



ALGORITMO DE GERAÇÃO DE COLUNAS COM RESTRIÇÕES DE SOFT
FIXING APLICADO NO PROBLEMA DE CUTTING STOCK
UNIDIMENSIONAL

Matheus Degliomini Silva

Dissertação de Mestrado apresentada ao
Programa de Pós-graduação em Engenharia
de Sistemas e Computação, COPPE, da
Universidade Federal do Rio de Janeiro, como
parte dos requisitos necessários à obtenção do
título de Mestre em Engenharia de Sistemas e
Computação.

Orientadores: Abilio Pereira de Lucena Filho
Pedro Henrique González Silva

Rio de Janeiro
Setembro de 2025

ALGORITMO DE GERAÇÃO DE COLUNAS COM RESTRIÇÕES DE SOFT
FIXING APLICADO NO PROBLEMA DE CUTTING STOCK
UNIDIMENSIONAL

Matheus Degliomini Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Abilio Pereira de Lucena Filho
Pedro Henrique González Silva

Aprovada por: Prof. Pedro Henrique González Silva
Prof. Luidi Gelabert Simonetti
Prof. Yuri Abitbol de Menezes Frota

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2025

Degliomini Silva, Matheus

Algoritmo de Geração de Colunas com Restrições de Soft Fixing aplicado no Problema de Cutting Stock Unidimensional/Matheus Degliomini Silva. – Rio de Janeiro: UFRJ/COPPE, 2025.

IX, 61 p.: il.; 29,7cm.

Orientadores: Abilio Pereira de Lucena Filho

Pedro Henrique González Silva

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2025.

Referências Bibliográficas: p. 53 – 55.

1. Geração de Colunas. 2. Soft Fixing. 3. Cutting Stock Unidimensional. I. de Lucena Filho, Abilio Pereira *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Primeiramente, agradeço aos meus pais, que sempre se dedicaram para oferecer a mim e à minha irmã as melhores condições possíveis. Tudo o que conquistei até aqui é, sem dúvida, fruto do esforço e apoio deles.

Aos meus orientadores, Abilio Pereira de Lucena Filho e Pedro Henrique González Silva, deixo meu sincero reconhecimento por toda a orientação, paciência e incentivo ao longo do mestrado.

Também sou grato aos professores do PESC que contribuíram, direta ou indiretamente, com sugestões e observações valiosas para o desenvolvimento deste trabalho, em especial aos professores Luidi Simonetti e Nelson Maculan. Estendo esse agradecimento ao professor Renan Pinto, cuja orientação começou ainda no período da graduação na UFRRJ.

Aos amigos feitos na graduação pela UFRRJ que contribuíram mesmo menos próximos, em especial Amanda Azevedo, Alexandre Almeida, Daniel Rezende e Roberta Moraes. E também às amizades feitas na UFRJ, principalmente os amigos de Labotim e do Capgov, especialmente Aline Veríssimo, Amanda Matos, Beatris Xavier, Caroline Lima, Gabriel Souto, Jéssica Costa, Julia Gratz, Juliana Rangel, Leonardo Dias, Lucas Fonseca, Marcella Braga, Rafael Schneider e Victor Hugo. A todos, agradeço pelas conversas, colaborações e pelos momentos de descontração ao longo do mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMO DE GERAÇÃO DE COLUNAS COM RESTRIÇÕES DE SOFT
FIXING APLICADO NO PROBLEMA DE CUTTING STOCK
UNIDIMENSIONAL

Matheus Degliomini Silva

Setembro/2025

Orientadores: Abilio Pereira de Lucena Filho

Pedro Henrique González Silva

Programa: Engenharia de Sistemas e Computação

O Problema de *Cutting Stock* (PCS) é um problema clássico de otimização combinatória, caracterizado por pertencer à classe NP -difícil e servir de modelo para inúmeras aplicações de relevância prática ou teórica. Este trabalho investiga um procedimento de Busca Local (BL) para o PCS. Mais especificamente, uma BL que resulta da incorporação de Restrições de *Soft Fixing* (RSF) ao algoritmo padrão de Geração de Colunas aplicado ao problema. O procedimento foi testado em 1600 instâncias geradas artificialmente, tendo a formulação de Kantorovich como base de obtenção do valor ótimo dessas instâncias. Nos testes empíricos que conduzimos, a aplicação da metodologia resultou em melhores soluções viáveis para 44% das instâncias ainda sem valores ótimos conhecidos. Além disso, obteve soluções ótimas para 3% destas. Da mesma forma, o método se mostrou eficaz tanto na quantidade de soluções que conseguiu aprimorar quanto no tempo de execução. Embora apresente um custo computacional um pouco maior em comparação ao uso do algoritmo de Geração de Colunas de forma isolada, esse aumento é compensado pelos ganhos em qualidade das soluções. Em particular, a RSF que se mostrou mais atraente em nossos experimentos computacionais, combinou menores tempos de execução com um maior número de soluções que conseguiu melhorar. Nossos resultados indicam que a estratégia proposta é promissora para instâncias do PCS empiricamente mais difíceis de se resolver.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

COLUMN GENERATION ALGORITHM WITH SOFT FIXING CONSTRAINTS APPLIED TO THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM

Matheus Degliomini Silva

September/2025

Advisors: Abilio Pereira de Lucena Filho

Pedro Henrique González Silva

Department: Systems Engineering and Computer Science

The Cutting Stock Problem (PCS) is a classic combinatorial optimization problem, characterized by belonging to the NP -hard class and serving as a model for numerous applications of practical or theoretical relevance. This work investigates a Local Search (LB) procedure for PCS. More specifically, a LB that results from incorporating Soft Fixing Constraints (SFR) into the standard Column Generation algorithm applied to the problem. The procedure was tested on 1600 artificially generated instances, using the Kantorovich formulation as the basis for obtaining the optimal value of these instances. In our empirical tests, the application of the methodology resulted in better feasible solutions for 44% of the instances still without known optimal values. Furthermore, it obtained optimal solutions for 3% of these. Likewise, the method proved effective both in the number of solutions it was able to improve and in its execution time. Although it presents a slightly higher computational cost compared to using the Column Generation algorithm alone, this increase is offset by gains in solution quality. In particular, RSF, which proved most attractive in our computational experiments, combined shorter runtimes with a greater number of solutions it managed to improve. Our results indicate that the proposed strategy is promising for empirically more difficult-to-solve PCS instances.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
1 Introdução	1
2 Revisão Bibliográfica	3
2.1 Análise Quantitativa	3
2.2 Análise Qualitativa	6
3 Definição e Decomposição do Problema	11
3.1 Motivação do Problema	11
3.2 Formulação Matemática	12
4 Metodologia	17
4.1 Geração de Colunas	17
4.1.1 Solução Inicial	17
4.1.2 Problema Mestre Restrito Relaxado	18
4.1.3 Subproblema de Geração de Colunas	18
4.2 Restrições de Soft Fixing	19
5 Experimentos Computacionais	26
5.1 Instâncias Geradas	26
5.2 Análise com Geração de Colunas	28
5.3 Solução Ótima	31
5.4 Resultados Computacionais	33
6 Conclusões e Trabalhos Futuros	50
Referências Bibliográficas	53
A Demonstração do Custo Reduzido do PCS	56
B Decomposição de Dantzig-Wolfe do PCS	58

Lista de Figuras

2.1	Quantidade de artigos publicados por ano entre os selecionados. . . .	6
3.1	Peça Mestre e as Peças Menores do Exemplo 1.	13
3.2	Padrões de Corte pela Formulação de Kantorovich.	14
4.1	Fluxograma do Geração de Colunas com <i>Soft Fixing</i>	22
5.1	Tempo (s) de execução do Geração de Colunas por tamanho de ins- tância.	30
5.2	Tempo (s) de execução do Geração de Colunas por tipo de instância.	30
5.3	Tempo (s) de execução para resolver a formulação de Kantorovich. . .	31
5.4	Quantificação das instâncias que finalizam antes do tempo limite por tamanho.	32
5.5	Tempo (s) de execução da formulação de Kantorovich por tipo. . . .	33
5.6	Tempo (s) de execução do <i>Soft Fixing</i> 4.	37
5.7	Tempo (s) de execução do <i>Soft Fixing</i> 5.	38
5.8	Tempo (s) de execução do <i>Soft Fixing</i> 4&5.	38
5.9	Tempo (s) de execução do <i>Soft Fixing</i> 5&4.	39
5.10	Tempo (s) de execução do <i>Soft Fixing</i> 4 por tipo.	40
5.11	Tempo (s) de execução do <i>Soft Fixing</i> 5 por tipo.	41
5.12	Tempo (s) de execução do <i>Soft Fixing</i> 4&5 por tipo.	42
5.13	Tempo de execução do <i>Soft Fixing</i> 5&4 por tipo.	43
5.14	Tempo (s) total de execução com o <i>Soft Fixing</i> 4.	44
5.15	Tempo (s) total de execução com o <i>Soft Fixing</i> 5.	44
5.16	Tempo (s) total de execução com o <i>Soft Fixing</i> 4&5.	45
5.17	Tempo (s) total de execução com o <i>Soft Fixing</i> 5&4.	46
5.18	Tempo (s) total de execução com o <i>Soft Fixing</i> 4 por tipo.	46
5.19	Tempo (s) de execução com o <i>Soft Fixing</i> 5 por tipo.	47
5.20	Tempo (s) de execução com o <i>Soft Fixing</i> 7 (4&5) por tipo.	48
5.21	Tempo (s) de execução com o <i>Soft Fixing</i> 8 (5&4) por tipo.	48

Lista de Tabelas

2.1	Análise Quantitativa dos Repositórios.	5
2.2	Artigos Seleccionados.	5
2.3	Técnicas utilizadas no PCS nos trabalhos seleccionados.	10
5.1	Instâncias sem ótimo por tamanho.	28
5.2	Instâncias sem ótimo por tipo de distribuição.	28
5.3	Desempenho das Restrições de <i>Soft Fixing</i>	34
5.4	Tamanho das instâncias melhoradas.	34
5.5	Tipos de distribuições das instâncias melhoradas.	35
5.6	Restrições de <i>Soft Fixing</i> que chegaram ao ótimo.	35
5.7	Tamanho das instâncias que chegaram à solução ótima com <i>Soft Fixing</i>	35
5.8	Desempenho das Restrições de <i>Soft Fixing</i> por tamanho.	36
5.9	Desempenho das Restrições de <i>Soft Fixing</i> por tipo.	36
5.10	Estatísticas do tempo total de execução de cada <i>Soft Fixing</i>	49

Capítulo 1

Introdução

O Problema de *Cutting Stock* Unidimensional (PCS) é um problema clássico em Pesquisa Operacional, tendo como objetivo otimizar o corte de materiais para atender à demanda por peças menores que o tamanho padrão, minimizando perdas e desperdícios. A primeira abordagem matemática para esse tipo de problema foi proposta por KANTOROVICH (1960), que introduziu uma formulação matemática para o PCS e um procedimento de solução para resolvê-la.

Desde então, o PCS tornou-se um tema amplamente estudado e aplicado em diversos setores industriais, principalmente naqueles que lidam com processos de corte, como a indústria siderúrgica, de papel e têxtil. Entre as principais contribuições para sua resolução, destaca-se o trabalho de GILMORE e GOMORY (1961), que propuseram o método de Geração de Colunas, uma estratégia eficaz para lidar com os exponencialmente muitos padrões de corte viáveis (ou seja, variáveis) associados ao problema.

O PCS tem como objetivo minimizar o desperdício de material ao cortar objetos de grandes dimensões, como bobinas, chapas, barras ou rolos, em itens menores, de acordo com uma demanda específica. Trata-se de um problema de otimização combinatória, cuja complexidade reside não apenas na escolha dos padrões de corte, mas também na demanda industrial de se obter soluções de alta qualidade, em tempos computacionais aceitáveis.

Além disso, como concluíram LÜBBECKE e DESROSIERS (2005), as relaxações lineares das formulações atualmente existentes para o PCS levam normalmente a soluções degeneradas, o que torna ainda mais desafiador encontrar soluções viáveis de boa qualidade para o problema, de forma rápida e eficiente.

Ao longo das últimas décadas, diversas extensões e variações do PCS foram estudadas, incluindo versões multiobjetivo, multidimensionais e estocásticas (PIERINI e POLDI, 2022; SENERGUES *et al.*, 2025; DE ARAÚJO SILVA OLIVEIRA *et al.*, 2025). No entanto, mesmo em sua versão unidimensional, estratégias de melhor desempenho empírico, especialmente para instâncias de grande porte, continuam

sendo investigadas intensamente.

Nesse contexto, MUSTAKEROV e BORISSOVA (2014, 2015) propuseram uma abordagem baseada em modelagem de Programação Inteira, capaz de definir simultaneamente o comprimento ideal das barras e os padrões de corte, com o objetivo de minimizar o desperdício. O estudo também evidencia a dificuldade computacional inerente ao PCS, decorrente de sua natureza NP-difícil, e valida a eficácia da formulação proposta por meio de experimentos com dados reais, utilizando *solvers* como LINGO e *MS Excel Solver*.

Mais recentemente, abordagens com aprendizado por reforço profundo surgiram como alternativas promissoras. Por exemplo, FANG *et al.* (2023) propuseram uma técnica baseada em *Deep Reinforcement Learning* (DRL) para resolver o PCS, demonstrando boa capacidade de generalização e eficiência em instâncias de grande porte.

Da mesma forma,, heurísticas primais, como o *Soft Fixing*, introduzida por FISCHETTI e LODI (2003), vêm ganhando espaço por seu bom desempenho em orientar a busca por soluções de boa qualidade, em ambientes associados a métodos de decomposição.

Este trabalho propõe uma abordagem híbrida, que combina a técnica de Geração de Colunas com estratégias de *Soft Fixing* associadas às variáveis que definem uma solução viável para o problema. O objetivo dessa combinação é acelerar a convergência para soluções de boa qualidade para PCS, preservando, ao mesmo tempo, a estrutura matemática da formulação original. A proposta foi avaliada por meio de experimentos computacionais, comparando seu desempenho com abordagens tradicionais e discutindo as implicações dessa técnica, especialmente em problemas de grande porte.

O restante desta dissertação está organizado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica, abordando os principais estudos sobre o PCS, o algoritmo de Geração de Colunas e a técnica de *Soft Fixing*. O Capítulo 3 discute as formulações matemáticas propostas para o Problema de Cutting Stock, incluindo a decomposição sugerida por Gilmore e Gomory. O Capítulo 4 descreve a metodologia adotada, detalhando o funcionamento do algoritmo de Geração de Colunas e a incorporação das Restrições de *Soft Fixing* ao mesmo. No Capítulo 5, são apresentados os experimentos computacionais obtidos com a aplicação da abordagem proposta, seguidos da análise de desempenho e do tempo de execução. Por fim, o Capítulo 6 apresenta as conclusões do trabalho, bem como sugestões para pesquisas futuras.

Capítulo 2

Revisão Bibliográfica

Seguindo a metodologia proposta por PAGE *et al.* (2021), este capítulo apresenta uma revisão sistemática da literatura sobre o Problema do Cutting Stock (PCS), conduzida a partir de uma abordagem estruturada para identificar os estudos mais relevantes sobre o tema.

Para organizar a apresentação dos resultados, o capítulo foi dividido em duas seções complementares: a Seção 2.1 traz uma análise quantitativa da produção científica identificada, enquanto a Seção 2.2 apresenta uma análise qualitativa dos trabalhos mais relevantes para os objetivos desta dissertação.

2.1 Análise Quantitativa

A revisão bibliográfica foi conduzida nos repositórios Scopus, Web of Science (WOS) e IEEE Xplore, abrangendo publicações no período de 1981 a 2025. Embora o PCS tenha sido originalmente proposto em 1939 (em uma publicação russa, com sua difusão ocidental ocorrendo apenas em 1960), a busca nesta revisão foi iniciada a partir de 1981. Esse recorte temporal se justifica por marcar o primeiro registro de estudos que combinam os termos utilizados nesta pesquisa, como Geração de Colunas, com o contexto específico do PCS. A escolha desse ponto de partida permite focar em trabalhos mais alinhados ao escopo metodológico adotado neste estudo.

A escolha desses repositórios se deve à sua credibilidade reconhecida na comunidade científica e ao fato de serem frequentemente utilizados em revisões sistemáticas de trabalhos relevantes já consolidados na literatura (PAGE *et al.*, 2021; DYCKHOFF, 1981; WÄSCHER e GAU, 1996a; BELOV e SCHEITHAUER, 2006; SILVA *et al.*, 2014; DA SILVA e SCHOUERY, 2023).

O repositório Scopus, mantida pela Elsevier, é uma das maiores plataformas de dados acadêmicos do mundo, reunindo periódicos e anais de conferências de diversas áreas do conhecimento, como economia, ciência da computação, engenharia e matemática. Sua abrangência e qualidade editorial a tornam uma fonte amplamente

utilizada em revisões sistemáticas.

A base de dados *Web of Science* (WOS) é conhecida por seu perfil altamente seletivo, com uma curadoria criteriosa voltada para periódicos de alto impacto. Os artigos incluídos passam por um rigoroso processo de revisão por pares, e a base mantém um dos índices de citação mais utilizados para avaliar a relevância e a influência científica de publicações.

O repositório IEEE Xplore foi incluído nesta revisão sistemática por sua relevância consolidada nas áreas de engenharia, computação e tecnologias emergentes. Além de reunir artigos revisados por pares e publicações de conferências reconhecidas, o acervo do IEEE cobre temas diretamente relacionados a este trabalho. A escolha por essa base se justifica não apenas pela qualidade técnica dos materiais disponíveis, mas também pela atualidade e aplicabilidade dos estudos encontrados, o que a torna uma fonte valiosa para o desenvolvimento desta pesquisa.

A estratégia de busca utilizou as palavras-chave “One-Dimensional Cutting Stock”, “Column Generation”, “Local Branching” e “Soft Fixing”. A string aplicada foi a seguinte: (“One-Dimensional Cutting Stock Problem” OR “1-D Cutting Stock Problem”) AND (“Column Generation” OR “Local Branching” OR “Soft Fixing”).

Os resultados da busca foram organizados e analisados a partir dos estudos recuperados nas bases selecionadas, seguindo uma estrutura sistemática descrita a seguir:

- 1. Encontrados:** Quantidade de estudos encontrados em cada repositório selecionado, com base na string de busca utilizada na revisão.
- 2. Duplicados:** Quantidade de artigos identificados simultaneamente em mais de uma base de dados.
- 3. Filtrados:** Para refinar o conjunto de artigos da revisão, adotaram-se critérios de exclusão baseados no desalinhamento temático, ausência de abordagem metodológica compatível e baixa qualidade científica, avaliada pela relevância do periódico, rigor metodológico e clareza dos resultados.
- 4. Selecionados:** Para cada artigo filtrado, foi realizada uma leitura mais aprofundada e sistemática, com o objetivo de avaliar sua relevância e a qualidade metodológica do estudo.

Os resultados obtidos após a aplicação dos critérios descritos estão apresentados na Tabela 2.1, que mostra a quantidade de trabalhos publicados que selecionados para compor esta revisão bibliográfica. A tabela permite visualizar a distribuição dos trabalhos nos repositórios e a percepção de qual repositório forneceu mais estudos considerados para essa revisão.

Repositório	Encontrados	Duplicados	Filtrados	Selecionados
Scopus	48	29	13	7
Web of Science (WOS)	41	28	3	2
IEEE Xplore	3	2	0	0
Total	92	59	16	9

Tabela 2.1: Análise Quantitativa dos Repositórios.

A Tabela 2.2 apresenta um conjunto de informações relevantes sobre os trabalhos selecionados para compor esta revisão. São explicitados os nomes dos autores, os repositórios onde os estudos foram encontrados, bem como os veículos de publicação, como periódicos ou anais de conferências, em que os trabalhos foram originalmente divulgados. Essa organização permite uma visão mais clara da origem, da diversidade e da credibilidade das fontes utilizadas ao longo da pesquisa.

Autor	Repositório	Revista / Conferência
SCHEITHAUER <i>et al.</i> (2001)	WOS	Journal of the Operational Research Society
HOLTHAUS (2002)	Scopus	European Journal of Operational Research
BELOV e SCHEITHAUER (2006)	Scopus	European Journal of Operational Research
WONGPRAKORNKUL e CHARNSETHIKUL (2007)	Scopus	Journal of Mathematics and Statistics
ANDRADE <i>et al.</i> (2021)	Scopus	Applied Mathematical Modelling
SILALAHY <i>et al.</i> (2022)	Scopus	BAREKENG: Jurnal Ilmu Matematika dan Terapan
XIAO <i>et al.</i> (2023)	WOS	Mathematical Biosciences and Engineering
DE ARAÚJO SILVA OLIVEIRA <i>et al.</i> (2025)	Scopus	Journal of Heuristics
KAYHAN e TEKEZ (2025)	Scopus	Engineering Optimization

Tabela 2.2: Artigos Selecionados.

A Figura 2.1 ilustra a quantidade de publicações por ano entre os trabalhos selecionados nesta revisão. É possível notar que os estudos estão distribuídos de forma bastante espaçada ao longo do tempo, com aparições pontuais nos anos de 2001, 2002, 2006, 2007, 2021, 2022 e 2023, cada um contribuindo com apenas uma publicação. O ano de 2025 surge como uma exceção discreta, com dois trabalhos identificados, indicando um leve aumento no interesse recente sobre o tema.



Figura 2.1: Quantidade de artigos publicados por ano entre os selecionados.

Apesar de o Problema do Cutting Stock Unidimensional (PCS) ser objeto de estudo desde a década de 1960, quando foi inicialmente formalizado, a combinação da técnica de Geração de Colunas com abordagens complementares, como heurísticas, metaheurísticas ou estratégias de refinamento como o *Soft Fixing*, só passou a ganhar maior destaque nas últimas duas décadas. Essa integração mais sofisticada de métodos reflete uma busca por soluções mais robustas e eficientes, especialmente em instâncias mais complexas.

Portanto, embora o PCS seja um problema clássico e consolidado na literatura, o uso de estratégias híbridas ainda é relativamente recente e mostra-se como um caminho promissor para avanços futuros. A frequência modesta de publicações ao longo dos anos pode indicar tanto a especificidade do tema quanto os desafios técnicos envolvidos em integrar essas abordagens de forma eficaz.

2.2 Análise Qualitativa

Esta seção é dedicada à análise qualitativa dos estudos selecionados, previamente identificados na Seção 2.1. O objetivo dessa análise é examinar com mais profundidade o conteúdo, os métodos e as contribuições de cada trabalho, evidenciando os motivos que justificam sua inclusão nesta revisão e seu valor para a compreensão do estado da arte sobre o PCS.

A abordagem apresentada por SCHEITHAUER *et al.* (2001) propõe uma técnica exata para resolver o PCS utilizando um algoritmo de planos de corte. Em vez de aplicar o tradicional método de Geração de Colunas, os autores adotam uma estratégia baseada na formulação original de Kantorovich, reforçando-a iterativamente com

desigualdades válidas que reduzem a região viável do problema e conduzem a uma solução ótima. O algoritmo desenvolvido evita a necessidade de gerar explicitamente todos os padrões de corte possíveis, o que contribui para maior eficiência computacional, especialmente em instâncias de tamanho intermediário. Ao contrário de muitas abordagens híbridas recentes, o estudo não faz uso de heurísticas inteiras nem da técnica de *Soft Fixing*, concentrando-se exclusivamente em métodos exatos. Os resultados computacionais apresentados demonstram que, apesar da simplicidade conceitual, a abordagem é competitiva e se destaca pela precisão na resolução de instâncias clássicas e de *benchmark* do problema.

O estudo apresentado por HOLTHAUS (2002) explora abordagens baseadas em decomposição para a resolução do Problema de Cutting Stock Unidimensional (PCS), com ênfase em cenários práticos nos quais os comprimentos padrão das barras disponíveis para corte não são fixos, mas sim variados. Para lidar com essa complexidade adicional, os autores propõem o uso da técnica de Geração de Colunas na resolução do problema mestre contínuo, complementada por heurísticas inteiras capazes de construir soluções viáveis a partir dos resultados fracionários. A aplicação de heurísticas de arredondamento e estratégias específicas de ramificação dentro de um esquema de branch-and-price mostrou-se eficaz para aprimorar a qualidade das soluções inteiras obtidas. Embora o trabalho não utilize a técnica de *Soft Fixing*, ele contribui de forma significativa ao demonstrar que a combinação entre decomposição e heurísticas pode ser uma alternativa poderosa na resolução de instâncias mais realistas do PCS, reforçando a importância de estratégias híbridas na otimização de problemas com estruturas complexas.

O trabalho de BELOV e SCHEITHAUER (2006) apresenta um algoritmo de *Branch-and-Cut-and-Price* para os problemas de Cutting Stock Unidimensional (1D-PCS) e Bidimensional em duas etapas (2D-2SC). O método combina a formulação de Gilmore e Gomory com Geração de Colunas, utilizada para tratar o grande número de padrões de corte potenciais, e reforça a relaxação linear por meio da adição de cortes gerais, como os de Chvátal-Gomory e Gomory mistos. Para lidar com a obtenção de soluções inteiras viáveis, os autores incorporam heurísticas específicas, como procedimentos de arredondamento e o *Sequential Value Correction* (SVC), que ajusta soluções fracionárias a configurações factíveis. Os experimentos computacionais evidenciam que, no caso do 1D-PCS, a relaxação já é bastante forte, tornando os cortes de utilidade limitada, enquanto para o 2D-2SC a integração de colunas, cortes e branching leva a resultados superiores em relação a métodos tradicionais, mostrando a robustez e eficiência da abordagem proposta.

WONGPRAKORNKUL e CHARNSETHIKUL (2007) propõem uma formulação integrada e realista para o PCS com múltiplos comprimentos de barras, incorporando simultaneamente as decisões de transporte associadas à entrega dos itens cortados.

Essa integração reflete situações reais em ambientes industriais, onde não apenas o aproveitamento de material é relevante, mas também os custos logísticos envolvidos na distribuição dos produtos finais. Para enfrentar a complexidade do modelo, os autores desenvolveram três abordagens distintas: uma abordagem exata utilizando geração de colunas (A0), uma heurística baseada em geração de colunas com arredondamento de soluções fracionárias (A1) e uma abordagem híbrida que combina a decomposição de Benders com geração de colunas (A2). A estratégia A1 se destacou como a mais promissora, sendo capaz de gerar soluções de boa qualidade em menor tempo computacional, o que a torna atrativa para aplicações práticas em ambientes com alta demanda por eficiência operacional. Os resultados computacionais indicam que a adoção de abordagens heurísticas bem planejadas pode superar limitações de desempenho enfrentadas por métodos exatos, especialmente em instâncias de grande escala. Dessa forma, o artigo contribui significativamente ao avanço de técnicas de resolução aplicadas a variantes integradas e mais próximas da realidade industrial do PCS.

O trabalho de ANDRADE *et al.* (2021) apresenta um modelo de programação inteira mista voltado para a resolução integrada do problema de dimensionamento de lotes e do PCS, com aplicação específica em uma fábrica de molas automotivas. O estudo propõe uma abordagem que visa minimizar, de forma simultânea, os custos de preparação das máquinas e o desperdício de material resultante dos cortes. Para isso, os autores utilizam dados reais extraídos do ambiente produtivo da indústria, conferindo maior robustez e aplicabilidade ao modelo. Embora não sejam empregadas técnicas como Geração de Colunas ou *Soft Fixing*, a proposta destaca-se por integrar dois problemas clássicos da pesquisa operacional em uma única formulação matemática, permitindo uma resolução exata por meio de um solver comercial. Essa contribuição é particularmente relevante por demonstrar como modelos integrados podem melhorar a eficiência operacional em contextos industriais reais.

O artigo de SILALAH *et al.* (2022) apresenta uma abordagem baseada em geração de colunas para resolver o problema de corte unidimensional com foco na minimização do número de padrões distintos utilizados no corte. A proposta se diferencia das abordagens clássicas ao incorporar os custos de setup no modelo, tornando-o mais realista para aplicações industriais. A solução é dividida em duas etapas: inicialmente, determina-se o número mínimo de barras a serem utilizadas; em seguida, utiliza-se essa solução como base para um segundo modelo que minimiza os padrões distintos, aproximando-se do custo real de corte. O estudo não utiliza heurísticas inteiras nem técnicas de *Soft Fixing*, concentrando-se exclusivamente na geração de colunas e na resolução exata via programação linear. Os resultados mostram que é possível obter soluções com baixa variabilidade de padrões e alta eficiência econômica, destacando o potencial da abordagem para ambientes industriais

com alto custo de reconfiguração.

O artigo de XIAO *et al.* (2023) propõe uma abordagem inovadora para o PCS, com ênfase na minimização do número de diferentes padrões de corte utilizados, ao invés de focar exclusivamente no número total de barras ou no desperdício de material. Essa mudança de foco busca alinhar a modelagem matemática às necessidades operacionais das indústrias, onde a variedade de padrões pode representar um gargalo significativo em termos de *setup*, armazenamento e complexidade na linha de produção. Para isso, o autor desenvolve uma formulação de Programação Linear Inteira que incorpora variáveis auxiliares responsáveis por identificar a diversidade de padrões na solução final. A contribuição central do trabalho está em oferecer uma solução mais aderente ao contexto prático industrial, com foco na simplificação dos processos e no controle da variedade de padrões. Essa abordagem mostra-se especialmente útil em ambientes de produção que valorizam a padronização e a redução de complexidade operacional, ainda que em alguns casos isso possa vir acompanhado de um leve aumento no uso de matéria-prima.

O trabalho de KAYHAN e TEKEZ (2025) propõe uma nova heurística denominada OPTICUT, voltada para a resolução do PCS com o objetivo adicional de minimizar o número de padrões de corte utilizados. A motivação central do trabalho é baseada na observação de que, em contextos industriais, a simplicidade e repetibilidade dos padrões são desejáveis, tanto para facilitar a operação quanto para reduzir o tempo de *setup*. A heurística OPTICUT adota uma abordagem iterativa e construtiva, na qual são geradas soluções iniciais que são posteriormente refinadas. O algoritmo se baseia em técnicas de busca local e métodos de melhoria que avaliam o impacto de substituições e realocações de peças entre os padrões de corte. A eficiência da proposta é validada por meio de experimentos computacionais com conjuntos de instâncias clássicas da literatura, onde o OPTICUT demonstrou bom desempenho, especialmente em termos de simplicidade de padrões e tempo de execução.

O estudo de DE ARAÚJO SILVA OLIVEIRA *et al.* (2025) propõe uma abordagem baseada em *Local Branching* para resolver o problema de corte unidimensional com múltiplos períodos, levando em consideração custos de antecipação, atraso e *setup*. Os autores modelam o problema como uma programação inteira mista, utilizando uma heurística inteira que explora vizinhanças de soluções viáveis com o objetivo de reduzir o tempo de resolução e melhorar a qualidade das soluções. O trabalho contribui ao demonstrar a eficácia da técnica de *Local Branching* em ambientes complexos e realistas, consolidando-a como uma estratégia promissora para problemas de corte com múltiplos critérios operacionais.

Autores	Geração de Colunas	Heurística Inteira	Soft Fixing
SCHEITHAUER <i>et al.</i> (2001)		X	
HOLTHAUS (2002)	X	X	
BELOV e SCHEITHAUER (2006)	X	X	
WONGPRAKORNKUL e CHARNSETHIKUL (2007)	X	X	
ANDRADE <i>et al.</i> (2021)		X	
SILALAHY <i>et al.</i> (2022)		X	
XIAO <i>et al.</i> (2023)	X	X	
KAYHAN e TEKEZ (2025)		X	
DE ARAÚJO SILVA OLIVEIRA <i>et al.</i> (2025)	X	X	
Nosso Método	X	X	X

Tabela 2.3: Técnicas utilizadas no PCS nos trabalhos selecionados.

A Tabela 2.3 evidencia como, ao longo dos anos, as heurísticas inteiras dominaram as estratégias adotadas para resolver o Problema de Cutting Stock Unidimensional (PCS). Apesar disso, observa-se que o algoritmo de Geração de Colunas segue como uma ferramenta central na literatura, mantendo sua relevância mesmo com o passar do tempo. No entanto, o uso de abordagens híbridas, que combinem Geração de Colunas com técnicas mais recentes ou sofisticadas, como Local Branching ou *Soft Fixing*, ainda aparece com pouca frequência. Essa escassez revela uma lacuna importante nas investigações atuais. É justamente nesse ponto que esta dissertação se insere, ao propor uma integração entre Geração de Colunas e Restrições de *Soft Fixing*, complementada pelo uso de uma heurística inteira. Essa proposta contribui para preencher esse espaço ainda pouco explorado, mostrando uma alternativa promissora para melhorar a qualidade das soluções e o desempenho computacional.

Capítulo 3

Definição e Decomposição do Problema

Neste capítulo, apresenta-se o Problema de Cutting Stock (PCS), originalmente formulado por KANTOROVICH (1960), bem como uma abordagem de decomposição desenvolvida posteriormente para facilitar sua resolução. A Seção 3.1 discute a definição do problema e suas principais aplicações em contextos industriais. Já a Seção 3.2 aborda a formulação matemática tradicional do problema, além da técnica de decomposição proposta por GILMORE e GOMORY (1961), leva à obtenção de soluções ótimas do Problema de Cutting Stock, de uma forma que se mostrou, empiricamente, mais eficiente e escalável.

3.1 Motivação do Problema

O Problema de Cutting Stock Unidimensional (PCS) surge em diversos contextos industriais onde materiais compridos, como bobinas de papel, barras metálicas, rolos de tecido ou cabos elétricos, precisam ser cortados em pedaços menores para atender a uma demanda específica. A princípio, pode parecer uma tarefa puramente operacional, mas por trás dessa simplicidade aparente, esconde-se um desafio combinatório complexo, com impactos diretos em custos, desperdício de material e eficiência produtiva.

Em muitas dessas indústrias, o desperdício resultante de cortes mal planejados representa uma parcela significativa do custo total (JARIYAVAJEE *et al.*, 2025; LAMERS, 2021). Assim, encontrar formas de minimizar a sobra de material e, ao mesmo tempo, atender com precisão à demanda de itens com diferentes tamanhos se torna uma prioridade logística e financeira. À medida que a variedade de itens cresce, o número de possíveis combinações de corte aumenta drasticamente, o que torna inviável a resolução do problema por métodos ingênuos ou puramente enumerativos.

Do ponto de vista científico, o problema tem sido estudado há décadas, sendo considerado clássico na área de Pesquisa Operacional. Soluções baseadas em Programação Linear Inteira e, mais especificamente, na técnica de Geração de Colunas se destacam por sua capacidade de lidar com instâncias de grande escala, ao decompor o problema em um problema mestre e subproblemas estruturados. Essa abordagem permite uma exploração mais eficiente do espaço de soluções, especialmente quando combinada com estratégias de otimização dedicadas.

3.2 Formulação Matemática

O PCS pode ser descrito considerando peças mestres de tamanho W , que devem ser cortadas em peças menores de tamanhos w_i ($i = 1, \dots, m$), respeitando a condição $w_i \leq W$ para todo i . Cada tipo de peça w_i possui uma demanda associada d_i . Define-se $y_k \in \{0, 1\}$ como uma variável binária que indica se a peça mestre k , onde $k \in \{1, \dots, K\}$, foi utilizada, e $x_{ik} \in \mathbb{Z}_+$ como o número de peças do tipo w_i obtidas do corte da peça mestre k . Dessa forma, a combinação dos valores de x_{ik} estabelece um padrão de corte para cada peça mestre. A formulação de Kantorovich para o problema se baseia nas variáveis que acabamos de discutir e é definida como:

$$(PCS) \min \quad \sum_{k=1}^K y_k \quad (3.1)$$

sujeito a

$$\sum_{k=1}^K x_{ik} \geq d_i \quad i = 1, \dots, m \quad (3.2)$$

$$\sum_{i=1}^m w_i x_{ik} \leq W y_k \quad k = 1, \dots, K \quad (3.3)$$

$$x_{ik} \in \mathbb{Z} \quad k = 1, \dots, K, i = 1, \dots, m \quad (3.4)$$

$$y_k \in \{0, 1\} \quad k = 1, \dots, K \quad (3.5)$$

Na formulação do PCS, a função objetivo (3.1) busca minimizar a quantidade de peças mestres utilizadas na solução do problema. A família de Restrições (3.2) asseguram que a demanda de cada peça w_i seja completamente atendida, enquanto a família de Restrições (3.3) garantem que o número de cortes de cada w_i em uma peça mestre não exceda o comprimento disponível, respeitando o limite físico de corte.

Na prática, essa formulação é aplicada definindo previamente um limite superior K para o número de peças mestre utilizadas, geralmente por meio de alguma heurística. Além disso, para reduzir o impacto da simetria presente na formula-

ção, que pode prejudicar o desempenho do solver, são adicionadas restrições do tipo $y_{k+1} \leq y_k$, para todo $k \in \{1, \dots, K - 1\}$, limitando a geração de soluções equivalentes.

Exemplo 1. Considere o seguinte exemplo do PCS: deseja-se cortar peças mestres de comprimento 10.000 unidades para produzir cinco tipos de peças menores, com tamanhos 654, 1393, 2157, 2680 e 4586. As respectivas demandas para cada tipo de peça são 5, 6, 8, 1 e 5. O objetivo é encontrar combinações de corte que atendam a essas demandas com o menor desperdício possível.

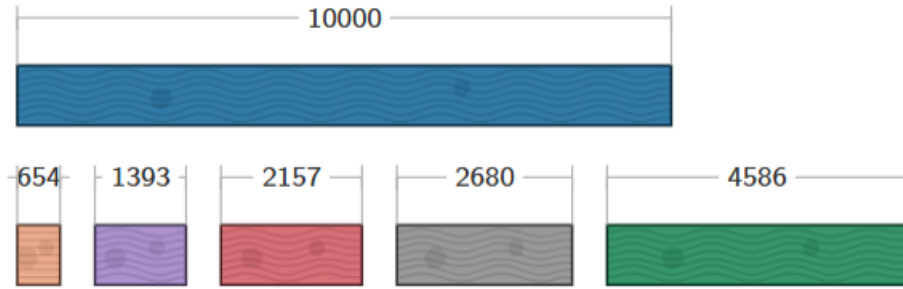


Figura 3.1: Peça Mestre e as Peças Menores do Exemplo 1.

Fonte: O autor.

Ao resolver o exemplo utilizando a formulação de Kantorovich, adotou-se como valor de K a soma das demandas ($K = \sum_i^m d_i$), o que fornece um limitante superior viável ainda que não necessariamente o mais eficiente. Com esse valor fixado, a formulação foi resolvida por meio do método Branch-and-Bound. O resultado foi um conjunto de padrões de corte que atende exatamente às demandas estabelecidas, minimizando o desperdício de material. Os padrões utilizados estão apresentados a seguir:

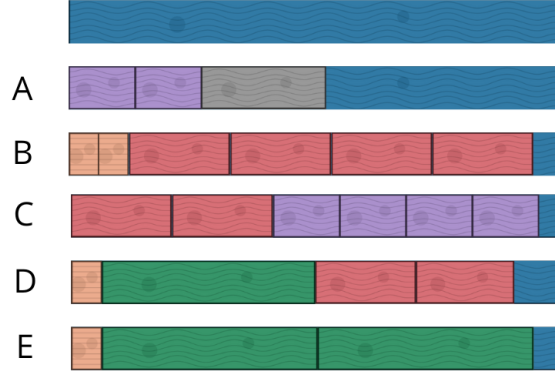


Figura 3.2: Padrões de Corte pela Formulação de Kantorovich.

Fonte: O autor.

Para atender às demandas de cada uma das peças menores, foram necessárias 6 peças mestres, distribuídas da seguinte forma: 1 unidade de cada nos padrões A, B, C e D, e 2 unidades no padrão E.

Para lidar com a complexidade do PCS, que é um problema pertencente à classe dos problemas NP-difíceis, GILMORE e GOMORY (1961) propuseram uma estratégia de decomposição: em vez de resolver o problema completo de uma vez só, eles o dividiram em duas partes mais fáceis de tratar, um Problema Mestre (PM) e um subproblema auxiliar.

O Problema Mestre, considera todos os padrões de corte possíveis, de forma que toda a demanda de peças menores w_i seja atendida, usando o menor número possível de peças mestres. Cada padrão de corte é representado por uma coluna j , onde a_{ij} indica quantas peças do tipo w_i aparecem nesse padrão. A variável x_j indica quantas vezes aquele padrão específico será usado na solução final. Além disso, associamos um custo c_j para cada padrão, normalmente relacionado a custos de *set-up* para cortar a peça mestre no padrão desejado e ao desperdício de material que dele resulta.

$$(PM) \quad \min \quad \sum_{j=1}^n c_j x_j \quad (3.6)$$

sujeito a

$$\sum_{j=1}^n a_{ij} x_j \geq d_i \quad \forall i = 1, \dots, m \quad (3.7)$$

$$x_j \in \mathbb{Z}_+ \quad \forall j = 1, \dots, n \quad (3.8)$$

A função objetivo (3.6) busca minimizar o custo total associado à utilização dos

padrões de corte escolhidos. Caso os custos de todas as peças mestres sejam iguais (o que é feito neste trabalho, onde $c_j = 1$ é assumido para todo $j \in J$). A função objetivo pode ser simplificada para a minimização da quantidade total de peças utilizadas, sendo representada por $\sum_{j=1}^n x_j$, onde $n = |J|$. As Restrições em (3.7) garantem que a solução atenda integralmente à demanda de cada tipo de peça w_i . Por fim, as condições expressas em (3.8) asseguram que as variáveis x_j assumam apenas valores inteiros, respeitando a natureza discreta do problema.

Um algoritmo de Geração de Colunas, se inicia com a seleção de um subconjunto $J' \subseteq J$, de todas as colunas (padrões de corte) existentes para a instância, formando um Problema Mestre Restrito (PMR). Numa resolução do PM, através do PMR, trabalha-se, como é usual, com a relaxação linear do PMR, resultando no Problema Mestre Restrito Relaxado (PMRR). Por sua vez, a resolução do PMRR, que é feita em tempo polinomial, fornece as informações necessárias para a formulação de um subproblema de geração de colunas, em um procedimento que iterativamente nos levará a aumentar o número colunas de PMR/PMRR e chegar, eventualmente, a soluções comprovadamente ótimas para a relaxação linear de PM.

O subproblema é formulado como um Problema da Mochila Inteira, onde o que se deseja é preencher a peça mestre de tamanho W com combinações de peças w_i que resultem em novos padrões de corte capazes de reduzir (ou pelo menos, não aumentar) o valor da solução ótima do PMRR que se tem em mãos. Mais especificamente, a função do subproblema é encontrar colunas que possam melhorar a solução atual do PMRR, ou seja, identificar novos padrões de corte, ainda não incluídos no PMR/PMRR, que ajudem a reduzir o número total de peças mestre a utilizar numa eventual solução ótima do PM.

A cada iteração, o algoritmo avalia se há necessidade de incluir novos padrões de corte ao PMR/PMRR. Ao final do processo, obtém-se uma solução do PMRR que: (a) é comprovadamente ótima para a relaxação linear do PM e (b) fornece um limite inferior para a solução inteira ótima do PM. Além disso, como o conjunto de colunas J' gerado tende a ser pequeno, pode-se resolver o PMR diretamente, para tentar obter uma solução viável para o PM, assim formulado, contenha de fato uma solução viável para o problema original.

A formulação do Problema da Mochila, utilizada como subproblema no PCS, é construída com base na solução dual ótima obtida a partir do PMRR ($\bar{\pi}$). Essa solução dual permite definir a função objetivo do subproblema, que representa o custo reduzido associado a uma variável (ou padrão de corte) ainda não incluída no PMRR. Caso esse custo reduzido seja negativo, a variável é considerada promissora e deve ser adicionada ao modelo. A seguir, essa formulação é apresentada pelas equações (3.9)(3.11).

$$1 - \max \sum_{i=1}^m \bar{\pi}_i y_i \quad (3.9)$$

sujeito a

$$\sum_{i=1}^m w_i y_i \leq W \quad (3.10)$$

$$y_i \in \mathbb{Z}_+ \quad \forall i = 1, \dots, m \quad (3.11)$$

Para a resolução do subproblema, pode-se recorrer tanto a *solvers* comerciais quanto a algoritmos específicos de Programação Dinâmica, como os propostos por PISINGER (1995), que se destacam pela eficiência e simplicidade na implementação em contextos de Problemas da Mochila.

A formulação de Gilmore-Gomory para o PCS resulta da aplicação da decomposição de Dantzig-Wolfe, detalhada no Apêndice B.

Capítulo 4

Metodologia

Este capítulo descreve a metodologia proposta para incorporar Restrições de *Soft Fixing* ao algoritmo de Geração de Colunas. A Seção 4.1 apresenta os principais componentes e estruturas envolvidos na construção do algoritmo base de Geração de Colunas. Em seguida, a Seção 4.2 detalha as Restrições de *Soft Fixing* desenvolvidas especificamente para o contexto do Problema de Cutting Stock (PCS).

4.1 Geração de Colunas

Nesta seção, são apresentados os principais elementos que estruturam o algoritmo de Geração de Colunas, começando pela definição da solução inicial, que serve como ponto de partida, até a formulação do PMRR e do Subproblema. A Seção 4.1.1 mostra como essa solução inicial é construída e qual o seu papel dentro do algoritmo. Depois, na Seção 4.1.2, é explorada a modelagem do PMRR, que funciona como a base para aplicar a técnica de Geração de Colunas de maneira eficiente. Por fim, a Seção 4.1.3 trata do Subproblema, responsável por identificar novos padrões de corte que podem ser inseridos no PMRR, permitindo refinar progressivamente a solução obtida.

4.1.1 Solução Inicial

De forma a garantir a viabilidade do PMR e, conseqüentemente, de sua relaxação linear, PMRR, deve-se inicializa-lo com colunas (variáveis) que contenham uma solução viável para o mesmo. No presente trabalho, uma solução viável inicial utilizada foi definida da seguinte forma:

$$A = \begin{bmatrix} \left\lfloor \frac{W}{w_1} \right\rfloor & 0 & \cdots & 0 \\ 0 & \left\lfloor \frac{W}{w_2} \right\rfloor & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \left\lfloor \frac{W}{w_m} \right\rfloor \end{bmatrix} \quad (4.1)$$

Como é possível observar, a matriz apresentada em (4.1) define uma base viável para a solução inicial do PCS, formada exclusivamente por m padrões de corte distintos, um para cada tipo de peça $\{w_i : i = 1, \dots, m\}$ que se deseja obter.

4.1.2 Problema Mestre Restrito Relaxado

No PMR e, da mesma forma, na sua versão relaxada, PMRR (que desconsidera as restrições de integralidade das variáveis imposta por PMR), o vetor de custos c foi definido atribuindo o valor $c_j = 1$ para todo j , ou seja, todos os padrões de corte têm o mesmo custo. Sob essa definição, a formulação do PMRR é dada por:

$$(PMRR) \min \sum_{j=1}^n x_j \quad (4.2)$$

sujeito a

$$\sum_{j=1}^n a_{ij} x_j \geq d_i \quad \forall i = 1, \dots, m \quad (4.3)$$

$$x_j \in \mathbb{R}_+ \quad \forall j \in J' \quad (4.4)$$

Deve-se observar que a formulação do PMR distingue-se da do PMRR simplesmente por impor que $x_j \in \mathbb{Z}_+$, $\forall j \in J'$.

4.1.3 Subproblema de Geração de Colunas

O subproblema de geração de colunas associado à PMRR corresponde ao seguinte Problema da Mochila Inteira (PMI):

$$1 - \max \sum_{i=1}^m \bar{\pi}_i y_i \quad (4.5)$$

sujeito a

$$\sum_{i=1}^m w_i y_i \leq W \quad (4.6)$$

$$y_i \in \mathbb{Z}_+ \quad \forall i = 1, \dots, m \quad (4.7)$$

Na formulação (4.5)-(4.7), a função objetivo é construída a partir dos coeficientes $\bar{\pi}_i$, para $i = 1, \dots, m$, que correspondem aos valores das variáveis duais obtidos na solução do PMRR. As variáveis y_i , por sua vez, indicam quantas peças de dimensão w_i compõem o novo padrão de corte que, em conjunto, não podem exceder o comprimento da peça mestre, como imposto por (4.6). Assim, a função objetivo corresponde ao custo reduzido associado à variável não básica vinculada ao padrão gerado. Caso esse custo reduzido seja positivo, o critério de parada do algoritmo de Geração de Colunas é alcançado, sinalizando que a solução que se tem em mãos corresponde a uma solução ótima para a relaxação linear do PMR, considerando-se todos os padrões de corte distintos existentes para a instância considerada. Isso inclui tanto os padrões explicitamente considerados pelo PMRR quanto aqueles que este considera de forma indireta, através do Subproblema de Geração de Colunas. Caso contrário, a coluna (variável) gerada deve ser acrescida ao PMR/PMRR, expandindo J' com um índice a ela associado.

Dessa forma, a matriz A recebe, como nova coluna, a solução y obtida no subproblema. Em seguida, o PMRR, expandido com esta nova coluna, é resolvido novamente, e os valores das variáveis duais são atualizados e enviados de volta para o subproblema, repetindo-se esse ciclo até que a solução encontrada apresente um custo reduzido maior ou igual a zero. Quando isso acontece, significa que não há mais variáveis não básicas x_j , $j \in J \setminus J'$, com custo reduzido negativo, atendendo, portanto, ao critério de otimalidade do PMRR.

O Exemplo Ilustrativo (4.8) ilustra essa atualização na matriz de restrições do PMRR após uma iteração do algoritmo, mostrando como a nova coluna correspondente ao padrão y é incorporada. Além disso, como cada nova coluna adicionada representa um novo padrão de corte, também é incluída uma variável associada, cujos coeficientes correspondem aos elementos dessa coluna. Da mesma forma, a função objetivo passa a considerar essa variável, atribuindo-lhe o custo referente à utilização da respectiva peça mestre.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ \vdots & \ddots & & \vdots \\ a_{m1} & \cdots & & a_{mj} \end{bmatrix} \Rightarrow A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & y_1 \\ \vdots & \ddots & & \vdots & \vdots \\ a_{m1} & \cdots & & a_{mj} & y_m \end{bmatrix} \quad (4.8)$$

4.2 Restrições de Soft Fixing

Considerando a definição adotada para o PMRR, e assumindo que o processo de geração de colunas foi concluído, a solução obtida corresponde à solução ótima da relaxação linear do PM. Além disso, como o PMR é construído a partir dos padrões viáveis gerados ao longo do processo, ele tende a conter soluções inteiras viáveis para

o PCS, o que o torna uma base promissora para a busca da otimalidade inteira.

Considerando a resolução ótima do PMR, a solução obtida é garantidamente viável para o PCS e pode ser considerada ótima se seu valor for inteiro ou estiver a menos de uma unidade da solução da relaxação (PMRR). Caso contrário, aplica-se uma Restrição de Soft Fixing (RSF) para explorar soluções inteiras na vizinhança da atual, na tentativa de melhorar o resultado mantendo a viabilidade.

A técnica de *Soft Fixing* é uma heurística do tipo *Large Neighborhood Search* (LNS) que busca explorar a vizinhança de uma solução conhecida, restringindo o espaço de busca por meio de cortes controlados. Outras abordagens baseadas em LNS incluem a *Relaxation Induced Neighborhood Search* (RINS), proposta por DANNA *et al.* (2005), e a *Feasibility Pump* (FP), apresentada por FISCHETTI *et al.* (2005).

A técnica RINS pode ser interpretada como uma forma de *Hard Fixing*, pois fixa um subconjunto das variáveis e resolve o subproblema resultante, explorando uma vizinhança mais restrita. Por outro lado, as Restrições de *Soft Fixing* introduzem um corte, forçando a nova solução a permanecer em um raio de proximidade em relação à solução atual. Essa abordagem é mais flexível, embora possa gerar subproblemas maiores e computacionalmente mais exigentes.

Já a *Feasibility Pump* atua nas etapas iniciais da resolução de problemas de Programação Inteira Mista, com o objetivo de encontrar rapidamente soluções viáveis. O método alterna entre a resolução da relaxação linear e o arredondamento de soluções fracionárias, em um processo iterativo. Diferentemente do *Soft Fixing*, que é empregado para refinar soluções conhecidas, a FP busca viabilidade antes da obtenção de uma solução de boa qualidade.

Para explorar soluções viáveis do PCS, foram consideradas diferentes variantes de Restrições de *Soft Fixing* (RSF), avaliadas de forma individual. Essa abordagem permitiu identificar, com mais clareza, os pontos fortes e limitações de cada uma em termos de desempenho e impacto na qualidade da solução.

A Figura 4.1 ilustra o fluxograma que descreve o funcionamento do algoritmo de Geração de Colunas com *Soft Fixing*, desde a construção da solução inicial até a resolução final do problema inteiro restrito. O processo começa com a obtenção de uma solução básica viável, que serve de ponto de partida para a execução do algoritmo. Em seguida, resolve-se o Problema Mestre Restrito em sua versão inteira. Assim, o algoritmo de Geração de Colunas fornece um limitante inferior (Lb) e a solução do PMR fornece um limitante superior (Ub).

Se a otimalidade não for alcançada nesse momento ($Ub - Lb \geq 1$), inicia-se a fase de *Soft Fixing*. Nessa etapa, um laço de repetição é ativado sobre o parâmetro α , que define a taxa de fixação da solução corrente, inicialmente estabelecida em 0,9 nos casos em que há decremento. A cada iteração, o algoritmo é executado novamente, agora com as Restrições de *Soft Fixing* restringindo o PMRR.

A atualização de α segue uma lógica simples: se uma nova coluna for gerada, o parâmetro retorna ao valor inicial (0,9); caso contrário, é reduzido em 0,1. O processo se repete até que α atinja um valor inferior a 0,1, momento em que ocorre o critério de parada desta fase. Por fim, realiza-se uma última resolução do Problema Mestre Restrito, novamente em sua versão inteira, encerrando assim o algoritmo.

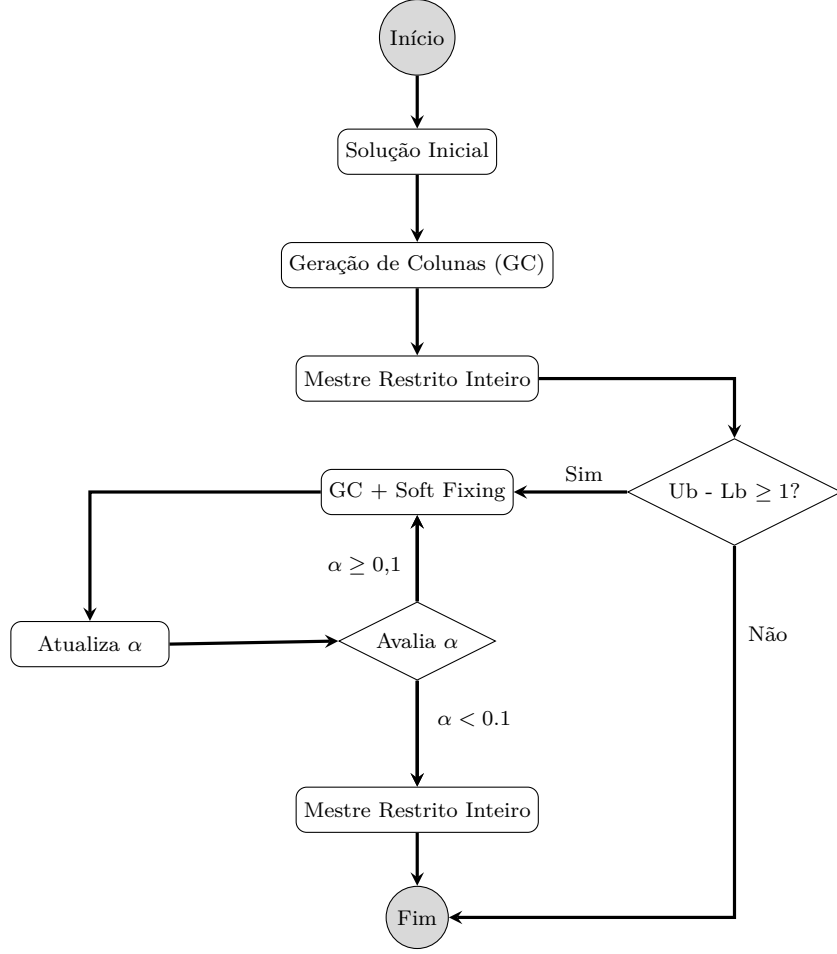


Figura 4.1: Fluxograma do Geração de Colunas com *Soft Fixing*.

I. Soft Fixing 1: Soma das Variáveis Não Nulas

A primeira abordagem aplicada consistiu em fixar uma parte do somatório das variáveis associadas à solução encontrada pelo algoritmo de Geração de Colunas. Para isso, foi adicionada a seguinte restrição:

$$\sum_{j \in J^*} x_j \geq \left\lceil \alpha \sum_{j \in J^*} \bar{x}_j \right\rceil, \quad \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.9)$$

Na Restrição (4.9), o parâmetro α define a porcentagem da solução que será mantida para a próxima iteração. Inicialmente, α é fixado em 0,9, o que corresponde a preservar 90% da solução encontrada pelo algoritmo de Geração de Colunas. Com essa restrição aplicada ao PMR, o algoritmo é executado novamente. Se nenhuma nova coluna for gerada, o valor de α é reduzido em 0,1, e o processo se repete. O ciclo continua até que α atinja 0,1 e ainda assim nenhuma coluna seja encontrada, momento em que o critério de parada é atingido. Caso uma nova coluna seja gerada em qualquer iteração, α é

redefinido para 0,9 e o ciclo recomeça. Por fim, na restrição, $\bar{x}_j$ representa o valor que a variável x_j assume na solução atual do algoritmo de Geração de Colunas. O conjunto das variáveis não nulas é definido por J^* .

II. Soft Fixing 2: Limitante pelas Demandas

Nessa abordagem, foi adicionada a Restrição (4.10) que utiliza as demandas de cada peça como limite inferior para o somatório dos valores das variáveis associadas à solução encontrada pelo algoritmo de Geração de Colunas:

$$\sum_{j \in J^*} x_j \geq \lceil \alpha d_i \rceil, \quad i = 1, \dots, m, \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.10)$$

As operações de decremento do parâmetro α utilizado seguem a mesma lógica adotada na fixação da soma de variáveis e d_i é a demanda da i -ésima peça.

III. Soft Fixing 3: Soma das Variáveis Nulas

A abordagem da Restrição (4.11) é semelhante à primeira, porém, neste caso, são consideradas as variáveis que não fazem parte da solução atual do problema.

$$\sum_{j \in J/J^*} x_j \geq \left\lceil \alpha \sum_{j \in J/J^*} \bar{x}_j \right\rceil, \quad \alpha \in \{0,1, 0,2, \dots, 0,9\} \quad (4.11)$$

Nesta abordagem, o parâmetro α segue o mesmo esquema das restrições anteriores, com a diferença de que, neste caso, α é inicializado em 0,1 e incrementado a cada iteração, em vez de ser decrementado. O conjunto de variáveis nulas é definido por J/J^*

IV. Soft Fixing 4: Demanda Atendida

Com a Restrição de *Soft Fixing* (4.12), são mantidas partes das demandas de cada peça que já foram atendidas na solução obtida pelo algoritmo de Geração de Colunas.

$$\sum_{j \in J^*} a_{ij} x_j \geq \left\lceil \alpha \sum_{j \in J^*} a_{ij} \bar{x}_j \right\rceil, \quad i = 1, \dots, m, \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.12)$$

Dessa maneira, são estabelecidos limites inferiores para as variáveis associadas às colunas presentes na solução, determinando um valor mínimo para a soma dessas variáveis na iteração em que a restrição é aplicada. Esse valor mínimo

corresponde a um percentual da quantidade de peças cortadas na solução corrente, definido pelo parâmetro α . Caso nenhuma nova coluna seja gerada, α é decrementado sucessivamente até que atinja o critério de parada estabelecido.

V. Soft Fixing 5: Uso de Padrões

Com essa abordagem, é estabelecido um limite mínimo para a utilização dos padrões de corte que compõem a solução obtida pelo algoritmo de Geração de Colunas.

$$x_j \geq \lceil \alpha \bar{x}_j \rceil, \quad \forall j \in J^*, \quad \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.13)$$

Dessa forma, é definido um percentual mínimo α para a utilização de cada padrão de corte presente na solução original, antes da adição dessa restrição. O parâmetro α segue o mesmo procedimento de decremento aplicado nas abordagens anteriores, até que o critério de parada do algoritmo seja atingido.

VI. Soft Fixing 6: Soma das Entradas da Matriz de Restrições

Nesta abordagem, é adicionada a Restrição (4.14) que impõe um limite inferior para as variáveis, baseado no somatório das entradas da matriz de restrições do problema, considerando apenas as colunas que compõem a solução obtida pelo algoritmo de Geração de Colunas.

$$\sum_{j \in J^*} \sum_{i=1}^m a_{ij} x_j \geq \left\lceil \alpha \sum_{j \in J^*} \sum_{i=1}^m a_{ij} \bar{x}_j \right\rceil, \quad \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.14)$$

Nesse caso, o parâmetro α é ajustado seguindo a lógica de decremento, até que o critério de parada do algoritmo seja atingido.

VII. Soft Fixing 7: Combinação de Soft Fixing 4 e 5

Nesta abordagem, são aplicadas sequencialmente as Restrições de *Soft Fixing* 4 e 5. Inicialmente, a Restrição (4.12) é incorporada ao PMR e o algoritmo de Geração de Colunas é executado. Em seguida, (4.12) é removida e a Restrição (4.13) é adicionada, dando início a uma nova execução do algoritmo. Ao final, a Restrição (4.13) também é removida e verifica-se se o parâmetro α deve ser decrementado conforme o critério de atualização do algoritmo.

VIII. Soft Fixing 8: Combinação de Soft Fixing 5 e 4

Nesta abordagem, são aplicadas sequencialmente as Restrições de *Soft Fixing* 5 e 4. Primeiro, a Restrição (4.13) é incorporada ao PMR e o algoritmo de Geração de Colunas é executado. Em seguida, a Restrição (4.13) é removida e

a Restrição (4.12) é inserida, dando início a uma nova execução do algoritmo. Por fim, a Restrição (4.12) é removida, e então verifica-se se o parâmetro α deve ser decrementado conforme o critério de atualização estabelecido.

IX. Soft Fixing 9: Demanda Atendida por Novos Padrões

Na Restrição (4.15), aplica-se a mesma abordagem utilizada em (4.12), mas considerando as variáveis que não fazem parte da solução obtida pelo algoritmo de Geração de Colunas.

$$\sum_{j \in J/J^*} a_{ij} x_j \geq \left\lceil \alpha \sum_{j \in J/J^*} a_{ij} \bar{x}_j \right\rceil, \quad i = 1, \dots, m, \alpha \in \{0,9, 0,8, \dots, 0,1\} \quad (4.15)$$

Com isso, concluímos a apresentação da metodologia proposta. A seguir, são mostrados os resultados obtidos a partir de sua aplicação, com destaque para a análise do desempenho das Restrições de *Soft Fixing* nas instâncias testadas.

Capítulo 5

Experimentos Computacionais

Com a metodologia previamente definida, este capítulo apresenta os principais resultados obtidos a partir da aplicação do algoritmo de Geração de Colunas em conjunto com as Restrições de Soft Fixing. Os desempenhos foram analisados tendo como referência os valores ótimos fornecidos pela formulação de Kantorovich. As instâncias de teste foram geradas conforme o método descrito na Seção 5.1, enquanto os resultados da execução do algoritmo de Geração de Colunas são detalhados na Seção 5.2.

Em seguida, a Seção 5.3 aborda a obtenção das soluções ótimas de referência, e a Seção 5.4 apresenta os resultados obtidos com a aplicação da etapa de *Soft Fixing*.

Os experimentos foram realizados em um ambiente com processador Intel Core i7-11800H (4,60 GHz) e 16 GB de memória RAM. A implementação foi realizada em Python 3.11.3, utilizando o solver Gurobi Optimizer 9.1.2 para a resolução do PMRR e do subproblema de Geração de Colunas. Para garantir maior controle sobre o desempenho, os parâmetros de pré-solver, heurísticas e planos de corte do Gurobi foram desativados.

5.1 Instâncias Geradas

Para o PCS, as situações reais utilizadas nas indústrias são consideradas muito pequenas e simples de resolver, como foi concluído por WÄSCHER e GAU (1993). Para contornar esse problema, WÄSCHER e GAU (1995, 1996b) desenvolveram um algoritmo para gerar instâncias para o PCS mais difíceis de se resolver empiricamente (ou seja, instâncias onde, com maior frequência, soluções ótimas não inteiras estão associadas ao PMRR). No algoritmo, é possível definir a quantidade de peças que serão particionadas e seus tamanhos e definir a demanda de atendimento de cada uma dessas peças. As instâncias são geradas passando como parâmetros do algoritmo: o comprimento da peça mestre, a quantidade de peças de comprimentos distintos que serão particionadas, o intervalo que contenha os comprimentos

possíveis para as peças e um fator de multiplicidade. O algoritmo é descrito a seguir:

I) Determinar o tamanho das instâncias

O tamanho da instância é determinado pelo número de peças de tamanhos distintos que serão cortados a partir da peça mestre. No algoritmo, são geradas instâncias de tamanho 10, 20, 30, 40 e 50.

II) Determinar o tamanho de cada uma das peças a serem cortadas e da peça mestre

Como tamanho de peça mestre, foi estabelecido $L = 10000$, e para determinar os tamanhos das peças que serão cortadas foram estabelecidos valores aleatórios dentro de uma distribuição uniforme. Assim, foram estabelecidos 4 tipos de instâncias baseadas nessas distribuições, onde o parâmetro b assume um valor para a distribuição que é utilizado para obter os valores l_i de cada uma das peças de forma que $l_i \in [1, b.L]$. Onde $b \in \{0.25, 0.5, 0.75, 1.0\}$.

III) Atribuir a totalidade das demandas

É definido um parâmetro T , que representa a soma das demandas de todas as peças menores. Esse parâmetro T é calculado como $T = m.\bar{d}$, onde m é o número de peças de tamanhos distintos, e $\bar{d}$ é um fator de multiplicidade. Para cada instância, $\bar{d}$ assume um valor pertencente ao conjunto $\{10, 20, 30, 40, 50\}$.

Após calcular o parâmetro T , essa demanda total é distribuída entre as peças menores, determinando assim a demanda a ser atendida para cada peça a ser cortada.

IV) Determinar a demanda de cada peça

As demandas de cada peça são determinadas utilizando parâmetros gerados aleatoriamente. Esses parâmetros são representados por R_i , onde $i = 1, \dots, m$, ou seja, há um parâmetro para cada peça. O valor de cada R_i é um número aleatório pertencente ao intervalo $[1, T]$. Com esses parâmetros, cada demanda d_i é definida como:

$$d_i = \left\lfloor \frac{R_i}{R_1 + R_2 + \dots + R_m} \right\rfloor \cdot T \quad \text{onde } i = 1, \dots, m-1$$

e

$$d_m = T - \sum_{i=1}^{m-1} d_i$$

Com esse algoritmo, foram geradas 1600 instâncias, sendo 320 para cada um dos tamanhos mencionados anteriormente. O processo de distribuição uniforme descrito no algoritmo foi utilizado para definir o comprimento das peças. Assim, foram geradas 80 instâncias para cada tipo de distribuição em cada tamanho de instância.

5.2 Análise com Geração de Colunas

Executando o algoritmo de geração de colunas nas 1600 instâncias geradas, foi encontrada a solução ótima para 1.148 instâncias, ou seja, a diferença entre o limitante inferior e o superior foi inferior a 1 unidade. Nas 452 instâncias em que não foram encontradas as soluções ótimas foram aplicadas as Restrições de *Soft Fixing*.

Tamanho	Instâncias
10	61
20	78
30	100
40	105
50	108

Tabela 5.1: Instâncias sem ótimo por tamanho.

A Tabela 5.1 apresenta a distribuição das instâncias que não alcançaram a otimalidade com o algoritmo de Geração de Colunas, classificadas conforme o tamanho das instâncias. Essa organização facilita a análise sobre a relação entre o porte do problema e a dificuldade em obter soluções ótimas.

Como se pode observar, mesmo para instâncias menores, como as de tamanho 10 e 20, ainda há uma quantidade considerável de casos em que, em tese, enumeração implícita é necessária para se obter uma solução ótima para o PCS. Vale notar que o número de instâncias nesta situação tende a aumentar, com o aumento do tamanho da instância, se mostrando mais expressivo para tamanho 50.

Tipo de Distribuição	Instâncias
0.25	155
0.50	46
0.75	113
1.00	138

Tabela 5.2: Instâncias sem ótimo por tipo de distribuição.

A Tabela 5.2 apresenta as mesmas instâncias que não alcançaram a otimalidade, desta vez classificadas conforme as dimensões das peças a serem cortadas. Essa

organização permite analisar se há alguma relação entre o tipo de distribuição das peças e a dificuldade em resolver as instâncias.

A Figura 5.1 mostra o tempo de execução dessas 452 instâncias utilizando o método de Geração de Colunas de acordo com o tamanho da instância.

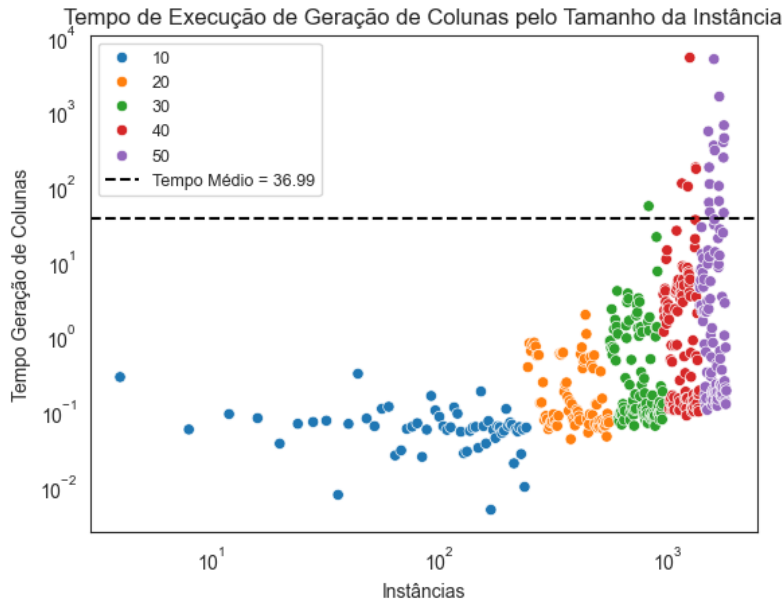


Figura 5.1: Tempo (s) de execução do Geração de Colunas por tamanho de instância.

Um padrão observado foi que as instâncias menores, em geral, foram resolvidas rapidamente, consumindo poucos recursos computacionais. Por outro lado, à medida que o tamanho do problema aumentava, o tempo de execução também crescia, refletindo a maior complexidade envolvida na geração e avaliação de padrões de corte. Esse comportamento reforça a relação direta entre o tamanho da instância e o esforço computacional necessário para alcançar a solução.

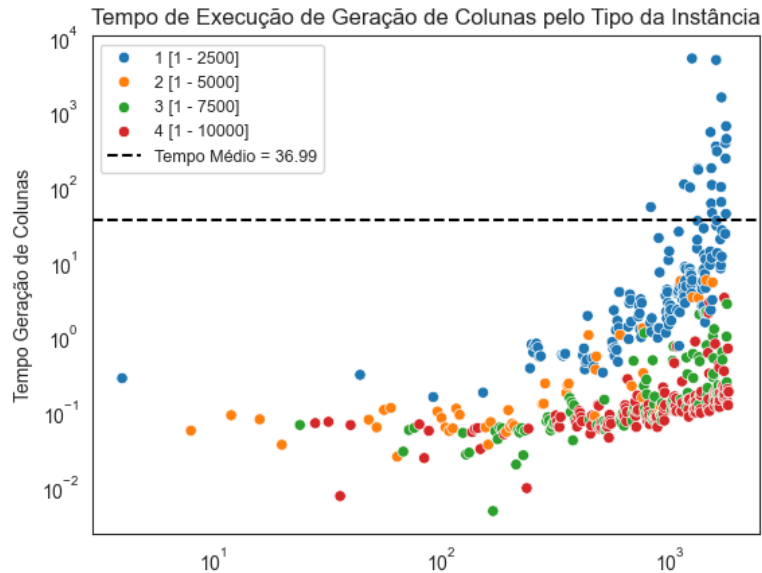


Figura 5.2: Tempo (s) de execução do Geração de Colunas por tipo de instância.

A Figura 5.2 destaca um padrão claro: independentemente da quantidade de peças em uma instância, quanto menores forem as dimensões das peças, maior será o tempo necessário para encontrar a solução do problema. Esse comportamento

ocorre porque, à medida que as peças diminuem de tamanho, o número possível de combinações de novos padrões de corte aumenta significativamente.

5.3 Solução Ótima

O algoritmo de Geração de Colunas fornece um limitante inferior para a solução do problema. Para determinar a solução ótima de cada instância, a formulação de Kantorovich, definida por (3.1)-(3.5), foi aplicada às 452 instâncias restantes, com imposição de um tempo limite de execução de uma hora. A Figura 5.3 ilustra o tempo necessário para encontrar a solução ótima dessas instâncias.

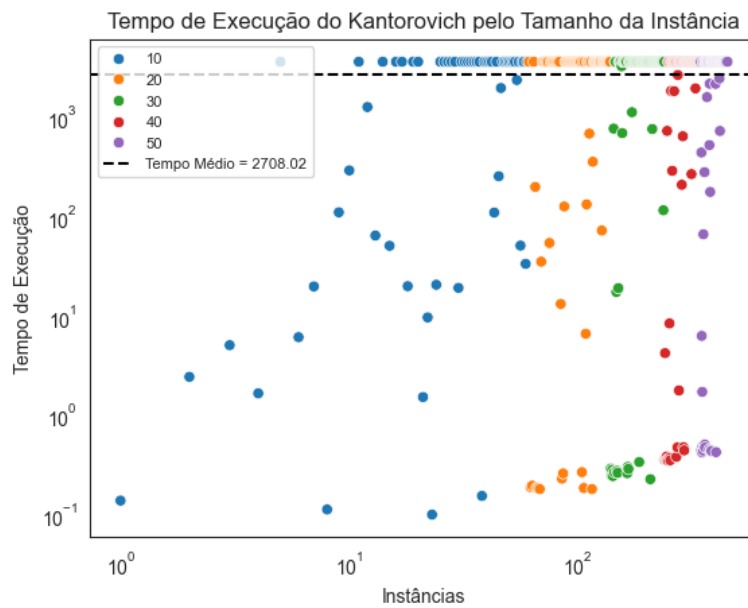


Figura 5.3: Tempo (s) de execução para resolver a formulação de Kantorovich.

A Figura 5.3 apresenta os tempos de execução das instâncias cuja otimalidade não foi comprovada pelo algoritmo de Geração de Colunas, utilizando, para isso, a formulação de Kantorovich como referência. Foi estipulado um limite de tempo de 1 hora para a execução da formulação de Kantorovich. Dentro desse intervalo, foram obtidas soluções primais de boa qualidade, que puderam ser comprovadas como ótimas por meio da comparação com os limitantes inferiores gerados pelo algoritmo de Geração de Colunas. Vale destacar que a aplicação da Kantorovich teve como único objetivo a obtenção dos valores ótimos das instâncias, e não a avaliação de seu desempenho computacional.

Entre as instâncias testadas, apenas 72 conseguiram finalizar antes do tempo limite estabelecido. Ao analisar a relação entre o tempo de execução e o tamanho das instâncias, percebe-se que o impacto do tamanho foi pouco expressivo, sugerindo que outros fatores podem ter exercido maior influência sobre o desempenho.

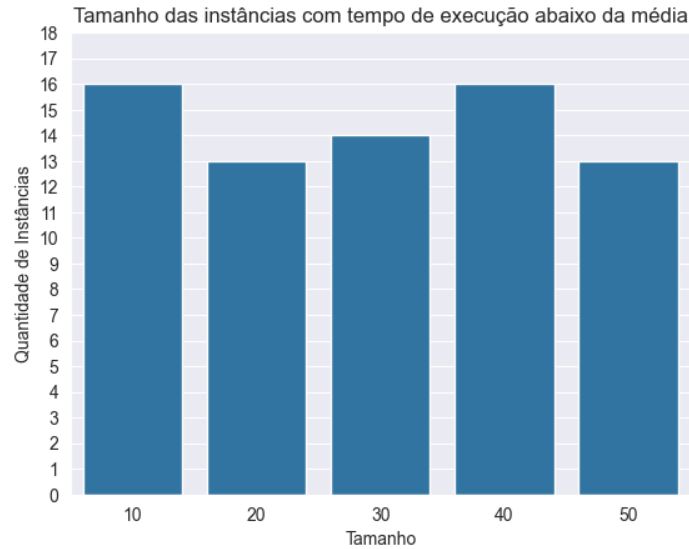


Figura 5.4: Quantificação das instâncias que finalizam antes do tempo limite por tamanho.

A Figura 5.4 confirma que o tamanho da instância não teve um impacto significativo no tempo de execução. Isso fica claro ao observar que a quantidade de instâncias, independentemente do tamanho, que encontraram solução antes do tempo limite foi bastante semelhante entre os diferentes grupos.

A combinação entre o melhor limitante superior obtido pela formulação de Kantorovich e o limitante inferior gerado pelo algoritmo de Geração de Colunas permitiu comprovar a otimalidade das instâncias analisadas. Embora a formulação de Kantorovich, isoladamente, não tenha alcançado esse resultado dentro do tempo estipulado, sua aplicação ainda foi eficaz para estimar o valor ótimo, mesmo sem o uso de versões mais refinadas do modelo.

Quando comparamos com os resultados da Figura 5.1, o contraste é claro: o algoritmo de Geração de Colunas de GILMORE e GOMORY (1961) apresentou um desempenho muito superior à formulação de Kantorovich. Enquanto o tempo médio de execução com Kantorovich foi de 2.708,02 segundos, o Geração de Colunas resolveu as instâncias em um tempo médio bem menor, apenas 36,99 segundos.

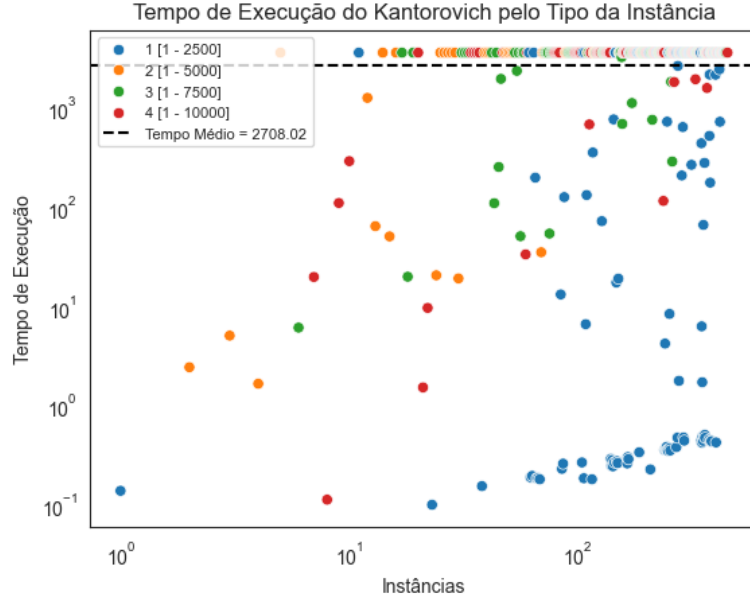


Figura 5.5: Tempo (s) de execução da formulação de Kantorovich por tipo.

A Figura 5.5 mostra que o tipo da instância, ou seja, as possíveis dimensões das peças a serem cortadas, influencia o tempo de execução. Nota-se uma concentração das instâncias com peças menores em uma faixa de menor tempo de execução, independentemente do tamanho total da instância.

5.4 Resultados Computacionais

Após a aplicação das Restrições de *Soft Fixing*, foi possível melhorar a solução em 199 das 452 instâncias que inicialmente não haviam alcançado a otimalidade. Essas melhorias aconteceram graças à redução do limitante superior das instâncias, obtida com a adição de uma das Restrições de *Soft Fixing*.

Os resultados detalhados, organizados por tipo de restrição aplicada, podem ser consultados na Tabela 5.3. Essa tabela permite visualizar de forma clara como cada uma das Restrições de *Soft Fixing* contribuiu para a melhoria das soluções, evidenciando quais abordagens se mostraram mais eficazes em termos de desempenho.

Tipo de Soft Fixing	Nº de Instâncias Melhoradas
1	0
2	0
3	0
4	118
5	182
6	0
7	191
8	189
9	0

Tabela 5.3: Desempenho das Restrições de *Soft Fixing*.

A distribuição dos tamanhos das instâncias que apresentaram melhorias com o uso das Restrições de *Soft Fixing* está apresentada na Tabela 5.4. Essa distribuição fornece uma visão mais detalhada sobre em quais faixas de tamanho as restrições foram mais eficazes, permitindo identificar padrões no comportamento do algoritmo em relação à complexidade das instâncias.

Tamanho	Nº de Instâncias
10	16
20	30
30	36
40	54
50	63

Tabela 5.4: Tamanho das instâncias melhoradas.

As instâncias que apresentaram melhoria no limitante superior estavam associadas, predominantemente, a distribuições com tamanhos de peças menores, conforme mostrado na Tabela 5.5. Esse comportamento sugere que as Restrições de *Soft Fixing* foram especialmente eficazes em cenários onde as peças possuem dimensões reduzidas, possivelmente devido à maior flexibilidade na combinação dos padrões de corte.

A Tabela 5.5 detalha essa distribuição, permitindo analisar como a variação nos tamanhos das peças influenciou diretamente o desempenho do algoritmo e a obtenção de soluções melhores.

Tipo de Distribuição	Nº de Instâncias
0.25	127
0.50	11
0.75	26
1.00	35

Tabela 5.5: Tipos de distribuições das instâncias melhoradas.

Em 15 dessas instâncias, a redução do limitante superior foi suficiente para que a solução alcançasse a otimalidade, resultado diretamente associado à aplicação das Restrições de *Soft Fixing*, como apresentado na Tabela 5.6. Esse resultado evidencia o potencial das restrições não apenas para melhorar a qualidade das soluções, mas também para levar o algoritmo a encontrar soluções ótimas em casos onde, inicialmente, isso não havia sido possível.

A Tabela 5.6 detalha essas instâncias, permitindo visualizar de forma clara o impacto positivo que a aplicação do *Soft Fixing* teve no desempenho do algoritmo.

Tipo de Soft Fixing	Nº de Instâncias
4	8
5	12
7	12
8	12

Tabela 5.6: Restrições de *Soft Fixing* que chegaram ao ótimo.

Quanto ao tamanho das instâncias que atingiram a otimalidade com o uso das Restrições de *Soft Fixing*, a Tabela 5.7 apresenta a distribuição correspondente. Essa distribuição permite observar em quais faixas de tamanho as restrições foram mais efetivas, indicando a relação entre a complexidade das instâncias e a capacidade das restrições de contribuir para a obtenção de soluções ótimas.

Essa análise é relevante para entender melhor o comportamento do algoritmo frente a diferentes tamanhos de problema, podendo orientar ajustes ou escolhas estratégicas no uso das restrições em cenários práticos.

Tamanho	Nº de Instâncias
10	2
20	2
30	3
40	6
50	2

Tabela 5.7: Tamanho das instâncias que chegaram à solução ótima com *Soft Fixing*.

A aplicação das Restrições de *Soft Fixing* nas instâncias em que o algoritmo de Geração de Colunas, isoladamente, não encontrou a solução ótima resultou em melhorias em 44% desses casos. Em relação ao tamanho das instâncias, o desempenho detalhado dessas melhorias está apresentado na Tabela 5.8.

Tamanho	% de Melhoria por Tamanho
10	26,22
20	38,46
30	36,00
40	51,49
50	58,33

Tabela 5.8: Desempenho das Restrições de *Soft Fixing* por tamanho.

O desempenho das restrições aplicadas, considerando especificamente os diferentes tipos que resultaram em melhorias, está apresentado de forma detalhada na Tabela 5.9. Essa tabela permite visualizar como cada tipo de Restrição de *Soft Fixing* contribuiu para a melhoria das soluções, evidenciando quais delas foram mais eficazes em reduzir o limitante superior ou até mesmo em alcançar a otimalidade em determinadas instâncias.

A análise desses resultados é importante para compreender o impacto individual de cada tipo de restrição no desempenho geral do algoritmo, além de auxiliar na identificação das abordagens mais promissoras para aplicações futuras ou adaptações do método.

Tipo de Soft Fixing	% de Melhoria por tamanho
4	26,10
5	40,26
7	42,25
8	41,81

Tabela 5.9: Desempenho das Restrições de *Soft Fixing* por tipo.

Com base na Tabela 5.8, observa-se que as Restrições de *Soft Fixing* apresentaram melhor desempenho nas instâncias de maior porte, alcançando 58,33% de melhorias entre aquelas com tamanho 50 e 51,49% nas de tamanho 40. Além de registrarem os maiores percentuais de melhoria, esses dois grupos também concentraram o maior número absoluto de casos.

Assim, o desempenho das restrições foi superior tanto proporcionalmente quanto em termos do volume total de instâncias com soluções aprimoradas.

Em relação ao tempo de execução, foi analisado o desempenho de cada uma das Restrições de *Soft Fixing* nas 452 instâncias mencionadas anteriormente. A análise

considerou apenas o tempo gasto na etapa iniciada após a execução do algoritmo de Geração de Colunas, ou seja, o período correspondente à inclusão e aplicação das restrições.

A análise do tempo de execução na etapa de *Soft Fixing* foi realizada inicialmente com base no tamanho das instâncias. Em seguida, o tempo também foi avaliado conforme o tipo de instância, que determina os tamanhos possíveis das peças a serem cortadas.

A Figura 5.6 apresenta os resultados obtidos para a Restrição de *Soft Fixing* do tipo 4, destacando o tempo de execução correspondente.

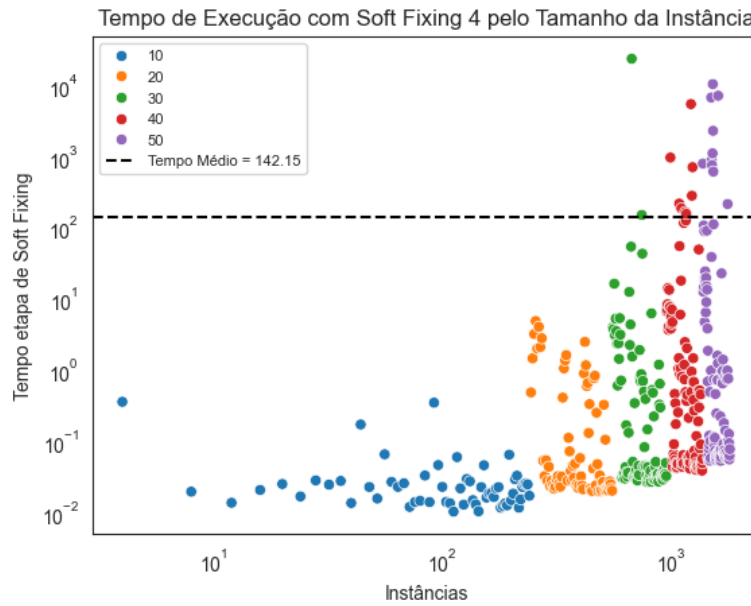


Figura 5.6: Tempo (s) de execução do *Soft Fixing* 4.

Com a aplicação dessa restrição, o desempenho em relação ao tempo apresentou um comportamento semelhante ao observado na Figura 5.1. No entanto, o tempo médio de execução aumentou consideravelmente, atingindo 142,15 segundos.

Além disso, mesmo com o aumento no tempo, a aplicação da restrição proporcionou uma melhoria significativa, visto que foram obtidas soluções inteiras aprimoradas em 118 instâncias, destacando o impacto positivo do método.

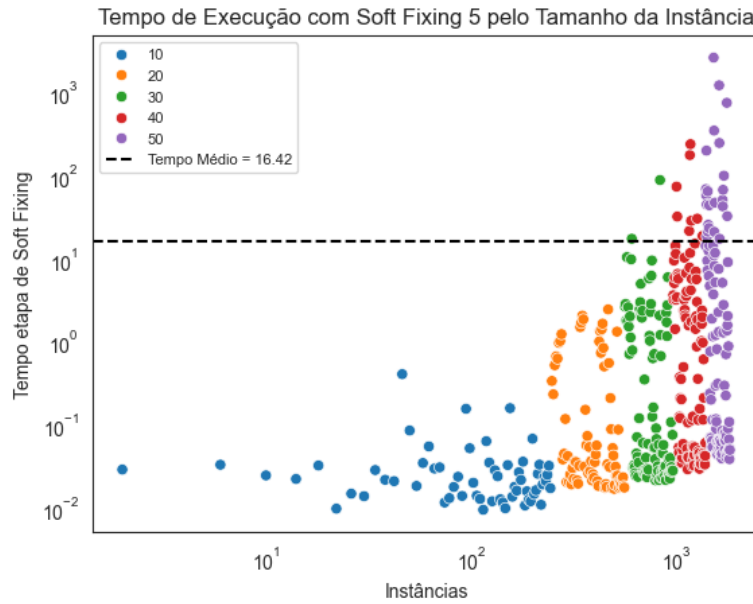


Figura 5.7: Tempo (s) de execução do *Soft Fixing* 5.

A Figura 5.7 apresenta o tempo de execução da Restrição de *Soft Fixing* do tipo 5, que teve um tempo médio de apenas 16,42 segundos. Em comparação com a restrição anterior, esse desempenho foi significativamente mais eficiente, reduzindo de forma expressiva o tempo necessário.

Além do ganho em tempo, a aplicação dessa restrição também resultou em uma melhoria considerável na qualidade das soluções, com 182 instâncias apresentando aprimoramentos. Esses resultados demonstram a eficácia do *Soft Fixing* 5, tanto em termos de eficiência computacional quanto na obtenção de soluções melhores.

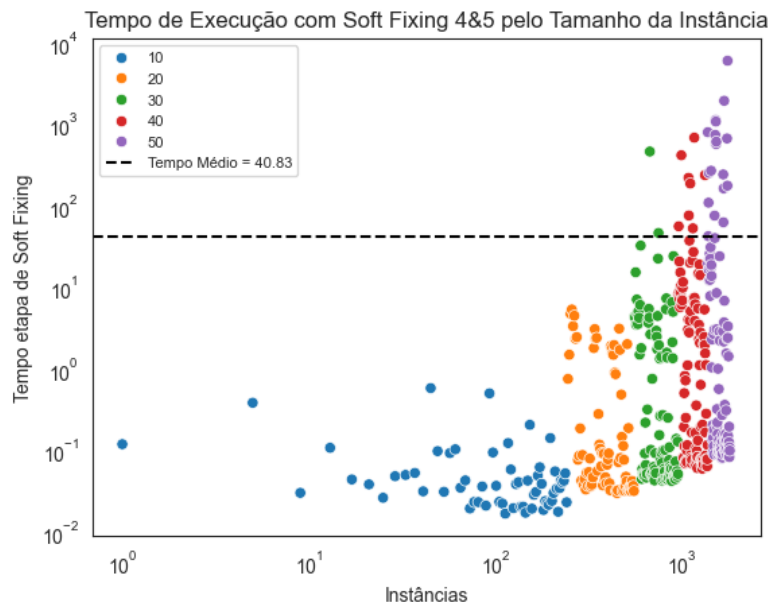


Figura 5.8: Tempo (s) de execução do *Soft Fixing* 4&5.

A Figura 5.8 mostra os tempos de execução do *Soft Fixing* 7 (4&5), que combina as abordagens 4 e 5, aplicando cada uma de maneira intercalada. Essa estratégia resultou em um tempo médio de execução de 40,83 segundos. Apesar de o *Soft Fixing* 4, quando usado sozinho, apresentar um tempo médio bem mais elevado que o do *Soft Fixing* 5, a combinação das duas restrições se mostrou significativamente mais eficiente, mantendo o tempo médio muito abaixo do que seria obtido com o uso exclusivo do *Soft Fixing* 4.

Além de tornar o processo mais ágil, essa combinação também teve um impacto positivo na qualidade das soluções, com 191 instâncias apresentando melhorias.

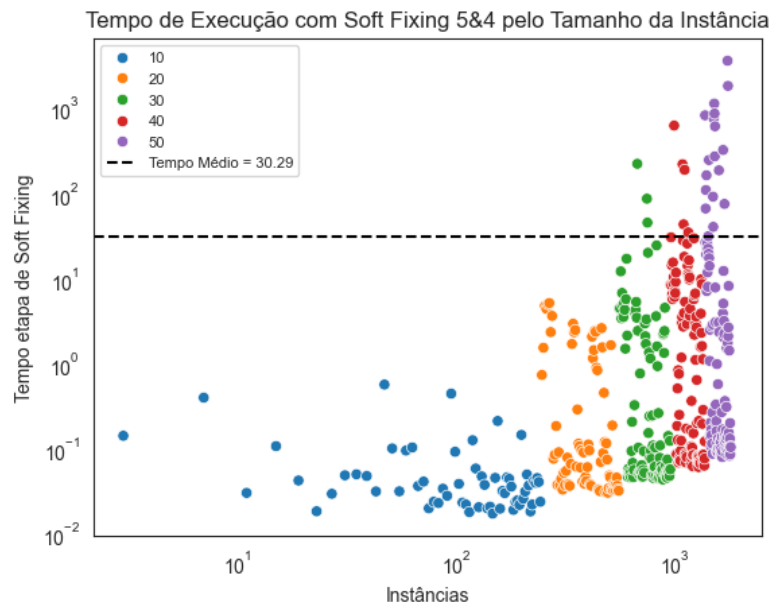


Figura 5.9: Tempo (s) de execução do *Soft Fixing* 5&4.

A Figura 5.9 mostra os tempos de execução da etapa de *Soft Fixing* 8 (5&4), que combina as abordagens 5 e 4. Embora essa combinação tenha resultado em um tempo médio elevado, de 484,50 segundos, o pior desempenho entre todas as restrições efetivas em termos de tempo, ela ainda apresentou um bom resultado na melhoria das soluções inteiras.

Com essa combinação, 189 instâncias tiveram suas soluções aprimoradas, o que evidencia sua eficácia, mesmo com o alto custo computacional envolvido.

Agora, a análise dos tempos de execução será feita considerando o tipo de instância, ou seja, focando no tamanho das peças a serem cortadas, em vez do tamanho total da instância. Essa mudança de perspectiva permite avaliar de forma mais precisa como o tamanho das peças individuais influencia o desempenho do algoritmo.

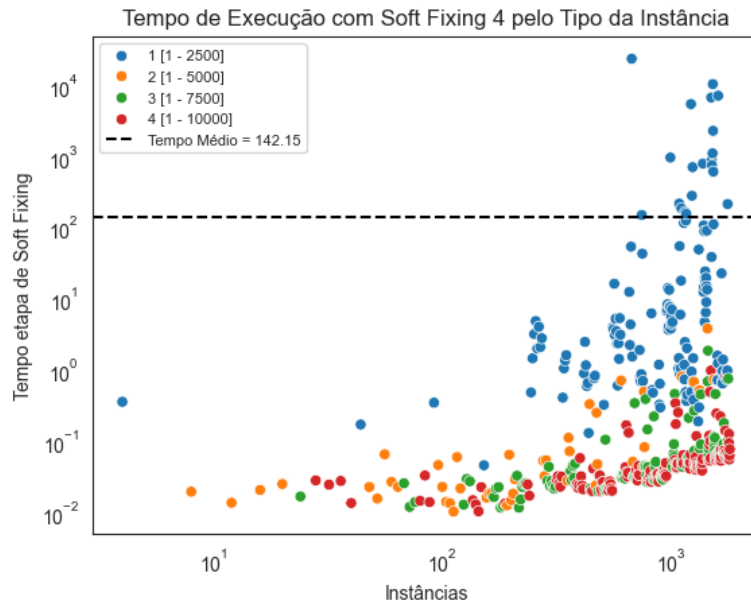


Figura 5.10: Tempo (s) de execução do *Soft Fixing* 4 por tipo.

A Figura 5.10 mostra os tempos de execução da etapa com a aplicação da Restrição de *Soft Fixing* 4, semelhante à Figura 5.6, mas agora organizados com base no tipo de instância. Assim como na etapa de Geração de Colunas, observa-se que as instâncias compostas por peças de dimensões menores demandaram mais tempo para serem concluídas, enquanto aquelas formadas por peças maiores apresentaram tempos de execução significativamente mais baixos.

Esse comportamento se deve à maior complexidade gerada pelo número elevado de padrões de corte possíveis nas instâncias com peças menores, o que torna o processo computacionalmente mais custoso.

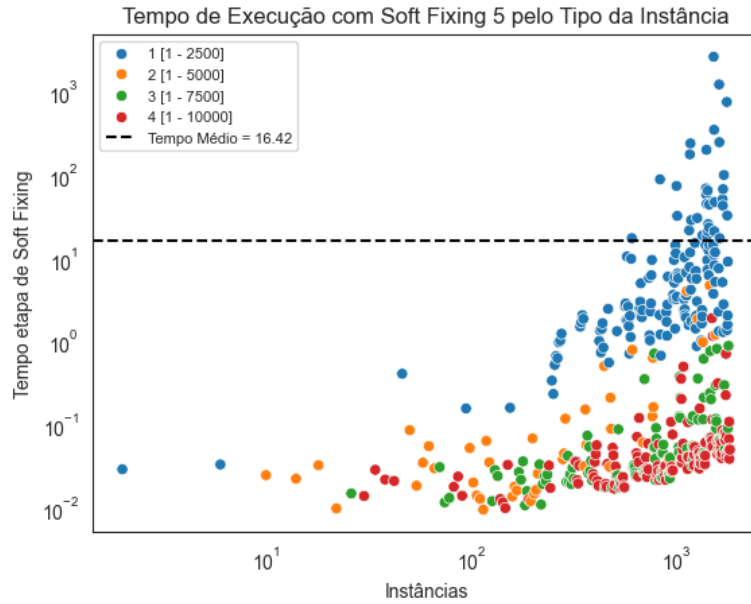


Figura 5.11: Tempo (s) de execução do *Soft Fixing* 5 por tipo.

A Figura 5.11 mostra o tempo de execução do *Soft Fixing* 5 em função das dimensões das peças a serem cortadas. Nota-se que, em torno do tempo médio, predominam as instâncias com peças menores, especialmente nos tamanhos 40 e 50, como já observado na Figura 5.7.

Com essa restrição, as instâncias compostas por peças maiores tendem a se concentrar em uma faixa de baixo tempo de execução, sem uma separação clara entre diferentes grupos. Já as instâncias com peças menores se destacam por uma concentração mais evidente na faixa de maiores tempos de execução, indicando que, nesses casos, o processamento é mais intenso para alcançar soluções satisfatórias.

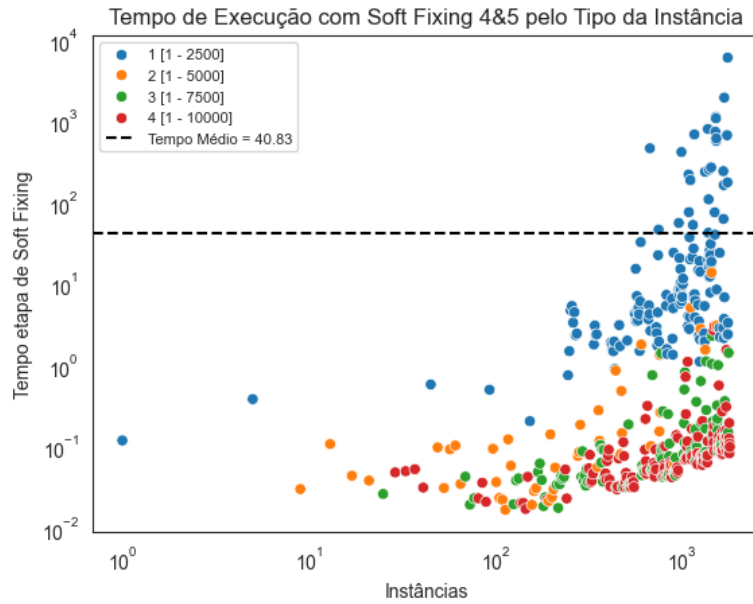


Figura 5.12: Tempo (s) de execução do *Soft Fixing* 4&5 por tipo.

A Figura 5.12 mostra os tempos de execução da etapa de *Soft Fixing* utilizando a restrição do tipo 7 (4&5), organizados conforme as dimensões das peças. Assim como nos casos anteriores, as instâncias compostas por peças menores exigiram mais tempo de processamento.

Nota-se uma concentração de instâncias com tempos próximos à média, enquanto todas aquelas formadas por peças maiores apresentaram tempos de execução inferiores a esse valor. Além disso, quanto maiores as dimensões das peças, menor tende a ser o tempo necessário para a execução, o que reforça um padrão claro de eficiência associado ao tamanho das peças.

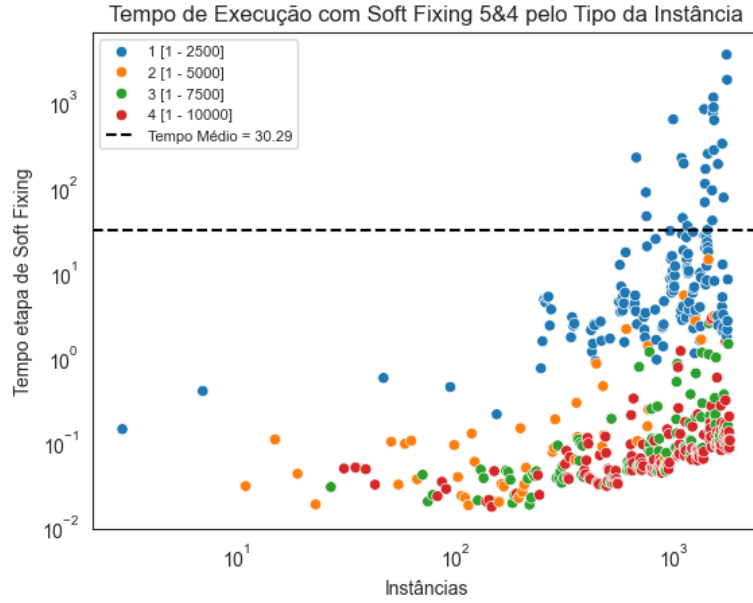


Figura 5.13: Tempo de execução do *Soft Fixing* 5&4 por tipo.

A Figura 5.11 apresenta um comportamento semelhante, mas com uma diferença importante: este foi o caso com o menor número de instâncias acima do tempo médio, todas associadas às menores dimensões das peças. Ao contrário dos cenários anteriores, aqui a concentração de instâncias não está tão próxima da média. Em vez disso, percebe-se uma distribuição mais dispersa, especialmente entre as instâncias com peças menores, indicando uma variabilidade maior no tempo de execução.

A seguir, será analisado o desempenho das Restrições de *Soft Fixing* considerando a execução completa do processo. Isso inclui não apenas a etapa específica de *Soft Fixing*, mas também todo o tempo de execução do algoritmo, englobando a Geração de Colunas, a aplicação das restrições e a resolução final do Problema Mestre Restrito Inteiro.

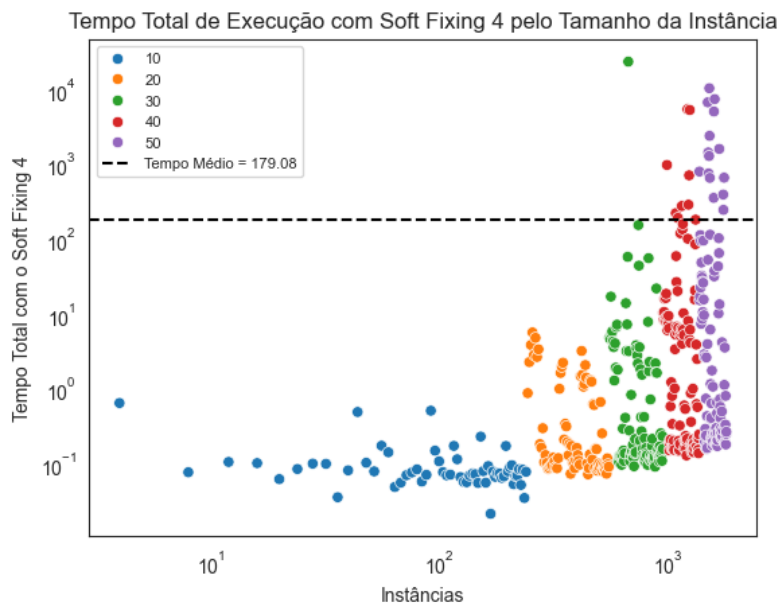


Figura 5.14: Tempo (s) total de execução com o *Soft Fixing* 4.

A Figura 5.14 mostra o tempo total de execução considerando o uso da Restrição de *Soft Fixing* 4. O tempo médio total, que já inclui todas as etapas do processo, geração de colunas, aplicação do *Soft Fixing* e resolução do Mestre Restrito Inteiro, foi de 179,08 segundos.

Com essa restrição, destaca-se que 94,13% das instâncias foram executadas em um tempo abaixo da média, evidenciando a eficiência geral dessa abordagem em termos de tempo de processamento.

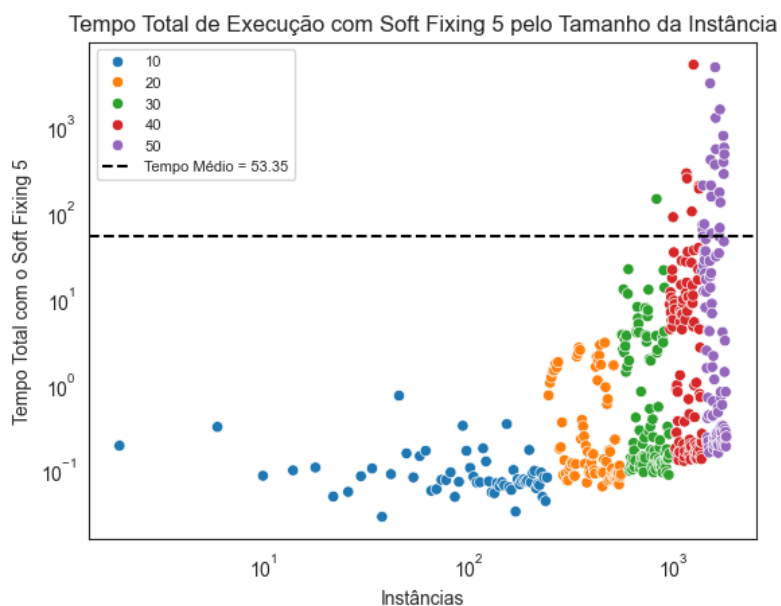


Figura 5.15: Tempo (s) total de execução com o *Soft Fixing* 5.

A Figura 5.15 mostra o tempo total de execução com a aplicação da Restrição de *Soft Fixing* 5, que apresentou um tempo médio de 53,35 segundos. Esse valor foi significativamente inferior ao observado com o uso do *Soft Fixing* 4, mostrado na Figura 5.14, o que evidencia um desempenho mais eficiente.

Assim como na Restrição do tipo 4, a maioria das instâncias foi resolvida em um tempo inferior à média, mais especificamente, 93,11% delas, reforçando a consistência e a eficácia dessa abordagem em termos de tempo de execução.

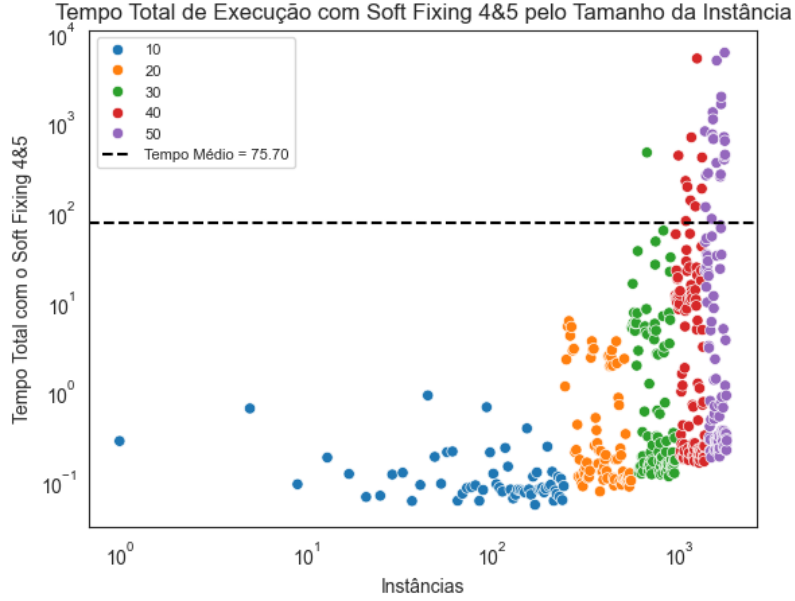


Figura 5.16: Tempo (s) total de execução com o *Soft Fixing* 4&5.

A Figura 5.16 mostra o tempo total de execução ao utilizar a restrição de *Soft Fixing* do tipo 7 (4&5), com um tempo médio de 75,70 segundos. Nesse cenário, 92,60% das instâncias foram executadas em um tempo inferior à média.

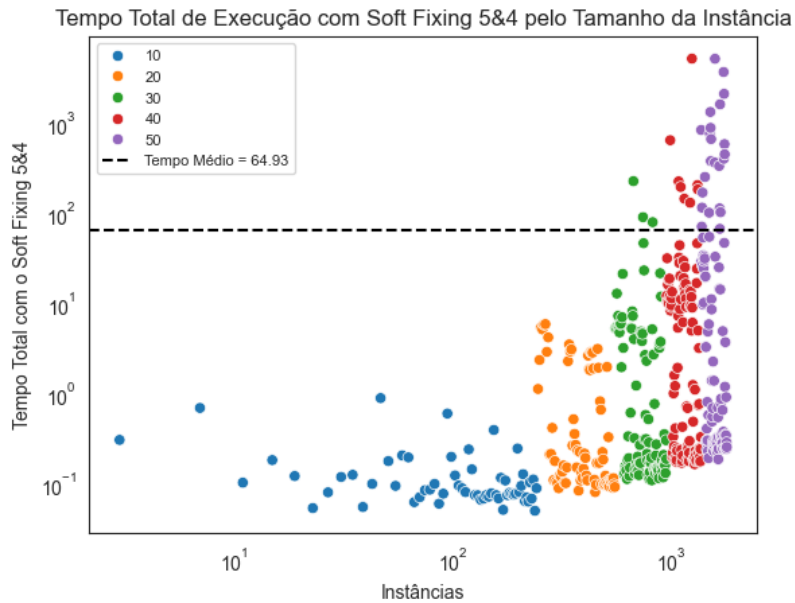


Figura 5.17: Tempo (s) total de execução com o *Soft Fixing* 5&4.

A Figura 5.17 mostra o tempo total de execução com a aplicação da Restrição de *Soft Fixing* do tipo 8 (5&4), que resultou em um tempo médio de 64,93 segundos. Nesse cenário, 92,39% das instâncias foram executadas em um tempo inferior à média.

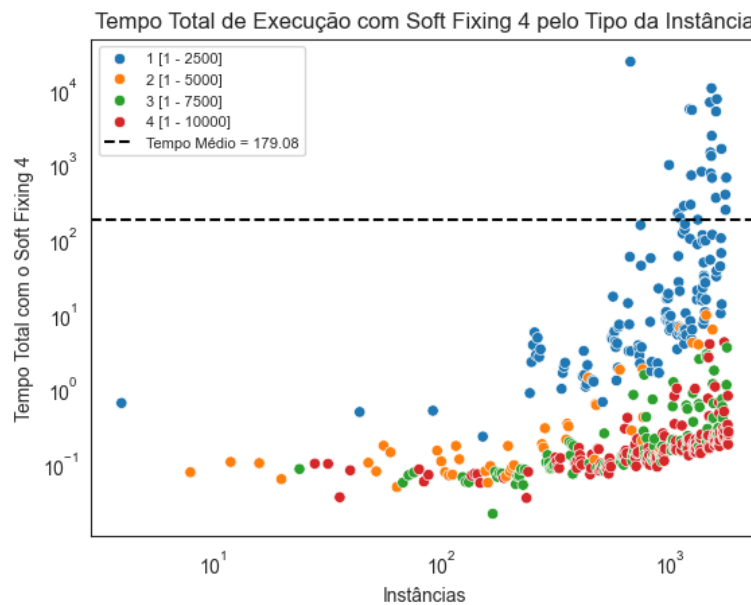


Figura 5.18: Tempo (s) total de execução com o *Soft Fixing* 4 por tipo.

A Figura 5.18 exibe o tempo total de execução do algoritmo de Geração de

Colunas com a Restrição de *Soft Fixing* do tipo 4. É possível observar o mesmo comportamento identificado em outras execuções: instâncias compostas por peças maiores tendem a exigir menos esforço computacional para serem resolvidas.

De modo geral, as instâncias se concentram em uma faixa de tempo consideravelmente baixa, com exceção daquelas cujas peças estão na faixa de $[1 - 2500]$, que apresentaram tempos de execução mais elevados.

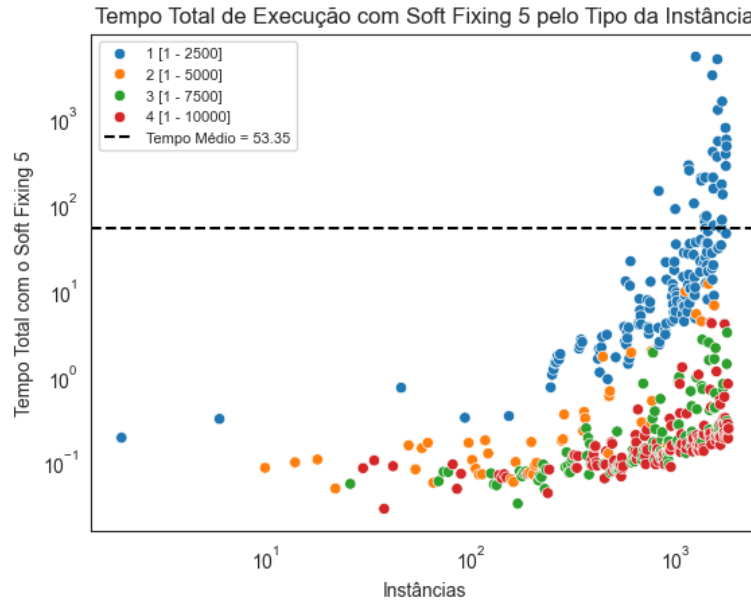


Figura 5.19: Tempo (s) de execução com o Sotf Fixing 5 por tipo.

A Figura 5.19 mostra os tempos de execução com a aplicação da Restrição de *Soft Fixing* do tipo 5, organizando as instâncias de acordo com o tamanho das peças do problema. Assim como na Figura 5.18, percebe-se que as instâncias com peças menores demandaram significativamente mais esforço computacional para serem solucionadas.

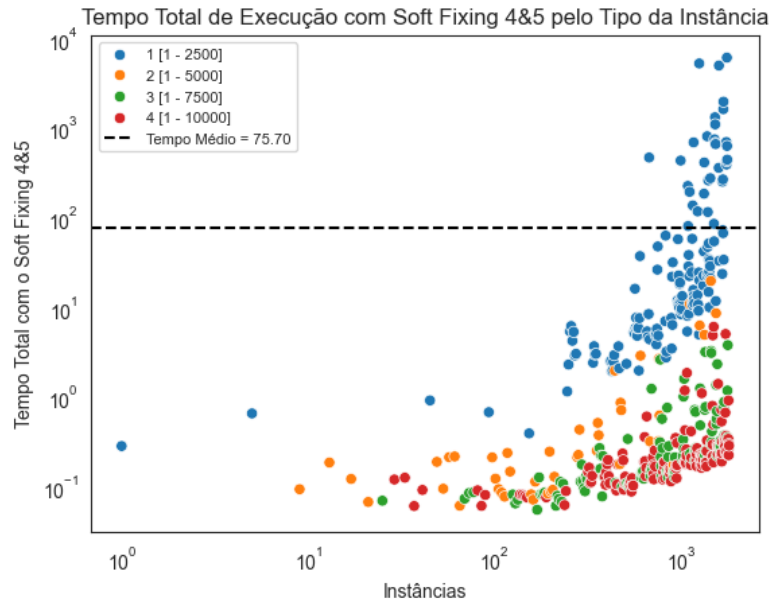


Figura 5.20: Tempo (s) de execução com o *Soft Fixing* 7 (4&5) por tipo.

A Figura 5.20 exibe o tempo total de execução das instâncias com a aplicação da Restrição de *Soft Fixing* do tipo 7 (4&5). Mais uma vez, percebe-se o mesmo comportamento observado em abordagens anteriores: as instâncias com peças de maiores dimensões foram resolvidas rapidamente, com tempos em torno de 1 segundo ou menos, enquanto as instâncias com peças menores demandaram mais tempo de execução, sendo as únicas que ultrapassaram o tempo médio.

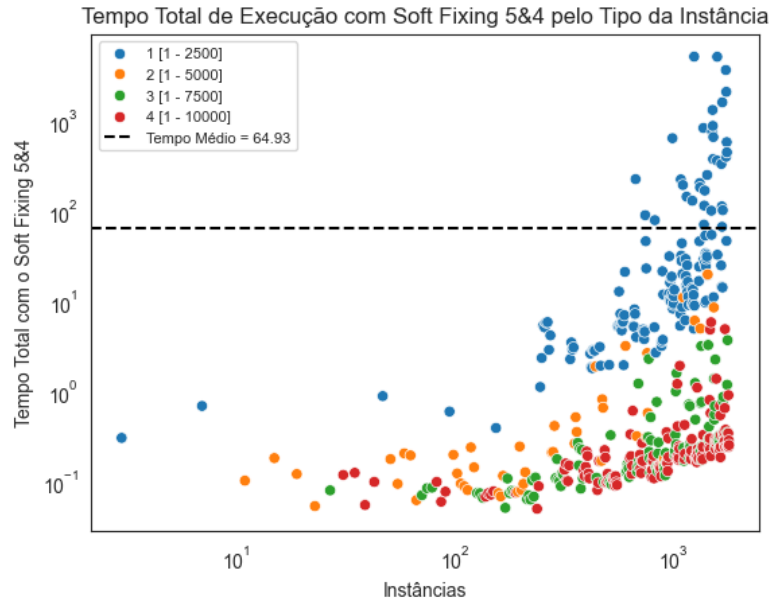


Figura 5.21: Tempo (s) de execução com o *Soft Fixing* 8 (5&4) por tipo.

A Figura 5.21 apresenta um comportamento muito semelhante ao da Figura 5.20, com um tempo médio de execução ligeiramente inferior. Ainda assim, o padrão se manteve: as instâncias compostas por peças menores continuaram a demandar mais tempo para serem resolvidas, concentrando-se acima da média de execução. Esse resultado reforça a tendência já observada nas demais abordagens, indicando que a dimensão das peças exerce um papel determinante no esforço computacional necessário.

Na execução do algoritmo de Geração de Colunas sem a aplicação das Restrições de *Soft Fixing*, observou-se um tempo médio de 36,99 segundos. O menor tempo registrado foi de apenas 0,0048 segundos, refletindo a alta eficiência em algumas instâncias simples. Em contraste, a instância mais complexa exigiu 5228,11 segundos para ser resolvida, evidenciando o impacto do aumento da complexidade. Ainda assim, o coeficiente de variação de 9,49% indica uma boa estabilidade no desempenho do algoritmo ao longo do conjunto de testes.

A Tabela 5.10 apresenta as medidas estatísticas referentes às Restrições de *Soft Fixing* que se mostraram mais efetivas. Entre elas, a restrição do tipo 7 (4&5) se destacou por ter a menor variação nos tempos de execução, indicando uma maior consistência no desempenho. Além disso, foi a que apresentou os melhores resultados em termos de melhoria das soluções, tornando-se a alternativa mais equilibrada entre eficiência e qualidade.

Soft Fixing	Média (s)	Mínimo (s)	Máximo (s)	Coef. de Variação (%)
4	179,08	0,0213	23.115,26	7,60
5	53,35	0,0281	5.260,23	7,24
7 (4&5)	75,70	0,0546	5.995,16	6,21
8 (5&4)	64,93	0,0495	5.092,74	6,35

Tabela 5.10: Estatísticas do tempo total de execução de cada *Soft Fixing*.

Os valores mínimos de tempo são bastante baixos e próximos entre si, o que mostra que, em alguns casos, as restrições puderam ser executadas com grande rapidez. O menor tempo registrado foi da restrição do tipo 4, com apenas 0,0213 segundos.

Por outro lado, ao observar os valores máximos de tempo, nota-se que, com exceção da restrição do tipo 4, que apresentou um tempo significativamente elevado, de 23.115,26 segundos, os demais tempos são relativamente próximos, girando em torno de 5.000 segundos.

Capítulo 6

Conclusões e Trabalhos Futuros

Nesta dissertação, foi proposta a incorporação de Restrições de *Soft Fixing* ao algoritmo de Geração de Colunas, como uma estratégia para resolver o Problema de Cutting Stock Unidimensional (PCS). A eficácia da metodologia foi avaliada a partir de um conjunto de 1.600 instâncias geradas artificialmente, com variação nos tamanhos e nas características dos problemas.

Utilizando apenas o algoritmo de Geração de Colunas, 452 instâncias (28% do total) permaneceram sem a solução ótima confirmada, indicando uma eficiência de aproximadamente 72%. Para esse grupo de instâncias, a aplicação das Restrições de *Soft Fixing* conseguiu melhorar as soluções em 44% dos casos (199 instâncias), embora a otimalidade tenha sido efetivamente alcançada em apenas 3% delas (15 instâncias).

A análise dos resultados, como mostrado na Tabela 5.8, revelou que as Restrições de *Soft Fixing* tiveram um desempenho especialmente positivo nas instâncias de maior porte, com 58,33% de melhoria nas de tamanho 50 e 51,49% nas de tamanho 40. Além de alcançarem os maiores percentuais de melhoria, esses grupos também concentraram o maior número absoluto de instâncias aprimoradas.

Assim, os resultados confirmam a hipótese levantada na introdução de que a incorporação das Restrições de *Soft Fixing* pode, de fato, contribuir significativamente para o aprimoramento das soluções, especialmente nas instâncias mais complexas do PCS.

Considerando as melhorias obtidas, a abordagem mais atrativa foi a restrição do tipo 7, que apresentou o maior número de instâncias aprimoradas, embora com resultados próximos aos das restrições dos tipos 8 e 5, segunda e terceira colocadas, respectivamente.

Em relação ao tempo de execução, a Restrição de *Soft Fixing* do tipo 5 se destacou como a mais eficiente, superando as demais estratégias testadas. No entanto, suas estatísticas de desempenho, como tempo máximo e média de execução, ainda foram ligeiramente superiores às observadas na versão original do algoritmo de Ge-

ração de Colunas, o que implica um pequeno custo adicional associado à aplicação da técnica.

Durante a etapa de implementação, foram identificadas algumas limitações relevantes. Uma delas refere-se à formulação de Kantorovich utilizada para obter o valor ótimo das instâncias geradas, que, por ser uma versão mais fraca, apresenta menor capacidade de exploração eficiente do espaço de soluções. Isso ocorre devido a definição do valor inicial de K , calculado apenas como a soma das demandas, o que não contribui para o fortalecimento do modelo, podendo introduzir simetrias indesejadas e impactar negativamente o desempenho do método.

Outra limitação importante está relacionada à estratégia adotada para resolver o subproblema de mochila, que neste trabalho foi delegada a um solver generalista. Embora funcional, essa abordagem não explora o potencial de algoritmos especializados para o problema, como técnicas de Programação Dinâmica mais eficientes, já propostas por autores como PISINGER (1995). O uso de métodos mais específicos poderia reduzir significativamente o tempo de execução e aumentar a eficiência do processo de geração de colunas.

Para trabalhos futuros, uma possibilidade é implementar o algoritmo na linguagem C++, visando à melhoria do tempo de execução. Ainda dentro da mesma técnica utilizada neste trabalho, também se sugere substituir a abordagem de resolução do subproblema por um algoritmo de programação dinâmica, o que pode reduzir significativamente o tempo de processamento, especialmente em instâncias mais complexas, principalmente aquelas compostas por peças menores, já que, como foi observado, quanto menores as peças, maior tende a ser o tempo de execução.

A formulação de Kantorovich pode ser fortalecida por meio de duas modificações importantes. A primeira consiste na inclusão de restrições para eliminação de simetria, como $y_{k+1} \leq y_k$ para todo $k \in \{1, \dots, K - 1\}$, o que reduz o espaço de busca do Branch-and-Bound. A segunda envolve a definição mais eficiente do parâmetro K , em vez de utilizar $\sum_{i=1}^m d_i$, que tende a superestimar o problema, recomenda-se adotar $\lceil PMRR \rceil + 1$, com base na solução do Problema Mestre Restrito Relaxado. Juntas, essas estratégias tornam a formulação mais robusta e computacionalmente eficiente.

Outra possibilidade relevante seria a incorporação das Restrições de *Soft Fixing* ao método de Branch-and-Price, ampliando seu potencial de exploração em estruturas combinatórias mais complexas. Essa integração poderia contribuir para acelerar a convergência em problemas de maior escala, ao direcionar a busca para regiões promissoras do espaço de soluções.

Além disso, propõe-se a incorporação de Cortes Fracionários de Gomory ao algoritmo de Geração de Colunas, com o objetivo de aproximar a região viável do problema à sua envoltória convexa, aumentando assim a chance de atingir a otima-

lidade em um número ainda maior de instâncias.

Referências Bibliográficas

- KANTOROVICH, L. V. “Mathematical methods of organizing and planning production”, *Management science*, v. 6, n. 4, pp. 366–422, 1960.
- GILMORE, P. C., GOMORY, R. E. “A linear programming approach to the cutting-stock problem”, *Operations research*, v. 9, n. 6, pp. 849–859, 1961.
- LÜBBECKE, M. E., DESROSIERS, J. “Selected topics in column generation”, *Operations research*, v. 53, n. 6, pp. 1007–1023, 2005.
- PIERINI, L. M., POLDI, K. C. “A bi-objective multiperiod one-dimensional cutting stock problem”, *Pesquisa Operacional*, v. 42, pp. e258432, 2022.
- SENERGUES, V., BRAHIMI, N., CHERRI, A., et al. “Cutting stock problem with usable leftovers: A review”, *European Journal of Operational Research*, 2025.
- DE ARAÚJO SILVA OLIVEIRA, E., WANNER, E., DE SÁ, E. M., et al. “A local branching-based solution for the multi-period cutting stock problem with tardiness, earliness, and setup costs”, *Journal of Heuristics*, v. 31, n. 1, pp. 16, 2025. ISSN: 1572-9397. doi: 10.1007/s10732-025-09547-4.
- MUSTAKEROV, I., BORISSOVA, D. “One-dimensional cutting stock model for joinery manufacturing”, *Proc. Advanced Information Science and Applications*, v. 1, pp. 17–21, 2014.
- MUSTAKEROV, I. C., BORISSOVA, D. I. “Combinatorial optimization modeling approach for one-dimensional cutting stock problems”, *International Journal of Systems Applications, Engeneering & Development*, v. 9, pp. 13–18, 2015.
- FANG, J., RAO, Y., LUO, Q., et al. “Solving one-dimensional cutting stock problems with the deep reinforcement learning”, *Mathematics*, v. 11, n. 4, pp. 1028, 2023.
- FISCHETTI, M., LODI, A. “Local branching”, *Mathematical programming*, v. 98, pp. 23–47, 2003.

- PAGE, M. J., MCKENZIE, J. E., BOSSUYT, P. M., et al. “Updating guidance for reporting systematic reviews: development of the PRISMA 2020 statement”, *Journal of clinical epidemiology*, v. 134, pp. 103–112, 2021.
- DYCKHOFF, H. “A new linear programming approach to the cutting stock problem”, *Operations Research*, v. 29, n. 6, pp. 1092–1104, 1981.
- WÄSCHER, G., GAU, T. “Heuristics for the integer one-dimensional cutting stock problem: A computational study”, *Operations-Research-Spektrum*, v. 18, n. 3, pp. 131–144, 1996a.
- BELOV, G., SCHEITHAUER, G. “A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting”, *European journal of operational research*, v. 171, n. 1, pp. 85–106, 2006.
- SILVA, V., GRANDE, A. J., CARVALHO, A. P. V. D., et al. “Overview of systematic reviews-a new type of study. Part II”, *Sao Paulo Medical Journal*, v. 133, n. 3, pp. 206–217, 2014.
- DA SILVA, R. F., SCHOUERY, R. “A Branch-and-Cut-and-Price Algorithm for Cutting Stock and Related Problems”, *arXiv preprint arXiv:2308.03595*, 2023.
- SCHEITHAUER, G., TERNO, J., MÜLLER, A., et al. “Solving one-dimensional cutting stock problems exactly with a cutting plane algorithm”, *Journal of the Operational Research Society*, v. 52, n. 12, pp. 1390–1401, 2001.
- HOLTHAUS, O. “Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths”, *European Journal of Operational Research*, v. 141, n. 2, pp. 295–312, set. 2002. ISSN: 0377-2217. doi: 10.1016/S0377-2217(02)00126-1.
- WONGPRAKORNKUL, S., CHARNSETHIKUL, P. “Optimization Based Heuristic Approaches for Solving an Integrated One-dimensional Cutting Stock-Transportation Problem”, *Journal of Mathematics and Statistics*, v. 3, n. 3, pp. 142–150, 2007.
- ANDRADE, P. R. D. L., DE ARAUJO, S. A., CERRI, A. C., et al. “The integrated lot sizing and cutting stock problem in an automotive spring factory”, *Applied Mathematical Modelling*, v. 91, pp. 1023–1036, 2021.

- SILALAH, B. P., HANUM, F., SETYAWAN, F., et al. “One-Dimensional Cutting Stock Problem that Minimizes The Number of Different Patterns”, *BA-REKENG: Jurnal Ilmu Matematika dan Terapan*, v. 16, n. 3, pp. 805–814, 2022.
- XIAO, H., LIANG, Q., ZHANG, D., et al. “A method for demand-accurate one-dimensional cutting problems with pattern reduction”, *Mathematical Biosciences and Engineering*, v. 20, n. 4, pp. 7453–7486, 2023.
- KAYHAN, N., TEKEZ, E. K. “OPTICUT: a new heuristic algorithm for the one-dimensional cutting stock problem with pattern minimization”, *Engineering Optimization*, pp. 1–23, 2025.
- JARIYAVAJEE, C., FAIREE, S., KHOMPATRAPORN, C., et al. “Efficient cutting stock optimization strategies for the steel industry”, *PloS one*, v. 20, n. 3, pp. e0319644, 2025.
- LAMERS, B. *Reducing waste by improving sheet metal efficiency*. B.S. thesis, University of Twente, 2021.
- PISINGER, D. *Algorithms for Knapsack Problems*. DIKU rapport. Univ., Department of Computer Science, 1995. Disponível em: <<https://books.google.com.br/books?id=x72zHAAACAAJ>>.
- DANNA, E., ROTHBERG, E., PAPE, C. L. “Exploring relaxation induced neighborhoods to improve MIP solutions”, *Mathematical Programming*, v. 102, n. 1, pp. 71–90, 2005.
- FISCHETTI, M., GLOVER, F., LODI, A. “The feasibility pump”, *Mathematical Programming*, v. 104, n. 1, pp. 91–104, 2005.
- WÄSCHER, G., GAU, T. “Test data for the one-dimensional cutting stock problem”, 1993.
- WÄSCHER, G., GAU, T. “CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem”, *European journal of operational research*, v. 84, n. 3, pp. 572–579, 1995.
- WÄSCHER, G., GAU, T. “Heuristics for the integer one-dimensional cutting stock problem: A computational study”, *Operations-Research-Spektrum*, v. 18, pp. 131–144, 1996b.

Apêndice A

Demonstração do Custo Reduzido do PCS

A partir da formulação do Problema Mestre temos como custo reduzido das variáveis é dado por $1 - \max \sum_{i=1}^m \bar{\pi}_i a_i$

Demonstração.

Seja P um problema de Programação Linear no formato padrão:

$$\begin{aligned} (P) \quad & \max \quad c x \\ & \text{sujeito a} \\ & A x = b \end{aligned}$$

Particionando as variáveis e a matriz de coeficientes em duas partições, B para variáveis básicas e N para as variáveis não básicas.

$$\begin{aligned} (P) \quad & \max \quad c_B x_B + c_N x_N \\ & \text{sujeito a} \\ & B x_B + N x_N = b \end{aligned}$$

Multiplicando $B x_B + N x_N = b$ por B^{-1} pela esquerda :

$$x_B + B^{-1} N x_N = B^{-1} b$$

Isolando x_B , temos:

$$x_B = B^{-1} b - B^{-1} N x_N$$

Substituindo x_B na função objetivo:

$$\begin{aligned}
& c_B (B^{-1} b - B^{-1} N x_N) + c_N x_N \\
& c_B B^{-1} b - c_B B^{-1} N x_N + c_N x_N \\
& c_B B^{-1} + (c_N - c_B B^{-1} N) x_N
\end{aligned}$$

Tomando $\bar{\pi} = c_B B^{-1}$ a solução dual de P .

Dessa forma, a expressão do custo reduzido fica determinada por :

$$c_N - \bar{\pi} N$$

Como não são conhecidas todas as variáveis não básicas, então é preciso encontrar a coluna a_j de N que forneça um custo reduzido negativo. Assim, temos como expressão para o custo reduzido:

$$c_j - \bar{\pi} a_j$$

Considerando o Cutting Stock Unidimensional como aplicação, temos que $c_B = c_N = 1$. Dessa forma, temos:

$$1 - \bar{\pi} a_j \Rightarrow 1 - \sum_{i=1}^m \bar{\pi}_i a_i$$

Como queremos $1 - \sum_{i=1}^m \bar{\pi}_i a_i < 0$, então podemos ter como objetivo minimizar $1 - \sum_{i=1}^m \bar{\pi}_i a_i$.

$$\min \left(- \sum_{i=1}^m \bar{\pi}_i a_i \right) \Rightarrow 1 - \max \sum_{i=1}^m \bar{\pi}_i a_i$$

Assim, temos a expressão do custo reduzido do Cutting Stock Unidimensional como $1 - \max \sum_{i=1}^m \bar{\pi}_i a_i$ □

Apêndice B

Decomposição de Dantzig-Wolfe do PCS

Dada a formulação de Kantorovich do PCS

$$\min \sum_{j=1}^n y_j \quad (\text{B.1})$$

sujeito a

$$\sum_{j=1}^n x_{ij} = d_i \quad \forall i = 1, \dots, m, \quad (\text{B.2})$$

$$\sum_{i=1}^m w_i x_{ij} \leq W y_j \quad \forall j = 1, \dots, n \quad (\text{B.3})$$

$$x_{ij} \in \mathbb{Z}_+, \quad \forall i = 1, \dots, m, j = 1, \dots, n \quad (\text{B.4})$$

$$y_j \in \{0, 1\} \quad \forall j = 1, \dots, n \quad (\text{B.5})$$

Para cada uma das peças mestres $j \in J$, construímos os conjuntos $\mathcal{P}_j$, a partir de todas as restrições dependentes de j , definido por:

$$\mathcal{P}_j = \left\{ \begin{pmatrix} x_{1j} \\ \vdots \\ x_{mj} \\ y_j \end{pmatrix} \left| \begin{array}{l} \sum_{i=1}^m w_i x_{ij} \leq W y_j \\ x_{ij} \in \mathbb{Z}_+, \forall i = 1, \dots, m \\ y_j \in \{0, 1\} \end{array} \right. \right\}$$

Como todas as peças mestres possuem o mesmo tamanho, podemos considerar um único conjunto $\mathcal{P}$ que representa todos os padrões de corte possíveis. Suponha que $\mathcal{P}$ tenha cardinalidade $K + 1$, incluindo o vetor nulo (em que $y_j = 0$ para todo j) e os demais K padrões viáveis, denotados por p_k , com $k = 1, \dots, K$.

Seja p_0 o vetor nulo com $k = 0$ e $p_k = (a_{1k}, \dots, a_{mk}, 1)$ com $k = 1, \dots, K$

Cada elemento do conjunto $\mathcal{P}$, composto por vetores inteiros que representam padrões de corte viáveis, pode ser representado da seguinte forma:

$$p = \sum_{k=0}^K \lambda_{jk} p_k$$

$$\sum_{k=0}^K \lambda_{jk} = 1$$

$$\lambda_{jk} \in \{0, 1\} \quad \forall k = 0, \dots, K$$

Desconsiderando o vetor nulo (p_0), podemos fazer as substituições para cada um dos elementos p_k :

$$x_{ij} = \sum_{k=1}^K \lambda_{jk} a_{ik} \tag{B.6}$$

$$y_j = \sum_{k=1}^K \lambda_{jk} \tag{B.7}$$

Substituindo (B.6) e (B.7) na formulação de Kantorovich (B.1)-(B.5):

$$\min \quad \sum_{j=1}^n \sum_{k=1}^K \lambda_{jk} \tag{B.8}$$

sujeito a

$$\sum_{j=1}^n \sum_{k=1}^K \lambda_{jk} a_{ik} \geq d_i \quad \forall i = 1, \dots, m \tag{B.9}$$

$$\sum_{k=0}^K \lambda_{jk} = 1 \quad \forall j = 1, \dots, n \tag{B.10}$$

$$\lambda_{jk} \in \{0, 1\} \quad \forall j = 1, \dots, n \tag{B.11}$$

A quantidade de vezes que o padrão de corte k é utilizado pode ser expressa como

$x_k = \sum_{j=1}^n \lambda_{jk}$. Assim, reescrevemos a função objetivo (B.8) em termos das variáveis x_k .

$$\sum_{j=1}^n \sum_{k=1}^K \lambda_{jk} = \sum_{k=1}^K x_k \tag{B.12}$$

A família de restrições de atendimento das demandas (B.9) pode ser rescrita como:

$$\sum_{j=1}^n \sum_{k=1}^K \lambda_{jk} a_{ik} \geq d_i \quad \Rightarrow \quad \sum_{k=1}^K a_{ik} \sum_{j=1}^n \lambda_{jk} \geq d_i \quad \Rightarrow \quad \sum_{k=1}^K a_{ik} x_k \geq d_i \quad (\text{B.13})$$

Com relação a restrição (B.10) , somando com relação a j :

$$\sum_{k=0}^K \sum_{j=1}^n \lambda_{jk} = \sum_{j=1}^n 1 \quad \Rightarrow \quad \sum_{k=0}^K x_k = |J| \quad (\text{B.14})$$

Como para $k = 0$ estamos nos referindo ao vetor nulo, a variável x_0 corresponde às peças mestres não utilizadas. Separando x_0 na restrição (B.14), temos:

$$x_0 + \sum_{k=1}^K x_k = |J| \quad \Rightarrow \quad \sum_{k=1}^K x_k \leq |J| \quad (\text{B.15})$$

Como a formulação é um problema de minimização em que a função objetivo é a expressão de (B.15), então é possível omitir essa restrição da formulação. E dessa forma, temos o Problema Mestre da formulação de Gilmore e Gomory ao utilizar (B.13) e (B.12) na formulação (B.8)-(B.11):

$$\begin{aligned} \min \quad & \sum_{k=1}^K x_k \\ \text{sujeito a} \quad & \sum_{k=1}^K a_{ik} x_k \geq d_i \quad \forall i = 1, \dots, m \\ & x_k \in \mathbb{Z}_+ \quad \forall k = 1, \dots, K \end{aligned} \quad (\text{B.16})$$

Tomando a relaxação linear de (B.16) :

$$\begin{aligned} \min \quad & \sum_{k=1}^K x_k \\ \text{sujeito a} \quad & \sum_{k=1}^K a_{ik} x_k \geq d_i \quad \forall i = 1, \dots, m \end{aligned} \quad (\text{B.17})$$

$$\sum_{k=1}^K a_{ik} x_k \geq d_i \quad \forall i = 1, \dots, m \quad (\text{B.18})$$

$$x_k \in \mathbb{R}_+ \quad \forall k = 1, \dots, K \quad (\text{B.19})$$

Sejam π_i as variáveis duais associadas às restrições (B.18) e o custo reduzido de uma coluna p (padrão de corte viável) seja dado por :

$$\bar{c}_p = c_p - \sum_{i=1}^m \pi_i a_{ip} \quad (\text{B.20})$$

Na formulação que estamos utilizando temos $c_p = 1$ e que se $\bar{c}_p < 0$ significa que a inclusão desse padrão reduziria o valor da função objetivo do Problema Mestre.

Para isso se resolve o subproblema

$$\min_{a \in \mathcal{P}} \left(1 - \sum_{i=1}^m \pi_i a_{ip} \right) \Rightarrow 1 - \max_{a \in \mathcal{P}} \left(\sum_{i=1}^m \pi_i a_{ip} \right) \quad (\text{B.21})$$

Onde $\mathcal{P}$ considera somente os padrões não nulos possíveis. E dessa forma podemos reescrever (B.21) como:

$$\begin{aligned} & 1 - \max \sum_{i=1}^m \pi_i a_{ip} \\ & \text{sujeito a} \\ & \sum_{i=1}^m w_i a_{ip} \leq W \\ & a_{ip} \in \mathbb{Z}_+ \end{aligned}$$

Assim, temos a Decomposição de Gilmore e Gomory do PCS através da Decomposição de Dantzig Wolfe.