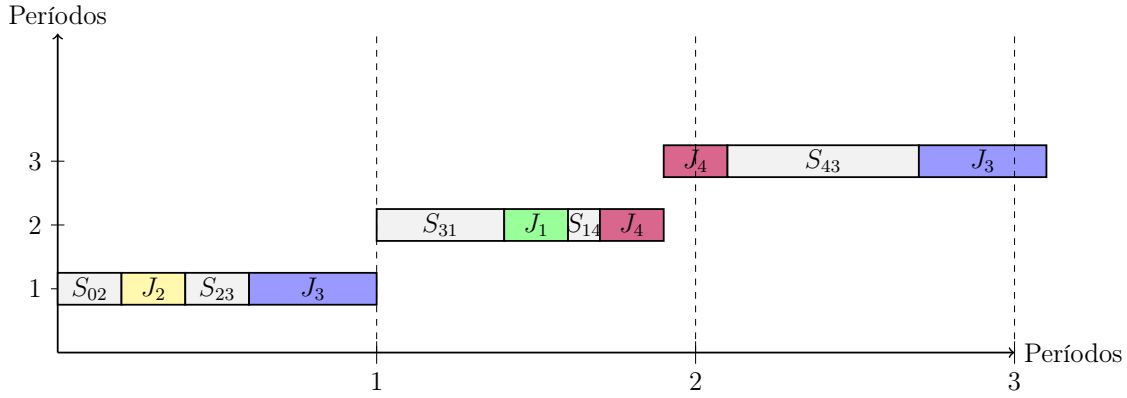


Figura 5.1: Exemplo de um planejamento com a divisão dos períodos de corte e um conjunto de planos de corte sequenciados. Fonte: Própria da Autora.

operações de corte, criando uma dependência sequencial que torna a modelagem e a solução do problema desafiadoras.

A Figura 5.1 apresenta o exemplo de uma programação com a divisão dos períodos de corte e um conjunto de planos de corte sequenciados, apresentando os tempos de *setups* da máquina dependentes do último corte do padrão.

A partir da Figura 5.1, sabendo a demanda inicialmente de cada pedido, observamos que:

(i) No primeiro período há um tempo inicial para configurar a máquina para o corte do primeiro padrão de corte que será cortado (J_2). Depois desse corte, corta-se o padrão de corte J_3, com custo de preparação S_{23}. Cada um dos padrões de corte leva um certo tempo para ser cortado. Ao final desse período, o tempo de entrega coincidiu com a data de entrega dele, não havendo atraso e nem adiantamento na entrega, sendo a data de conclusão igual ao final do período, ou seja, $d_1 = 1$, $d_2 = 2$ e $d_3 = 2$.

(ii) No segundo período, há um padrão de corte na máquina de corte (J_3). Para cortar o padrão de corte J_1, há um custo de *setup* da máquina S_{31}, pois os padrões de corte são diferentes. Com o corte do padrão de corte J_4 antes do final do segundo período, houve um adiantamento na entrega dos pedidos;

(iii) No terceiro período, o padrão de corte J_4 já iniciou na máquina de corte e ele será cortado nesse período. Como não ocorreu a troca dos padrões de corte, não há necessidade de haver um custo de *setup* da máquina para esse primeiro corte nesse período. Como o final do corte do último padrão de corte ocorreu após o final do terceiro período, há um atraso na entrega desse pedido.

As conclusões extraídas da Figura 5.1 ressaltam pontos críticos e indicam caminhos para estudos futuros, que podem ser resumidos nos seguintes aspectos. No primeiro período, a entrega foi exatamente na data prevista, evidenciando que uma boa sequência pode eliminar penalidades por atraso ou adiantamento. No segundo período, a necessidade de um novo *setup* devido à mudança de padrão de corte gera um custo extra, mas também possibilita um adiantamento na entrega. Essa situação

destaca a importância de se ponderar os custos adicionais de *setup* contra os benefícios ou prejuízos logísticos e financeiros de entregas antecipadas. No terceiro período, a continuidade do mesmo padrão de corte elimina custos de *setup*, mas o corte finalizado após o período acarreta atraso, o que pode implicar penalidades ou custos adicionais. Essa discrepância reforça a relevância de alinhar o tempo de produção com as datas de entrega planejadas.

Dada a natureza combinatória do 1D-PMCE-DE-SD, resolver este problema de forma eficiente para instâncias de grande porte pode ser impraticável por métodos exatos de Programação Inteira Mista (MIP). Embora esses métodos forneçam garantias de otimalidade, o tempo computacional para encontrar uma solução ótima pode ser proibitivo.

Com base nos resultados apresentados no Capítulo 4, decidimos permitir a ociosidade das máquinas. Essa escolha foi feita com o objetivo de reduzir o tempo de *setup* e melhorar a eficiência global do sistema, pois há uma escolha entre adiantar itens e penalizar na função objetivo ou deixar por um tempo a máquina de corte parada. Embora a ociosidade não seja ideal em termos de utilização plena das máquinas, ela foi considerada uma estratégia vantajosa em cenários onde a flexibilidade na alocação de recursos leva a um ganho maior no cumprimento dos prazos e na minimização de custos associados aos processos de *setup*. A decisão de permitir essa ociosidade, portanto, se alinha aos resultados do capítulo anterior, onde se demonstrou que, em muitos casos, a flexibilidade em termos de recursos pode levar a soluções mais eficientes.

5.2 Revisão da literatura para o 1D-PMCE-DE-SD

Nesta seção, serão apresentados e analisados dois artigos fundamentais que abordam os desafios e avanços no tema em questão. O primeiro artigo explora uma abordagem integrada, que une o dimensionamento de lotes, a programação da produção e o corte de estoque, destacando as particularidades dos tempos e custos de *setup* dependentes da sequência de operações. Já o segundo trabalho oferece uma perspectiva complementar, enfatizando metodologias inovadoras e estratégias de decomposição para a resolução de problemas complexos em ambientes de produção. A análise desses estudos permite uma compreensão aprofundada das técnicas aplicadas, bem como dos benefícios e limitações identificados na literatura atual.

Em Garraffa et al. (2016) os autores apresentam uma nova generalização do problema de estoque de corte unidimensional (1D-PCE) que considera perdas de corte dependentes da sequência de corte dos itens. Além disso, é apresentada uma heurística baseada em padrões (denominada por HSD) que considera especificamente perdas de corte dependentes da sequência (SDCL). Por fim, dois estudos de caso ilustram a relevância dessa nova generalização.

No artigo de Yanasse e Lamosa (2007), um problema integrado formulado como um problema de programação linear inteira é apresentado para encontrar uma solução ótima para o problema de corte de estoque unidimensional sob restrições específicas de sequenciamento de padrões. A solução utiliza uma abordagem lagrangiana. O problema dual é resolvido utilizando um método de subgradiente modificado. Uma heurística para o problema integrado também é apresentada. Os resultados compu-

tacionais obtidos a partir de um conjunto de instâncias unidimensionais que utilizam esses procedimentos são relatados.

Wuttke e Heese (2018), motivados pelo contexto têxtil, propõem um modelo matemático inovador que integra o problema de corte de estoque à programação, considerando tempos de *setup* dependentes da sequência de operações. Nesse modelo, as localizações das facas são fundamentais, pois cada inserção ou remoção requer um tempo específico de processamento, o que adiciona uma camada de complexidade à otimização do sistema. Utilizando a formulação matemática, os pesquisadores conseguiram resolver pequenas instâncias do problema até a otimalidade, demonstrando a eficácia da abordagem teórica. Para instâncias de tamanho real, nos quais a complexidade computacional aumenta significativamente, foi desenvolvida uma heurística sequencial, que oferece soluções de alta qualidade em um tempo razoável, destacando o potencial da metodologia para aplicações práticas no setor têxtil.

Wang et al. (2020) neste artigo, abordam uma nova variante do problema de corte de estoque, na qual os custos de instalação são considerados. O objetivo do estudo é minimizar o custo do material e o número de configurações. Os experimentos computacionais mostram que esses algoritmos são eficientes e de alta qualidade.

No estudo realizado por Melega et al. (2020), os autores desenvolvem um modelo matemático e propõem estratégias de solução para um problema integrado que abrange o dimensionamento de lotes, a programação da produção e o corte de estoque, estruturado em dois estágios. Essa formulação contempla ainda tempos e custos de *setup* que variam em função da sequência de operações realizadas.

No primeiro estágio do processo produtivo, utiliza-se uma máquina de corte especializada para subdividir objetos de grandes dimensões em porções menores. Nesta fase, são gerados padrões de corte que orientam a divisão das peças, os quais precisam ser devidamente sequenciados para compor um plano de corte global e consistente com os requisitos do problema. As peças resultantes dessa etapa são, subsequentemente, direcionadas ao segundo estágio, onde se procede à montagem dos produtos finais. Nesta última fase, a programação da produção é realizada de forma a alinhar a oferta com as demandas dos clientes.

Para resolver o desafio inerente ao problema integrado de dois estágios, os autores implementam métodos de solução baseados em uma abordagem combinada de "*Pricing and Branching*". Nesta metodologia, um procedimento de geração de colunas é proposto para introduzir novas variáveis ou padrões que possam enriquecer o conjunto de soluções possíveis, enquanto o problema global é decomposto em subproblemas, tornando-o mais tratável do ponto de vista computacional.

Adicionalmente, o artigo apresenta um estudo computacional fundamentado em dados gerados aleatoriamente, o qual permite analisar minuciosamente o impacto das diferentes abordagens de solução sobre o desempenho do problema integrado. Por fim, os resultados obtidos por meio da abordagem integrada são comparados com uma simulação empírica que reflete as práticas convencionais do setor, evidenciando, assim, os ganhos de desempenho e a eficácia da metodologia proposta.

Na tese Silva et al. (2023) o problema de corte de estoque unidimensional multiperíodo com minimização dos custos de preparação nos padrões de corte é estudado. Primeiro propuseram um extenso conjunto de formulações pseudo-polinomiais baseadas em padrões de corte, principalmente adaptando formulações conhecidas para problemas de corte de estoque da literatura.

Tabela 5.1: Uma Visão Geral Comparativa das Características do 1D-PMCE-DE-SD

Tipo	Artigo de Referência	Características abordadas					
		Dimensionalidade	Objetos cortados	*Setup*	Data de entrega(multiperíodo)	Atraso	Adiantamento
1	Yanasse e Lamosa (2007)	1D	Sim	Sim	–	–	–
1	Garraffa et al. (2016)	1D	Sim	Sim	–	–	–
1	Wuttke e Heese (2018)	2D	Sim	Sim	–	–	–
1	Wang et al. (2020)	2D	Sim	Sim	–	–	–
2	Melega et al. (2022)	1D	Sim	Sim	Sim	–	–
2	Silva et al. (2023)	1D	Sim	Sim	Sim	–	–
2	Silva et al. (2025)	1D	Sim	Sim	Sim	–	–
3	**Esta Tese**	1D	Sim	Sim	Sim	Sim	Sim

No artigo de Silva et al. (2025) os autores apresentam um procedimento Otimizador de Chave Aleatória para resolver o Problema Multiperíodo de Corte de Estoque Unidimensional (1D-PMCE) com custos de configuração e restrições de capacidade. Uma análise de sensibilidade também é conduzida, examinando o impacto dos custos de instalação e da capacidade de produção.

A Tabela 5.1 apresenta uma visão geral comparativa entre a formulação proposta nesta tese e os principais trabalhos da literatura que abordam o problema de corte de estoque com *setup* dependente da sequência. Observa-se que, embora vários artigos tratem de aspectos relevantes como a dimensionalidade do problema, a minimização do número de objetos cortados e a consideração de *setups*, poucos incorporam simultaneamente múltiplas características operacionais frequentemente encontradas em ambientes industriais reais. Entre essas características, destacam-se o tratamento explícito de múltiplos períodos com datas de entrega, bem como a consideração de penalidades associadas a atrasos e adiantamentos. A formulação 1D-PMCE-DE-SD apresentada nesta tese se diferencia ao integrar todas essas dimensões de forma unificada, o que amplia suas possibilidades de aplicações práticas e contribui significativamente para o avanço da modelagem integrada de problemas de corte e programação da produção.

5.3 Modelagem matemática do 1D-PMCE-DE-SD

A modelagem do Problema Multiperíodo de Corte de Estoque Unidimensional com Datas de Entregas e *Setup* Dependente da Sequência (1D-PMCE-DE-SD) é uma abordagem fundamental na otimização de processos industriais e logísticos. Para abordar esse problema, consideramos a minimização do somatório dos seguintes custos:

(i) Objetos cortados;

(ii) Estoque de itens;

(iii) Antecipação ou Atraso na entrega dos pedidos;

(iv) Tempo de *setup* dependente da sequência.

A gestão eficiente desses fatores permite atender a demanda dentro do prazo, reduzindo atrasos e antecipações, enquanto otimiza a transição entre padrões de corte, minimizando custos e desperdícios. Dessa forma, a solução desse problema exige o desenvolvimento de modelos matemáticos e métodos de otimização que equilibrem a produtividade, a economia de materiais e a pontualidade nas entregas, contribuindo para a competitividade das empresas no mercado.

A modelagem proposta nesta tese considera os seguintes conjuntos, parâmetros e variáveis de decisão para o modelo.

Indices:

- i: representa os tipos de itens a serem cortados, para $i = 1, ..., n$;
- t: refere-se aos períodos de tempo ao longo do horizonte de planejamento, para $t = 1, ..., M$. Define as janelas temporais em que as operações de corte podem ser realizadas;
- j: corresponde aos padrões de corte viáveis, para $j = 1, ..., N$.

Parâmetros:

- b_i^t : demanda do item do tipo i no período t;
- d^t : data de entrega dos pedidos no período t;
- α_{ji}^t : números de itens do tipo i cortados de acordo com o padrão de corte j no período t;
- B: valor suficientemente grande;
- c_j^t: o custo de cortar um padrão de corte j no período t;
- γ^t: custo por unidade de atraso na entrega do período t;
- q^t : tempo de corte no período t;
- β^t: custo por unidade de adiantamento na entrega do período t;
- r_i^t : custo por item estocar itens do tipo i em cada período t;
- CS_{jk}^t : custo de troca de *setup* entre um padrão de corte j e outro k;
- S_{jk}^t: tempo de preparação da máquina, necessário para processar o padrão de corte k imediatamente após o padrão de corte j no período de tempo t.

Variáveis de Decisão:

- x_j^t: número de objetos cortados no período t segundo o padrão de corte j;
- T^t: tempo de atraso na entrega do pedido no período t;
- E^t: tempo de adiantamento na entrega do pedido no período t;
- C_j^t: tempo de conclusão do corte do padrão de corte j no período t;

- e_i^t : estoque de itens do tipo i no período t;
- P^t: (*makespan*) tempo despendido para cortar todos os padrões de corte que atendem a demanda dos itens no período t;
- v_j^t : variável binária que indica se o padrão de corte j esteve na máquina de corte no período t.
- $start^t$: tempo de início de corte de um período t;
- y_{jk}^t é uma variável binária que indica se o padrão de corte k é sequenciado imediatamente após o padrão de corte j no período t ($y_{jk}^t = 1$) ou não ($y_{jk}^t = 0$). Caso j seja igual a 0, então $y_{0k}^t = 1$ indica que o padrão de corte k é o padrão de corte que inicia na máquina de corte no período t. Observe que, para $t > 1$, $y_{0k}^t = 1$ indica que k é o padrão de corte que permanece na máquina de corte no final do período $t - 1$;
- SI^t : tempo de *setup* realizado no início do período t.

O modelo matemático para o 1D-PMCE-DE-SD é uma formulação que integra problemas de dimensionamento de lotes, programação da produção e sequenciamento de estoque. O objetivo é desenvolver uma abordagem que otimize a utilização do material, reduza custos operacionais e atenda às demandas dentro dos prazos estabelecidos, garantindo maior eficiência e competitividade no processo produtivo. Ele considera custos de produção e *setup*, além de aspectos relacionados a inventário e prazos de entrega. Este modelo matemático aborda um problema de corte unidimensional, no qual há produção escalar e um único tipo de objeto em quantidade suficiente para atender à demanda. Em algumas aplicações práticas, no entanto, as peças resultantes do corte podem atender a uma demanda compartilhada entre diferentes tipos de objetos, ou até mesmo o problema poderia se estender para duas ou três dimensões.

Para essa formulação usamos como base a formulação para o 1D-PMCE-DE-SI e a forma de modelamento do *setup* dependente da sequência descrita em Melega et al. (2022). Dessa maneira, a modelagem proposta para o 1D-PMCE-DE-SD é dada então por:

$$\min \quad \sum_{t=1}^{H}\sum_{j=1}^{N} c_j^t x_j^t + \sum_{t=1}^{H} \beta^t E^t + \sum_{t=1}^{H} \gamma^t T^t + + \sum_{t=1}^{H}\sum_{i=1}^{R} r_i^t e_i^t + \sum_{k=1}^{N} CS_{0k}^1 y_{0k}^1 + \sum_{t=1}^{H}\sum_{j=1}^{N}\sum_{\substack{k=1 \\ k\neq j}}^{N} CS_{jk}^t y_{jk}^t \tag{5.1}$$

$$\text{suj. a:} \quad \sum_{j=1}^{N} a_{ji}^t x_j^t + e_i^{t-1} = b_i^t + e_i^t \quad \forall t, \forall i, \tag{5.2}$$

$$start^t \geq d^{t-1} - E^{t-1} + T^{t-1} \quad \forall t \tag{5.3}$$

$$start^t + P^t + E^t - T^t = d^t \quad \forall t, \tag{5.4}$$

$$x_j^t \leq B v_j^t \quad \forall t, j = 1, \ldots, N \tag{5.5}$$

$$\sum_{\substack{j=0 \\ j\neq k}}^{N} y_{jk}^{t} = v_{k}^{t} \quad \forall t,\ k = 1,\ldots,N \tag{5.6}$$

$$\sum_{\substack{k=1 \\ j\neq k}}^{N} y_{jk}^{t} + y_{0j}^{t+1} = v_{j}^{t} \quad \forall t = 1,\ldots,H;\ j = 1,\ldots,N \tag{5.7}$$

$$y_{0j}^{t} + \sum_{\substack{k=1 \\ j\neq k}}^{N} y_{kj}^{t} = \sum_{\substack{k=1 \\ j\neq k}}^{N} y_{jk}^{t} + y_{0j}^{t+1}\ \forall t = 1,\ldots,H\text{-}1;\ j = 1,\ldots,N \tag{5.8}$$

$$\sum_{k=1}^{N} y_{0k}^{t} = 1 \quad \forall t \tag{5.9}$$

$$C_{k}^{1} \geq S_{0k}^{1} y_{0k}^{1} + q_{k}^{1} x_{k}^{1} \quad \forall k = 1,\ldots,N \tag{5.10}$$

$$C_{k}^{t} \geq q_{k}^{t} x_{k}^{t} + \sum_{j=1}^{N} S_{jk}^{t} y_{jk}^{t} \quad \forall t,k = 1,\ldots,N \tag{5.11}$$

$$C_{k}^{t} \geq C_{j}^{t} + S_{jk}^{t} + q_{k}^{t} x_{k}^{t} - B(1 - y_{jk}^{t})\ \forall t,j,k = 1,..,N\ :\ k \neq j \tag{5.12}$$

$$P^{t} = \sum_{k=1}^{N} q_{k}^{t} x_{k}^{t} + \sum_{j=1}^{N} \sum_{\substack{k=1 \\ k\neq j}}^{N} S_{jk}^{t} y_{jk}^{t} \quad \forall t \tag{5.13}$$

$$P^{t} \geq C_{j}^{t} \quad \forall t,j = 1,\ldots,N \tag{5.14}$$

$$C_{j}^{t} \leq B v_{j}^{t} \quad \forall t,j = 1,\ldots,N \tag{5.15}$$

$$C_{0}^{t} = 0 \tag{5.16}$$

$$d^{0} = 0 \tag{5.17}$$

$$x_{j}^{t} \in \mathbb{Z}_{+} \quad \forall j = 1,\ldots,N;\ \forall t \tag{5.18}$$

$$E^{t}, T^{t}, \in \mathbb{Z}_{+} \quad \forall t \tag{5.19}$$

$$P^{t} \in \mathbb{Z}_{+} \quad \forall t \tag{5.20}$$

$$start^{t} \in \mathbb{Z}_{+} \quad \forall t \tag{5.21}$$

$$e_{i}^{t} \in \mathbb{Z}_{+} \quad \forall t,i \tag{5.22}$$

$$v_{j}^{t} \in \{0,1\} \quad \forall t,j \tag{5.23}$$

$$y_{jk}^{t} \in \{0,1\} \quad \forall t,j,k. \tag{5.24}$$

A função objetivo (5.1), a ser minimizada, compreende cinco componentes: (i) custo de produção, (ii) custo associado a adiantar a produção no período, (iii) custo de atraso no período, (iv) penalidades associadas a estoques de itens e (v) custo de *setup* dependente da sequência. O conjunto (5.2) garantem que a produção e o estoque do período anterior supram a demanda e o estoque do período corrente. As restrições (5.3) garantem que a produção no período t só inicie após a conclusão da produção do período anterior, levando em conta antecipações e atrasos. Além disso, essa restrição permite que haja tempo ocioso na máquina. Por exemplo, no primeiro período se o $start_1 = 0$, afirma que não ouve tempo ocioso, caso contrário, há um tempo em que a máquina esteja parada.

O conjunto de restrições (5.4) tem como objetivo garantir a coerência temporal do processo modelado, estabelecendo uma relação entre o tempo de início, o tempo

de processamento, as antecipações e os atrasos para determinar a data final. Se não houver antecipação e nem atraso, a equação garante que a atividade termine exatamente no prazo. Essa restrição é essencial para sincronizar eventos ao longo do tempo e assegurar que os prazos sejam respeitados ou devidamente ajustados, conforme as condições operacionais.

As restrições (5.5) utilizam a técnica *big-B* para condicionar a relação entre variáveis contínuas e binárias, em que B é um parâmetro suficientemente grande. O conjunto de restrições (5.6) aborda a precedência de padrões de cortes. Ele assegura que cada máquina atenda apenas a padrões de corte ativos no período.

As expressões (5.7) são restrições de sucessão de transferências que indicam que a soma dos *setups* deve ser igual à quantidade de trocas no período t, garantindo a continuidade das transferências entre os períodos. O conjunto de restrições (5.8) assegura o balanceamento de fluxo de padrão de corte na máquina para cada período. As restrições (5.9) garantem que um padrão de corte só será produzido se a máquina de corte tiver sido configurada no início do período.

O conjunto de restrições (5.10) assegura que o tempo de conclusão do padrão de corte k no primeiro período deve levar em conta o tempo de *setup* inicial (caso ele seja o primeiro padrão de corte a ser executado) e o tempo de corte necessário. O conjunto de restrições (5.11) garante que o padrão de corte k só pode ser concluído após o tempo de corte e o *setup* que o precede. Esta restrição assegura que o tempo de conclusão leva em conta a troca do padrão de corte anterior. O conjunto de restrições (5.12) garante que se $y_{jk}^t = 1$, então o padrão de corte k só pode começar após o término de j, somado ao tempo de *setup* S_{jk}^t e ao tempo de corte de k. Se $y_{jk}^t = 0$, a restrição é relaxada.

O conjunto de restrições (5.13) assegura o cálculo do tempo total de despendido para cortar todos os padrões do período. O conjunto de restrições (5.14) assegura que o *makespan* seja pelo menos tão grande quanto o maior tempo de conclusão. O conjunto de restrições (5.15) assegura que os tempos de conclusão sejam válidos apenas para máquinas ativas.

O conjunto de restrições (5.16) define o tempo de conclusão inicial como zero. O conjunto de restrições (5.17) define a data inicial como zero. As expressões (5.18)-(5.24) são as condições de não-negatividade e integralidade das variáveis do modelo.

Diante do exposto, o modelo proposto apresenta uma formulação matemática para a resolução do problema em questão, integrando variáveis de decisão, restrições operacionais e critérios de otimização. A inclusão de restrições de acoplamento, precedência e sucessão garante a consistência das decisões ao longo do horizonte de planejamento, enquanto a função objetivo contempla múltiplos custos relevantes, proporcionando uma abordagem abrangente e realista. A estrutura do modelo possibilita sua aplicação em cenários práticos, permitindo simulações e ajustes conforme necessidades específicas.

5.4 Método de Solução para o 1D-PMCE-DE-SD

Esta seção apresenta os procedimentos heurísticos desenvolvidos para solucionar o problema de corte de estoque unidimensional, considerando restrições adicionais, como datas de entrega e tempos de *setup* dependentes da sequência de corte. Para

a construção dessas heurísticas, foram realizadas adaptações e aprimoramentos em métodos clássicos amplamente estudados na literatura, originalmente voltados para o problema de corte de estoque convencional.

Nos tópicos seguintes, detalhamos os passos para a resolução do problema, descrita da seguinte maneira:

(i) Fase 1: Processo de Geração de Padrões de Corte com a heurística *First-Fit-Decreasing*-Modificada (FFD-M) (subseção 5.4.1)

(ii) Fase 2: Geração da Solução Inicial (subseção 5.4.2)

(iii) Fase 3: Melhorando a solução inicial com a matheurística *Local Branching* (subseção 5.4.3)

5.4.1 Fase 1: Processo de Geração de Padrões de Corte com a heurística *First-Fit-Decreasing*-Modificada (FFD-M)

Para a resolução do 1D-PMCE-DE-SD, precisamos de padrões de corte para serem usados na formulação proposta e com isso ser possível a escolha de qual padrão de corte será usado e quantas vezes ele será usado (cortado). Nos modelos anteriores (ou seja, Capítulos 3 e 4), propomos o método simplex com geração de colunas para determinar os possíveis padrões de corte para resolver o problema. No presente capítulo, envolvendo a solução do 1D-PMCE-DE-SD, propomos a heurística *First-Fit-Decreasing*-Modificada-(FFD) para gerar possíveis padrões de corte a serem utilizados, ao invés de utilizar o método anteriormente utilizado.

A Heurística *First-Fit-Decreasing* (FFD) consiste em colocar o maior item em um padrão de corte tantas vezes quanto for possível, ou seja, até que não haja mais espaço para colocar esse item, ou até que sua demanda já tenha sido atendida (os itens maiores são colocados em primeiro lugar, uma vez que são mais difíceis de serem combinados). Assim, quando não for mais possível ou necessária a inclusão do maior item, o segundo maior item é considerado e assim por diante, até o menor item. Quando nenhum novo item pode ser incluído, um padrão de corte é construído e repetido tantas vezes quanto for possível. Este procedimento de gerar um padrão de corte e repeti-lo tantas vezes quanto possível é conhecido na literatura como heurística de repetição exaustiva (Hinxman, 1980) e é bastante empregado na área de corte e empacotamento.

De posse dessa ideia, fizemos a modificação para que encontre padrões de corte para vários períodos de tempo. A heurística *First-Fit-Decreasing*-Modificada - (FFD-M) consiste na heurística *First-Fit-Decreasing* de modo que o processo de apresentação dos padrões de corte será repetido para cada um dos períodos de tempo do problema trabalhado. Dessa maneira, é possível que haja padrões de corte iguais entre os períodos de tempo. Após serem gerados os padrões de corte para cada um dos períodos, juntamos todos esses padrões de corte para fazer parte da solução inicial dos padrões de corte. Esse novo conjunto maior ou igual de padrões de corte serão repetidos para todos os períodos de tempo. Dessa maneira, essas colunas estarão disponíveis para a resolução do modelo ao iniciarmos a abordagem de solução.

Assim, resta apenas encontrar quantas vezes esses padrões de corte serão utilizados, uma vez que o problema não é apenas determinar esse valor, mas sim um plano

de corte que atenda às demais restrições do modelo. A Fase 2 do processo de solução fica encarregada de encontrar uma solução inicial, para que então a Fase 3 do método possa melhorá-la, se possível.

Vejamos, como ilustração, um pequeno exemplo desse método para encontrar os padrões de corte.

Exemplo 2. *Suponha que temos disponíveis em estoque objetos de comprimento* $L = 100$ *para atender às demandas de itens indicadas na Tabela 3.1, que mostra as solicitações da empresa.*

Tabela 5.2: Exemplo Numérico: demanda de item em cada período

	Demanda dos itens		
Período	1(40)	2(30)	3(20)
1	2	0	1
2	1	1	1
3	1	3	3
Comprimento	40	30	20

Os itens a serem ordenados (maior para o menor) no primeiro período seriam 40, 40 *e* 20, *com a seguinte alocação nos objetos que serão cortados:*

(1) Objeto 1: Comprimento dos itens igual a 40 + 40 + 20

Alocamos o primeiro item no primeiro objeto no qual ele cabe. Em seguida, passamos para o segundo item e verificamos se ele cabe dentro do objeto corrente: caso não caiba, o alocamos em um novo objeto. O processo é repetido até acabar todos os itens que seriam demandados. Essa alocação gera o padrão de corte (2,0,1) *(ver Figura 5.2), que significa que cortaríamos duas vezes o item* 1(40), *nenhuma vez o item* 2(30) *e uma vez o item* 3(20). *Este padrão de corte atenderia à demanda dos pedidos sem levar em consideração mais nada, e não deixaria sobra no corte, uma vez que* $40 + 40 + 20 = 100 = L$.



Figura 5.2: Padrão de corte para o primeiro período do Exemplo Ilustrativo 2. Fonte: Própria do Autor.

Os itens a serem ordenados (maior para o menor) no segundo período seriam 40, 30 *e* 20, *com a seguinte alocação nos objetos que serão cortados:*

(2) Objeto 2: Comprimento dos itens igual a 40 + 30 + 20

Agindo da mesma maneira que no período anterior, alocamos os itens nos seus respectivos objetos, desde que caibam. Essa alocação produz o padrão de corte (1,1,1) *(ver Figura 5.3), que significa cortar um item de cada tipo. Este padrão de corte atenderia à demanda dos pedidos sem levar em consideração mais nada, e deixaria sobra de* 10 *no corte, uma vez que* $40 + 30 + 20 = 90 = L - 10$.

Continuando esse procedimento, os itens ordenados (maior para o menor) no terceiro período são 40, 30,30,30,10,10,10,10 e 10, com a seguinte alocação de comprimentos de itens a serem cortados nos objetos que serão cortados :

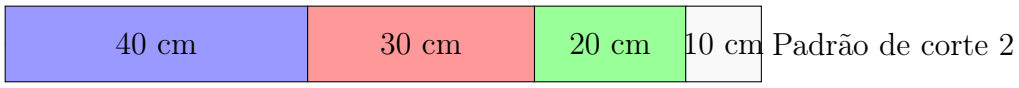


Figura 5.3: Padrão de corte para o segundo período do Exemplo Ilustrativo 2. Fonte: Própria do Autor.

1. *Objeto 3: Comprimento dos itens igual a 40 + 30 + 30*
2. *Objeto 4: Comprimento dos itens igual a 30 + 20 + 20 + 20*

Essa alocação apresenta os padrões de corte (1,2,0) e (0,1,3), conforme a Figura 5.3. Esses padrões atenderiam à demanda dos pedidos sem levar em consideração mais nada, sem deixar sobra no primeiro padrão de corte, uma vez que 40+30+30 = 100, mas permitindo sobra no segundo, pois 30 + 20 + 20 20 = 90.

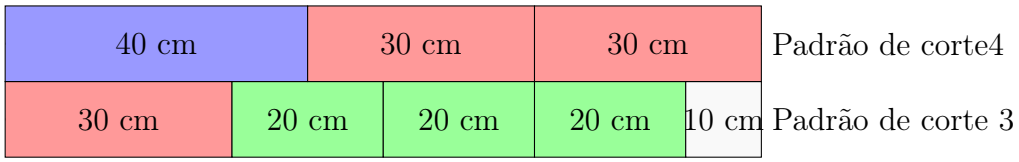


Figura 5.4: Padrões de corte para o terceiro período do Exemplo 2. Fonte: Própria do Autor.

Dessa maneira, os padrões de corte que resolveriam o problema básico 1D-PMCE-DE-SD seriam (2,0,1), (1,1,1), (1,2,0) e (0,1,3), sendo, portanto, 4 padrões de corte possíveis. Para resolver o problema proposto além desses padrões de corte, adicionamos os padrões de corte homogêneos que, para o problema, seriam (2,0,0), (0,3,0) e (0,0,5), conforme a Figura 5.4.

Com a junção de todos esses padrões de corte apresentados, há um total de 7 padrões de corte, que são dados inicialmente e utilizados em todos os períodos. A Figura 5.5 apresenta o corte dos itens nos objetos, considerando os 7 padrões como entrada para a resolução do 1D-PMCE-DE-SD. Cada padrão de corte é composto por segmentos de diferentes comprimentos (representados por cores distintas) e inclui uma porção de sobra (em cinza), que representa a parte do material não utilizada no corte específico. Ressalte-se que os três primeiros padrões de corte são os padrões homogêneos, nos quais apenas um tipo de item é cortado no objeto; já os quatro últimos são os padrões gerados pela heurística FFD-Modificada. É possível ver nesses cortes que 4 dos 7 padrões apresentam uma perda no objeto, pois esse pedaço que sobra do objeto não será reutilizado nessa abordagem.

5.4.2 Fase 2: Geração da Solução inicial

Para encontrarmos uma solução inicial para o 1D-PMCE-DE-SD, sugerimos uma solução apenas com os padrões de corte homogêneos. A ordem de corte escolhida foi o padrão de corte com o corte do item na forma decrescente.

Exemplo 3. *No Exemplo ilustrativo 2, teríamos os seguintes padrões de corte:*

(i) (2,0,0) seria o primeiro a ser cortado;

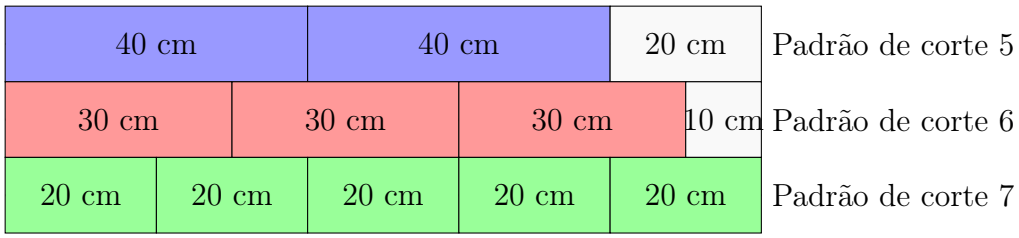


Figura 5.5: Apresentação dos padrões de corte homogêneos para o Exemplo Ilustrativo 2. Fonte: Própria da Autora.

(ii) (0,3,0) seria o segundo a ser cortado;

(iii) (0,0,5) seria o terceiro a ser cortado.

Agora já sabemos quais padrões seriam cortados e qual a ordem dos cortes. Restaria determinar quantas vezes esses padrões de corte deveriam ser repetidos no corte. A ideia proposta para que seja possível atender à demanda foi a seguinte:

$$x_j^t = \lceil \frac{b_i^t}{\ell_i} \rceil + 1. \tag{5.25}$$

Para o próximo período, como o padrão de corte 3 já estaria na máquina, então ele deveria ser o primeiro a ser cortado. A ordem de corte poderia ser:

1. *(0,0,5) seria o primeiro a ser cortado;*
2. *(0,3,0) seria o segundo a ser cortado;*
3. *(2,0,0) seria o terceiro a ser cortado.*

Dessa maneira para o terceiro período de corte, teríamos novamente o seguinte corte:

1. *(2,0,0) seria o primeiro a ser cortado;*
2. *(0,3,0) seria o segundo a ser cortado;*

Tabela 5.3: Plano de Corte Inicial.

	Itens					
Períodos	1	2	3	Ordem de Corte	#Corte	Atendimento da demanda
1	2	0	0	1	2	2
	0	3	0	2	1	3
	0	0	5	3	2	10
2	0	0	5	1	2	10
	0	3	0	2	2	6
	2	0	0	3	2	4
3	2	0	0	1	2	4
	0	3	0	2	2	6
	0	0	5	3	2	10

3. (0,0,5) seria o terceiro a ser cortado.

O processo irá se repetir para todos os períodos de corte. Dessa maneira, teríamos uma solução para o plano de corte de acordo com a Tabela 5.3.

Com isso, encontramos uma solução inicial, a qual tentaremos melhorar com a aplicação da matheurística Local Branching.

5.4.3 Fase 3: Melhorando a solução inicial com a matheurística *Local Branching*

Assim como realizamos a aplicação da matheurística *Local Branching* para o 1D-PMCE-DE, de igual modo propomos esse método para o 1D-PMCE-DE-SD. Ao definir vizinhanças de maneira eficiente, a matheurística *Local Branching* oferece uma forma de explorar soluções promissoras rapidamente, permitindo que se escape de mínimos locais e que a busca continue até encontrar uma solução próxima da ótima para problemas de grande porte.

Dessa maneira, as mesmas restrições e limitantes foram estabelecidos para a resolução do 1D-PMCE-DE-SD, assim como descrito no Algoritmo 3. Então, de acordo com Fischetti e Lodi (2003), uma restrição de ramificação local pode ser definida como:

$$\Delta(x, x') := \sum_{t=1}^{M} \sum_{j \in J_l(x')} \mu_{tj}(x_j^t - l_{tj}) + \sum_{t=1}^{M} \sum_{j \in J_u(x')} \mu_{tj}(u_{tj} - x_j^t) \leq k. \tag{5.26}$$

O valor de μ_{tj}, como sugerido por Fischetti e Lodi (2003), é dado por:

$$\mu_{tj} = \frac{1}{u_{tj} - l_{tj}} \quad \forall j, t.$$

Os valores l_{tj} e u_{tj} 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'_{tj}, o espaço de solução pode ser particionado por meio da disjunção:

$$\Delta(x,x') \leq k, \tag{5.27}$$

ou

$$\Delta(x,x') \geq k + 1. \tag{5.28}$$

5.5 Geração de Instâncias para o 1D-PMCE-DE-SD

Na literatura encontramos o CUTGEN1, proposto por Gau e Wäscher (1995), que é um gerador de problemas-teste para o problema de corte de estoque unidimensional. Apesar de sua ampla utilização para testes com problemas de corte de estoque unidimensional, não pode ser aplicado em nossos testes. O 1D-PMCE-DE-SD envolve muito mais parâmetros que um problema de corte clássico. Assim, desenvolvemos um gerador aleatório apropriado, que é descrito a seguir. Os modelos matemáticos foram

implementadas em C++, usando Concert, e foram resolvidos usando o solucionador IBM ILOG CPLEX com parâmetros padrão de corte. Impomos um limite de tempo computacional de 3600 segundos e uma tolerância de *gap* de otimalidade de 0,1%, para cada instância, visando encontrar uma solução viável de alta qualidade para o problema.

Para a execução dos testes computacionais para problemas multiperíodos de corte de estoque unidimensional, os seguintes parâmetros foram fixados, em que alguns deles foram baseados nos parâmetros de Melega et al. (2020):

(i) comprimento do objeto em estoque $L = 1000$;

(ii) número de períodos: $T = 3$ e 6;

(iii) número de tipos de itens: $n = 50$ e 100;

(iv) tempo para cortar um padrão de corte j: $c_j^t = 1$;

(v) custo de cortar objetos: L;

(vi) custo de troca de padrão de corte: $CS_{jk}^t = \Gamma_{jk}^t$;

(vii) custo por item estocado: $r_i^t = 0{,}1\ell_i$;

(viii) custo por unidade de atraso na entrega: $\gamma = 0{,}1L$;

(ix) custo por unidade de adiantamento na entrega: $\beta^t = 0{,}1L$;

(x) data de entrega: d^t, de acordo com a Tabela 5.4.

Outros parâmetros necessários para os testes foram gerados aleatoriamente nos seguintes intervalos:

(i) comprimento ℓ_i dos itens demandados: conforme a Tabela **??**, há três classes de comprimentos de itens. Em cada classe, o comprimento ℓ_i dos itens é gerado aleatoriamente segundo uma distribuição uniforme.

Por simplicidade, usaremos a notação (P,M,G) para representar as classes. Observe que a classe P é formada por itens pequenos e pouco heterogêneos em relação ao objeto; a classe M, por sua vez, é uma classe mais heterogênea, pela maior amplitude na variação de valores; e, finalmente, a classe G possui itens variados, de dimensões que envolvem dimensões de itens nas demais classes.

(ii) demanda dos itens: $b_i^t \in [0{,}125]$;

(iii) tempo de preparação (tempo de *setup*) da máquina, necessário para processar padrões diferentes: $S_{jk}^t = [1{,}10] \times \Gamma_{jk}^t$.

A Tabela 5.4 apresenta os intervalos de valores utilizados para definir as datas de entrega nas instâncias em estudo, que são D1, D2 e D3, com base na capacidade de corte, utilizando uma distribuição uniforme. O intervalo D1, caracterizado como apertado ou estreito, representa um planejamento com menor flexibilidade. O intervalo D2 é considerado médio, enquanto o intervalo D3, classificado como largo, permite maior margem para ajustes na programação da produção.

Essas estruturas permitem avaliar o impacto da variação do tamanho dos itens e das restrições nas datas de entrega sobre a eficiência do planejamento e da alocação da capacidade de corte.

Tabela 5.4: Intervalo de Valores para as datas de entrega

Descrição	Distribuição Uniforme
Datas D1 (Apertada)	$d^t \in U[0{,}5\ CAP^t,\ \ 1{,}0\ CAP^t]$
Datas D2 (Média)	$d^t \in U[1{,}0\ CAP^t,\ \ 2{,}0\ CAP^t]$
Datas D3 (Larga)	$d^t \in U[2{,}0\ CAP^t,\ \ 4{,}0\ CAP^t]$

Os tempos de *setup* da máquina de corte são dependentes da sequência e relacionados aos padrões de corte consequentes. Esses tempos de *setup* são calculados considerando um valor fixo (uniformemente distribuído em $[1, 10]$), multiplicado pelo número de alterações entre dois padrões de corte (Γ). A matriz Γ representa os custos de *setup* associados à troca entre diferentes padrões de corte. Cada entrada da matriz indica o custo de troca de um padrão de corte j para o padrão de corte k. Essa matriz serve de base para entender como é a distribuição desses custos.

Neste trabalho, as alterações são contabilizadas por qualquer adição ou remoção de uma peça de um padrão de corte para outro, como por exemplo na Figura 5.6. Para ilustrar os cálculos de transição para as configurações, um conjunto de padrões de corte foram gerados e a sua matriz de mudança Γ são apresentados nessa figura. Esses valores estão de acordo com os padrões de corte do Exemplo ilustrativo 2.

$$\Gamma = \begin{bmatrix} - & SC_1 & SC_2 & SC_3 & SC_4 & SC_5 & SC_6 & SC_7 \\ SC_1 & 0 & 5 & 7 & 1 & 3 & 3 & 6 \\ SC_2 & 5 & 0 & 8 & 6 & 4 & 2 & 5 \\ SC_3 & 5 & 8 & 0 & 6 & 6 & 8 & 3 \\ SC_4 & 1 & 6 & 6 & 0 & 2 & 4 & 5 \\ SC_5 & 3 & 4 & 6 & 3 & 0 & 2 & 6 \\ SC_6 & 3 & 2 & 8 & 4 & 2 & 0 & 5 \\ SC_7 & 6 & 5 & 3 & 5 & 3 & 5 & 0 \end{bmatrix}$$

Figura 5.6: Matriz dos custos de *setup* para o Exemplo ilustrativo 2.

A diagonal principal contém valores zero, pois não há custo de troca ao manter o mesmo padrão de corte. Os custos variam conforme a complexidade da mudança entre padrões de corte, dependendo das diferenças nos tipos e quantidades de cortes necessários. Mudanças entre padrões de corte mais similares tendem a ter custos menores, enquanto transições para padrões de corte com maior variação de cortes resultam em custos mais altos.

Na matriz Γ de custos de *setup*, as entradas correspondem ao número de peças diferentes entre uma mudança do padrão de corte j para o padrão de corte k. Por exemplo, a primeira linha da matriz de mudança exibe as configurações necessárias para uma mudança do padrão de corte 1 para todos os outros padrões de corte. Para mudar do padrão de corte 1 ((2,0,0)) para o padrão de corte 2 ((0,3,0)), conforme exibição dos padrões de corte apresentados na Figura 5.6, trocaríamos dois cortes do

item 1 e adicionaríamos três cortes do item 2 e nada do item 3; assim, $2 + 3 = 5$ configurações. Já para o corte do padrão 1 para o padrão 3 ((0,0,5)) será necessário retirar dois cortes do item 1 e adicionar cinco cortes com item 3, de modo que um total de $2 + 5 = 7$ configurações e para o padrão de corte 4, seria apenas a inserção de um corte do item três, e assim por diante.

A Tabela 5.5 apresenta a classificação das instâncias utilizadas no estudo, organizadas em 12 classes diferentes. Cada classe é definida com base em três parâmetros principais: (i) o número de itens distintos considerados na instância; (ii) a data de entrega associada aos itens; (iii) e o número de períodos considerados no planejamento. A estrutura da tabela é dividida em duas seções para facilitar a visualização das diferentes combinações desses parâmetros.

Tabela 5.5: Classes de Instâncias

Classes	Descrição			Classes	Descrição		
	Itens	Data	Períodos		Itens	Data	Períodos
C1	50	D1	3	C7	50	D1	6
C2	100	D1	3	C8	100	D1	6
C3	50	D2	3	C9	50	D2	6
C4	100	D2	3	C10	100	D2	6
C5	50	D3	3	C11	50	D3	6
C6	100	D3	3	C12	100	D3	6

Na primeira parte da tabela (classes C1 a C6), as instâncias possuem três períodos de planejamento, variando o número de tipos de itens entre 50 e 100 e utilizando três datas de entrega distintas (D1, D2 e D3). Já na segunda parte (classes C7 a C12), as instâncias seguem a mesma estrutura, porém com seis períodos de planejamento. Essa variação permite avaliar o impacto do aumento do horizonte de planejamento sobre os resultados do modelo.

5.6 Experimentos Computacionais para o 1D-PMCE-DE-SD

Para garantir uma apresentação clara e organizada dos resultados, estruturamos a discussão dos testes em duas seções distintas. A Seção 5.6.1 abrange os testes realizados com um horizonte de planejamento de 3 períodos, enquanto a Seção 5.6.2 trata dos testes com 6 períodos. Essa divisão permite uma análise mais detalhada e comparativa entre os diferentes cenários avaliados. Além disso, facilita a identificação de padrões e variações no desempenho dos métodos utilizados. Separar os testes dessa forma também contribui para uma melhor compreensão das diferenças na qualidade das soluções obtidas em cada caso.

5.6.1 Solução para as classes de instâncias com 3 períodos de tempo

Esta seção apresenta, na Subseção 5.6.1.1, os resultados obtidos usando a matheurística *Local Branching*. A Subseção 5.6.1.2 mostra uma comparação desses resultados com os encontrados usando o *solver* CPLEX.

5.6.1.1 Solução usando *Local Branching*

Na Tabela 5.6 estão apresentadas as médias dos resultados para os 10 instâncias das classes com 3 períodos de tempo, para 6 categorias de problemas descritos na tabela, uma vez que há 6 classes e cada classe tem 3 categorias de tipos de itens, denominados P, M e G, segundo Tabela **??**. Além disso, a diferenciação de cada classe é baseada na diferença dos 3 tipos de datas de entrega.

Tabela 5.6: Resultados dos componentes da função objetivo para o *Local Branching* (LB) para as classes com 3 períodos.

		Termos da Função Objetivo - LB					LB	
Itens	Classe	Atraso	Adiantamento	Estoque	*Setup*	Objetos	FO	Tempo
50	C1-P	9530,60	0,00	6347,20	97,80	1031,10	2054564,00	3591,46
	C1-M	4742,50	0,00	1521,80	124,50	4206,40	4716189,00	3592,96
	C1-G	5981,90	0,00	692,80	139,30	5933,70	6560465,00	1506,96
100	C2-P	72615,70	0,00	11692,50	234,40	2284,10	9818468,00	3596,67
	C2-M	24954,70	0,00	6898,20	218,90	9589,30	12283870,00	3593,65
	C2-G	14396,10	0,00	1498,90	270,40	11914,50	13427220,00	3597,67
50	C3-P	8807,50	0,00	5717,70	110,50	1027,30	1975056,00	3593,34
	C3-M	1048,40	0,00	1704,60	141,60	4301,40	4450653,00	3018,54
	C3-G	738,70	0,00	746,90	134,00	5454,40	6031051,00	1052,58
100	C4-P	60318,10	0,00	11913,60	219,20	2265,80	8564899,00	3597,58
	C4-M	16625,70	0,00	4537,50	260,40	9679,80	11493113,00	3590,92
	C4-G	2893,60	0,00	1477,80	267,40	12002,90	12365710,00	3590,95
50	C5-P	11880,40	0,00	5540,10	118,10	1040,60	2306003,00	3594,84
	C5-M	0,00	0,00	1378,20	173,90	3849,80	3889478,00	1927,54
	C5-G	0,00	0,00	544,20	160,40	5931,80	5957533,00	714,74
100	C6-P	63019,60	0,00	9205,00	224,80	2169,40	8706590,00	3597,81
	C6-M	5600,00	165,80	4275,70	275,20	10139,60	10860582,00	3586,72
	C6-G	0,00	0,00	2073,90	304,50	11766,80	11876250,00	2591,26

A análise dos termos da função objetivo na Tabela 5.6 revela pontos críticos na abordagem e possíveis deficiências no modelo. Vejamos então alguns desses pontos. O primeiro é o custo de atraso, que é consistentemente o maior fator na função objetivo, especialmente nas classes de tipos de itens P e M. Algumas observações relevantes: as classes C2-P (72615,7) e C6-P (63019,6) indicam uma dificuldade grave no planejamento da produção. Em classes maiores, como C3-G (738,7) e C5-G (0), o plano de corte não está conseguindo cumprir os prazos de entrega de forma eficiente.

O custo de estoque mostra uma grande variação entre as classes. Em algumas instâncias, ele é alto, indicando um excesso de produção, enquanto em outras as ordens de grandeza são menores que para as classes do tipo P. Por exemplo, os valores nas classes C2-P (11692,5) e C4-P (11913,6) indicam que o modelo gera produção excessiva para mitigar atrasos. Já para as classes C5-G (544,2) e C3-G (746,9) demonstra-se que a produção está sendo feita de maneira mais controlada.

O custo de *setup* representa o impacto da troca de padrão de corte na produção. Nas classes com tipos de itens G, ele tende a ser mais alto, sugerindo uma maior necessidade de mudanças na sequência de cortes. Alguns exemplos de alto custo de *setup* são as classes C6-G (304,5) e C4-G (267,4), que indicam que há muitas trocas na configuração de produção. Já as classes C1-P (97,8) e C3-P (110,5) mostram que classes de itens de menor comprimento exigem menor número de reconfigurações.

O tempo de execução é muito elevado na maioria das instâncias, ficando em torno de 3500 segundos, o que pode ser um problema para aplicações práticas que demandam tempos menores. Para as classes C5-G (714,74) e C3-G (1052,58), o algoritmo é notavelmente mais rápido. No caso de execução lenta, as classes C1-M (3592,96) e C4-M (3590,92) têm tempos altos, mesmo considerando que os tipos de itens são da classe M.

Para detalhar de melhor forma as variações dos valores dos termos da função objetivo para o *Local Branching*, apresentaremos, nas Figuras 5.7 e 5.8, cada um dos componentes da função para as classes C1-P, C1-M e C1-G e C6-P, C6-M e C6-G.

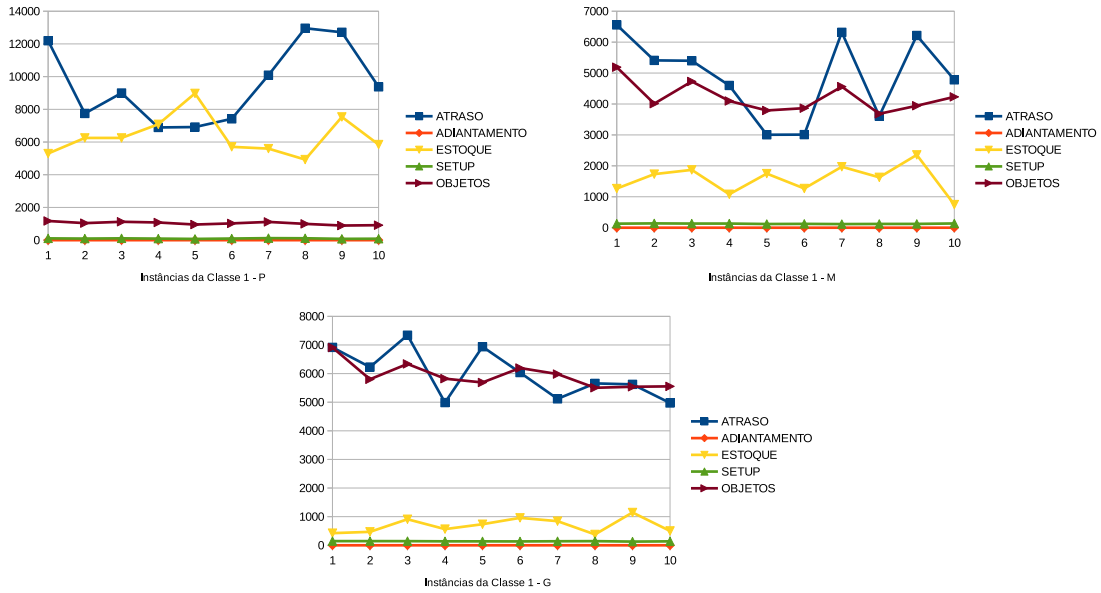


Figura 5.7: Comparação dos valores das componentes da função objetivo para as classes C1-P, C1-M e C1-G. Fonte: Própria da Autora.

Observando a Figura 5.7 é possível perceber que o valor do maior componente na função objetivo é o atraso e que com o aumento do comprimento dos itens o número de objetos vai diminuindo. Na classe P, esse número ficou em torno de 6000; na classe M, aproximadamente 2000; e na classe G, cerca de 1000.

5.6.1.2 Comparação da Abordagem de solução proposta com o CPLEX

Nesta subseção fazemos uma comparação dos resultados que foram obtidos com a nossa proposta de solução e a solução fornecida pelo *solver* CPLEX. Nessa análise, foi utilizado o *solver* para resolver o problema proposto nessa tese (1D-PMCE-DE-SD) sem realizar nenhuma ajuste nos seus parâmetros, o que chamamos de CPLEX. Dessa

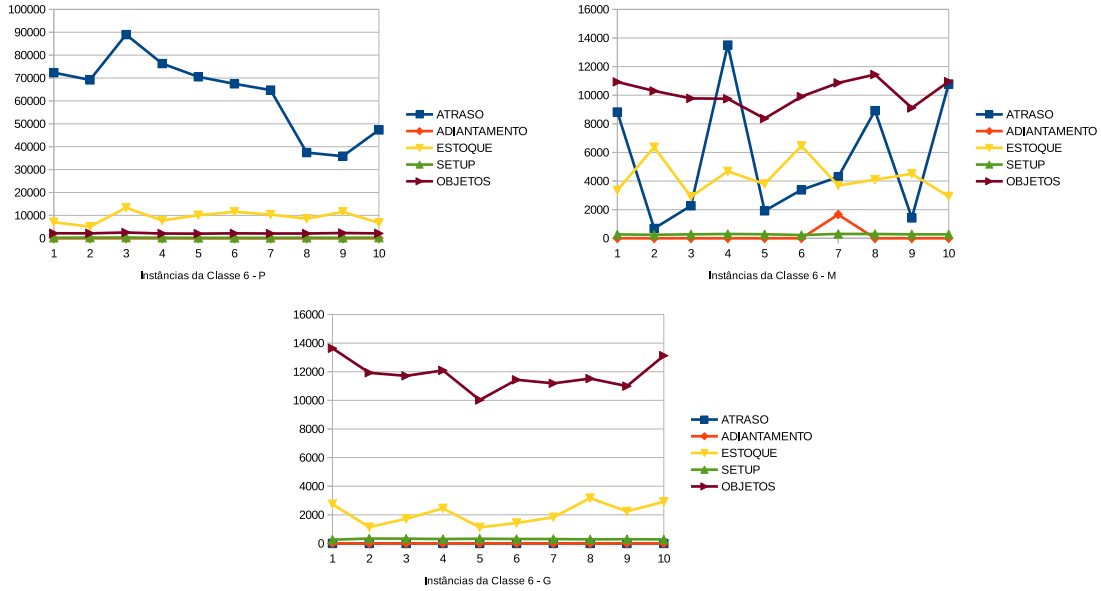


Figura 5.8: Comparação dos valores das componentes da função objetivo para a classe C6-P, C6-M e C6-G. Fonte: Própria do Autor.

maneira, como descrito na quinta coluna da Tabela 5.7, o *solver* não conseguiu encontrar nenhuma solução no tempo computacional limite de 3600 segundos. Após esses resultados, em um novo experimento, fornecemos, como solução inicial para o *solver*, um MIPSTART, a qual chamamos de $CPLEX_{Mip}$. Essa solução inicial de referência foi gerada da mesma maneira que a solução inicial usada para a matheurística *Local Branching*, conforme descrito na Seção 5.4.2.

Quando se fornece uma solução inicial para o *solver* CPLEX, esse a processa antes de iniciar o *Branch-and-cut* durante uma otimização. Se ao processar a solução MIP inicial, uma ou mais soluções são encontradas, o CPLEX utiliza a melhor dessas soluções como a solução incumbente para continuar a resolução do problema. Ter uma solução incumbente desde o início do *Branch-and-cut* permite que o CPLEX elimine partes do espaço de busca e, portanto, pode resultar em árvores menores de *Branch-and-cut*. Com a utilização do MIPSTART, foi possível encontrar um solução melhorada para o problema dentro do limite de tempo fornecido, como mostrado nas colunas CPLEX_Mip da tabela.

Na Tabela 5.7, a Solução de Referência apresenta tempos de execução extremamente baixos (sempre inferiores a 2 segundos), o que a torna muito rápida para obter uma solução inicial. No entanto, a qualidade dessas soluções é bastante inferior às obtidas pelos outros métodos. Por exemplo, na instância C1-P, a função objetivo (FO) da solução de referência é 579836283, enquanto as soluções encontradas pelo *Local Branching* (LB) e pelo CPLEX com MIPSTART são ordens de grandeza menores (2054564 e 4110711, respectivamente). Isso demonstra que a solução de referência serve apenas como um ponto de partida, mas não é competitiva em termos de qualidade.

A metaheurística *Local Branching* (LB) tem um impacto significativo na melhoria da função objetivo. Em praticamente todas as instâncias, o valor médio de função

Tabela 5.7: Resultados com termos da função objetivo para o *Local Branching* em comparação com o valor do CPLEX para as classes com 3 períodos

		Solução Referência		LB		CPLEX		CPLEX_Mip		*Gap*	
Itens	Classe	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	Gap_F	Gap_T
50	C1-P	579836283,00	0,21	**2054564,00**	3591,46	–	3600,00	4110711,00	3599,77	-100,07	-0,23
	C1-M	7499104,00	0,23	4716189,00	3592,96	–	3600,00	**4308523,00**	3591,30	9,00	0,08
	C1-G	8012825,00	0,14	6560465,00	1506,96	–	3600,00	**6431551,00**	1488,89	98,95	1,19
100	C2-P	1323767780,00	1,33	**9818468,00**	3596,67	–	3600,00	12992590,00	3301,82	-32,32	8,19
	C2-M	15270390,00	1,35	**12283870,00**	3593,65	–	3600,00	13953658,00	3298,12	-13,59	8,22
	C2-G	15526130,00	0,96	**13427220,00**	3597,67	–	3600,00	15014111,00	3100,19	-11,81	13,82
50	C3-P	6484473,00	0,21	**1975056,00**	3593,34	–	3600,00	5313022,00	3599,80	-169,00	-0,17
	C3-M	6987284,00	0,22	**4450653,00**	3018,54	–	3600,00	4478934,00	3438,16	-0,63	-13,90
	C3-G	7375915,00	0,15	6031051,00	1052,58	–	3600,00	**5932477,00**	1948,55	1,63	-85,12
100	C4-P	12760360,00	1,29	**8564899,00**	3597,58	–	3600,00	12418213,00	3277,90	-44,98	8,88
	C4-M	14077460,00	1,37	**11493113,00**	3590,92	–	3600,00	13499845,00	3271,38	-17,46	8,89
	C4-G	14414710,00	0,97	**12365710,00**	3590,95	–	3600,00	13487171,00	3263,07	-9,06	9,13
50	C5-P	6204843,00	0,21	**2306003,00**	3594,84	–	3600,00	4815112,00	3598,09	-108,80	-0,09
	C5-M	5930064,00	0,22	**3889478,00**	1927,54	–	3600,00	3907036,00	3437,36	-0,45	-78,32
	C5-G	7525355	0,14	5957533	714,74	–	3600,00	**5874252**	1802,94	1,39	-152,24
100	C6-P	12300500	1,32	**8706590**	3597,81	–	3600,00	12286879	3599,95	-41,12	-0,05
	C6-M	12118350	1,36	**10860582**	3586,72	–	3600,00	10881093	3571,16	-0,18	0,43
	C6-G	14393030	0,95	**11876250**	2591,26	–	3600,00	12032837	3063,25	-1,31	-18,21
	Média	115026936,44	0,70	7629871,88	2996,45	–	3600,00	8985445,27	3125,09	-24,43	-16,08

objetivo (FO) obtido pelo LB é substancialmente menor que o da Solução de Referência. Por exemplo, na classe C2-P, o FO inicial é 1323767780, enquanto o LB reduz esse valor para 9818468, uma queda impressionante. No entanto, essa melhoria tem um custo: o tempo de execução do LB é sempre muito alto, variando entre 714 e 3597 segundos, tornando-o um método caro em termos computacionais.

Outra característica relevante dos resultados obtidos pelo método LB é seu desempenho superior ao do CPLEX com MIPSTART em 14 instâncias, nas quais foram encontradas soluções de melhor qualidade. Embora o CPLEX seja um solucionador exato, seu desempenho prático pode ser limitado por restrições de tempo computacional ou por pontos de partida subótimos. Nesse contexto, a abordagem heurística proposta mostra-se capaz de explorar eficientemente o espaço de soluções viáveis, superando, em certos casos, a qualidade das soluções encontradas pelo solver comercial dentro do tempo estipulado.

Na classe C2-P, por exemplo, o LB encontrou um FO de 9818468, enquanto o CPLEX com MIPSTART ficou em 12992590, mostrando que o LB pode, em alguns casos, obter soluções mais otimizadas. Entretanto, há exceções: em algumas instâncias, como C3-G e C5-G, o LB não conseguiu melhorar significativamente a solução quando comparado ao CPLEX MIPSTART, indicando que o desempenho do método pode variar dependendo do problema específico.

O CPLEX sem MIPSTART não conseguiu encontrar soluções dentro do limite de 3600 segundos em nenhuma das instâncias, o que indica que essa abordagem não é eficiente para a resolução do problema. A incapacidade do solucionador de convergir dentro do tempo limite sugere que ele precisa de um ponto de partida mais refinado para melhorar sua performance. Isso reforça a importância de métodos heurísticos como o *Local Branching* e a inicialização via MIPSTART, que permitem ao CPLEX explorar o espaço de busca de maneira mais eficiente.

O CPLEX com MIPSTART apresenta um equilíbrio entre tempo de execução e qualidade da solução. Em geral, ele encontra soluções melhores que a Solução de Referência e que o CPLEX sem MIPSTART, mas, em algumas instâncias, fica atrás do LB. Em 10 casos, o CPLEX com MIPSTART obtém valores de FO próximos aos do

LB, mas com um tempo de execução ligeiramente menor. Por exemplo, na instância C1-G, o LB encontra uma FO de 6.560.465, enquanto o CPLEX MIPSTART obtém 6.431.551, uma diferença pequena.

Entretanto, em 4 situações, o CPLEX com MIPSTART supera o LB. Na instância C3-G, por exemplo, o CPLEX com MIPSTART encontrou uma FO de 5932477, enquanto o LB ficou em 6031051, mostrando que o CPLEX pode, em certos casos, otimizar melhor a solução. Outro aspecto interessante é que, enquanto o LB pode ter tempos de execução muito altos, o CPLEX MIPSTART mantém um tempo mais consistente, geralmente inferior ao LB em algumas instâncias, como em C2-G, em que o LB leva 3597,67 segundos e o CPLEX MIPSTART apenas 3100,19 segundos.

Os *gaps* apresentados na última coluna mostram a variação percentual do FO entre o *Local Branching* e o CPLEX (Gap_F), bem como a variação do tempo de execução entre os dois métodos (Gap_T). O Gap_F é majoritariamente negativo, indicando que o LB reduz significativamente o valor do FO em relação ao CPLEX. A maior melhoria ocorre em C1-P, em que o FO é reduzida em impressionantes $-100{,}07\%$. O Gap_T varia bastante. Em algumas instâncias, como C3-G ($-85{,}12\%$), o tempo do LB é muito menor que o da solução de referência, o que é uma exceção, pois, na maioria dos casos, o LB tem tempos de execução muito superiores. Esses valores indicam que, apesar do alto custo computacional, o LB é muito eficiente na redução da FO, tornando-o uma estratégia poderosa para otimização.

5.6.2 Solução para as classes de instâncias com 6 períodos de tempo

Nessa subseção são apresentados os resultados obtidos com a solução usando a matheurística *Local Branching* na Seção 5.6.2.1 e uma comparação da abordagem com o CPLEX, na seção 5.6.2.2.

5.6.2.1 Solução via Matheurística *Local Branching*

A Tabela 5.8 apresenta os resultados do valor médio de função objetivo obtidos a partir do método de *Local Branching* (LB). Os dados estão organizados por classe (C7 a C12), com a análise de diferentes componentes da função objetivo, como "Atraso", "Adiantamento", "Estoque", "*setup*" e "Objetos". Além disso, são fornecidos os valores da função objetivo (FO) e o tempo computacional necessário para a execução. A tabela abrange tanto os casos com 50 como com 100 tipos de itens por classe.

A coluna "Atraso" representa o custo associado ao não cumprimento dos prazos de entrega. Observa-se que, em diversas instâncias, há valores significativos de atraso, o que indica a dificuldade de se atender a demanda no tempo estipulado com as instâncias propostas. Isso pode ser mais evidente nas classes C7-P, C8-P e C10-P, nas quais os valores são elevados. O custo de atraso é um dos fatores mais elevados na função objetivo, indicando dificuldades na pontualidade das entregas. As instâncias (P) tendem a ter os maiores valores de atraso, como visto em C7-P (160485,3), C8-P (367269,2), e C10-P (256108,7). Isso sugere que a abordagem usada pode estar mais ajustada para as classes de comprimentos de itens M e G. Para algumas dessas instâncias, como C11-G (0 de atraso), o plano de corte fornecido deu conta de atender dentro do prazo os pedidos.

Tabela 5.8: Resultados da função objetivo da Solução Referência e do *Local Branching* para as classes com 6 períodos

		Termos da Função Objetivo - LB					LB	
Itens	Classe	Atraso	Adiantamento	Estoque	*Setup*	Objetos	FO	Tempo
50	C7-P	160485,30	0,00	9722,80	292,00	2212,50	18601250,00	3488,53
	C7-M	39848,70	0,00	10576,98	247,23	6643,32	14055710,00	3460,96
	C7-G	22602,20	0,00	5361,94	247,07	11005,25	14478260,00	3522,14
100	C8-P	367269,20	0,00	40376,00	600,00	4909,00	42581300,00	3499,09
	C8-M	79065,50	0,00	28960,64	378,82	14472,83	35241920,00	3474,58
	C8-G	88308,60	0,00	12584,15	467,51	24225,16	36184360,00	3525,22
50	C9-P	148207,20	0,00	12791,20	263,80	2384,90	18992520,00	3491,17
	C9-M	15790,50	0,00	12773,12	236,63	6599,28	11503674,00	3466,24
	C9-G	1142,00	0,00	6316,47	248,68	10782,42	12210890,00	3142,60
100	C10-P	256108,70	0,00	30916,50	420,10	4943,30	41669820,00	3513,10
	C10-M	43278,70	0,00	22245,26	328,77	14589,37	31972260,00	3543,41
	C10-G	40790,70	0,00	7972,13	435,22	24207,50	31073040,00	3589,16
50	C11-P	155912,00	0,00	11340,30	300,40	2284,30	18240570,00	3490,67
	C11-M	10356,90	0,00	9007,84	278,25	6652,13	11052232,00	3397,43
	C11-G	0,00	0,00	5413,52	276,19	10621,30	11781340,00	2626,10
100	C12-P	338153,20	0,00	45275,30	599,90	4949,80	39725050,00	3501,15
	C12-M	23219,10	160,29	27729,66	504,81	14511,50	24656860,00	3533,92
	C12-G	33,20	246,13	9483,27	600,00	24352,91	27300500,00	3587,05
	Média	99476,21	22,58	17158,17	373,63	10574,82	24517864,22	3436,25

A coluna "Adiantamento" reflete os custos relacionados à entrega antecipada de pedidos. Na maioria dos casos, o valor é zero ou próximo de zero, o que sugere que o modelo não prioriza a antecipação de entregas, focando mais no cumprimento exato dos prazos ou lidando com atrasos, exceto por valores pequenos em C12-M (160,29) e C12-G (246,13). Isso sugere que o modelo prioriza atender a demanda no prazo exato ou com atraso, em vez de antecipar entregas.

A coluna "Estoque" representa os custos de manutenção de itens produzidos que ainda não foram entregues. Classes como C7-P, C8-P e C12-P apresentam estoques mais elevados, indicando que, para minimizar atrasos, há uma necessidade de produção antecipada, resultando em custos adicionais de armazenamento. Em alguns casos, como C8-P (40376) e C12-P (45275,3), há estoques elevados, o que pode indicar que o modelo está tentando minimizar atrasos produzindo antecipadamente.

A coluna "*Setup*" mostra os custos decorrentes de *setup* dependente da sequência de produção, e seu impacto pode ser visto em diferentes classes. Notavelmente, classes com maior número de itens e maior complexidade tendem a apresentar valores de *setup* mais elevados, como C12-G e C8-G, evidenciando a necessidade de um planejamento eficiente para minimizar essas transições. Instâncias como C8-G (467,51) e C12-G (600) apresentam *setups* mais altos, o que sugere maior complexidade na organização das sequências de corte.

A coluna "Objetos" indica a quantidade de recursos (como barras ou rolos de material) utilizados no processo de corte. Valores elevados podem indicar maior desperdício ou necessidade de mais matéria-prima para atender à demanda. A variação desse termo entre diferentes classes sugere que alguns problemas requerem maior eficiência no uso de material. Em instâncias como C12-G (24352,91) há um uso maior de objetos, o que pode indicar maior desperdício ou dificuldade em encaixar os cortes de forma eficiente.

A coluna "FO" representa o valor total otimizado da função objetivo, que é impactado pelos componentes acima. Observa-se que as instâncias de maior porte (C12-P,

C10-P, etc.) possuem valores significativamente altos, refletindo a complexidade e o custo do problema quando há um grande número de itens a serem processados. Já a coluna "Tempo" mostra que o tempo de processamento para encontrar a solução ótima pelo *Local Branching* está na intervalo de 2600 a 3600 segundos, o que indica que a complexidade computacional aumenta para casos maiores, como nas classes C12-G e C12-M.

A análise dos componentes da função objetivo evidencia a importância de um equilíbrio entre os diferentes fatores para garantir eficiência no processo de corte de estoque. O alto custo de atraso em várias classes sugere que estratégias para melhor planejamento da produção podem ser necessárias. Além disso, a variação nos custos de *setup* e estoque indica a relevância da otimização da sequência de produção para reduzir gastos operacionais.

Vejamos então algumas comparações entre as classes P, M, G. As classes P apresentam os maiores custos gerais da função objetivo, devido ao alto impacto do atraso. As classes M mostram custos mais balanceados, com destaque para o estoque. O tempo de execução do método LB varia pouco entre as classes, permanecendo na intervalo de 2600 a 3600 segundos.

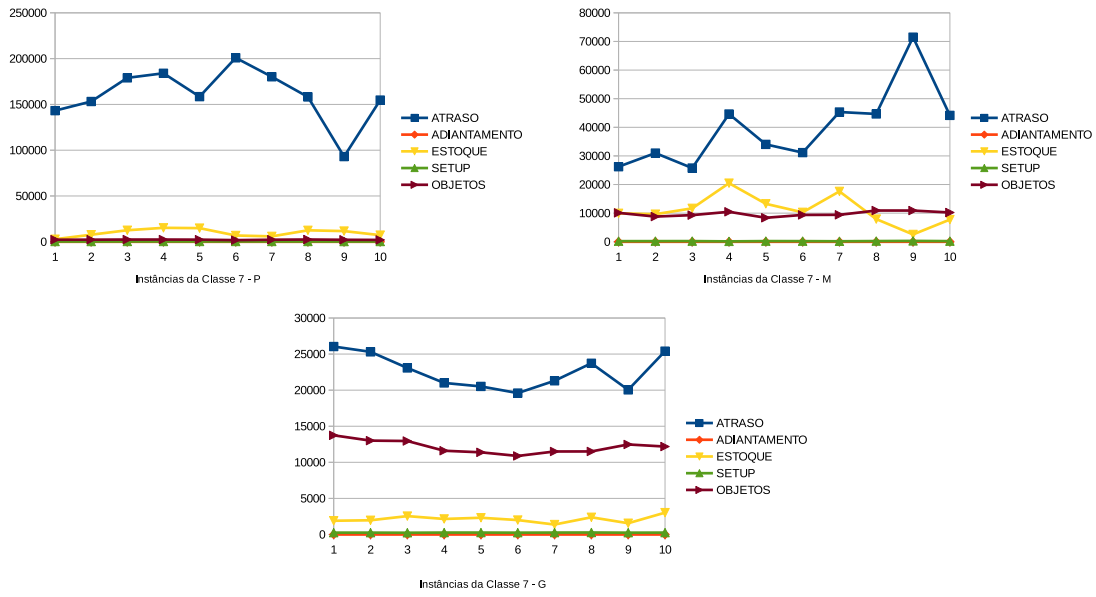


Figura 5.9: Comparação dos valores das componentes da função obejtivo para as classe C7-P, C7-M e C7-G. Fonte: Própria do Autor.

A Figura 5.9 descreve os valores de cada um dos componentes da função objetivo para as classes C7-P, C7-M e C7-G, detalhando os resultados obtidos para cada uma das 10 instâncias dentro dessas classes. Ao analisar os dados, observa-se que o fator predominante entre os componentes da função objetivo é o atraso dos pedidos, que se destaca significativamente em relação aos demais termos. Em contraste, os valores relativos aos termos de adiantamento, estoque e *setup* apresentam-se mais próximos entre si, sem grandes variações. No entanto, na classe C7-G, verifica-se um comportamento diferente, pois o valor associado ao número de objetos começa a se distanciar dos outros termos da função objetivo, mostrando um crescimento

mais expressivo. Este fenômeno sugere que, nesta classe específica, a quantidade de objetos tem um impacto mais relevante nos resultados, superando a influência dos termos de *setup*, estoque e adiantamento. Essa observação destaca as particularidades do modelo para cada classe de instância, indicando como diferentes componentes da função objetivo podem ter pesos variados conforme as configurações específicas dos problemas.

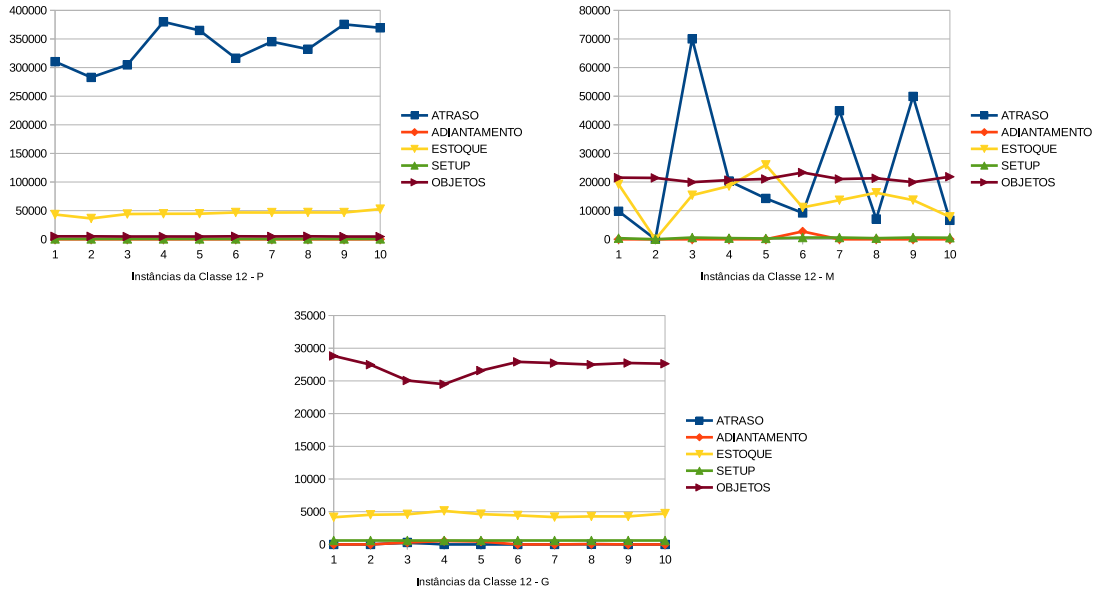


Figura 5.10: Comparação dos valores das componentes da função objetivo para as classe C12-P, C12-M e C12-G. Fonte: Própria do Autor.

A Figura 5.10 apresenta os valores de cada uma das componentes da função objetivo para a classe C12-P, C12-M e C12-G, descrevendo os valores para cada uma das 10 instâncias dessa classe. Percebe-se que o fator dominante é de fato o atraso dos pedidos, enquanto os outros termos ficam mais próximos. Apenas na classe C12-G que o valor do número de objetos começa a crescer e se distanciar um pouco mais do *setup*, estoque e adiantamento.

5.6.2.2 Comparação da Abordagem Proposta com o CPLEX

A Tabela 5.9 compara diferentes abordagens para a resolução do problem em estudo com seis períodos, destacando a eficácia de cada método em relação aos valores da função objetivo (FO) e ao tempo de execução. As colunas principais incluem a Solução de Referência, a abordagem *Local Branching* (LB), o CPLEX sem MIPSTART e o CPLEX com MIPSTART. Além disso, as últimas colunas indicam os *gaps* de função objetivo (Gap_F) e *gaps* de tempo (Gap_T), permitindo avaliar a eficiência de cada abordagem.

A Solução de Referência representa um ponto de partida inicial para a resolução do problema. Seu principal destaque é a extrema rapidez no tempo de execução, sendo sempre inferior a 3 segundos. No entanto, a qualidade das soluções obtidas é notadamente inferior às geradas pelos outros métodos. Isso se evidencia ao comparar

Tabela 5.9: Resultados da Solução de Referência com a do *Local Branching* (LB) em comparação com o valor do CPLEX para as classes com 6 períodos.

		Solução de Referência		LB		CPLEX		CPLEX_Mip		*Gaps*	
Itens	Classe	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	Gap_F	Gap_T
50	C7-P	20130150,00	0,43	**18601250,00**	3488,53	–	3600,00	20075364	3401,60	-7,92	2,49
	C7-M	18747180,00	0,45	**14055710,00**	3460,96	–	3600,00	18646047,00	3400,03	-32,65	1,76
	C7-G	17638700,00	0,30	14478260,00	3522,14	–	3600,00	**13711429,00**	3430,64	5,29	2,59
100	C8-P	42749410,00	2,68	**42581300,00**	3499,09	–	3600,00	42690773,00	3600,00	-0,25	-2,88
	C8-M	3770952810,00	2,75	**35241920,00**	3474,58	–	3600,00	38666060,00	3600,00	-9,71	-3,62
	C8-G	3462063790,00	1,89	**36184360,00**	3525,22	–	3600,00	36380653,00	3598,44	-0,54	-2,07
50	C9-P	19673170,00	0,45	**18992520,00**	3491,17	–	3600,00	19552010,00	3148,88	-2,94	9,80
	C9-M	16710970,00	0,46	**11503674,00**	3466,24	–	3600,00	15374357,00	3142,25	-33,64	9,34
	C9-G	15096960,00	0,30	12210890,00	3142,60	–	3600,00	**12193232,00**	2699,83	0,14	14,08
100	C10-P	18678600,00	0,47	**41669820,00**	3513,10	–	3600,00	41715591,00	3600,00	-0,10	-2,50
	C10-M	13417660,00	0,48	**31972260,00**	3543,41	–	3600,00	33836676,00	3505,46	-5,83	1,07
	C10-G	31778730,00	1,94	**31073040,00**	3589,16	–	3600,00	31759391,00	3006,09	-2,20	16,24
50	C11-P	18678600,00	0,47	**18240570,00**	3490,67	–	3600,00	18534269,00	3599,51	-1,61	-3,11
	C11-M	13417660,00	0,48	**11052232,00**	3397,43	–	3600,00	12267560,00	2996,61	-10,99	11,79
	C11-G	16522340,00	0,30	**11781340,00**	2626,10	–	3600,00	11999785,00	1849,28	-1,85	29,58
100	C12-P	3823050400,00	2,66	**39725050,00**	3501,15	–	3600,00	39780510,00	3600,00	-0,13	-2,82
	C12-M	26046540,00	2,77	**24656860,00**	3533,92	–	3600,00	26073320,00	3003,39	-5,74	15,01
	C12-G	32600360,00	1,97	**27300500,00**	3587,05	–	3600,00	27303285,00	3513,87	-0,01	2,04
	Média	632108557,22	1,18	24517864,22	3436,25	–	3600,00	25586684,00	3260,88	-6,15	5,49

a função objetivo da solução de referência com a obtida pelo *Local Branching* e pelo CPLEX com MIPSTART, em que há reduções significativas nos valores da função objetivo, indicando que essa abordagem inicial não encontra soluções otimizadas, porém servem de um limitante superior para comparar a solução.

O *Local Branching* surge como uma técnica de refinamento para melhorar a solução de referência. Observa-se que, em quase todas as instâncias, essa abordagem reduz expressivamente a função objetivo, tornando as soluções mais eficientes. Por exemplo, na classe C7-M, a FO cai de 18747180 para 14055710, uma melhoria de 32,65%, o que demonstra o potencial do método para encontrar soluções de alta qualidade. Entretanto, essa melhoria vem acompanhada de um custo computacional elevado, já que o tempo de execução do LB permanece em torno de 3500 segundos para a maioria dos casos.

O CPLEX sem MIPSTART apresenta um desempenho questionável, uma vez que não consegue encontrar soluções dentro do tempo limite de 3600 segundos. Em todas as instâncias, o tempo indicado é 3600 segundos, sugerindo que o solucionador não consegue convergir de forma eficiente. Isso indica que, sem uma solução inicial de qualidade, o CPLEX pode enfrentar dificuldades para explorar o espaço de busca e alcançar bons resultados dentro do limite de tempo estipulado. Assim, essa abordagem se mostra ineficiente e pouco prática para a resolução desse problema.

Comparando com o CPLEX, percebe-se que o LB consegue, em 16 situações, encontrar melhores valores médios de FO que o CPLEX com MIPSTART. Em C7-P, por exemplo, o LB obteve 18601250, enquanto o CPLEX com MIPSTART ficou com 20075364, mostrando que o LB pode superar o CPLEX em determinadas circunstâncias. No entanto, há casos em que o CPLEX com MIPSTART encontra valores um pouco menores, como em C9-G, em que o LB obtém média igual a 12210890, enquanto o CPLEX com MIPSTART obtém 12193232, com uma diferença mínima de 0,14%.

A variante CPLEX com MIPSTART apresenta um desempenho mais consistente, sendo, na maioria dos casos, a melhor alternativa em termos de equilíbrio entre tempo e qualidade da solução. Quando comparado ao *Local Branching*, o CPLEX com MIPS-

TART encontra 2 soluções um pouco melhores ou muito próximas. Por exemplo, na classe C8-P, a FO do CPLEX MIPSTART é 42690773, ligeiramente superior à obtida pelo LB (42581300). Além disso, a eficiência do CPLEX com MIPSTART se reflete no tempo de execução, que, em 13 instâncias, é ligeiramente menor que o do LB. Isso indica que a estratégia de iniciar a busca com uma solução prévia melhora a convergência do algoritmo, reduzindo a necessidade de explorações extensivas do espaço de busca. Entretanto, há exceções. Em casos como C7-P, o *Local Branching* conseguiu um valor médio de FO significativamente melhor que o CPLEX MIPSTART, mostrando que há situações em que a abordagem heurística pode ser mais eficaz do que a estratégia baseada no *solver* CPLEX.

Os *gaps* apresentados na última coluna indicam a variação percentual entre as soluções. O Gap_F (*gap* da função objetivo) mostra a diferença da FO entre o LB e o CPLEX, enquanto o Gap_T (*gap* de tempo) reflete a variação de tempo entre essas abordagens. Em quase todos os casos, o Gap_F é negativo, reforçando que o LB consegue melhorar substancialmente a solução inicial. A maior redução ocorre em C9-M, com uma variação de $-33{,}64\%$ na FO, evidenciando a superioridade do LB sobre o CPLEX.

Por outro lado, o Gap_T é majoritariamente positivo, indicando que o LB exige muito mais tempo de execução que a Solução do CPLEX. Há exceções, como em C8-P ($-2{,}88\%$), em que o LB foi um pouco mais rápido. No entanto, em casos como C11-G (29,58%), o tempo do LB foi maior em 13 instâncias, sugerindo que, em algumas situações, essa abordagem pode ser inviável em termos computacionais.

A análise detalhada dos resultados permite destacar alguns pontos essenciais. A Solução de Referência é rápida, mas subótima – é útil como ponto de partida, mas suas soluções não são adequadas para problemas mais complexos. O *Local Branching* melhora significativamente o FO, mas seu tempo de execução é elevado, o que pode limitar sua aplicação prática. O CPLEX sem MIPSTART é ineficiente, pois não consegue resolver os problemas dentro do tempo limite, tornando essa abordagem inviável. O CPLEX com MIPSTART equilibra tempo e qualidade, sendo geralmente a melhor escolha para encontrar boas soluções em prazos razoáveis. Em alguns casos, o LB supera o CPLEX MIPSTART, mostrando que sua estratégia heurística pode ser útil para refinar soluções.

5.7 Conclusões e Discussões Parciais

Nesse capítulo trabalhamos com o modelo que atende características como a demandas dos itens ocorrendo em períodos diversos, sendo possível antecipar ou não a produção e a inserção dos custos de estoques. Foi considerado que os itens que forem estocados podem ser usados nos próximos períodos, além desse valor ser penalizado na função objetivo.

Além dessas características, o fator de dependência na troca entre padrões aumentou a dificuldade do problema de corte, uma vez que o CPLEX com as configurações próprias, no tempo estabelecido já não conseguiu encontrar uma solução. Uma vez que foi necessário a ajuda de uma solução usada como MIPSTART para gerar uma solução de referência, para que fosse possível encontrar novas soluções incumbentes e melhorar a solução inicial fornecida. A solução inicial proposta e a mesma fornecida

no MIPSTART, ou seja, o CPLEX e a matheurística saem de uma mesma solução de referência.

A Solução de Referência é obtida de forma muito rápida, mas suas soluções são extremamente ruins e não são úteis para problemas complexos. O *Local Branching* encontra soluções de alta qualidade, mas seu tempo de execução é muito alto, tornando-o um método caro. O CPLEX sem MIPSTART não é eficiente, pois não consegue resolver os problemas dentro do tempo limite. O LB supera o CPLEX com MIPSTART em algumas instâncias, mas em outras o CPLEX pode encontrar soluções melhores ou equivalentes.

Observa-se que, na maioria das classes, a média dos resultados obtidos pelo *Local Branching* é inferior à média apresentada pelo CPLEX. Como se trata de um problema de minimização, isso indica que o *Local Branching* encontrou soluções de melhor qualidade. No entanto, é importante destacar que os tempos computacionais ficaram próximos ao limite estabelecido de 3600 segundos na maioria das classes, o que pode impactar a viabilidade prática da abordagem.

Um fator importante a ser destacado é a influência do termo de atraso na função objetivo, uma vez que esses valores são penalizados, e sendo altos afeta muito o valor final. No entanto, apenas as classes C5-M, C5-G, C6-G e C11-G apresentaram atraso igual a zero em todas as instâncias analisadas. Isso ocorre especificamente nas datas D3, que são consideradas largas, ou seja, possuem um tempo de produção maior. Esse tempo adicional facilitou o processo produtivo dessas classes. Além disso, todas essas classes correspondem a itens em estoque classificados como "grandes" e ocorrem em três períodos, com exceção da classe C11-G, que se estende por seis períodos. Outro ponto relevante é que, para todas essas classes, além do atraso ter sido nulo, o adiantamento também foi igual a zero, indicando um equilíbrio no planejamento da produção.

Como proposta de aprimoramento no método de solução proposto, uma ideia seria aprimorar a matheurística *Local Branching*, podendo adicionar uma busca local ou algum outro tipo de busca para que ela seja capaz de melhorar a solução de referência inicial.

Capítulo 6

Comparação entre as Modelagens Propostas

Este capítulo apresenta uma análise comparativa entre três modelagens distintas desenvolvidas para o Problema Multiperíodo de Corte de Estoque Unidimensional (1D-PMCE), denominadas 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD. Cada uma dessas modelagens incorpora diferentes níveis de complexidade e detalhamento, ampliando progressivamente o escopo das restrições e dos objetivos considerados no processo de otimização. Embora as funções objetivo de cada modelo estejam expressas na mesma unidade de medida, possibilitando a comparação direta dos seus valores, é importante destacar que as modelagens mais completas incorporam termos adicionais que representam aspectos não contemplados nas versões anteriores. Tais termos abrangem fatores operacionais e estratégicos relevantes, como penalizações associadas à estoque, que contribuem para uma representação mais realista e abrangente do problema estudado.

Assim, esta análise não busca comparar os resultados do valor da função objetivo, mas sim avaliar qualitativamente as contribuições e implicações decorrentes da inclusão dos novos termos, destacando o gasto de tempo computacional entre cada uma delas. Logo, são detalhados os resultados obtidos para cada modelagem, bem como suas respectivas vantagens e limitações, proporcionando uma visão integrada que subsidia a escolha do modelo mais adequado para diferentes contextos de aplicação.

A Seção 6.1 apresenta uma análise das soluções para um exemplo numérico com as modelagens com *setup* independente e dependente da sequência com os itens considerados P e G com relação ao objeto em estoque (Subseção 6.1.1), além de apresentar uma análise dos resultados da influência do *setup* independente da sequência e do dependente da sequência para itens das classes P, M e G (Subseção 6.1.2). Na Seção 6.3 são apresentados os resultados para os testes com custos iguais a 1 em todas as modelagens. Além disso, trabalhamos com o modelo com tempo ocioso. O tempo ocioso da máquina refere-se ao período em que a máquina não está operando ou produzindo, podendo ser causado por diversos fatores. A Seção 6.4 apresenta esses resultados para custos variados. Essa separação será realizada para verificar as diferenças entre as modelagens propostas e o impacto de diferentes custos. Para conduzir uma análise comparativa entre as modelagens apresentadas, cujos resultados são detalhados nas tabelas subsequentes, é fundamental estabelecer uma nomenclatura clara para cada uma delas:

(a) 1D-PMCE-DE: refere-se ao modelo que considera apenas as penalizações associadas ao atraso e ao adiantamento na entrega dos itens, conforme descrito no Capítulo 3;

(b) 1D-PMCE-DE-SI: corresponde ao modelo que, além das penalizações mencionadas, incorpora uma restrição de estoque e *setup* independente da sequência, sendo detalhado no Capítulo 4;

(c) 1D-PMCE-DE-SD: representa o modelo mais abrangente, que adiciona, além das penalizações de atraso e adiantamento, a penalização do estoque e *setup* dependente da sequência, conforme exposto no Capítulo 5.

6.1 Comparação entre as modelagens com *setup*

Esta seção apresenta um exemplo numérico para as modelagens do Problema Multiperído de Corte de Estoque Unidimensional com Datas de Entrega, tendo *setup* independente da sequencia (1D-PMCE-DE-SI) e tendo *setup* dependente da sequência de corte (1D-PMCE-DE-SD), descritos na Subseção 6.1.1. Na Subseção 6.1.2 é feita uma análise dos resultados obtidos a partir das modelagens propostas com *setup* para os itens variando entre P, M e G.

6.1.1 Exemplo Numérico

Exemplo 4. *Suponha que temos objetos em estoque disponíveis de comprimento* $L = 100$ *para atender aos tipos de itens dados na Tabela 6.1.*

A Tabela 6.1 mostra as solicitações da empresa para os pedidos com os tipos de itens considerados P e G em relação ao objeto em estoque de comprimento $L = 100$*. O pedido deve ser entregue em 26 dias.*

Tabela 6.1: Dados para análise de *setup* e data de entrega igual a 26 dias.

Comprimentos (P)	Demanda (P)	Comprimentos (G)	Demanda (G)
10	10	10	10
15	6	30	6
20	5	60	5

Nessa análise vamos verificar o impacto dos itens serem P ou G, e como o fator de dependência ou não de setup impacta no resultado final. Inicialmente, foram utilizados os padrões de corte homogêneos nos resultados.
Para os itens P, os padrões seriam:

(i) $(10, 0, 0)^t$,

(ii) $(0, 6, 0)^t$,

(iii) $(0, 0, 5)^t$.

Para os itens G, os padrões seriam:

(i) $(10, 0, 0)^t$,

(ii) $(0, 3, 0)^t$,

(iii) $(0, 0, 1)^t$.

Para o setup dependente da sequência, o custo é calculado pelas trocas que a máquina precisar fazer para sair de um padrão de corte para outro. A expressão 6.1 apresenta, para os padrões de corte utilizados, a matriz dos custos de setup para o Exemplo numérico 4 com os itens P, enquanto a expressão 6.2 apresenta a matriz de custos de setup para o mesmo exemplo numérico no caso de itens G.

$$\Gamma_P = \begin{bmatrix} - & SC_1 & SC_2 & SC_3 \\ SC_1 & 0 & 16 & 15 \\ SC_2 & 16 & 0 & 11 \\ SC_3 & 15 & 11 & 0 \end{bmatrix} \tag{6.1}$$

$$\Gamma_G = \begin{bmatrix} - & SC_1 & SC_2 & SC_3 \\ SC_1 & 0 & 13 & 11 \\ SC_2 & 13 & 0 & 4 \\ SC_3 & 11 & 4 & 0 \end{bmatrix} \tag{6.2}$$

A Tabela 6.2 apresenta esses resultados para a solução das modelagens para o caso de itens P. Nesta tabela, foram utilizadas as modelagens 1D-PMCE-DE-SI (Setup Independente) ou 1D-PMCE-DE-SD (Setup Dependente). A tabela mostra os padrões de corte utilizados. A coluna OB representa a quantidade de vezes que cada padrão de corte foi utilizado na solução; a coluna S é o custo de setup, associado à utilização daquele padrão de corte, que pode variar conforme o modelo; a coluna E mostra o valor de estoque; a coluna AT mostra os valores de Atraso Total; a coluna AD mostra os valores de Adiantamento na entrega; e a coluna FO apresenta os valores de função objetivo.

Esses experimentos utilizaram 3 padrões de corte diferentes (um para cada tipo de item), com custo de setup fixo (2 por padrão) para o 1D-PMCE-DE-SI. O valor de FO alcançou 26 unidades, com 15 unidades de tempo entregue antes da data de entrega (AD = 0), e sem atraso total (AT = 0). Por outro lado, para a modelagem 1D-PMCE-DE-SD, o custo de setup varia conforme a expressão 6.1. Apesar de não haver adiantamento (AD = 0), houve 9 unidades em atraso durante o processo (AT = 9), e o custo total (FO) foi mais alto (35). Esses resultados destacam o impacto da forma de modelagem do setup no custo total e no cumprimento dos prazos de entrega, evidenciando um trade-off entre eficiência econômica e pontualidade logística.

Tabela 6.2: Resultados para o exemplo numérico para os itens P.

	Padrão de Corte			Solução					
Modelagem	1	2	3	OB	S	E	AT	AD	FO
1D-PMCE-DE-SI	0	0	5	1	2	0	0	15	26
	0	6	0	1	2	0			
	10	0	0	1	2	0			
1D-PMCE-DE-SD	0	0	5	1	5	0	9	0	35
	0	6	0	1	11	0			
	10	0	0	1	16	0			

A Tabela 6.3 apresenta os resultados para a solução das modelagens para o caso dos itens G, com as mesmas descrições das colunas já feitas para a Tabela 6.2.

Tabela 6.3: Resultados para o exemplo numérico para os itens G.

	Padrão de Corte			Solução					
Modelagem	1	2	3	OB	Custo de *setup*	E	AT	AD	FO
1D-PMCE-DE-SI	0	0	1	5	2	0	0	12	26
	0	3	0	2	2	0			
	10	0	0	1	2	0			
1D-PMCE-DE-SD	0	0	1	5	1	0	0	0	26
	0	3	0	2	4	0			
	10	0	0	1	13	0			

Na modelagem 1D-PMCE-DE-SI, o custo total da solução (FO = 26) resulta da aplicação de três padrões de corte com setups fixos, totalizando cinco ordens de corte no padrão de corte 3, duas no padrão de corte 2 e uma no padrão de corte 1, com custo de setup constante igual a 2 por padrão. Apesar de não haver atraso total (AT = 0), houve 12 unidades com adiantamento em relação à data de entrega (AD = 12). Por outro lado, na modelagem 1D-PMCE-DE-SD, o custo total também é de 26, porém sem nenhum tipo de atraso ou adiantamento (AT = 0 e AD = 0). Isso ocorreu pois o custo de setup é mais elevado e variável conforme a quantidade: 1, 4 e 13 unidades, respectivamente, para os padrões 3, 2 e 1.

Esses resultados indicam que, para os itens G, ambas as modelagens foram capazes de atingir o mesmo valor da função objetivo, porém com comportamentos distintos quanto à pontualidade e à distribuição dos custos de setup. Enquanto o modelo com setup dependente favorece a pontualidade das entregas. Dessa forma, a escolha da modelagem mais adequada dependerá, novamente, das prioridades estratégicas do tomador de decisão, como o foco em custo total ou cumprimento rigoroso de prazos.

Ao comparar os resultados obtidos para os itens P e G, percebe-se que a modelagem 1D-PMCE-DE-SD (com setup dependente) teve comportamentos distintos em cada cenário. No caso dos itens P, foram utilizados apenas 3 objetos de corte (ou seja, 3 padrões diferentes), mas ocorreu um atraso total de 9 unidades, ainda que nenhuma tenha sido entregue fora do prazo final (AD = 0). Já para os itens G, foram necessários 8 objetos de corte, um número significativamente maior, mas, em contrapartida, não houve nenhum tipo de atraso ou adiantamento (AT = 0 e AD = 0).

Essa diferença pode ser explicada, sobretudo, pelo impacto do setup associado à troca de facas entre os padrões de corte. Os itens P, por apresentarem maior variedade de tamanhos menores, permitem composições mais flexíveis nos padrões de corte, possibilitando a utilização de um número reduzido de objetos. No entanto, essa flexibilidade cobra um preço. A produção desses itens exige mais trocas frequentes de ferramentas (facas) para atender às pequenas variações de dimensão, o que gera um custo elevado de setup na modelagem 1D-PMCE-DE-SD.

Já no caso dos itens G, os padrões de corte foram mais restritos, o que reduz a necessidade de trocas frequentes de facas. Embora isso leve a um aumento no número de objetos utilizados (um total de 8 objetos na modelagem 1D-PMCE-DE-SD), o

modelo consegue absorver melhor o custo de setup proporcional. Isso ocorre porque os cortes para itens G produzem volumes maiores em cada padrão, o que dilui o custo de setup por unidade e permite uma operação mais eficiente mesmo com maior número de ordens de corte.

Essa análise evidencia que a eficiência da solução está diretamente relacionada ao perfil de produção. Em contextos com muitos itens P e variados, a gestão de setups deve ser cuidadosamente balanceada, pois a economia de cortes pode causar atrasos indesejados. Em contrapartida, para itens G e padronizados, mesmo com mais objetos, é possível um desempenho satisfatório com menor impacto nos prazos.

6.1.2 Análise da influência dos itens P, M e G nas modelagens propostas

Nessa seção será feita uma análise dos resultados obtidos a partir das modelagens com *setup* independente da sequência (1D-PMCE-DE-SI) e *setup* dependente da sequência (1D-PMCE-DE-SD). Essa análise evidencia diferenças significativas no desempenho operacional e no atendimento da demanda, no caso do *setup* independente.

Por outro lado, quando o *setup* é dependente da sequência, ou seja, o custo e o tempo de troca de facas variam conforme a transição entre os padrões utilizados, o modelo tende a buscar uma ordenação que minimize essas trocas, o que pode levar a um aumento no número total de objetos utilizados e, eventualmente, a atrasos na entrega de itens com maior variedade de demandas.

A Tabela 6.4 apresenta os diferentes tipos de itens (1 até 10), classificados de acordo com seus respectivos comprimentos e agrupados em três categorias de tamanho: P, M e G. Essa estrutura de dados é útil para realizar testes e simulações em problemas de corte de estoque, permitindo observar como diferentes distribuições e variações nos comprimentos influenciam o desempenho dos modelos de otimização, especialmente no contexto de *setups*.

Tabela 6.4: Tipos de Itens e Comprimentos

Tipo de Item	1	2	3	4	5	6	7	8	9	10
Comprimento - P	16	4	3	-	-	-	-	-	-	-
Comprimento - M	28	11	10	-	-	-	-	-	-	-
Comprimento - G	65	24	14	-	-	-	-	-	-	-
Comprimento - P1	16	11	9	4	3	-	-	-	-	-
Comprimento - M1	60	51	28	11	10	-	-	-	-	-
Comprimento - G1	65	55	45	24	14	-	-	-	-	-
Comprimento - P2	18	16	12	11	10	9	8	4	3	2
Comprimento - M2	60	53	51	41	28	20	17	14	11	10
Comprimento - G2	66	65	55	45	44	29	24	15	9	3

A Tabela 6.5 apresenta os resultados do modelo com *setup* dependente da sequência (SD) nas três configurações de instâncias (P, M, G). É possível notar tendências importantes sobre o comportamento do modelo conforme a complexidade e quantidade de itens aumentam.

Tabela 6.5: Comparação das soluções para os modelos SD e SI usando os dados da Tabela 6.4: valores da função objetivo (FO), de termos da função objetivo e do tempo computacional.

Modelo	Instância	Objetos	*Setup*	Estoque	Atraso	Adiantamento	FO	Tempo(s)
SD	P	32	7	199	782	0	12372,3	0,62
	M	74	9	99	307	0	10753,6	0,27
	G	166	9	23	134	0	18046,2	0,43
SI	P	33	5	372	72	0	4040,0	0,91
	M	72	5	123	44	0	7664,0	1,26
	G	158	7	12	41	0	16246,0	0,55
SD	P1	82	11	453	987	0	19845,1	821,67
	M1	332	17	244	10	0	34003,0	0,37
	G1	404	14	135	249	0	43174,6	0,27
SI	P1	81	8	333	132	0	9464,0	15,97
	M1	326	5	911	0	0	32624,0	12,56
	G1	404	9	263	157	0	42018,0	7,64
SD	P2	168	20	761	1358	0	33445,2	3600
	M2	663	27	147	41	0	67157,0	29,31
	G2	758	25	262	151	0	77732,8	22,17
SI	P2	168	14	1087	160	0	18464,0	6,15
	M2	603	11	818	15	0	60514,0	49,65
	G2	669	12	779	0	0	66960,0	1354,64

Há um crescimento progressivo no número de objetos utilizados, especialmente notável nas instâncias P2, M2 e G2 (168, 663 e 758 objetos, respectivamente). Os valores de *setups* também crescem proporcionalmente à diversidade de padrões de corte: de 7 *setups* em P até 27 *setups* em M2, indicando uma maior exigência na organização das sequências. O estoque diminui à medida que se altera de P para G. Contudo, o atraso é maior nas instâncias P, especialmente em P2 (1358 unidades), o que indica que alocar comprimentos do tipo P de forma eficiente impacta diretamente a solução, uma vez que as trocas que ocorrem entre um padrão de corte e outro dependem do número de mudanças que a máquina precisa realizar. O tempo computacional varia, com destaque para o tempo limite alcançado na instância P2 (3600 segundos), evidenciando que a modelagem com *setup* dependente da sequência (SD) se torna computacionalmente mais custosa.

O modelo com *setup* independente da sequência (SI), por outro lado, apresenta comportamento distinto, favorecendo flexibilidade e, como consequência, melhoria de alguns indicadores operacionais. O número de objetos tende a ser menor que no modelo SD, principalmente em G2 (669 objetos contra 758 do SD), o que mostra um aproveitamento mais eficiente do material. O número de *setups* é inferior, reforçando a vantagem operacional da independência da sequência no agendamento dos cortes. O estoque, entretanto, é mais elevado, especialmente em P2 e G2 (1087 e 779 unidades), uma vez que ele não está sendo penalizado na função objetivo. O atraso é significativamente menor, com destaque para a eliminação total do atraso em G2 (zero unidades), o que mostra que a flexibilidade do modelo SI favorece o atendimento pontual.

A função objetivo é consistentemente inferior nas instâncias iniciais (P, M, G), mas a vantagem se reduz nas instâncias com maior número de itens, como M2 e G2. O tempo de solução cresce substancialmente nessas instâncias, com destaque para G2 (1354,64 segundos), evidenciando a complexidade computacional associada

à flexibilidade quando o número de itens aumenta.

No modelo SD (*Setup* dependente), cada troca de padrão de corte envolve ajustes manuais na configuração da máquina, especialmente a troca de facas. A complexidade dessas trocas pode ser medida pela diferença entre o padrão atual e o próximo, como apresentado na Seção 5.1. Os valores de *setup* crescem conforme a complexidade do reconfiguramento dos padrões de corte. Os itens do tipo P têm dimensões curtas e grande variedade, especialmente na instância P2 (10 tipos, com comprimentos de 2 a 18). Essa característica gera um maior número de padrões de corte diferentes para atender à demanda.

As instâncias do tipo G (como G2), embora também tenham variedade, lidam com itens maiores, o que reduz o número de itens por objeto (menos padrões possíveis), facilita o agrupamento dos pedidos em menor número de configurações diferentes e diminui a necessidade de reconfigurar a máquina frequentemente. Assim, o número de *setups* ainda é relevante (25 *setups* em G2 para modelo SD), mas os atrasos são muito menores (151 unidades em atraso), pois o modelo consegue organizar os padrões com mais eficiência e menor número de trocas.

Conclui-se que a modelagem com *setup* dependente da sequência revela que o tipo de item influencia diretamente na viabilidade operacional. Itens do tipo P, por exigirem maior diversidade de padrões de corte e mais ajustes manuais entre eles, encarecem e limitam a operação quando a troca de facas é considerada como parte do tempo de *setup*. Isso aumenta os valores de atraso e mostra que, embora se usem menor número de objetos, o custo de tempo computacional e de atendimento pode ser muito maior. Essa análise reforça a importância de considerar não apenas o número de padrões de corte ou de objetos, mas o custo de transição entre eles, especialmente quando o *setup* reflete processos físicos, como a troca de facas em máquinas de corte.

6.2 Dados para a comparação entre as três modelagens propostas: análise da inclusão de termos

Para aprofundar a comparação entre as modelagens propostas, principalmente envolvendo a inclusão de novos termos nas formulações apresentadas, gerou-se um conjunto de 10 instâncias a serem utilizados na análise a segui, apresentada nas Seções 6.3 e 6.4. Essa geraçãoo de instâncias seguiu um critério sistemático, visando garantir a diversidade e representatividade das instâncias testadas. Cada instância foi formulada de maneira estruturada, considerando diferentes configurações e parâmetros relevantes ao contexto do estudo. A Tabela 6.6 apresenta os dados das instâncias geradas.

Tabela 6.6: Geração de Dados para a comparação das Modelagens

Parâmetro	Valor
Comprimento do objeto	1000
Número de tipos de itens	20
Número de períodos de tempo	3
Comprimento dos itens	M $\in$ [0,01L; 0,8L]
Demanda	[0,125]
Data de entrega	D2

6.3 Resultados Computacionais para as instâncias para custos iguais a 1

A Tabela 6.7, juntamente com a Figura 6.2, apresentam as resultados obtidos para os instâncias geradas ssegundo os dados da Tabela 6.6, considerando todos os custos e tempos de cortes iguais a 1. A tabela mostra uma comparação entre os resultados para cada modelo estudado nessa tese (1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD) para a classe com 20 tipos de itens. A tabela descreve também cada um dos termos da função objetivo, ou seja, valores de atraso, adiantamento, estoque, *setup* e número de objetos cortados. Já a Figura 6.2 mostra os valores dos tempos computacionais para a solução das formulações.

Entre as dez instâncias analisadas, verificou-se que apenas as instâncias 2-M, 4-M, 5-M e 6-M apresentaram atrasos na entrega dos itens. Dentre essas, a instância 5-M mostrou-se particularmente crítica, pois, independentemente do modelo utilizado, os pedidos foram entregues com atraso em todas as configurações testadas. Em relação ao adiantamento das entregas, o modelo 1D-PMCE-DE evidenciou um comportamento mais acentuado, apresentando sete instâncias com entregas antecipadas. Já no modelo 1D-PMCE-DE-SD, o fenômeno do adiantamento ocorreu de forma significativamente reduzida, sendo registrado apenas na instância 1-M. Vale ressaltar que, no modelo 1D-PMCE-DE, não foram considerados os fatores de estoque de itens nem os custos associados ao *setup* de troca de padrões de corte, o que impacta diretamente os resultados obtidos.

As modelagens 1D-PMCE-DE-SI e 1D-PMCE-DE-SD incorporam restrições de balanço de estoque, permitindo uma melhor gestão dos itens armazenados ao longo do horizonte de planejamento. No entanto, observa-se uma diferença significativa entre essas duas abordagens: enquanto o modelo 1D-PMCE-DE-SI não penaliza explicitamente os itens mantidos em estoque na função objetivo, resultando em valores consideravelmente elevados para essa componente, o modelo 1D-PMCE-DE-SD introduz essa penalização, o que leva a uma redução substancial na quantidade de itens armazenados. Essa tendência é verificada de forma consistente em todas as dez instâncias analisadas, evidenciando o impacto direto da penalização do estoque na otimização do modelo.

No modelo 1D-PMCE-DE-SD, que considera *setups* dependentes da sequência, observou-se um maior número de padrões de corte utilizados. Isso ocorre porque o custo associado ao corte de um padrão de corte é influenciado pelo padrão de corte anterior, tornando a escolha dos padrões mais criteriosa e estratégica. Em contra-

Tabela 6.7: Comparação entre a evolução das modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD para a classe com 20 tipos de itens e penalizações de custos iguais a 1.

Dados		Termos da Função Objetivo				
Instância	Modelagem	Atraso	Adiantamento	Estoque	*setup*	Objetos
1-M	1D-PMCE-DE	0	226	–	–	2097
	1D-PMCE-DE-SI	0	0	1561	**42**	2010
	1D-PMCE-DE-SD	0	31	**16**	61	**1695**
2-M	1D-PMCE-DE	0	4	–	–	2689
	1D-PMCE-DE-SI	22	0	3068	**37**	2674
	1D-PMCE-DE-SD	0	0	**34**	64	**1955**
3-M	1D-PMCE-DE	0	168	–	–	2286
	1D-PMCE-DE-SI	0	0	4440	**42**	2365
	1D-PMCE-DE-SD	0	0	**11**	73	**1702**
4-M	1D-PMCE-DE	0	0	–	–	2023
	1D-PMCE-DE-SI	43	0	5454	**44**	2018
	1D-PMCE-DE-SD	0	0	**11**	65	**1676**
5-M	1D-PMCE-DE	1	0	–	–	1572
	1D-PMCE-DE-SI	127	0	6187	**37**	**1565**
	1D-PMCE-DE-SD	185	0	**88**	60	1632
6-M	1D-PMCE-DE	0	204	–	–	1862
	1D-PMCE-DE-SI	144	0	6582	**55**	1802
	1D-PMCE-DE-SD	0	0	**20**	64	**1415**
7-M	1D-PMCE-DE	0	0		–	–2862
	1D-PMCE-DE-SI	0	0	8421	**37**	2539
	1D-PMCE-DE-SD	0	0	**10**	65	**1755**
8-M	1D-PMCE-DE	0	356	–	–	2527
	1D-PMCE-DE-SI	0	0	10098	**37**	2257
	1D-PMCE-DE-SD	0	0	**20**	60	**1651**
9-M	1D-PMCE-DE	0	30	–	–	2200
	1D-PMCE-DE-SI	0	0	11659	**36**	2080
	1D-PMCE-DE-SD	0	0	**54**	63	**1716**
10-M	1D-PMCE-DE	0	158	–	–	1931
	1D-PMCE-DE-SI	0	0	13555	**33**	1639
	1D-PMCE-DE-SD	0	0	**31**	71	**1461**
	Média	17,40	39,10	3564	41,50	1988,53

partida, no modelo com *setup* independentes da sequência, esse custo é determinado isoladamente para cada padrão, sem influência de cortes anteriores, o que torna as decisões menos restritivas. Essa diferença impactou diretamente o número de trocas realizadas, como evidenciado na coluna de *setup* da tabela. Além disso, ao analisar a última coluna, nota-se que, em todas as instâncias, exceto na ins5-M, o modelo 1D-PMCE-DE-SD apresentou o menor número de objetos cortados, evidenciando uma maior eficiência no aproveitamento do material. Por outro lado, o modelo 1D-PMCE-DE, que não considera *setup* e restrições adicionais, foi a que resultou no maior número de objetos cortados em quase todos os casos analisados, exceto na instância 5-M.

A Figura 6.1 apresenta os valores da função objetivo para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD. A partir da análise dessa figura, observa-se que o modelo 1D-PMCE-DE-SD resultou no menor valor de função ob-

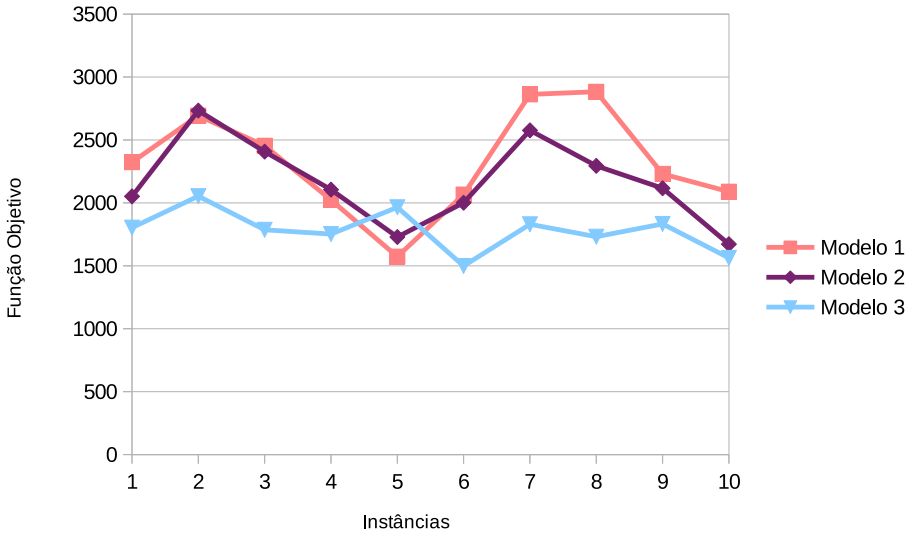


Figura 6.1: Comparação entre os valores das funções objetivo para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD com custos iguais a 1.

jetivo em nove dos dez instâncias analisados. A única exceção foi o instância 5, no qual o modelo 1D-PMCE-DE obteve um valor inferior, fato que pode ser explicado pelo alto nível de atraso na entrega dos itens, conforme indicado na Tabela 6.7. Já o modelo 1D-PMCE-DE-SI apresentou valores intermediários da função objetivo em todas as instâncias analisadas, demonstrando um comportamento entre as duas abordagens extremas. Esses resultados evidenciam a influência das diferentes estratégias de modelagem na minimização dos custos envolvidos no instância .

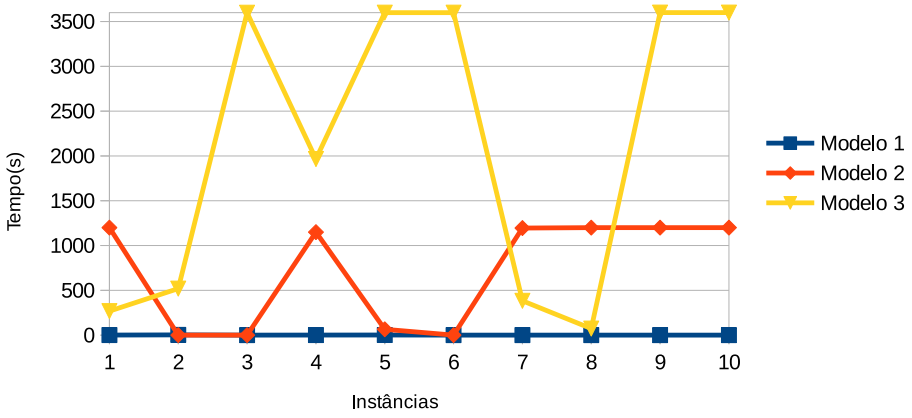


Figura 6.2: Comparação entre os tempos computacionais para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD com custos iguais a 1.

A Figura 6.2 apresenta os valores dos tempos computacionais consumidos para a resolução das instâncias, considerando as modelagens 1D-PMCE-DE, 1D-PMCE-DE-

SI e 1D-PMCE-DE-SD. A análise dessa figura evidencia que a adição de novos termos na função objetivo impactou diretamente o tempo de processamento, resultando em um aumento significativo. Esse efeito foi mais perceptível no modelo 1D-PMCE-DE-SD, na qual o tempo computacional atingiu o limite de 3600 segundos em 5 das 10 instâncias avaliadas. Em contraste, o modelo 1D-PMCE-DE demonstrou um desempenho computacional significativamente superior, com tempos de execução inferiores a 3 segundos em todas as instâncias analisadas. Esses resultados indicam que a complexidade introduzida pelas diferentes formulações influencia diretamente a eficiência da resolução dos instâncias.

6.4 Resultados Computacionais para as instâncias para custos variados

A Tabela 6.8, juntamente com as Figuras 6.3 e 6.4, apresentam os resultados obtidos para os instâncias escolhidas, considerando custos variados. Esta tabela apresenta os resultados para custos variados, na forma:

(i) custo de cortar um objeto = 1000;

(ii) custo de atraso e adiantamento = 100;

(iii) custo de estoque = $\lceil 0.1 \times L \rceil$;

(iv) tempo de cortar um objeto = 1;

(v) tempo de troca de *setup* independente = [1,10]; e

(vi) tempo e custos do *setup* dependentes, como descritos no Capítulo 5.

Na tabela são apresentados os resultados para cada uma dos modelos. Esses resultados estão descritos para os valores de custos variados, gerados de acordo com o descrito na Seção 5.5 para a Data D2. A tabela compara os três modelos (1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD) considerando cinco critérios principais:

(i) Atraso;

(ii) Adiantamento;

(iii) Estoque;

(iv) *Setup*; e

(v) Objetos cortados.

A seguir, analisamos detalhadamente cada um desses critérios e comparamos o desempenho das modelagens para diferentes instâncias.

O modelo 1D-PMCE-DE apresenta atraso zero na maioria dos casos, exceto na instância 6, na qual ocorre um pequeno atraso de 78. O modelo 1D-PMCE-DE-SI tem atrasos mínimos, mas não valores iguais a zero, como nas instâncias 2 (1 unidade

Tabela 6.8: Comparação da evolução das modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD para a classe com 20 tipos de itens e penalizações de custos variados.

Dados		Termos da Função Objetivo				
Instância	Modelagem	Atraso	Adiantamento	Estoque	*setup*	Objetos
1-M	1D-PMCE-DE	0	52	–	–	2323
	1D-PMCE-DE-SI	0	0	3453	**30**	**1626**
	1D-PMCE-DE-SD	143	0	**756**	63	1660
2-M	1D-PMCE-DE	0	207	–	–	2819
	1D-PMCE-DE-SI	1	0	5657	**26**	2098
	1D-PMCE-DE-SD	40	0	**1253**	69	**1840**
3-M	1D-PMCE-DE	0	20	–	–	2444
	1D-PMCE-DE-SI	0	0	8501	**33**	1965
	1D-PMCE-DE-SD	104	0	**660**	76	**1563**
4-M	1D-PMCE-DE	0	118	–	–	2135
	1D-PMCE-DE-SI	0	0	10317	**27**	**1552**
	1D-PMCE-DE-SD	0	0	**458**	69	1588
5-M	1D-PMCE-DE	0	176	–	–	1751
	1D-PMCE-DE-SI	6	0	11052	**42**	**1315**
	1D-PMCE-DE-SD	590	0	**783**	53	1581
6-M	1D-PMCE-DE	78	0	–	–	2003
	1D-PMCE-DE-SI	63	0	12853	**36**	1486
	1D-PMCE-DE-SD	134	0	**422**	73	**1375**
7-M	1D-PMCE-DE	0	50	–	–	2863
	1D-PMCE-DE-SI	0	0	15515	**26**	1988
	1D-PMCE-DE-SD	0	0	**618**	75	1607
8-M	1D-PMCE-DE	0	0	–	–	2883
	1D-PMCE-DE-SI	1	0	17623	**29**	2067
	1D-PMCE-DE-SD	56	0	**355**	86	**1607**
9-M	1D-PMCE-DE	0	137	–	–	2239
	1D-PMCE-DE-SI	0	0	20313	**20**	**1183**
	1D-PMCE-DE-SD	130	0	**816**	74	1642
10-M	1D-PMCE-DE	0	0	–	–	2089
	1D-PMCE-DE-SI	0	0	3013	**18**	**1612**
	1D-PMCE-DE-SD	143	0	**756**	63	1660
	Média	49,63	25,33	5758,70	49,25	1885,46

de atraso), 5 (6 unidades), 6 (63 unidades) e 8 (1 unidade). o modelo 1D-PMCE-DE-SD tem os maiores atrasos, com valores expressivos em algumas instâncias, como 590 (instância 5) e 143 (instância 10). Conclui-se que a 1D-PMCE-DE é o modelo mais eficiente para evitar atrasos, porém é o modelo que possui menor número de considerações em sua modelagem.

O modelo 1D-PMCE-DE tem altos níveis de adiantamento em algumas instâncias, como 207 (instância 2) e 176 (instância 5). A 1D-PMCE-DE-SI elimina completamente o adiantamento em todas as instâncias, sugerindo que esse modelo prioriza a entrega no prazo exato. A 1D-PMCE-DE-SD também reduz significativamente o adiantamento em relação a 1D-PMCE-DE, sendo zero em todas as instâncias analisadas. O modelo 1D-PMCE-DE tem um alto nível de adiantamento, enquanto 1D-PMCE-DE-SI e 1D-PMCE-DE-SD conseguem eliminá-lo. Se a entrega no prazo exato for essencial para reduzir custos, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD são superiores a

1D-PMCE-DE.

O modelo 1D-PMCE-DE não apresenta estoque. A 1D-PMCE-DE-SI apresenta os maiores valores de estoque em todas as instâncias, chegando a 20313 unidades (instância 9), 17623 unidades (instância 8) e 12853 unidades (instância 6). Isso indica que 1D-PMCE-DE-SI mantém altos estoques para minimizar outros custos, uma vez que esse termo não esta sendo penalizado na função objetivo desse modelo. A 1D-PMCE-DE-SD reduz drasticamente o estoque em relação a 1D-PMCE-DE-SI. Por exemplo, na instância 8, o estoque cai de 17623 (1D-PMCE-DE-SI) para apenas 355 (1D-PMCE-DE-SD). Na instância 9, a redução é de 20313 (1D-PMCE-DE-SI) para 816 (1D-PMCE-DE-SD). Conclui-se que o modelo 1D-PMCE-DE-SI tem os maiores estoques, o que pode ser problemático em ambientes com limitação de espaço ou custos elevados de armazenagem. O modelo 1D-PMCE-DE-SD consegue equilibrar melhor a necessidade de estoque, reduzindo significativamente os volumes sem comprometer tanto os outros critérios.

O modelo 1D-PMCE-DE não apresenta valores de *setup*. A 1D-PMCE-DE-SI tem os menores valores de *setup* em quase todas as instâncias, como 20 (instância 9), 18 (instância 10) e 26 (instância 2 e 7). Isso sugere que esse modelo prioriza a redução de *setups*, tornando a produção mais contínua. A 1D-PMCE-DE-SD tem valores de *setup* consideravelmente maiores que 1D-PMCE-DE-SI, como 73 (instância 6), 75 (instância 7) e 86 (instância 8). Isso indica que 1D-PMCE-DE-SD troca lotes de produção com mais frequência, uma vez que a escolha depende do ultimo corte que foi feito, e essa escolha é bem mais criteriosa.

A coluna 7 dessa tabela apresenta o número de objetos cortados. O modelo 1D-PMCE-DE tem os maiores custos totais em quase todas as instâncias, como 2863 (instância 7), 2444 (instância 3) e 2819 (instância 2). O modelo 1D-PMCE-DE-SI apresenta valores mais baixos que 1D-PMCE-DE, mas ainda altos em algumas instâncias. Por exemplo, na instância 9, 1D-PMCE-DE-SI reduz de 2239 (1D-PMCE-DE) para 1183, uma grande melhora. O modelo 1D-PMCE-DE-SD frequentemente tem o menor custo total, como nas instâncias 1, 2, 3, 4, 5, 6 e 8. Como exemplo, na instância 8, o modelo 1D-PMCE-DE-SD reduz o custo de 2883 (1D-PMCE-DE) para 1607, o que é uma queda significativa. Conclui-se que o modelo 1D-PMCE-DE-SD geralmente apresenta os menores custos totais, mostrando que esse modelo pode oferecer um equilíbrio melhor entre atraso, adiantamento, estoque e *setup*.

A Figura 6.3 apresenta os valores da função objetivo obtidos para as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD. A partir da análise dessa figura, observa-se que o modelo 1D-PMCE-DE obteve os menores resultados, sendo considerada o modelo mais básico neste estudo. Além disso, verifica-se que, em cinco das instâncias analisadas, o modelo 1D-PMCE-DE-SD apresentou valores inferiores aos do modelo 1D-PMCE-DE-SI, indicando um desempenho superior nesses casos específicos. Esse comportamento reforça a influência das características de cada modelo na qualidade das soluções encontradas.

A Figura 6.4 apresenta, de maneira detalhada, os tempos computacionais consumidos pelas modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD. A análise dessa figura demonstra que a inclusão sequencial de termos na função objetivo resultou em um aumento considerável no tempo de execução. Em três dos dez instâncias analisados no modelo 1D-PMCE-DE-SD, foi atingido o limite máximo de tempo, estipulado em 3600 segundos. Ao realizar uma comparação entre as modelagens, ob-

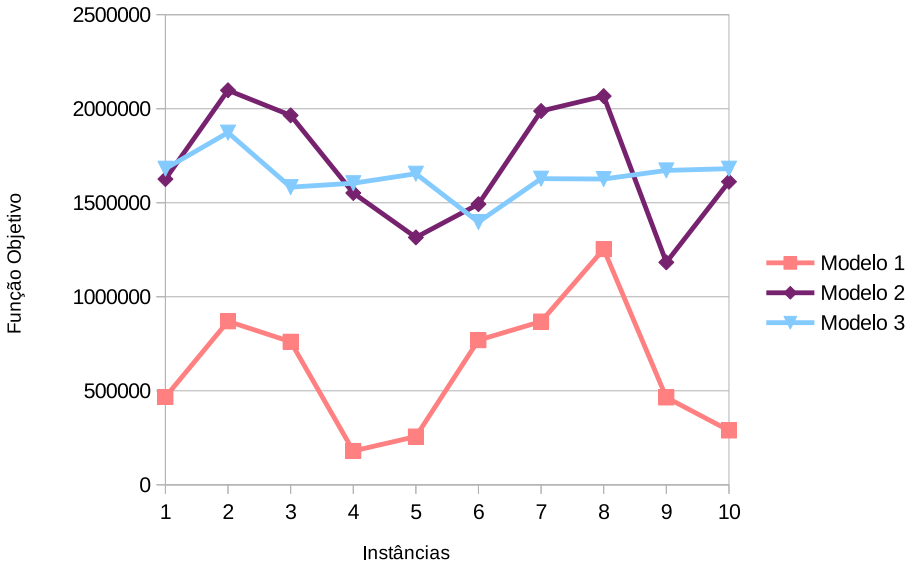


Figura 6.3: Comparação entre o valores de função objetivo para as modelagens com custos variados.

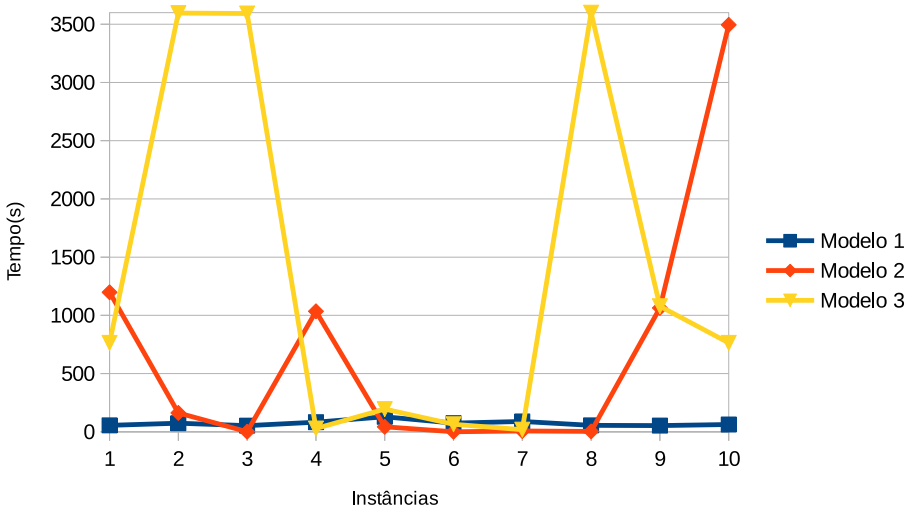


Figura 6.4: Comparação entre os valores de tempo de processamento para as modelagens com custos variados.

servamos que o modelo 1D-PMCE-DE, que se caracteriza como a mais simples dentre as três, obteve os menores tempos computacionais. Esse achado sugere uma relação direta entre a complexidade do modelo e o tempo necessário para sua resolução, evidenciando um crescimento progressivo no tempo de execução, conforme a modelagem se torna mais complexa. Assim, a introdução de cada novo termo na função objetivo, à medida que o modelo se torna mais sofisticado, demanda um aumento na utilização dos recursos computacionais, destacando o impacto significativo da complexidade no desempenho temporal das soluções propostas.

6.5 Conclusões Parciais

Neste capítulo apresentamos uma comparação entre as modelagens 1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD propostas nessa tese e descritas nas Seções 5.3, 5.4 e 5.5, respectivamente.

Na Seção 6.3, são apresentados os resultados provenientes das diferentes modelagens aplicadas à classe de problemas selecionada, com o objetivo de realizar uma análise dos desempenhos obtidos. O modelo 1D-PMCE-DE, considerado o mais simples neste estudo, contempla apenas as penalizações adicionais associadas ao atraso e ao adiantamento, o que lhe confere a capacidade de gerar soluções de maneira extremamente rápida. Dependendo dos custos atribuídos aos parâmetros trabalhados, essa modelagem pode, em certos casos, resultar em uma função objetivo com valor superior àquela gerada pelo modelo 1D-PMCE-DE-SD. Esta última, até o momento, é reconhecido como o modelo mais abrangente, uma vez que incorpora um maior número de termos em sua função objetivo, conferindo-lhe uma maior complexidade.

Observa-se que a inserção de termos relacionados ao estoque, juntamente com a penalização na função objetivo, resultou em uma redução substancial no número de itens estocados, um efeito que já era antecipado. O gerenciamento adequado do estoque de itens em um problema de corte desempenha um papel fundamental na minimização de custos, pois impacta diretamente tanto a eficiência na utilização dos materiais quanto os gastos operacionais. Um estoque excessivo pode acarretar custos elevados de armazenamento, incluindo despesas com espaço físico e manutenção. Portanto, um planejamento eficiente que equilibre a demanda com a disponibilidade de estoque torna-se essencial para a redução de custos e a otimização do processo produtivo.

A análise dos dados revela que as diferentes modelagens (1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD) exercem impactos distintos na função objetivo, variando conforme os critérios avaliados, como atraso, adiantamento, estoque, *setup* e número de objetos. O modelo 1D-PMCE-DE-SD resulta em menores estoque e número de objetos utilizados, ao mesmo tempo em que elimina ou reduz substancialmente os atrasos e adiantamentos. Embora o 1D-PMCE-DE-SI tenha apresentado um desempenho intermediário, com menor custo de *setup* em determinados casos, ele frequentemente resultou em custos elevados relacionados ao estoque. A 1D-PMCE-DE, por sua vez, embora simples, não considera o estoque nem o *setup*, o que pode levar a soluções menos otimizadas, caso o custo real destes componentes não seja nula. Assim, a inclusão de termos adicionais, como estoque e *setup*, no modelo 1D-PMCE-DE-SD permite um equilíbrio mais eficaz entre os diversos critérios, resultando em soluções mais econômicas e eficientes.

Em síntese, os resultados obtidos ao longo deste estudo evidenciam que a introdução de elementos como termos de *scheduling*, estoque e *setup* nas modelagens de corte tem um impacto substancial na eficiência operacional e nos custos totais. O modelo 1D-PMCE-DE-SD, ao incorporar uma maior quantidade de termos em sua função objetivo, se destacou por proporcionar soluções mais equilibradas, com menores custos de estoque e maior redução de atrasos e adiantamentos, demonstrando-se a mais eficiente para a maioria dos cenários analisados. Embora o modelo 1D-PMCE-DE, com sua simplicidade, apresente soluções rápidas, sua falta de consideração para o estoque e *setup* pode resultar em soluções subótimas. O modelo 1D-PMCE-DE-SI, por sua

vez, apresentou um desempenho intermediário, com um custo de *setup* mais baixo, mas frequentemente comprometeu a eficiência do estoque. Assim, pode-se concluir que, ao equilibrar adequadamente os diferentes critérios de otimização, como estoque e *setup*, é possível alcançar uma produção mais econômica, eficiente e alinhada às necessidades operacionais, reforçando a importância de um planejamento cuidadoso e da consideração de múltiplos fatores na otimização de processos.

Capítulo 7

Conclusões finais e Direções futuras

Neste capítulo são apresentadas as conclusões finais (Seção 7.1) e propostas de trabalhos futuros, mostrando os caminhos que esta pesquisa pode seguir a partir do presente tese (Seção 7.2).

7.1 Conclusões finais

O objetivo central desta tese foi a introdução de termos de tempo de atraso, tempo de adiantamento e tempo de *setup* no 1D-PCE, com o intuito de investigar de que maneira essa inserção influenciaria os resultados obtidos pelas diferentes modelagens. Observamos que a inclusão de tais termos na função objetivo do problema de corte unidimensional pode ter um impacto considerável na eficiência operacional, ao integrar de maneira mais eficaz aspectos relacionados ao tempo e à priorização dentro do processo de otimização. Ao incorporar variáveis como prazos de entrega, tempos de processamento e sequência de produção, torna-se possível alinhar o planejamento do corte às exigências da demanda e à disponibilidade de recursos, o que resulta na redução de atrasos e no aumento da produtividade. Esta abordagem não só permite otimizar a utilização das máquinas, como também otimizar a utilização dos materiais disponíveis, aprimorando simultaneamente o atendimento ao cliente. Contudo, a introdução desses termos acarreta um aumento na complexidade computacional do problema, o que exige o desenvolvimento de modelos matemáticos mais sofisticados e de algoritmos eficientes, capazes de fornecer soluções viáveis dentro de um intervalo de tempo aceitável.

Na Seção 6.3 são apresentados os resultados provenientes das diferentes modelagens aplicadas à classe de problemas selecionada, com o objetivo de realizar uma análise minuciosa dos desempenhos obtidos. A modelagem 1D-PMCE-DE, considerada a mais simples neste estudo, contempla apenas as penalizações adicionais associadas ao atraso e ao adiantamento, o que lhe confere a capacidade de gerar soluções de maneira extremamente rápida. Dependendo dos custos atribuídos aos parâmetros trabalhados, essa modelagem pode, em certos casos, resultar em uma função objetivo com valor superior àquela gerada pela modelagem 1D-PMCE-DE-SD. Esta última, até o momento, é reconhecida como a modelagem mais abrangente, uma vez que incorpora um maior número de termos em sua função objetivo, conferindo-lhe uma maior complexidade.

Observa-se que a inserção de termos relacionados ao estoque, juntamente com a penalização na função objetivo, resultou em uma redução substancial no número de itens estocados, um efeito que já era antecipado. O gerenciamento adequado do estoque de itens em um problema de corte unidimensional desempenha um papel fundamental na minimização de custos, pois impacta diretamente tanto a eficiência na utilização dos materiais quanto os gastos operacionais. Um estoque excessivo pode acarretar custos elevados de armazenamento, incluindo despesas com espaço físico e manutenção. Portanto, um planejamento eficiente que equilibre a demanda com a disponibilidade de estoque torna-se essencial para a redução de custos e a otimização do processo produtivo.

A análise dos dados revela que as diferentes modelagens (1D-PMCE-DE, 1D-PMCE-DE-SI e 1D-PMCE-DE-SD) exercem impactos distintos na função objetivo, variando conforme os critérios avaliados, como atraso, adiantamento, estoque, setup e número de objetos. A modelagem 1D-PMCE-DE-SD destaca-se como a mais eficiente na maioria dos cenários, apresentando custos significativamente menores relacionados ao estoque e ao número de objetos utilizados, ao mesmo tempo em que elimina ou reduz substancialmente os atrasos e adiantamentos. Embora a 1D-PMCE-DE-SI tenha apresentado um desempenho intermediário, com menor custo de setup em determinados casos, ela frequentemente resultou em estoques elevados. A 1D-PMCE-DE, por sua vez, embora simples, não considera o estoque nem o setup. Assim, a inclusão de termos adicionais, como estoque e *setup*, na modelagem 1D-PMCE-DE-SD permite um equilíbrio mais eficaz entre os diversos critérios, resultando em soluções mais econômicas e eficientes.

Em síntese, os resultados obtidos ao longo deste estudo evidenciam que a introdução de elementos como minimização de atraso, adiantamento, estoque e *setup* no 1D-PCE tem um impacto substancial na eficiência operacional e nos custos totais. A modelagem 1D-PMCE-DE-SD, ao incorporar uma maior quantidade de termos em sua função objetivo, se destacou por proporcionar soluções mais equilibradas, com menores custos de estoque e maior redução de atrasos e adiantamentos, demonstrando-se a mais eficiente para a maioria dos cenários analisados. Embora a modelagem 1D-PMCE-DE, com sua simplicidade, apresente soluções rápidas, sua falta de consideração para o estoque e setup pode resultar em soluções subótimas. A modelagem 1D-PMCE-DE-SI, por sua vez, apresentou um desempenho intermediário, com um custo de setup mais baixo, mas frequentemente comprometeu a eficiência do estoque. Assim, pode-se concluir que, ao equilibrar adequadamente os diferentes critérios de otimização, como estoque e *setup*, é possível alcançar uma produção mais econômica, eficiente e alinhada às necessidades operacionais, reforçando a importância de um planejamento cuidadoso e da consideração de múltiplos fatores na otimização de processos.

7.2 Trabalhos futuros

Como propostas de trabalhos futuros, sugere-se dar continuidade a esta pesquisa nas seguintes direções:

(i) Primeiramente, a complexidade computacional das modelagens, especialmente a 1D-PMCE-DE-SD, pode ser um desafio em cenários de grande escala, o que

abre espaço para o desenvolvimento de algoritmos mais eficientes ou técnicas de decomposição que permitam soluções viáveis em um tempo reduzido.

(ii) Além disso, a análise realizada focou em um conjunto específico de parâmetros e variáveis, mas há uma gama de outros fatores que poderiam ser considerados, como a variabilidade nos custos de produção, mudanças nas condições de mercado, dentre outros.

(iii) Outro aspecto que merece ser aprofundado é a integração de outras abordagens de otimização, como métodos híbridos que combinem programação matemática com heurísticas ou metaheurísticas. Portanto, futuras pesquisas podem ampliar o escopo desta tese, investigando novas variáveis, métodos e cenários que possibilitem uma otimização ainda mais eficaz e realista dos processos estudados.

(iv) Os resultados obtidos indicam que o método *Local Branching* se apresenta como uma alternativa promissora ao CPLEX, especialmente quando aplicado a instâncias de grande escala e alta complexidade. Contudo, a qualidade das soluções geradas é, em muitas situações, inferior àquela obtida usando o CPLEX, e o tempo de execução pode se tornar um desafio, especialmente em casos específicos. Para tornar o método mais competitivo, melhorias substanciais podem ser implementadas, tanto em termos de eficiência computacional quanto na qualidade das soluções encontradas.

(v) O modelo atual pode ser expandido para incorporar incertezas relacionadas à demanda e à disponibilidade de recursos, o que tornaria a abordagem mais robusta e realista em contextos industriais dinâmicos. Assim, uma via para pesquisas futuras no contexto do Problema de Estoque de Corte Multiperíodo com Datas de Entrega e *Setup* envolve abordar a incerteza em parâmetros-chave, como tempos ou custos de corte e configuração. A contabilização de incertezas pode fornecer uma representação mais realista de cenários do mundo real e levar a soluções robustas que são eficazes sob várias condições.

(vi) Considerando a natureza conflitante dos objetivos de estoque, atraso, antecipação e setup, explorar uma formulação multiobjetivo pode oferecer oportunidades para obter soluções aprimoradas e obter maior volume de informações sobre as compensações entre esses objetivos conflitantes. Por outro lado, os resultados apresentados demonstram que diferentes modelagens impactam significativamente, mas de diferentes formas, as questões de como atraso, antecipação, estoque e *setup*. A formulação multiobjetivo pode trazer novos elementos para essas análises.

(vii) Outro ponto de grande relevância seria a aplicação prática do modelo em diferentes setores, como a fabricação de móveis (caso unidimensional), a produção de embalagens ou a construção civil. Estudos de caso reais nessas indústrias poderiam fornecer informações valiosas, validando os resultados obtidos e permitindo identificar ajustes necessários no modelo.

(viii) Modelar para os casos Bidimensional e Tridimensional.

(ix) Formulações baseadas em fluxo em arco.

Apêndice A

Formulações da literatura para o PCE com data de entrega

Neste apêndice são apresentadas algumas formulações para o PCE com data de entrega. Na Seção A.1 descrevemos a modelagem proposta por Reinertsen e Vossen (2010). Na Seção A.2 descrevemos a modelagem proposta por Arbib e Marinelli (2014). Por fim, na Seção A.3 descrevemos a modelagem proposta por Arbib e Marinelli (2017).

A.1 Proposta de modelagem de Reinertsen e Vossen (2010)

Reinertsen e Vossen (2010) estudam inicialmente o 1D-PCE com data de entrega com apenas um tipo de objeto em estoque e quantidade ilimitada, ou seja, há uma quantidade suficiente em estoque para atender à demanda. Os autores propuseram um modelo de otimização linear inteira para o PMCE-DE. Apresentaram modelos para outras formulações do problema, com diferentes tipos de objetos em estoque e com diferentes diâmetros dos objetos em estoque, e apresentam os procedimentos de solução que resolvem os modelos para o PMCE-DE propostos.

Suponha que d_i, $i = 1,..,n$, representa uma data em que o pedido deva ser entregue. Associa-se o valor da data de entrega ao número de objetos cortados no período considerado, ou seja, para cada unidade de tempo, admite-se que um único objeto é cortado. Assim, a data de entrega expressa a quantidade de objetos em estoque que podem ser cortados até a data de entrega do pedido, ou seja, se a data de entrega d_i for 5 unidades de tempo, será possível cortar apenas 5 objetos no período. A data de entrega, além disso, inclui a consideração de capacidade da máquina, pois, com máquinas automáticas de corte, o tempo de configuração e o tempo para troca de padrões de corte são mínimos, e pode ser desprezado, sem afetar a aplicação do plano de corte resultante. Algumas operações de corte requerem períodos de tempo para mudar a configuração da faca na máquina e isso aumenta bastante a dificuldade dos problemas. Este fato não é abordado por Reinertsen e Vossen (2010). Considere também que todos os pedidos que precisam ser produzidos, dentro do horizonte de planejamento, são conhecidos no momento em que um plano de corte é criado.

O horizonte de planejamento é o período de tempo ao qual um plano de corte

específico se relaciona. Ele é determinado pelo período de tempo que vai desde a data presente (início) até uma data futura (de entrega), na qual os pedidos devem ser entregues. No horizonte de planejamento, as datas de entrega d_i são organizadas em ordem não-decrescente, ou seja, $[d^0, d^1], [d^1, d^2], [d^2, d^3], \ldots, [d_{n-1}, d_n]$, sendo $d^0 = 0$ e $d_n = \max\{d_i\}$. Cada intervalo representa um período de tempo no qual os itens devem ser produzidos e entregues.

Reinertsen e Vossen (2010) apresentam um estudo de caso usando dados de um grande fabricante de aço. O objetivo é mostrar que o modelo proposto é capaz de resolver problemas de dimensão industrial, como a agregação de pedidos múltiplos, diferentes comprimentos de corte e tipos de material na mesma máquina. Além disso, busca-se avaliar o desempenho operacional em termos do resíduo resultante e do atraso de pedidos, usando o modelo em um quadro de horizonte rolante.

De cada execução de um padrão de corte j, temos um número α_{ij}, que indica a quantidade de itens i produzidos para cada pedido, de acordo com aquele padrão de corte j. As variáveis de decisão que indicam, no modelo, o número de objetos a serem cortados no padrão de corte j no período t são representadas por x_{jt}. Para cada pedido do item i, permite-se que uma violação da restrição sobre a data de entrega, ou seja, um atraso no cumprimento da data de entrega, denotado por y_i, é um limite para o atraso na entrega do item i e esta violação será penalizada na função objetivo com uma constante w_i, para cada pedido do item i em atraso. Ao considerarmos os problemas em que nem todas as datas de entrega podem ser atendidas, o objetivo é que o modelo consiga fornecer um plano de corte que determine um conjunto de padrões de corte e suas frequências correspondentes para cada período analisado, de modo que cada pedido deva ser concluído dentro do horizonte de planejamento, e que a soma do custo total por atraso na entrega dos pedidos e número de objetos cortados seja minimizado.

Considere, assim, a seguinte descrição do modelo matemático para este problema:

Índices:

- i: tipo de item, $i = 1,..,n$;
- t: período de tempo, $t = 1,...,M$;
- j: padrão de corte, $j = 1,..,N$.

Parâmetros:

- L: tamanho do objeto em estoque;
- ℓ_i: tamanho do item i ($\ell_i \leq L,\ \forall\ i$);
- b_i: demanda do item i;
- d_i: número de objetos que podem ser cortados até a data de entrega do item i;
- a_{ij} : quantidade de itens i produzidos no padrão de corte j;
- w_i: penalidade por atraso no corte do item i.
- w_0: penalidade por objeto cortado.

Variáveis de decisão:

- y_i: número de objetos cortados com atraso;
- x_{jt} : número de objetos a serem cortados no padrão de corte j no período t.

O modelo matemático em otimização linear inteira para o Problema de Corte de Estoque com data de entrega proposto por Reinertsen e Vossen (2010) é dado por:

$$\text{(Modelo RV) min} \quad w^0 \sum_{j=1}^{N} \sum_{t=1}^{M} x_{jt} + \sum_{i} w_i y_i \tag{A.1}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{N} \sum_{t=1}^{i} x_{jt} \leq d_i + y_i, \ \forall i \tag{A.2}$$

$$\sum_{j=1}^{N} \sum_{t=1}^{i} a_{ij} x_{jt} \geq b_i, \forall i, \tag{A.3}$$

$$x_{jt} \in \mathbb{Z}_+, \ \forall j,t \tag{A.4}$$

$$y_i \in \mathbb{Z}_+, \quad \forall i. \tag{A.5}$$

A função objetivo definida em (A.1) minimiza a penalização da combinação entre o número de objetos cortados e o atraso na entrega dos pedidos. O primeiro conjunto de restrições (A.2) garante que a soma dos objetos cortados de acordo com todos os padrões de corte, em todos os períodos antes da data de entrega, é menor ou igual ao número permitido de objetos que podem ser processados antes da demanda ser cumprida. No caso em que todos os pedidos não conseguem ser atendidos dentro de sua data de entrega, procura-se uma solução que minimize o atraso, permitindo o atraso e penalizando, na função objetivo (A.1), com custo w_i. O segundo conjunto de restrições (A.3) determina o número total de itens i produzidos antes da data de entrega, que é a quantidade total mínima que devemos ter para atender a demanda. Já o conjunto de restrições (A.4) e (A.5) são as condições de não negatividade e integralidade das variáveis do modelo.

A.2 Proposta de Arbib e Marinelli (2014)

Arbib e Marinelli (2014) estudam o 1D-PCE com datas de entrega. Neste artigo, os autores propuseram uma formulação de otimização linear inteira e desenvolveram heurísticas primais, procedimentos para obtenção de limitantes superiores e um esquema de enumeração implícita, além de estudaram o modelo matemático proposto por Reinertsen e Vossen (2010) e agregarem novas ideias a esse modelo.

O Problema estudado pelos autores tem a seguinte estrutura:

Problema 1. *Seja um conjunto $\mathcal{J}$ com n pedidos e as seguintes características:*

(i) Cada pedido $i \in \mathcal{J}$ está associado a uma data de entrega $d_i \geq 0$ e a uma penalidade w_i;

(ii) Cada pedido $i \in \mathcal{J}$ requer corte unidimensional b_i da peça normalizado com comprimento $l_i < 1$ de itens em estoque de comprimento unitário;

(iii) Cortar um objeto em estoque leva uma unidade de tempo, independentemente de seu padrão de corte.

Pretende-se encontrar um plano de corte, que é um conjunto de padrões de corte e suas frequências, e, de acordo com eles, uma sequência de custos que minimize a combinação de matéria prima usada (número de objetos cortados) e o atraso ponderado das tarefas.

Arbib e Marinelli (2014) propuseram uma formulação exata para o Problema 1 e, como em Reinertsen e Vossen (2010), atribuem padrões de corte para a formulação multiperíodos, usando equações de equilíbrio para acionar o atraso do pedido. A contribuição deles é dupla:

(i) a formulação proporciona um limitante para o problema dual, que pode ser utilizado para avaliar o desempenho de algoritmos heurísticos;

(ii) o algoritmo primal melhora o que foi proposto em Reinertsen e Vossen (2010), apresentando soluções que, em muitos casos, têm menor custo que a relaxação linear do modelo matemático de Reinertsen e Vossen (2010).

O modelo de Arbib e Marinelli (2014), chamado CF, representa uma formulação do problema de corte de estoque que mistura este problema com o problema de dimensionamento de lotes.

Considere, assim, a seguinte descrição do modelo matemático para este problema.

Índices:

- i: tipo de item, $i = 1,..,n$;
- t: período de tempo, $t = 1,...,M$;
- j: padrão de corte, $j = 1,..,N$.

Parâmetros:

- L: tamanho do objeto em estoque;
- ℓ_i: tamanho do item i ($\ell_i \leq L,\ \forall\ i$);
- a_{ij} : quantidade de itens i produzidos no padrão de corte j;
- w_i: penalidade por atraso no corte do item i;
- w^0: penalidade por corte;
- d_i: número de objetos que podem ser cortados até a data de entrega do item i;
- b_i^t: a demanda do item i no período t, de modo que, se $i = t$, então $b_i^t = b_i$; caso contrário, $b_i^t = 0$;
- u_i^t: limite superior do atraso acumulado para o item i no período t ($u_i^t = b_i$);

Variáveis de decisão:

- y_i^t: variável binária, que indica a existência ($y_i^t = 1$) ou não ($y_i^t = 0$) de atraso no período t considerado, para $t > i$.
- x_j^t: número de objetos a serem cortados no padrão de corte j no período t.
- s_i^t: nível de estoque do item i no período t;

Assim, o modelo matemático em otimização linear inteira para o Problema de Corte de Estoque com data de entrega proposto por Arbib e Marinelli (2014) é dado por:

$$\text{Modelo (CF) min} \quad w^0 \sum_{j=1}^{N} \sum_{t=1}^{M+1} x_{jt} + \sum_{i}^{n} w_i \sum_{t>i} \left(d^t - d^{t-1}\right) y_i^t \tag{A.6}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{N} x_{jt} \leq d^t - d^{t-1}, \;\; t = 1,...,n+1 \tag{A.7}$$

$$s_i^{t-1} + \sum_{j=1}^{N} a_{ij} x_{jt} = s_i^t + b_i^t \;, i = 1, \ldots, n, \; t = 1,...,n+1 \tag{A.8}$$

$$u_i^t y_i^t + s_i^{t-1} \geq 0, \; i = 1, \ldots n \;, t = i+1,...,n+1 \tag{A.9}$$

$$x_{jt} \in \mathbb{Z}_+, \quad j = 1, \ldots, N, t = 1,...,n+1 \tag{A.10}$$

$$y_i^t \in \{0,1\}, \; i = 1, \ldots, n \,, t = 1,...,n+1. \tag{A.11}$$

A função objetivo (A.6) busca atender a demanda, minimizando uma combinação linear de custo de matéria-prima e do atraso ponderado das tarefas. O conjunto de restrições (A.7) representa as restrições de capacidade, ou seja, o número total de objetos cortados no período deve ser inferior à capacidade de corte no período t. O conjunto de restrições (A.8) é uma equação de equilíbrio, que estabelece que o estoque do período anterior com o plano de corte do período t deve ser igual a quantidade de estoque no período t com o que será retirado naquele período. O conjunto de restrições (A.9) identifica se há atraso na produção, pois a produção do item i está atrasada se $s_i^t < 0$ para $t \geq i$. Assim, as datas de entrega seriam garantidas por $s_i^t \geq 0$, $\forall i$ e $t \geq i$, mas, desde que os pedidos sejam aceitas com atraso, desencadeia-se então os atrasos por uma variável binária y_i^t.

Como as extremidades dos intervalos de tempo correspondem à data de entrega, o modelo CF superestima o tempo de conclusão C_i pelo menor $d_i \geq C_i$, introduzindo um erro potencial no cálculo de y_i. Esta fonte de erro pode ser atenuada e, eventualmente, suprimida, utilizando períodos mais curtos.

O procedimento fornece uma formulação refinada RF do modelo (CF), na qual a função objetivo é reescrita como:

$$\min w^0 \sum_{j=1}^{N} \sum_{t=1}^{M+1} x_{tj} + \sum_{i=1}^{n} w_i \sum_{t \in K_i} \left(r^t - r^{t-1}\right) y_i^t \tag{A.12}$$

em que K_i contém os índices dos períodos após d_i. Além disso, as restrições de capacidade (A.7) são ajustadas, na forma:

$$\sum_{j=1}^{N} x_{tj} \leq r^t - r^{t-1} \forall t = 1, \ldots, n+1. \tag{A.13}$$

Os períodos de tempo $\Delta^t = [r^{t-1}, t^t]$ são inicializados segundo:

$$\begin{aligned} r^t &= d^t, \forall\ t = 0, \ldots, n; \\ r_{n+1} &= 1 + C_{\max} \end{aligned}$$

Dessa maneira, Δ^t está atrasado se, para algum i, $i = 1, \ldots, n$, o objeto i é produzido em Δ^t e $r^t > d_i$.

Os períodos são reduzidos usando o seguinte procedimento mostrado no Algoritmo 12.

Algoritmo 12 Processo de Refinamento - RF

1: **procedure** Processo de Refinamento - RF
2: **while** (existir um período em atraso em que mais de um padrão foi executado com comprimento > 0) **do**
   1. divida o período Δ^t em:
      - $\Delta^{t'} = \left[r^{t-1}, \lceil \frac{r^{t-1}+r^t}{2} \rceil\right]$
      - $\Delta^{t''} = \left[\lceil \frac{r^{t-1}+r^t}{2}, r^t \rceil\right]$
   2. resolva o modelo (A.6)-(A.11).
3: **end while**
4: **end procedure**

A.3 Proposta de Arbib e Marinelli (2017)

Arbib e Marinelli (2017) estudam o problema *Bin Packing Unidimensional* (1-BP). Nesse problema, temos itens de diferentes comprimentos que devem ser atribuídos a um número mínimo de caixas (*bin*). Cada item demandado requer um tempo específico de corte e alguma quantidade de recurso. Os autores generalizam esse problema para o caso no qual cada item i é demandado para uma data d_i, tendo, como objetivo, a minimização da combinação convexa entre o *makespan* $C_{\max} = \max_i \{C_i\}$ e o *lateness* $L_{\max} = \max_i \{(C_i - d_i), 0\}$. O problema *Bin Packing Unidimensional* é equivalente ao Problema de Corte de Estoque com uma unidade de demanda para o tipo de item i.

Arbib e Marinelli (2017) propõem uma formulação linear inteira mista indexada no tempo. O problema pode ser interpretado, então, como um problema de *scheduling* com n tarefas de comprimento de tempo unitário. Admite-se que, ao se realizar a tarefa do tipo de item i, consome-se uma parte da fração da discretização do *bin* disponível em uma unidade de tempo.

Suponha que um *bin* unidimensional esteja disponível. A descrição do modelo matemático segundo Arbib e Marinelli (2017) é apresentada a seguir.

Índices:

- i: tipo de item, $i = 1,..,n$;
- t: período de tempo, $t = 1,...,M$;
- j: padrão de corte, $j = 1,..,N$.

Parâmetros:

- d_i: data de entrega do item i;
- a_i : quantidade de itens i produzidos no *bin*;
- d_i: número de objetos i que podem ser cortados até a data de entrega;
- α^1 e α^2: fatores de ponderação da função objetivo, tais que $\alpha^1 + \alpha^2 = 1$.

Variáveis de Decisão:

- L_{max}: variável de decisão que indica o atraso máximo;
- C_{max}: variável de decisão que indica o tempo (máximo) necessário para concluir todos os pedidos;
- x_i^t: variável binária que recebe o valor $x_i^t = 1$ se, e somente se, o item i for atribuído à caixa no tempo t; em caso contrário, $x_i^t = 0$;
- z^t: variável binária que recebe o valor $z^t = 1$ se, e somente se, um compartimento da caixa for usado no período t; em caso contrário, $z^t = 0$.

O modelo matemático em otimização linear inteira mista proposto por Arbib e Marinelli (2017) para o Problema *Bin Packing* com data de entrega é dado por:

$$(KX)\ \min \quad \alpha^1 C_{\max} + \alpha^2 L_{\max} \tag{A.14}$$

$$\text{sujeito a:} \quad \sum_{t}^{M} x_i^t = 1, \quad \forall i, \tag{A.15}$$

$$\sum_{i}^{n} a_i x_i^t - z^t \leq 0, \quad \forall t, \tag{A.16}$$

$$\sum_{t}^{n} z^t - C_{\max} = 0, \tag{A.17}$$

$$t x_i^t - L_{\max} \leq d_i, \quad \forall t, i, \tag{A.18}$$

$$x_i^t \in \{0,1\}, \quad \forall t, i, \tag{A.19}$$

$$z^t \in \{0,1\}, \quad \forall t. \tag{A.20}$$

O conjunto de restrições (A.15) garante que todas as tarefas serão atribuída a um período; o conjunto de restrições (A.16) garante que os recursos são usados de acordo com a disponibilidade; os conjuntos de restrições (A.17) e (A.18) definem os termos da função objetivo dada em (A.14).

Referências Bibliográficas

Allahverdi, Ali. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, v. 246, n. 2, p. 345–378. doi: https://doi.org/10.1016/j.ejor.2015.04.004.

Allahverdi, Ali; Gupta, Jatinder N. D. e Aldowaisan, Tariq. (1999). A review of scheduling research involving setup considerations. *Omega*, v. 27, n. 2, p. 219–239. doi: https://doi.org/10.1016/S0305-0483(98)00042-5.

Allahverdi, Ali; Ng, C. T.; Cheng, T. C. E. e Kovalyov, Mikhail Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, v. 187, n. 3, p. 985–1032. doi: https://doi.org/10.1016/j.ejor.2006.06.060.

Alves, Cláudio e Valério de Carvalho, J.M. (2008). A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research*, v. 35, n. 4, p. 1315–1328. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2006.08.014.

Arbib, Claudio e Marinelli, Fabrizio. (2014). On cutting stock with due dates. *Omega*, v. 46, p. 11–20.

Arbib, Claudio e Marinelli, Fabrizio. (2017). Maximum lateness minimization in one-dimensional bin packing. *Omega*, v. 68, p. 76–84.

Arbib, Claudio; Marinelli, Fabrizio; Pferschy, Ulrich e Ranjbar, Fatemeh K. (2023). One-dimensional stock cutting resilient against singular random defects. *Computers & Operations Research*, v. 157, p. 106280.

Arenales, M.; Armentano, V. A.; Morabito, R. e Yanasse, H. H. (2007). *Pesquisa Operacional*. Campus-Elsevier, Rio de Janeiro, RJ, Brasil.

Bazaraa, Mokhtar S.; Jarvis, John J. e Sherali, Hanif D. (2011). *Linear Programming and Network Flows*. John Wiley & Sons.

Belov, Gleb e Scheithauer, Guntram. (2002). A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research*, v. 141, n. 2, p. 274–294.

Bennell, Julia A; Lee, Lai Soon e Potts, Chris N. (2013). A genetic algorithm for two-dimensional bin packing with due dates. *International Journal of Production Economics*, v. 145, n. 2, p. 547–560.

Bertolini, M; Mezzogori, D e Zammori, F. (2024). Hybrid heuristic for the one-dimensional cutting stock problem with usable leftovers and additional operating constraints. *International Journal of Industrial Engineering Computations*, v. 15, n. 1, p. 149–170.

Boschetti, Marco A.; Maniezzo, Vittorio; Roffilli, Matteo e Bolufé Röhler, Antonio. (2009). Matheuristics: Optimization, simulation and control. Blesa, María J.; Blum, Christian; Di Gaspero, Luca; Roli, Andrea; Sampels, Michael e Schaerf, Andrea, editors, *Hybrid Metaheuristics: Proceedings of the 6th International Workshop (HM 2009), Udine, Italy, October 16-17, 2009*, volume 5818, p. 171–177, Berlin, Heidelberg. Springer.

Braga, Nuno; Alves, Cláudio; Macedo, Rita e Valério de Carvalho, J. M. (2015). A model-based heuristic for the combined cutting stock and scheduling problem. Gervasi, Osvaldo; Murgante, Beniamino; Misra, Sanjay; Gavrilova, Marina L.; Rocha, Ana Maria Alves Coutinho; Torre, Carmelo; Taniar, David e Apduhan, Bernady O., editors, *Computational Science and Its Applications – ICCSA 2015*, volume 9156, p. 490–505, Cham. Springer International Publishing.

Braga, Nuno; Alves, Cláudio; Macedo, Rita e Valério de Carvalho, J. M. (2016)a. Combined cutting stock and scheduling: a matheuristic approach. *International Journal of Innovative Computing and Applications*, v. 7, n. 3, p. 135–146.

Braga, Nuno; Alves, Cláudio e Valério de Carvalho, José M. (2016)b. Exact solution of combined cutting stock and scheduling problems. Fonseca, Raquel J.; Weber, Gerhard-Wilhelm e Telhada, João, editors, *Computational Management Science*, volume 682, p. 131–139, Cham. Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-20430-7_17.

Cherri, Adriana Cristina; Arenales, Marcos Nereu e Yanasse, Horacio Hideki. (2009). The one-dimensional cutting stock problem with usable leftover–a heuristic approach. *European Journal of Operational Research*, v. 196, n. 3, p. 897–908.

Cherri, Adriana Cristina; Arenales, Marcos Nereu; Yanasse, Horacio Hideki; Poldi, Kelly Cristina e Vianna, Andréa Carla Gonçalves. (2014). The one-dimensional cutting stock problem with usable leftovers–a survey. *European Journal of Operational Research*, v. 236, n. 2, p. 395–402.

Cui, Yaodong e Yang, Yuli. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftover. *European Journal of Operational Research*, v. 204, n. 2, p. 245–250.

Cui, Yaodong; Zhong, Cheng e Yao, Yi. (2015). Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost. *European Journal of Operational Research*, v. 243, n. 2, p. 540–546. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2014.12.015.

Dantzig, George B. (1963). *Linear Programming and Extensions*, volume Book 48 of *Princeton Landmarks in Mathematics and Physics*. Princeton University Press, Princeton, New Jersey, EUA.

de Araujo, Silvio Alexandre; Poldi, Kelly Cristina e Smith, Jim. (2014). A genetic algorithm for the one-dimensional cutting stock problem with setups. *Pesquisa Operacional*, v. 34, n. 2, p. 165–187.

De Queiroz, Thiago A; Miyazawa, Flávio K; Wakabayashi, Yoshiko e Xavier, Eduardo C. (2012). Algorithms for 3d guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, v. 39, n. 2, p. 200–212.

Delorme, Maxence e Iori, Manuel. (2020). Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. *INFORMS Journal on Computing*, v. 32, n. 1, p. 101–119.

Delorme, Maxence; Iori, Manuel e Martello, Silvano. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, v. 255, n. 1, p. 1–20.

Dyckhoff, Harald. (1981). A new linear programming approach to the cutting stock problem. *Operations Research*, v. 29, n. 6, p. 1092–1104.

Dyckhoff, Harald. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, n. 2, p. 145–159.

Fischetti, Matteo e Lodi, Andrea. (2003). Local branching. *Mathematical programming*, v. 98, n. 1-3, p. 23–47.

Foerster, Hildegard e Wascher, Gerhard. (2000). Pattern reduction in one-dimensional cutting stock problems. *International journal of production research*, v. 38, n. 7, p. 1657–1676.

Garey, Michael R. e Johnson, David S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Garraffa, Michele; Salassa, Fabio; Vancroonenburg, Wim; Vanden Berghe, Greet e Wauters, Tony. (2016). The one-dimensional cutting stock problem with sequence-dependent cut losses. *International Transactions in Operational Research*, v. 23, n. 1-2, p. 5–24.

Gau, T. e Wäscher, G. (1995). CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, v. 84, n. 3, p. 572–579.

Gilmore, Paul C. e Gomory, Ralph E. (1961). A linear programming approach to the cutting-stock problem. *Operations research*, v. 9, n. 6, p. 849–859.

Gilmore, Paul C. e Gomory, Ralph E. (1963). A linear programming approach to the cutting stock problem – Part II. *Operations research*, v. 11, n. 6, p. 863–888.

Gilmore, Paul C. e Gomory, Ralph E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations research*, v. 13, n. 1, p. 94–120.

Haessler, R. e Talbot, F. B. (1983). A 0-1 model for solving the corrugator trim problem. *Management Science*, v. 29, p. 200–209.

Haessler, Robert W. (1975). Controlling cutting pattern changes in one-dimensional trim problems. *Operations Research*, v. 23, n. 3, p. 483–493.

Hao, Xinye; Liu, Changchun; Liu, Maoqi; Zhang, Canrong e Zheng, Li. (2023). Solving a real-world large-scale cutting stock problem: A clustering-assignment-based model. *IISE Transactions*, v. 55, n. 11, p. 1160–1173.

Harjunkoski, I. *Application of MINLP Methods on a Scheduling Problem in the Paper Converting Industry*. PhD Dissertation, Abo Akademi University, Abo, Finland, (1997).

Harjunkoski, Iiro; Westerlund, Tapio; Isaksson, Johnny e Skrifvars, Hans. (1996). Different formulations for solving trim loss problems in a paper-converting mill with ILP. *Computers & Chemical Engineering*, v. 20, p. S121–S126.

Harjunkoski, Iiro; Westerlund, Tapio e Pörn, Ray. (1999). Numerical and environmental considerations on a complex industrial mixed integer non-linear programming (MINLP) problem. *Computers & Chemical Engineering*, v. 23, p. 1545–1561.

Harjunkoski, Iiro; Westerlund, Tapio; Pörn, Ray e Skrifvars, Hans. (1998). Different transformations for solving non-convex trim-loss problems by MINLP. *European Journal of Operational Research*, v. 105, n. 3, p. 594–603.

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*, v. 47, n. 1, p. 38–47. ISSN 0160-5682.

Henn, Sebastian e Wäscher, Gerhard. May(2013). Extensions of cutting problems: setups. *Pesquisa Operacional*, v. 33, n. 2, p. 133–162. ISSN 0101-7438. doi: 10.1590/S0101-74382013000200001.

Hinxman, A. I. (1980). The trim-loss and assortment problems: A survey. *European Journal of Operational Research*, v. 5, n. 1, p. 8–18.

Johnson, David S. e Garey, Michael R. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman & Co.

Johnston, Robert E e Sadinlija, Enes. (2004). A new model for complete solutions to one-dimensional cutting stock problems. *European Journal of Operational Research*, v. 153, n. 1, p. 176–183. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(02)00704-X. Timetabling and Rostering.

Kantorovich, Leonid V. (1960). Mathematical methods of organizing and planning production. *Management science*, v. 6, n. 4, p. 366–422.

Li, Shanling. (1996). Multi-job cutting stock problem with due dates and release dates. *Journal of the Operational Research Society*, v. 47, n. 4, p. 490–510.

López-Ibánez, Manuel; Dubois-Lacoste, Jérémie; Cáceres, Leslie Pérez; Birattari, Mauro e Stützle, Thomas. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, v. 3, p. 43–58.

Luo, Qiang; Du, Bing; Rao, Yunqing e Guo, Xiaoqiang. (2022). Metaheuristic algorithms for a special cutting stock problem with multiple stocks in the transformer manufacturing industry. *Expert Systems with Applications*, v. 210, p. 118578.

Ma, Ning; Liu, Ya e Zhou, Zhili. (2019)a. Two heuristics for the capacitated multi-period cutting stock problem with pattern setup cost. *Computers & Operations Research*, v. 109, p. 218–229. doi: https://doi.org/10.1016/j.cor.2019.05.013.

Ma, Ning; Liu, Ya e Zhou, Zhili. (2019)b. Two heuristics for the capacitated multi-period cutting stock problem with pattern setup cost. *Computers & Operations Research*, v. 109, p. 218–229.

Ma, Ning; Liu, Ya; Zhou, Zhili e Chu, Chengbin. (2018). Combined cutting stock and lot-sizing problem with pattern setup. *Computers & Operations Research*, v. 95, p. 44–55. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2018.02.016.

Martello, Silvano e Toth, Paolo. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons.

Martin, Mateus; Moretti, Antonio; Gomes-Ruggiero, Marcia e Salles Neto, Luiz. (2018). Modification of haessler's sequential heuristic procedure for the one-dimensional cutting stock problem with setup cost. *Production*, v. 28, p. e20170105. ISSN 0103-6513. doi: 10.1590/0103-6513.20170105.

Martin, Mateus; Yanasse, Horacio Hideki e Salles-Neto, Luiz Leduíno. (2022). Pattern-based ilp models for the one-dimensional cutting stock problem with setup cost. *Journal of Combinatorial Optimization*, v. 44, p. 557–582. doi: https://doi.org/10.1007/s10878-022-00848-z.

Martinovic, John; Scheithauer, Guntram e Valério de Carvalho, J. M. (2018). A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems. *European Journal of Operational Research*, v. 266, n. 2, p. 458–471.

Melega, Gislaine Mara; de Araujo, Silvio Alexandre; Jans, Raf e Morabito, Reinaldo. (2022). Formulations and exact solution approaches for a coupled bin-packing and lot-sizing problem with sequence-dependent setups. *Flexible Services and Manufacturing Journal*, v. 35, p. 1276–1312. doi: https://doi.org/10.1007/s10696-022-09464-9.

Melega, Gislaine Mara; de Araujo, Silvio Alexandre e Morabito, Reinaldo. (2020). Mathematical model and solution approaches for integrated lot-sizing, scheduling and cutting stock problems. *Annals of Operations Research*, v. 295, n. 2, p. 695–736.

Mellouli, Ahmed e Dammak, Abdelaziz. (2008). An algorithm for the two-dimensional cutting-stock problem based on a pattern generation procedure. *Inf Manag Sci*, v. 19, n. 2, p. 201–218.

Mobasher, Azadeh e Ekici, Ali. (2013). Solution approaches for the cutting stock problem with setup cost. *Computers & Operations Research*, v. 40, n. 1, p. 225–235. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2012.06.007.

Montiel-Arrieta, Leonardo Javier; Barragan-Vite, Irving; Seck-Tuoh-Mora, Juan Carlos; Hernandez-Romero, Norberto; González-Hernández, Manuel e Medina-Marin, Joselito. (2023). Minimizing the total waste in the one-dimensional cutting stock problem with the african buffalo optimization algorithm. *PeerJ Computer Science*, v. 9, p. e1728.

Ngoo, Chong Man; Goh, Say Leng; Sze, San Nah; Sabar, Nasser R.; Hijazi, Mohd Hanafi Ahmad e Kendall, Graham. (2024). A survey of mat-heuristics for combinatorial optimisation problems: Variants, trends and opportunities. *Applied Soft Computing*, v. 164, p. 111947. doi: https://doi.org/10.1016/j.asoc.2024.111947.

Oliveira, Elisama de Araujo Silva. (2016). O problema de corte de estoque com data de entrega. Dissertação de mestrado, Universidade Estadual de Campinas, Campinas, São Paulo.

Oliveira, Elisama de Araújo Silva; de Sá, Elisângela Martins; de Souza, Sérgio Ricardo e Wanner, Elizabeth Fialho. (2020). Relax-and-fix aplicado ao problema de corte de estoque com data de entrega. *Proceedings Series of the Brazilian Society of Computational and Applied Mathematics*, v. 7, n. 1. doi: https://doi.org/10.5540/03.2020.007.01.0426.

Oliveira, Elisama de Araújo Silva; de Sá, Elisângela Martins; de Souza, Sérgio Ricardo e Wanner, Elizabeth Fialho. (2022). Uma proposta de solução via local branching do problema de corte de estoque com data de entrega e setup de troca de padrão de corte. *Proceedings Series of the Brazilian Society of Computational and Applied Mathematics*, v. 9, n. 1. doi: https://doi.org/10.5540/03.2022.009.01.0325.

Oliveira, Elisama de Araújo Silva; Wanner, Elizabeth; de Sá, Elisangela Martins e de Souza, Sérgio Ricardo. Matheuristic applied to the cutting stock problem with due dates, (2024). Aceito para publicação.

Oliveira, Elisama de Araújo Silva; Wanner, Elizabeth; de Sá, Elisangela Martins e de Souza, Sérgio Ricardo. (2025). A local branching-based solution for the multi-period cutting stock problem with tardiness, earliness, and setup costs. *Journal of Heuristics*, v. 31, n. 16. doi: https://doi.org/10.1007/s10732-025-09547-4.

Oliveira, Washington A.; Fiorotto, Diego J.; Song, Xiang e Jones, Dylan F. (2021). An extended goal programming model for the multiobjective integrated lot-sizing and cutting stock problem. *European Journal of Operational Research*, v. 295, n. 3, p. 996–1007. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2021.03.049.

Pitombeira-Neto, Anselmo R e Murta, Arthur HF. (2022). A reinforcement learning approach to the stochastic cutting stock problem. *EURO Journal on Computational Optimization*, v. 10, p. 100027.

Poldi, Kelly Cristina e Arenales, Marcos Nereu. (2006). Heurísticas para o problema de corte de estoque unidimensional inteiro. *Pesquisa Operacional*, v. 26, n. 3, p. 473–492.

Poldi, Kelly Cristina e Arenales, Marcos Nereu. (2009). Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers & Operations Research*, v. 36, n. 6, p. 2074–2081.

Poldi, Kelly Cristina e de Araujo, Silvio Alexandre. (2016). Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. *Annals of Operations Research*, v. 238, n. 1-2, p. 497–520.

Polyakovskiy, Sergey e M'Hallah, Rym. (2018). A hybrid feasibility constraints-guided search to the two-dimensional bin packing problem with due dates. *European Journal of Operational Research*, v. 266, n. 3, p. 819–839.

Rao, M. R. (1976). On the cutting stock problem. *Journal of the Computer Society of India*, v. 7, p. 35–39.

Reinertsen, Harald e Vossen, Thomas WM. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research*, v. 201, n. 3, p. 701–711.

Sá Santos, José Victor e Nepomuceno, Napoleão. (2022). Computational performance evaluation of column generation and generate-and-solve techniques for the one-dimensional cutting stock problem. *Algorithms*, v. 15, n. 11, p. 394.

Sezer, Zeynep e Muter, İbrahim. (2018). Two-stage cutting stock problem with due dates. *Operations Research Proceedings 2016*, p. 139–145. Springer.

Silva, Eduardo M; Chaves, Antônio A; deAraujo, Silvio A e Jans, Raf. (2025). Random-key optimizer with reinforcement learning for the capacited multi-period cutting stock problem with setup cost. *Computers & Operations Research*, v. 183, p. 107159.

Silva, Eduardo M.; Melega, Gislaine M.; Akartunali, Kerem e de Araujo, Silvio A. (2023). Formulations and theoretical analysis of the one-dimensional multi-period cutting stock problem with setup cost. *European Journal of Operational Research*, v. 304, n. 2, p. 443–460. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2022.04.023.

Stadtler, Hartmut. (1990). A one-dimensional cutting stock problem in the aluminium industry and its solution. *European Journal of Operational Research*, v. 44, n. 2, p. 209–223.

Umetani, Shunji; Yagiura, Mutsunori e Ibaraki, Toshihide. (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*, v. 86, n. 5, p. 1093–1102.

Umetani, Shunji; Yagiura, Mutsunoti e Ibaraki, Toshihide. (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*, v. 5, p. 43–64.

Valério de Carvalho, J. M. (1996). Exact solution of one-dimensional cutting stock problems using branch-and-bound and column generation. *Gestão & Produção*, v. 3, n. 1, p. 33–48.

Valério de Carvalho, J. M. (2002). LP models for bin packing and cutting stock problems. *European Journal of Operational Research*, v. 141, n. 2, p. 253–273.

Vanderbeck, François. (2000). Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, v. 48, n. 6, p. 915–926.

Vishwakarma, Ravi e Powar, PL. (2021). An efficient mathematical model for solving one-dimensional cutting stock problem using sustainable trim. *Advances in Industrial and Manufacturing Engineering*, v. 3, p. 100046.

Wang, Danni; Xiao, Fan; Zhou, Lei e Liang, Zhe. (2020). Two-dimensional skiving and cutting stock problem with setup cost based on column-and-row generation. *European Journal of Operational Research*, v. 286, n. 2, p. 547–563.

Wäscher, Gerhard e Gau, Thomas. (1996). Heuristics for the integer one-dimensional cutting stock problem: A computational study. *Operations-Research-Spektrum*, v. 18, n. 3, p. 131–144.

Wäscher, Gerhard; Haußner, Heike e Schumann, Holger. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, v. 183, n. 3, p. 1109–1130.

Wolsey, Laurence A. (1998). *Integer Programming*. John Wiley & Sons, New York, NY, EUA.

Wu, Wei; Ito, Mayu; Hu, Yannan; Goko, Hiromichi; Sasaki, Mihiro e Yagiura, Mutsunori. (2024). Heuristic algorithms based on column generation for an online product shipping problem. *Computers & Operations Research*, v. 161, p. 106403. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2023.106403.

Wuttke, David A. e Heese, H. Sebastian. (2018). Two-dimensional cutting stock problem with sequence dependent setup times. *European Journal of Operational Research*, v. 265, n. 1, p. 303–315. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2017.07.036.

Yanasse, Horacio Hideki e Lamosa, Maria Jose Pinto. (2007). An integrated cutting stock and sequencing problem. *European Journal of Operational Research*, v. 183, n. 3, p. 1353–1370.